IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 27, 2019, accepted January 10, 2020, date of publication January 22, 2020, date of current version January 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968576

Energy-Quality Scalable Memory-Frugal Feature Extraction for Always-On Deep Sub-mW Distributed Vision

ANASTACIA ALVAREZ[1,2], (Senior Member, IEEE), GOPALAKRISHNAN PONNUSAMY[1], (Member, IEEE), AND MASSIMO ALIOTO[1], (Fellow, IEEE)

[1]ECE Department, National University of Singapore, Singapore 117583
[2]EEE Institute, University of the Philippines-Diliman, Quezon City 1101, Philippines

Corresponding author: Anastacia Alvarez (anastacia.alvarez@eee.upd.edu.ph)

This work was supported by the Singapore National Research Foundation ''CogniVision'' under Grant NRF-CRP20-2017-0003. The work of Anastacia Alvarez was supported in part by the ERDT Faculty Development Program of the University of the Philippines and DOST.

ABSTRACT In this work, an energy-quality (EQ) scalable and memory-frugal architecture for video feature extraction is introduced to reduce circuit complexity, power and silicon area. Leveraging on the inherent resiliency of vision against noise and inaccuracies, the proposed approach introduces properly selected EQ tuning knobs to reduce the energy of feature extraction at graceful quality degradation. As opposed to prior art, the proposed architecture enables the adjustment of such knobs, and adapts its cycle-level timing to reduce the amount of computation per frame at lower quality targets. As further benefit, the approach adds opportunities for energy reduction via aggressive voltage scaling. The proposed architecture mitigates the traditionally dominant area/energy of the on-chip memory by reducing the number of pixels stored on chip, introducing memory access reuse and on-the-fly computation. At the same time, EQ tuning preserves the ability to conventionally operate at maximum quality, when required by the task or the visual context. A 0.55 mm^2 testchip in 40nm exhibits power down to 82μW at 5fps frame rate (i.e., 33X lower than prior art), while assuring successful object detection at VGA resolution. To the best of the authors' knowledge, this is the first feature extractor with sub-mW operation and sub-mm^2 area, making the proposed approach well suited for tightly power-constrained and low-cost distributed vision systems (e.g., video sensor nodes).

INDEX TERMS Low-power, energy-quality scaling, vision, video processing, feature extraction, Internet of Things, sensor nodes.

I. INTRODUCTION

Feature extraction is a fundamental task in integrated systems for vision, and is typically the front-end in computer vision systems based on machine learning algorithms [1]. Indeed, feature extraction reduces the dimensionality of compute-intensive vision tasks such as object classification, detection and tracking [2], as mandated in always-on and tightly-constrained computer vision systems [3] (e.g., real-time vision sensor nodes). In such always-on systems, recent deep learning frameworks are well known to be unsuitable, as their power (e.g., tens of mW or more [4]) vastly exceeds the power budget of self-powered vision sensor nodes [5].

Feature extraction accelerators reduce the pixels in a video frame into a smaller set of interest points (''keypoints'') with a description that is affine invariant (invariant against rotation, translation and scaling). This supports robust visual comprehension regardless of the position of individual objects in the scene [6]. Being always on even when no event is occurring in the scene, feature extraction dictates the system's minimum power consumption in self-powered vision sensor nodes [3], [7]. Unfortunately, feature extraction accelerators tend to be area-hungry, due to the high degree of parallelism and the large memory required by real-time operation [8]–[11]. On the other hand, self-powered low-cost

The associate editor coordinating the review of this manuscript and approving it for publication was Adnan M. Abu-Mahfouz.

vision sensor nodes are required to exhibit very low power and area due to battery life, form factor and cost requirements [3]. Accordingly, vision sensor nodes routinely have moderate resolutions, which are typically around VGA or slightly higher [12].

In this paper, an energy-quality (EQ) scalable memory-frugal architecture for low-power and low-area feature extraction is proposed for vision sensor nodes. As fundamental contribution, the proposed architecture is EQ scalable [13], [14] in that EQ tuning knobs are properly selected and inserted to dynamically adjust the balance between energy and feature extraction performance (i.e., good keypoint matches across rotated / translated / rescaled images, as popular metric to compare feature extraction algorithms [1]). Relying on the resiliency of vision algorithms against inaccuracies and the proper EQ knob selection, EQ scaling is shown to reduce power at a marginal feature extraction quality degradation in the common case (e.g., marginally lower number of detected keypoints). When EQ knobs are tuned for lower quality targets, the proposed architecture adapts its cycle-level timing to reduce the amount of computation required to complete a frame (i.e., number of cycles). This offers additional opportunities for aggressive voltage scaling at a given frame rate, and hence further energy reduction. At the same time, EQ knob adjustment preserves the ability to meet the maximum quality when needed. At the algorithm level, Oriented FAST Rotated BRIEF (ORB) [15] is adopted to mitigate the large complexity of SIFT (Scale-Invariant Feature Transform) and SURF (Speeded Up Robust Features) feature extraction algorithms in prior demonstrations [3], [6], [9], [16], at nearly the same feature extraction performance. At the architectural level, the energy penalty associated with the memory is reduced via access reuse, on-the-fly computation to store minimal number of pixels, and further simplifications in costly tasks to reduce the memory size.

This paper is structured as follows. Section II discusses background and prior art. The proposed architecture is discussed in Sections III-V. The testchip measurement results and the comparison with the state of the art are given in Section VI. Conclusions are drawn in Section VII.

II. BACKGROUND AND PRIOR ART ON FEATURE EXTRACTION

Video feature extraction invariably comprises three major steps: keypoint detection, description, and matching as in Fig. 1. Keypoint detection identifies a set of keypoints that may be unique to the objects to be detected or tracked, e.g. edges, corners and blobs [1], [2]. In keypoint description, keypoints are represented in the form of an affine-invariant descriptor, so that the same object can be associated with its many possible visual representations in the scene. Once keypoints are detected and described, they are used for the specific vision task at hand, such as image/object detection, classification and tracking [3], [6], [8]. For example, object classification is performed by matching the object class and the available set of keypoints, minimizing the distance of the related descriptors.

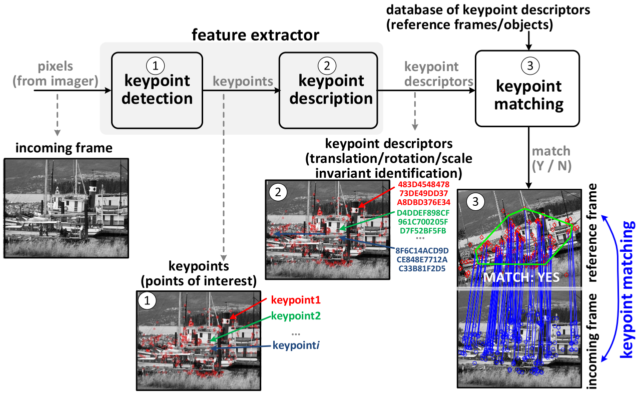

FIGURE 1. Feature extraction steps and subsequent keypoint matching.

The Scale Invariant Feature Transform (SIFT) is a popular keypoint detector and descriptor [6], [17]. Although it was proposed more than a decade ago, SIFT remains widely used because of its good matching performance. However, its circuit implementations are complex and hence area- and power-hungry, due to the need for massively parallel architectures [8], [9] (e.g., tens of mm^2, hundreds of mWs, including keypoint matching). A simplified feature extraction algorithm is SURF [16], [18]. A circuit demonstration of SURF was shown to reduce power down to a few mWs in [3].

As an alternative, the simplest keypoint detectors are corner detectors. A popular example is the Features from Accelerated Segment Test (FAST) [19]. A commonly used descriptor for FAST is the Binary Robust Independent Elementary Features (BRIEF) [20]. FAST detector and BRIEF descriptor involve significantly lower computational complexity compared to SIFT and SURF, and have been hence used for feature extraction at very high speed (e.g., 106fps with FHD resolution [8]). However, the FAST+BRIEF combination is neither scale nor rotation invariant, which is not acceptable in vision sensor nodes. Accordingly, the Oriented FAST Rotated BRIEF (ORB) algorithm was proposed in [15] as a FAST+BRIEF variant that is scale and rotation invariant.

ORB is based on FAST+BRIEF as in [15], [21], and adds non-maximal suppression (NMS) to reduce redundant keypoints and hence complexity, as in the FAST+BRIEF implementation in [8]. ORB is slightly more complex than FAST+BRIEF in [8], due to the addition of keypoint ranking to further reduce keypoints via importance-based selection, and the insertion of orientation/rotation sub-tasks to make the descriptor rotation-invariant. As well-known [15], [22], [23], and as confirmed by MATLAB and OpenCV experiments [24] with the publicly available benchmark in [25], ORB performs nearly as well as SIFT and SURF in terms of matching performance [26] (see example in Fig. 2).

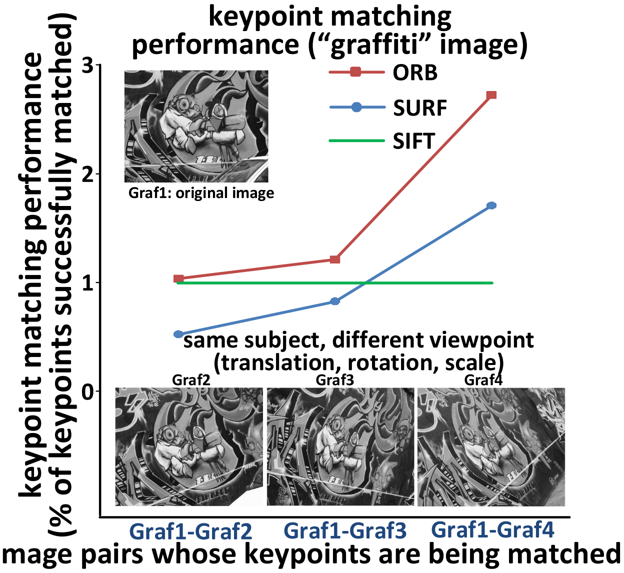

FIGURE 2. Example of keypoint matching on "graffiti" image [25] normalized to SIFT performance (i.e. percentage of keypoints in the original image that are matched in translated/rotated/scaled images).

III. ENERGY-QUALITY SCALABLE ARCHITECTURE FOR FEATURE EXTRACTION

The proposed architecture for the ORB-based feature extraction algorithm is depicted in Fig. 3 [27]. Keypoint detection is performed in three stages: Detector, NMS and Ranking. The Detector first identifies corners (see subsections below), which are then narrowed down to candidate keypoints in NMS, and reduced to actual keypoints in Ranking (Section IV). The successive keypoint description comprises of the Orientation and Descriptor (Section V). As will be discussed in the following section, the proper choice and insertion of EQ knobs in such fundamental blocks allows graceful quality degradation and significant energy reduction at given quality target. Architecturally, adaptation of the control flow is introduced to maintain correct operation under EQ knob tuning, while enabling an actual reduction in the number of cycles per frame when lower energy/quality is targeted.

A. KEYPOINT DETECTION: DETECTOR STAGE

Based on the ORB algorithm, the Detector block in Fig. 3 detects corners by comparing every pixel (interest point) with sixteen surrounding pixels placed circularly around it, as depicted in Fig. 4. Each of the sixteen pixels is preliminarily classified as either dark, light or grey, depending on the relative intensity of the pixel and the interest point. In ORB, the fixed *thresh* parameter defines this classification, as a pixel is considered dark (light) if its intensity is lower (higher) than the interest point by at least *thresh*. An interest point is a corner if there are at least 12 contiguous dark (light) pixels in the surrounding circle [15]. The resulting corner measure of the corner is defined as the sum of the magnitudes of differences between the surrounding pixels and the interest point [2], [6].

Conventionally, the *thresh* parameter is statically set to the fixed value of 20 in the ORB algorithm [15], whereas it is adjustable from 20 to 60 in EQSCALE. This enables EQ scaling at graceful quality degradation. Indeed, a larger *thresh* value discards more corners, decreasing the number of possible keypoints to be processed, execution time, and hence the overall energy across all blocks in Fig. 3. This is achieved at the cost of a mild quality degradation (see quantitative analysis in Section VI). Indeed, the matching quality between a reference image and its affine transformed image is marginally degraded, as long as a reasonably large number of keypoints are common to the two images, during object recognition or classification.[1] In the specific case of

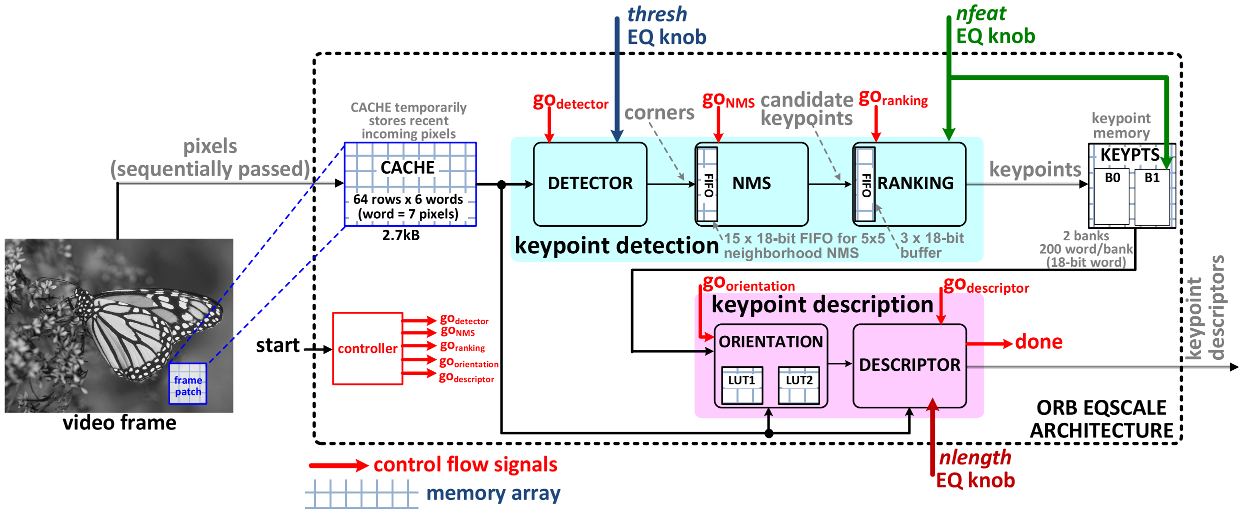

FIGURE 3. Proposed EQSCALE architecture for ORB, detailing feature extraction steps and subsequent keypoint matching.

[1] Feature extraction algorithms are generally conceived to be resilient against missing keypoints. Indeed, feature extraction must be robust against the inevitable presence of noise, which in turn corrupts keypoints.

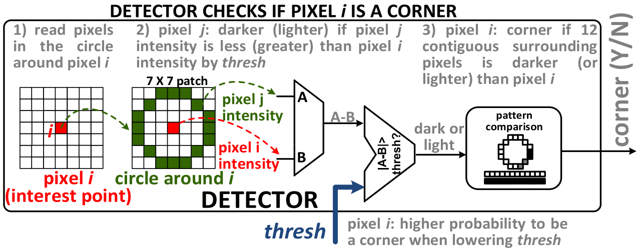
DETECTOR CHECKS IF PIXEL i IS A CORNER
1) read pixels in the circle around pixel i
2) pixel j: darker (lighter) if pixel j intensity is less (greater) than pixel i intensity by thresh
3) pixel i: corner if 12 contiguous surrounding pixels is darker (or lighter) than pixel i
7 X 7 patch
i
pixel i
(interest point)
circle around i
pixel j
intensity
pixel i
intensity
A
B
A-B
|A-B|>
thresh?
dark or
light
pattern
comparison
corner (Y/N)
DETECTOR
thresh
pixel i: higher probability to be a corner when lowering thresh

FIGURE 4. In keypoint detection, every interest point is compared with the surrounding 16 pixels, which are placed on a circle with 3-pixel radius. An interest point is classified as a corner if at least 12 contiguous surrounding pixels are dark or light.

the *thresh* knob, the number of generated keypoints at higher values is inherently reduced without modifying the control flow, and also reduces the number of cycles in later algorithm phases (e.g., description).

B. CACHE: MEMORY SIZE REDUCTION

The pixels and the interest points processed by the *Detector* stage are provided by the on-chip buffer *CACHE* memory in Fig. 1, which stores the incoming pixels. As feature extractors are generally memory dominated [9], [11], it is crucial to introduce techniques that reduce the *CACHE* energy and area. As a further challenge, the *Detector* stage turns out to be the performance (i.e., throughput) bottleneck in a straightforward implementation, as shown in Fig. 5. This is because most of other blocks turn out to be stalled most of the time, while waiting for the *Detector* outputs. To solve both challenges, the *CACHE* organization below was introduced to enable data reuse and *Detection* parallel operation, so that fewer pixels need to be kept in *CACHE* while processing pixels on the fly.

Regarding the *CACHE* organization, the number of read accesses was reduced by increasing the *CACHE* wordlength to seven 8-bit pixels per access (i.e., 56-bit word), corresponding to an entire row of contiguous pixels in the frame region fitted by the keypoint circle in Fig. 4. Indeed, the detection of a keypoint with lower wordlength would require more than one access per row, thus increasing the energy due to the repeated contribution of the *CACHE* decoder energy[2], as well as the added cycles on the remaining blocks. On the other hand, wordlength larger than seven pixels would consume more energy per keypoint without improving the throughput, since pixels exceeding the 7-pixel row width would not be used for the considered keypoint anyway.

To further reduce the number of read accesses and hence energy, read data reuse was introduced. In detail, *CACHE* stores a patch of the frame, where the oldest pixel is sequentially replaced one at a time, while receiving a new one. To reduce the *CACHE* energy, reuse of previous accesses was introduced by observing that each 7-pixel word contains part of up to 7 other interest points that are placed at horizontally adjacent locations, as shown in Fig. 5. Accordingly, the access of each word can be simultaneously used for the detection of 7 keypoints. This requires the simultaneous detection of 7 keypoints for each access, and hence a 7-replica parallel architecture for the *Detection* block (Fig. 5). A conventional architecture with single detection per access in Fig. 5 (left side) would require the same word to be accessed seven times, whereas the above data reuse scheme with parallel detection allow 7 simultaneous detections in 14 word accesses, i.e. two accesses per detection. As a result, the architecture in Fig. 5 enables 3.5X detection speedup and *CACHE* energy reduction, or equivalently 45% reduction in the overall energy.[3]

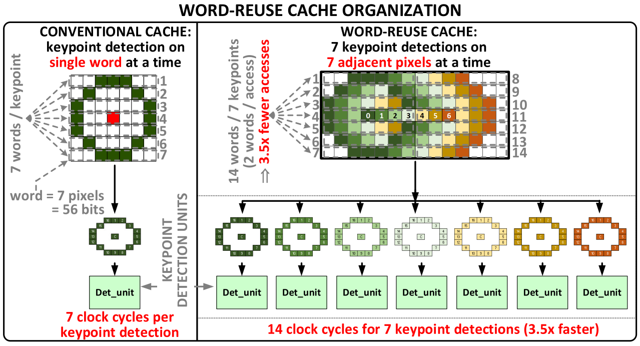

FIGURE 5. *CACHE* read data reuse through parallel detection, leveraging the fact that each word contains seven horizontally adjacent interest points that can be simultaneously computed.

To further reduce the *CACHE* size, the architecture was conceived to allow the simultaneous use of the same *CACHE* memory for both detection and description. This is achieved by optimizing the pipelined architecture and introducing simplifications to guarantee nearly-full overlap between detection and description in the common case. Such architectural optimization is statistical in nature, as the input rate of most blocks is image dependent. In detail, in Fig. 3 a *CACHE* row corresponds to 42 horizontally adjacent pixels. Among them, only 8.54% turn out to be corners on average, across images from the benchmark in [25]. Such percentage significantly varies across specific images, as shown in Fig. 6a. For each row, the number of detected corners ranges from 1 to 7 with the statistical distribution shown in Fig. 6a. From Fig. 6b, half of them (i.e., 4.34% of interest points) turn out to be candidate keypoints on average, and their specific percentage is again image dependent. Then, the interest points that turn out to be actual keypoints are selected by *Ranking*, which takes from 13 to 204 cycles per keypoint, and 61 cycles on average.

The interest points that are not corners, candidate keypoints or keypoints are not processed in the subsequent blocks, which are stalled by the controller in Fig. 3 as appropriate. Although needed to ensure correct operation, stalls are undesirable since the energy to keep the architecture running is consumed for a larger number of cycles, and limits opportunities to utilize time slack and hence voltage scaling.

[2] Indeed, compared to individual pixel access, consolidation of all 7 pixels in a single *CACHE* word permits to reduce the address bitwidth by two bits. Also, it permits to share the same access and hence decode energy among seven pixels, as opposed to individual pixel access. Similar considerations hold when accessing more than one and fewer than seven pixels at a time.

[3] Indeed, the *CACHE* energy in the proposed architecture accounts for 18% of the total energy after such 3.5X reduction (see Fig. 10).

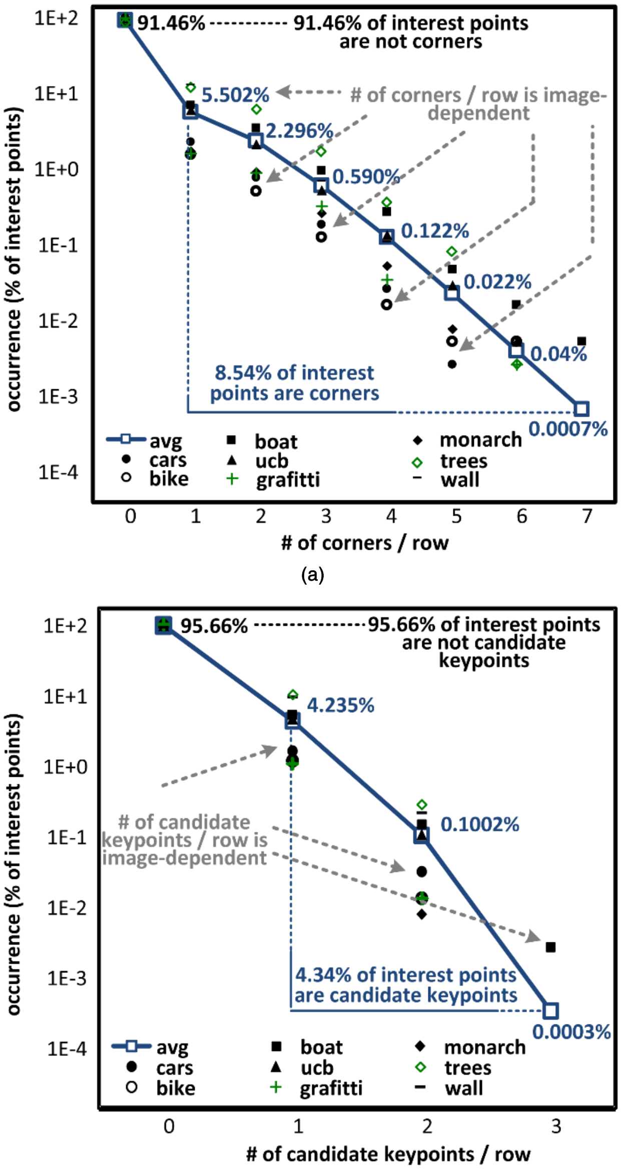

FIGURE 6. Statistical distribution of the number of interest points per *CACHE* row that turn out to be a) corners, b) candidate keypoints (based on images from benchmark in [25]).

In general, increasing the *CACHE* size increases the probability to successfully complete detection and description of keypoints without stalls, but leads to larger *CACHE* area and energy. Accordingly, *CACHE* was sized to the lowest capacity ensuring that computation for each keypoint is completed without any stall with 50% probability (i.e., with minimal number of cycles in the common case). This keeps energy and execution time lowest in the common case, while keeping memory area and energy reasonable, balancing computation and memory cost.

The above optimization permits to shrink the *CACHE* size down to 2.7kB, and hence substantially reduce both the overall energy and area. Indeed, a straightforward implementation would have needed the entire frame to be stored in *CACHE* [8], [9], requiring a much larger *CACHE* size of 300kB. In terms of organization, *CACHE* was arranged into 6 banks of 64 7-pixel words, which allow simultaneous access of different pixels for *Orientation* and *Descriptor*. This enables overlapped detection and description when they simultaneously need to access *CACHE*, while avoiding the extra complexity of dual-port bitcells. At the circuit level, *CACHE* is designed as a latch-based memory [28], to allow voltage scaling down to near-threshold like the logic circuits performing the detection and description.

IV. ARCHITECTURE FOR KEYPOINT DETECTION

The corners detected by the *Detector* are then filtered by the *NMS* block through non-maximal suppression as in Fig. 3, so that only one candidate keypoint is retained among the many similar corners that lie within the same neighborhood, due to how the FAST detection algorithm in ORB operates [15]. In detail, the corner measure coming from the *Detector* block is used to identify the most representative keypoint in each 5 x 5 pixel patch. This reduces the keypoints by 3X on average.

The *Ranking* block in Fig. 3 then sorts candidate keypoints according to their corner measure from the *Detector*, and retains only a specified number of keypoints with the highest corner measures (i.e., associated with the most important). Although it is fixed to 400 in ORB, this number was made adjustable to either 200 or 400 via the number of features/keypoints EQ knob *nfeat* (respective set to 0 or 1), allowing further EQ tradeoff. The lower value reduces the energy per frame since it halves the number of keypoints that are then accessed from the *KEYPTS* memory in Fig. 3 and then described. This comes at the cost of feature extraction quality degradation, as fewer keypoints are retained.

Interestingly, the choice of the *nfeat* knob assures once again graceful quality degradation, observing that the keypoints in *KEYPTS* are ranked by importance. Accordingly, the memory capacity is dynamically shrunk to retain the most important ones, whereas ignoring the least important leads to minor quality degradation. To translate the *nfeat* reduction into an actual reduction in the overall number of cycles per frame, the control flow of the architecture was made adaptive to minimize such number of cycles, according to the selected value of *nfeat*. In particular, when *nfeat* is set to 0 (i.e., 200 features are extracted), only one of the two available banks of the *KEYPTS* memory is used and accessed for comparison during Ranking. This reduces the worst-case number clock cycles taken by ranking by half, thus reducing the execution time and the energy by the same factor.

To reduce the complexity of *Ranking*, several sorting methods were considered, among which offline methods (e.g., *mergesort*) are the most computationally efficient with average complexity of $O(n \cdot log(n))$ [29]. However, for the targeted applications, offline sorting methods are impractical due to their high memory usage. Indeed, all 18-bit interest points need to be preliminarily stored in *KEYPTS*. In particular, the number of interest points was found to be easily

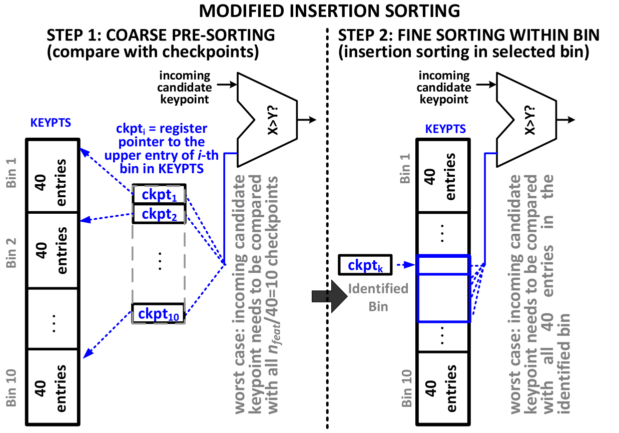
MODIFIED INSERTION SORTING
STEP 1: COARSE PRE-SORTING
(compare with checkpoints)
STEP 2: FINE SORTING WITHIN BIN
(insertion sorting in selected bin)
incoming candidate keypoint
X>Y?
KEYPTS
ckpt$_i$ = register pointer to the upper entry of *i*-th bin in KEYPTS
Bin 1
40 entries
Bin 2
40 entries
...
Bin 10
40 entries
ckpt$_1$
ckpt$_2$
ckpt$_{10}$
worst case: incoming candidate keypoint needs to be compared with all n_{feat}/40=10 checkpoints
incoming candidate keypoint
X>Y?
KEYPTS
ckpt$_k$
Identified Bin
Bin 1
40 entries
Bin 10
40 entries
worst case: incoming candidate keypoint needs to be compared with all 40 entries in the identified bin

FIGURE 7. **Proposed modified insertion sort with coarse pre-sorting and fine sorting, organizing the KEYPTS memory in 40-entry bins whose boundary address are specified by checkpoints. Each candidate keypoint is first binned in coarse pre-sorting, and then the relevant bin is sorted.**

in the order of thousands and up to 15,000, across benchmark images in [25]. This would require a *KEYPTS* capacity of hundreds of kb, which would lead to an unacceptably large area of 5X the area of the entire EQSCALE chip (see Section VI). Hence, memory and computation cost were rebalanced by resorting to the on-the-fly *insertion sort* algorithm [29]. This algorithm directly operates on incoming keypoint candidates, avoiding the need to preliminarily store interest points. In particular, *KEYPTS* needs to store only the final number of keypoints *nfeat*, thus reducing the *KEYPTS* capacity by at least an order of magnitude, compared to a conventional offline sorting method preliminarily storing the keypoint candidates (e.g., 200 or 400 as opposed to 15,000).

As downside of the adoption of *insertion sort*, its quadratic complexity rapidly increases the computational cost of sorting at larger number of keypoint candidates [29]. To mitigate its complexity, a modified two-stage *insertion sort* with coarse and fine sorting was adopted, as shown in Fig. 7. In the proposed modified insertion sort, *KEYPTS* is divided into 40- entry bins. The start and end points are tracked by checkpoints (i.e., pointer registers), which are stored in the *Ranking* block. The incoming keypoint candidates are first compared to checkpoints, thus executing a coarse pre-sorting (i.e., binning). Fine-grain insertion sorting is then performed only within the corresponding bin, thus reducing the number of items to be sorted at a time, and hence the computational cost. For each incoming candidate keypoint, this reduces the worst-case number of comparisons from *nfeat* down to *nfeat*/40+40 (see Fig. 7). This translates into an 8X energy reduction for the default value of *nfeat* $= 400$, compared to conventional insertion sorting.

V. KEYPOINT DESCRIPTION AND OVERALL MICROARCHITECTURE

A. KEYPOINT DESCRIPTION

Once ranked, keypoints are processed in the *Orientation* block in Fig. 3. According to the ORB algorithm, the latter computes the orientation of a 15x15 tile of pixels centered on the keypoint. The orientation is expressed in the form of moments as in (1a), where $I(x, y)$ the pixel intensity at (x,y) with the keypoint placed at (0,0). The orientation angle θ is computed as moment m_{01} (m_{10}) on the x (y) axis (atan2 is the quadrant-aware atan [15])

$$m_{pq} = \sum\nolimits_{x,y} x^p y^q I\left(x, y\right) \tag{1a}$$

$$\theta = atan2\left(m_{01}, m_{10}\right). \tag{1b}$$

As defined in ORB, the *Orientation* block contains a look-up table of pre-defined location pairs $P_j(x_1, y_1)$-$P_k(x_2, y_2)$ within a 31x31-pixel description patch (centered on the keypoint), which is used for comparison during description.

The 31x31-pixel description patch sets the minimum frame patch width stored in *CACHE* to 37 pixels per row (31 pixels + 6 other pixels for parallel detection). Indeed, the above 31-pixel width needs to be available for all the 7 simultaneously detected adjacent interest points (see Section III). Since the memory bank wordlength is 7 pixels, and considering that adjacent banks cover adjacent words, *CACHE* is organized in 6 banks to fit at least such 37 pixels. For a rotation-invariant description, these pairs are rotated using the transformation in (2) [15]

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{2}$$

The implementation of the trigonometric functions in (2) typically requires coarse and fine approximations such as in CORDIC [30], [31], thus requiring several cycles and a relatively complex hardware with few tens of kgates or larger [30]–[32]. To reduce the energy, the trigonometric functions in (2) were simplified by observing that a coarse evaluation of sinθsinθ and cosθcosθ is actually sufficient for ORB. Indeed, discretizing the angle from 0 to 360 degrees into 32 bins turned out to move the corresponding post-rotation pair location by at most 1 pixel away[4] from the exact location. Accordingly, rotation in (1)-(2) according to such coarse angle of $\pi/16$ was implemented with the simple look-up table in Table 1.

The intensities of the rotated pixel pairs $\mathrm{P}'_\mathrm{j}(\mathrm{x}'_1, \mathrm{y}'_1)$ and $\mathrm{P}'_\mathrm{k}(\mathrm{x}'_2, \mathrm{y}'_2)$ are compared in the *Descriptor* block as in (3a), generating the description vector $f_{nd}(p)$ in (3b)

$$\tau\left(p; j, k\right) = \begin{cases} 1 & \text{if } \mathrm{P}'_\mathrm{j} < P'_k \\ 0 & \text{otherwise} \end{cases} \tag{3a}$$

$$f_{nd}\left(p\right) = \sum\nolimits_{1 \leq i \leq \mathrm{nlength}} 2^{i-1} \cdot \tau\left(p; j_i, k_i\right) \tag{3b}$$

where p is the keypoint being described.

In (3b), the *nlength* EQ knob was inserted to define the number of pairs to be compared, or equivalently the descriptor length or bitwidth. Compared to the fixed ORB value of 256 (*nlength* $= 1$), adjusting *nlength* permits to reduce energy

[4] Detailed analysis showed that the maximum error in the evaluation of coordinates (x, y) of rotated coordinates monotonically decreases when increasing the number of such bins, and is equal to one under 32 bins.

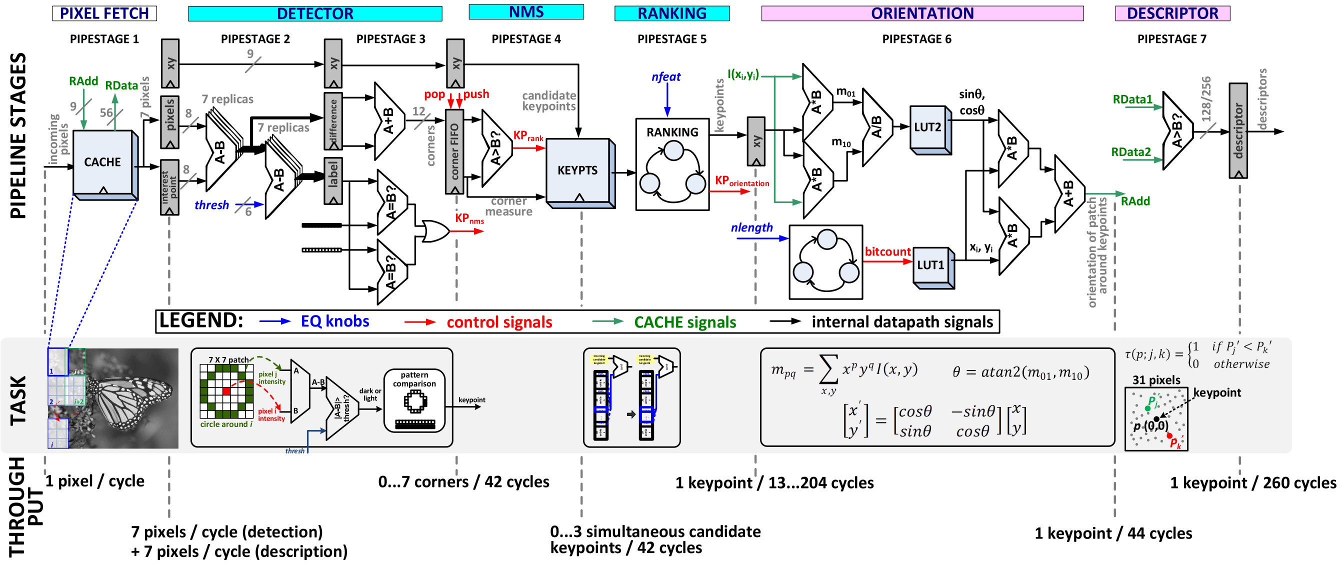

FIGURE 8. Detailed microarchitecture of the EQSCALE accelerator (top), and throughput at each stage (bottom).

TABLE 1. Lookup table for orientation bins to implement (2) (one quadrant is shown, and immediately extended to others).

θ	$cos\theta$	$sin\theta$
0	1	0
π/16	0.984375	0.195313
$\pi/8$	0.921875	0.382813
$3\pi/16$	0.828125	0.554688
$\pi/4$	0.7109375	0.710938
$5\pi/16$	0.5546875	0.828125
$3\pi/8$	0.3828125	0.921875
$7\pi/16$	0.1953125	0.984375

when lower quality is acceptable. Indeed, setting *nlength* = 0 reduces the number of comparisons in the *Descriptor* and related accesses to *CACHE* to 128, thus reducing their energy consumption and the overall execution time. This comes at the cost of a marginally lower quality, as a shorter descriptor (128-bit) reduces the distance (e.g.,Hamming, Euclidean) between the descriptors associated with different keypoints. This makes it marginally harder to discriminate pairs differing by few bits via thresholding, whereas discrimination remains straightforward in the majority of pairs, as they have larger difference. The control flow was again made adaptive to the value of *nlength*, to translate the *nlength* reduction in a reduced number of overall cycles per frame, and enable further voltage scaling opportunities described in Section VI.

B. SUMMARY OF OVERALL MICROARCHITECTURE

Based on the EQ scaling capabilities and simplifications enabled in Sections III-V, the detailed microarchitecture of the EQSCALE accelerator is shown in Fig. 8. It comprises seven stages, with *CACHE* and *KEYPTS* being synchronous (i.e., they lie at the boundary of pipestages). The critical path is in the *Detector* stage, whose logic depth is 155 *FO*4 (*FO*4 is the delay of an inverter gate with fan-out of 4). Fig. 8 also summarizes the resulting throughput of each block, which is expectedly image-dependent.

From a latency viewpoint, the complete processing of an interest point from fetching from *CACHE* to final description takes 337 clock cycles in the best case, 958 clock cycles in the worst case, and 407 clock cycles on average (i.e., 42 for *Detector* and *NMS*, 61 for *Ranking*, 44 for *Orientation*, and 260 for *Descriptor*). On the other hand, the 2.7-kB *CACHE* receives 1 pixel/cycle and hence takes 2,688 cycles, before a pixel is replaced. In other words, the microarchitecture is constructed in a way that pixels in *CACHE* are replaced only after the completion of the related computation, thus requiring only one download per pixel from the input. In turn, this allows to suppress the need for an off-chip buffer, whose energy contribution would easily become dominant. For example, an appropriate low-power low-cost LPDDR2 DRAM for mobile applications would consume 40-50pJ/bit [33] and hence 320-400pJ/pixel, which would be an order of magnitude higher than the energy achieved by EQSCALE, as discussed in Section VI.

The proposed mircroarchitecture meets the targeted maximum frame rate of 30fps at the nominal voltage, when EQ knobs are set for maximum quality. Operation at the minimum acceptable quality (see next section) allows 3X throughput excess at the same voltage, thus supporting the same frame rate down to near-threshold supply voltages (i.e., 0.5V). In turn, this throughput excess enabled by EQ scaling permits to further reduce the energy via voltage scaling, and adds further opportunities to reduce energy via coordinated EQ and voltage scaling, as discussed in Section VI.

VI. TESTCHIP DESIGN AND MEASUREMENTS

The proposed EQSCALE feature extraction accelerator was implemented in a 40nm testchip (see Fig. 9). Its overall area is 0.55 mm^2, 33% of which is due to core logic (i.e., *Detector*, *NMS*, *Ranking*, *Orientation*, *Descriptor*), 29% and 13% are

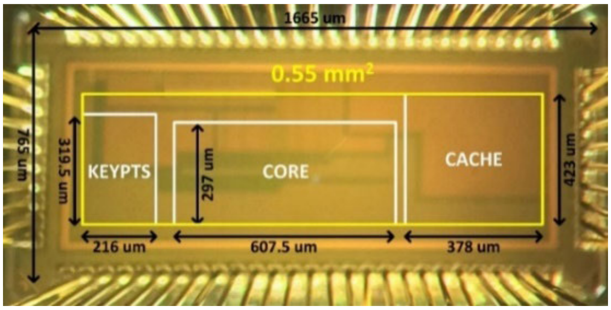

FIGURE 9. Chip microphotograph of the 40nm CMOS testchip.

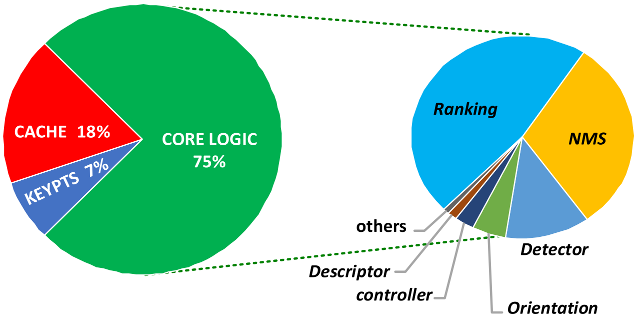

FIGURE 10. Power breakdown of EQSCALE at maximum quality.

respectively due to *CACHE* and *KEYPTS* memories, whereas the remaining 25% is due to setting storage, control, inter-module communication and testing harness. The EQSCALE power breakdown in Fig. 10 is dominated by the consumption of core logic, as opposed to previous chip demonstrations that were dominated by memory [8], [21].

The maximum operating frequency was evaluated across five dice at a supply voltage VDD ranging from 0.5V to 1V (see Fig. 11). The subsequent plots for a single die refer to die #4 unless otherwise specified,[5] as it is the closest to the average across dice, as indicated by solid line. Real-time operation at 30 frames per second is achieved at an energy per pixel of 366pJ at 1V, under EQ knobs tuned for maximum quality. In applications where lower frame rate is acceptable, pure voltage scaling down to 0.5V reduces energy to 107pJ/pixel at 7fps frame rate, as shown in Fig.12. From Fig. 11, sub-mW power consumption is achieved for $V_{DD} < 0.7$V.

Energy-quality scaling offers significant opportunities to further reduce energy, compared to the traditional trade-off between energy and frame rate via voltage scaling. Fig. 12 shows the energy E associated with the computation per pixel, which is normalized to its maximum value (obtained under nominal V_{DD} and EQ knobs tuned for best quality). Quality is defined by routinely assuming that the feature extractor is followed by a keypoint matching engine, as described in Fig. 1. Accordingly, quality Q is defined as the number of correct keypoint matches divided by their total number, when comparing a transformed image (or object) and the original one [1]. In the following, Q is normalized to its best value, which corresponds to the EQ knob configuration for maximum quality. Q was evaluated by post-processing the keypoints generated by the testchip, running keypoint matching based on the popular 3-nearest neighbors algorithm in software [34]. The RANSAC algorithm [26] was then run to identify the correct matches and evaluate Q.

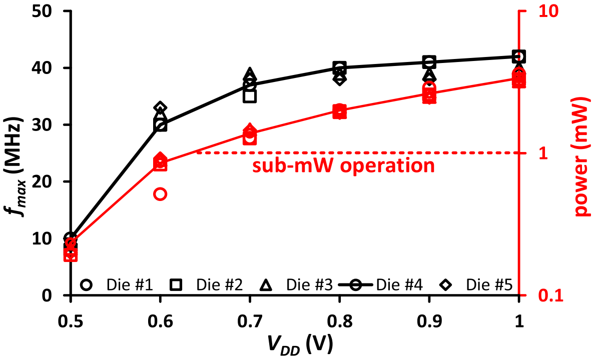

FIGURE 11. Maximum frequency f_{max} and power vs V_{DD} for 5 dice with EQ knobs set for maximum quality (f_{max} at nominal voltage limited by testing harness, which was designed for low-voltage testing with VGA video stream).

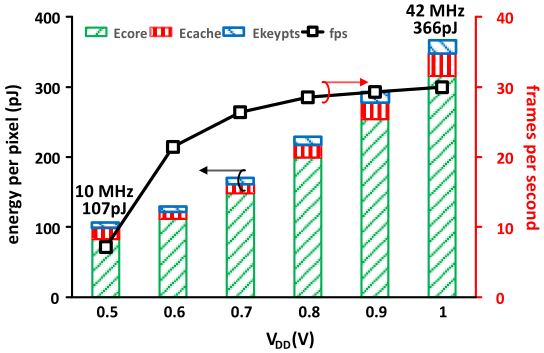

FIGURE 12. Energy per pixel (using average chip data) for maximum quality at VGA resolution, with varying frame rate and supply voltage.

Regarding the range of acceptable quality, correct object detection is expected when Q is sufficiently high, as this means that feature extraction can provide an adequate number of correct and useful keypoints describing the image. Based on the benchmark in [25], correct image matching was consistently found to be correctly performed if $0.4 < Q < 1$, across the above mentioned benchmark. As exemplified in Fig. 13, the bounding box around the matched object starts keypoints are being missed. For $Q < 0.4$, correct recognition shrinking below $Q = 0.4$, as a clear sign that some important keypoints are being missed. For $Q < 0.4$, correct recognition is no longer assured for all images, and the bounding box is progressively shrunk when Q is reduced down to 0.12, below which it completely disappears.

The effect of individual EQ knobs on energy and quality is plotted in Fig. 14. From this figure, reducing the length of the descriptor from 256 to 128 bits ($nlength = 0$) reduces the energy by 34% with 10% quality degradation, compared to the maximum-quality point. An increase in *thresh* from

[5] Only die #1 was tested in [27], which turned out to be significantly worse than all others in terms of process variations, as shown by its higher minimum V_{DD} (0.6V) in Fig. 11.

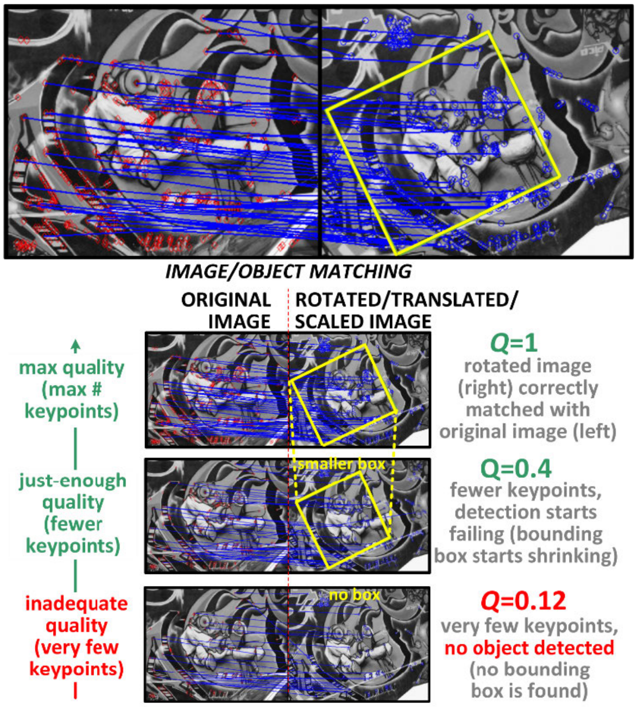
IMAGE/OBJECT MATCHING
ORIGINAL IMAGE
ROTATED/TRANSLATED/ SCALED IMAGE
max quality (max # keypoints)
just-enough quality (fewer keypoints)
inadequate quality (very few keypoints)
Q=1
rotated image (right) correctly matched with original image (left)
smaller box
Q=0.4
fewer keypoints, detection starts failing (bounding box starts shrinking)
no box
Q=0.12
very few keypoints, no object detected (no bounding box is found)

FIGURE 13. Illustration of image matching at different values of Q. Top image shows original image (left) keypoints in red triangle and transformed image (right) keypoint in blue circles. Blue limes connect matched keypoints between original and transformed image.

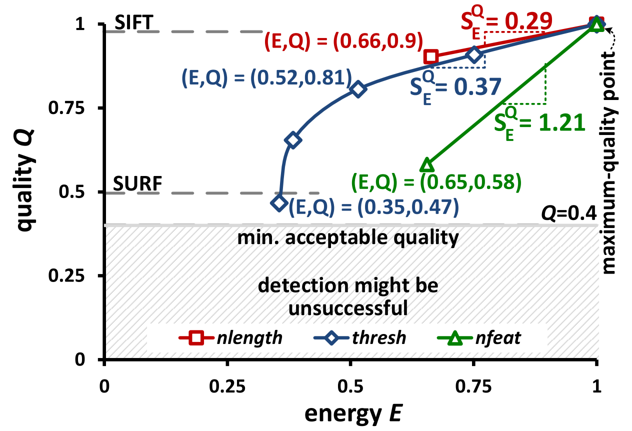

FIGURE 14. EQ tradeoff by adjusting individual EQ knobs at nominal voltage (VGA resolution, 30fps frame rate, V_{DD} = 1V).

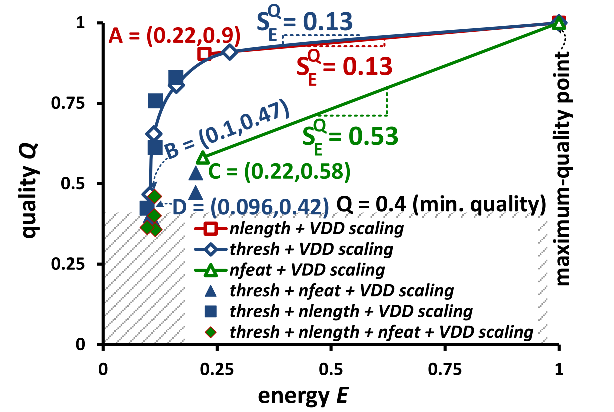

FIGURE 15. EQ tradeoff with joint energy-quality and voltage scaling (VGA resolution, 30fps frame rate).

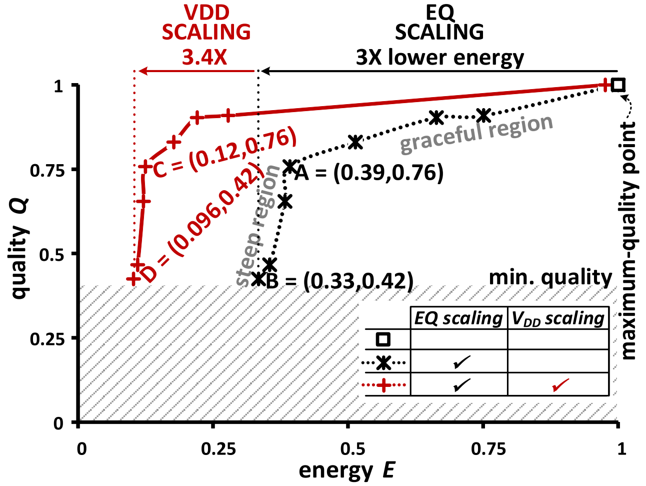

FIGURE 16. Energy-optimal curve from Figs. 14-15 under EQ scaling at nominal V_{DD} (30fps, black curve), and additional V_{DD} scaling enabled by simultaneous EQ and V_{DD} scaling (30fps, red curve).

20 to 40 (60) reduces the energy by 48% (65%) at 19% (53%) quality degradation. The reduction in the number of keypoints from 400 (*nfeat* = 1) to 200 (*nfeat* = 0) reduces the energy by 35%, at 42% lower quality. Such energy saving is enabled by the lower cycle count due to EQ knob adjustment and architectural adaptation, as clarified in Section III-V. The graceful quality degradation confirms the appropriate choice of such EQ tuning knobs.

The effectiveness of individual EQ knobs is quantified by the quality-energy sensitivity S_E^Q, i.e. the ratio of the relative quality change and relative energy change when adjusting the corresponding knob. Lower sensitivity indicates more graceful quality degradation for a given energy benefit. From Fig. 14, the most effective knob is *nlength* with $S_E^Q = 0.29$, followed by *thresh* with $S_E^Q = 0.37$. The *nfeat* EQ knob has 3X larger sensitivity and hence less effective than *nlength* and *thresh*.

Further energy benefits are offered by joint EQ and voltage scaling, as shown in Fig. 15. Indeed, as discussed in Sections III-V, the energy-quality reduction via EQ knob tuning and architectural adaptation reduce the number of execution cycles for feature extraction across a frame at a given frame rate, creating a task-level timing slack. For a fixed frame rate, reducing the number of execution cycles by a factor X allows to run at a clock frequency reduction by the same factor, without missing any necessary operation. Hence, energy-quality scaling adds further opportunities for aggressive voltage scaling while maintaining the same frame rate, as opposed to conventional voltage scaling that inevitably degrades the frame rate when energy is lowered.

Under joint adjustment of a single EQ knob and V_{DD}, the sensitivity of *nlength*, *thresh* and *nfeat* is respectively improved by 2.2X, 2.8X and 2.3X. As a result, joint tuning of *nlength* and V_{DD} allows 78% energy reduction at 10%

TABLE 2. Comparison (best result in bold).

	JSSC'13 [9]	TCASVT'16 [8]	JSSCC'14 [3]	this work
algorithm	SIFT	FAST+ BRIEF	SURF	ORB
technology	0.13 µm	65 nm	28 nm	40 nm
supply voltage	0.65-1.2V	1.2V	0.47V	0.5-1V
on-chip memory	382kB SRAM	128kB SRAM	~7kB FIFO	**~4kB** latch-based
clock	50~200MHz	200MHz	27MHz	8-45MHz
energy-quality scalable	NO	NO	NO	**YES**
frame rate	30fps	**94.3fps**	30fps	5 - 30 fps
power	320mW	185mW	2.7mW	**0.082 [d] (0.326 [e]) - 3.4 [f] mW**
energy/pixel	11,570pJ	930pJ	293pJ	**35.4 [d] - 366 pJ**
normalized energy/pixel [a]	3,970pJ	319pJ	419pJ	**35.4 [d] - 366 pJ**
area (mm^2)	32	6.76	2.22	**0.55**
normalized area [b] ($F^2/10^6$)	1,893	1,600	2,831	**344**
FOM [c]	617	42	97	**1 - 10**
on-chip operation	matching with external database	matching with descriptor cache	feature extraction up to description	feature extraction up to description

[a] Normalized to 40nm assuming 0.7X lower energy per CMOS generation (VGA)
[b] F = min. feature size of considered technology
[c] FOM=normalized area * normalized energy (energy is normalized to EQSCALE at min. acceptable quality and iso-technology)
[d] Obtained with joint EQ, V_{DD} and performance scaling (5fps)
[e] Obtained with joint EQ and V_{DD} scaling (V_{DD} adjusted for 30fps)
[f] Obtained with no EQ scaling at nominal V_{DD} (30fps)

quality degradation (point A in Fig. 15), compared to the maximum-quality point. Similarly, an increase in *thresh* to 60 (decrease in *nfeat* to 200) reduces the energy by 90% (56%) at 54% (42%) quality degradation, with reference to point B (C) in Fig. 15.

When the best combination of multiple EQ knobs and V_{DD} with lowest energy for a given quality is chosen from Figs. 14-15, the energy-optimal EQ curve in Fig. 16 is obtained. From this figure, pure EQ scaling allows an energy reduction of up to 3X compared to the maximum-quality point (point B in Fig. 16). Joint EQ and V_{DD} scaling allow an additional energy reduction up to 3.4X (point D in Fig. 16). In other words, the additional voltage scaling enabled by EQ scaling at iso-frame rate leads to an energy benefit that is nearly the same as the EQ scaling itself, leading to an overall 10X reduction, a minimum energy of 35.4pJ/pixel and 326μW power consumption. In applications where a reduced frame rate is acceptable, further power saving is allowed by combining conventional power-frame rate scaling and EQ scaling. For example, frame rate reduction down to 5fps was found to lead to a power consumption of 82μW.

To better understand the benefits brought by EQ scaling, Table 2 shows the energy compared to prior art at iso-technology. Without EQ scaling (i.e., at the maximum-quality point), is close to [8], as expected from the comparable complexity of the ORB and the FAST-BRIEF algorithms [15]. At the same maximum-quality point, VGA resolution and 30fps, the power of the proposed architecture is comparable to the previous best in class [3]. From the above considerations, the energy efficiency of the baseline architecture without EQ scaling is comparable to best-in-class demonstrations in prior art. On the other hand, energy-quality scaling enables an energy/bit reduction from hundreds of pJs [3], [8], [9] down to few tens of pJs, as shown in Table 2. Compared to the demonstration with lowest consumption without keypoint matching [3], EQSCALE achieves up to 11.2X energy reduction at 30fps at iso-technology (see Table 2). EQ scaling also enables 10X energy reduction compared to the maximum-quality point (Fig. 16).

From Table 2, EQ scaling enables deep sub-mW feature extraction for the first time, being power reduced to 326μW at 30fps (80μW at 5fps), which is 8.3X (33X) lower than 2.7mW exhibited by the previous best in class [3]. The area of EQSCALE at iso-technology is 4.7X lower than the smallest area reported [8], and 8.2X lower than [3]. As metric relevant to self-powered low-cost vision sensor nodes, the area-energy product of EQSCALE is improved by 97X compared to [3], which had the lowest power in prior art and very similar capabilities (see last row in Table 2).

VII. CONCLUSION

In this paper, an energy-quality scalable memory-frugal architecture for video feature extraction based on ORB has been presented. Properly selected tuning knobs were introduced to allow flexible and dynamic tradeoff between energy and quality, leveraging the inherent noise resiliency of vision applications. EQ scaling is shown to simultaneously allow cycle count reduction and V_{DD} scaling, and hence energy. Measurements from 40nm testchips show that the proposed memory-frugal architecture with joint EQ and V_{DD} scaling enables a 4.7-8.2X area reduction compared to prior art, when scaled at the same technology. Power consumption is reduced down to 82-326μW, which is 8.3-33X lower than the state of the art. At the same time, conventional operation at maximum quality is preserved.

To the best of the authors' knowledge, the proposed architecture enables deep sub-mW and sub-mm2 feature extraction for the first time. Accordingly, the proposed EQ- scalable architecture is well suited for vision systems with tightly-constrained consumption and area, such as vision sensor nodes and always-on cameras.

ACKNOWLEDGMENT

The authors acknowledge the support from TSMC for chip fabrication.

REFERENCES

[1] R. Szeliski, *Computer Vision—Algorithms and Applications*. New York, NY, USA: Springer, 2011.
[2] S. Krig, *Computer Vision Metrics—Survey, Taxonomy and Analysis*. New York, NY, USA: Apress, 2014.
[3] D. Jeon, M. B. Henry, Y. Kim, I. Lee, Z. Zhang, D. Blaauw, and D. Sylvester, "An energy efficient full-frame feature extraction accelerator with shift-latch FIFO in 28 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 49, no. 5, pp. 1271–1284, May 2014.

[4] M. Verhelst. *Deep Learning Processor Survey*. Accessed: Oct. 1, 2019. [Online]. Available: http://www.esat.kuleuven.be/~mverhels/DLICsurvey.html

[5] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, Mar. 2019.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[7] M. Alioto, *Enabling the Internet of Things-From Integrated Circuits to Integrated Systems*. Cham, Switzerland: Springer, 2017.

[8] J.-S. Park, H.-E. Kim, H.-Y. Kim, J. Lee, and L.-S. Kim, "A vision processor with a unified interest point detection and matching hardware for accelerating stereo matching algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 12, pp. 2328–2343, Jun. 2016.

[9] J. Oh, G. Kim, J. Park, I. Hong, S. Lee, J.-Y. Kim, J.-H. Woo, and H.-J. Yoo, "A 320 mW 342 GOPS real-time dynamic object recognition processor for HD 720p video streams," *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 33–45, Jan. 2013.

[10] A. Abbo, R. Kleihorst, V. Choudhary, L. Sevat, P. Wielage, S. Mouy, and M. Heijligers, "XETAL-II: A 107 GOPS, 600mW massively-parallel processor for video scene analysis," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2007, pp. 600–602.

[11] Y. Pu, "From Xetal-II to Xetal-Pro: On the road toward an ultralow-energy and high-throughput SIMD processor," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 4, pp. 472–484, Apr. 2011.

[12] P. W. On. *Products–Security Cameras & Surveillance Systems*. Accessed: Jun. 20, 2019. [Online]. Available: https://security.panasonic.com/products

[13] M. Alioto, "Energy-quality scalable adaptive VLSI circuits and systems beyond approximate computing," in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Lausanne, Switzerland, Mar. 2017, pp. 127–132.

[14] M. Alioto, V. De, and A. Marongiu, "Energy-quality scalable integrated circuits and systems: Continuing energy scaling in the twilight of Moore's Law," *IEEE Trans. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 4, pp. 653–678, Dec. 2018.

[15] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[16] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2006, pp. 404–417.

[17] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 2, 1999, pp. 1150–1157.

[18] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[19] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988. pp. 147–151.

[20] M. Calonder, V. Lepetit, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. Lect. Notes Comput. Sci.*, 2010, pp. 778–792.

[21] J.-S. Park, H.-E. Kim, and L.-S. Kim, "A 182 mW 94.3 f/s in Full HD pattern-matching based image recognition accelerator for an embedded vision system in 0.13-μm CMOS technology," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 5, pp. 832–845, May 2013.

[22] J. Hartmann, J. H. Klussendorff, and E. Maehle, "A comparison of feature descriptors for visual SLAM," in *Proc. Eur. Conf. Mobile Robots*, Sep. 2013, pp. 56–61.

[23] E. Karami, S. Prasad, and M. Shehata, "Image matching using SIFT, SURF, BRIEF and ORB: Performance comparison for distorted images," Oct. 2017, *arXiv:1710.02726*. [Online]. Available: https://arxiv.org/abs/1710.02726

[24] G. Bradski. (2000). *OpenCV*. Accessed: Feb. 11, 2015. [Online]. Available: https://opencv.org

[25] K. U. Leuven, I. Rhone-Alpes, and C. F. M. Perception. *Affine Covariant Features*. Accessed: Apr. 4, 2016. [Online]. Available: http://www.robots.ox.ac.uk/~vgg/research/affine/

[26] M. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applicatlons to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[27] A. B. Alvarez, G. Ponnusamy, and M. Alioto, "EQSCALE: Energy-quality scalable feature extraction engine for sub-mW real-time video processing with 0.55 mm^2 area in 40nm CMOS," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2017, pp. 241–244.

[28] P. Meinerzhagen, O. Andersson, B. Mohammadi, Y. Sherazi, A. Burg, and J. N. Rodrigues, "A 500 FW/bit 14 FJ/bit-access 4kb standard-cell based sub-VT memory in 65nm CMOS," in *Proc. ESSCIRC*, Sep. 2012, pp. 321–324.

[29] D. E. Knut, *The Art of Computer Programming*, vol. 3, 2nd ed. Boston, MA, USA: Addison Wesley, 1998.

[30] S. Rajan, S. Wang, and R. Inkol, "Efficient approximations for the four-quadrant arctangent function," in *Proc. Can. Conf. Electr. Comput. Eng.*, 2006, pp. 1043–1046.

[31] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron. Comput.*, vols. EC–8, no. 3, pp. 330–334, Sep. 1959.

[32] D. D. Hwang, D. Fu, and A. N. Willson, "A 400-MHz processor for the conversion of rectangular to polar coordinates in 0.25-μm CMOS," *IEEE J. Solid-State Circuits*, vol. 38, no. 10, pp. 1771–1775, Oct. 2003.

[33] K. T. Malladi, F. A. Nothaft, K. Periyathambi, B. C. Lee, C. Kozyrakis, and M. Horowitzy, "Towards energy-proportional datacenter memory with mobile DRAM," in *Proc. ISCA*, Portland, OB, USA, Jul. 2012, pp. 37–48.

[34] F. E. Uzyildirim and M. Özuysal, "Instance detection by keypoint matching beyond the nearest neighbor," *Signal, Image Video Process.*, vol. 10, no. 8, pp. 1527–1534, 2016.

ANASTACIA ALVAREZ (Senior Member, IEEE) received the B.S. degree in electronics and communications engineering and the M.S. degree in electrical engineering from the University of Philippines, Diliman, in 1998 and 2004, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, National University of Singapore, under the scholarship from the ERDT Faculty Development Program of the Department of Science and Technology, Philippines. She is currently a Professor with the Electrical and Electronics Engineering Institute, University of Philippines, Diliman. Her research interest includes energy efficient VLSI systems.

GOPALAKRISHNAN PONNUSAMY (Member, IEEE) received the B.Eng. degree in electronics and communications engineering from Anna University Coimbatore, India, in 2011, and the M.Sc. degree in electrical and computer engineering from the National University of Singapore, in 2014. He is currently working in Rockwell Automation, Singapore. His interests are in firmware development and embedded systems.

MASSIMO ALIOTO (Fellow, IEEE) received the Laurea (M.Sc.) degree in electronics engineering and the Ph.D. degree in electrical engineering from the University of Catania, Italy, in 1997 and 2001, respectively.

He held positions at the University of Siena, Intel Labs—CRL, in 2013, University of Michigan Ann Arbor, from 2011 to 2012, BWRC, University of California, Berkeley, from 2009 to 2011, and EPFL, Switzerland, in 2007. He is currently with the Department of Electrical and Computer Engineering, National University of Singapore, where he leads the Green IC group and is the Director of the Integrated Circuits and Embedded Systems area. He has authored or coauthored more than 270 publications. He is the coauthor of three books, including *Enabling the Internet of Things—from Circuits to Systems* (Springer, 2017). His primary research interests include self-powered wireless integrated systems, near-threshold circuits, widely energy-scalable systems, data-driven integrated systems, and hardware-level security, among the others.

Dr. Alioto is/was a Distinguished Lecturer, a member of the Board of Governors of the CASS Society, and the Chair of the VLSI Systems and Applications Technical Committee. He is/was the Technical Program Chair in several conferences (e.g., ISCAS 2023, SOCC, ICECS, and NEWCAS) and a TPC member (ISSCC and ASSCC). He served as a Guest Editor of several IEEE journal special issues (e.g., TCAS-I, TCAS-II, and JETCAS) and as an Associate Editor. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON VLSI SYSTEMS and the Deputy Editor-in-Chief of the IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS.