Computer Vision with a Superpixelation Camera

Sasidharan Mahalingam
Portland State University
samahali@pdx.edu

Rachel Brown*
Willamette University
rabrown3@willamette.edu

Atul Ingle*
Portland State University
atul.ingle@pdx.edu

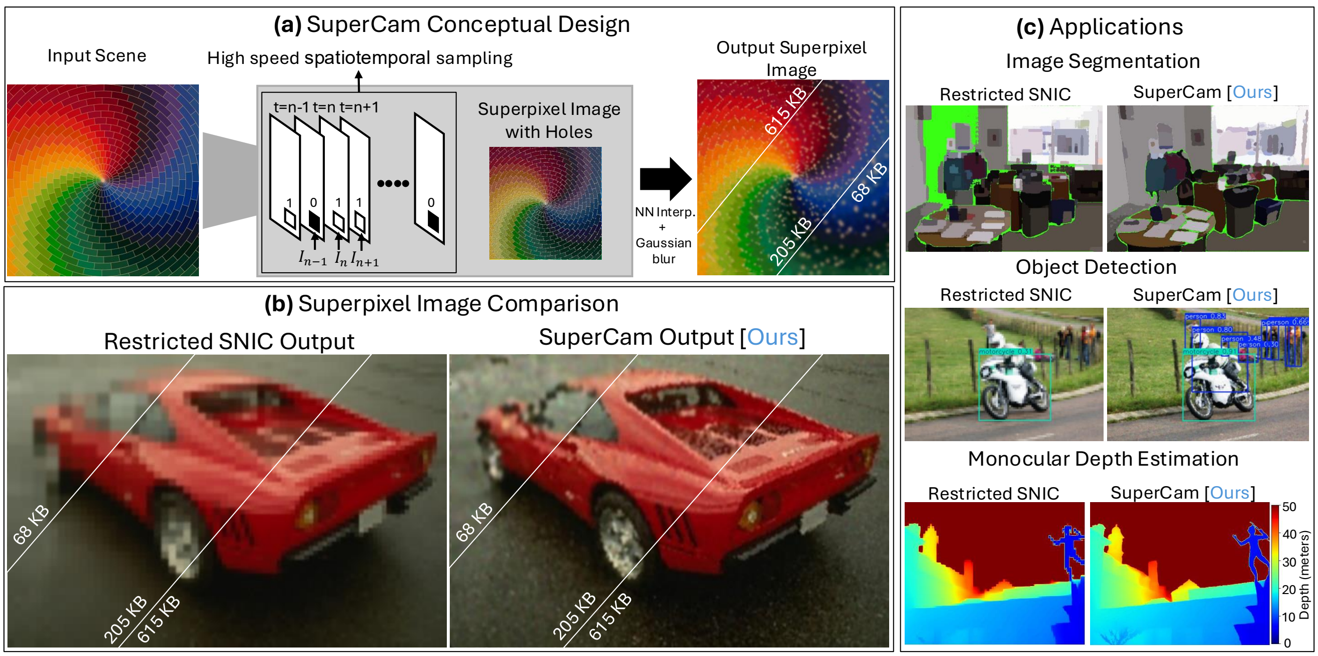

Figure 1. **Summary of SuperCam design, output, and applications.** We propose a new camera design that generates superpixel images on the fly without storing/capturing the full input image. **(a) SuperCam Conceptual Design.** The proposed design, which is implemented as a passive Single Photon Avalanche Diode (SPAD) Sensor. We adaptively sample the exposed pixels to generate a sparse superpixel image. This processing happens on-sensor. This sparse data is read out and filled off-sensor using nearest-neighbor interpolation and Gaussian blur to generate the superpixel image. **(b) Superpixel Image Comparison.** As SuperCam generates the superpixel image from sparse data on the fly, it performs better than the conventional superpixel algorithms under resource constrained scenarios. Shown are sample images with different memory settings; 68KB, 205KB and 615KB. **(c) Applications.** Three computer vision applications comparing SuperCam with a memory-constrained implementation of SNIC (Image Segmentation using SAMv2, Object Detection using YOLOv12 and Monocular Depth Estimation using DepthAnythingv2).

Abstract

Conventional cameras generate a lot of data that can be challenging to process in resource-constrained applications. Usually, cameras generate data streams on the order of the number of pixels in the image. However, most of this captured data is redundant for many downstream computer vision algorithms. We propose a novel camera design, which we call SuperCam, that adaptively processes captured data by performing superpixel segmentation on the fly. We show that SuperCam performs better than current state-of-the-art superpixel algorithms under memory-constrained situations. We also compare how well SuperCam performs when the compressed data is used for downstream computer vision tasks. Our results demonstrate that the proposed design provides superior output for image segmentation, object detection, and monocular depth estimation in situations where the available memory on the camera is limited. We posit that superpixel segmentation will play a crucial role as more computer vision inference models are deployed in edge devices. SuperCam would allow computer vision engineers to design more efficient systems for these applications.

* Equal contribution.

This material is based upon work supported by NSF ECCS-2138471; and the U.S. Department of Energy, Office of Fusion Energy Sciences, under Award Number DE-SC0025802.

1. Introduction

The notion of an image as a collection of square shaped pixels arranged in a uniform grid is a *fait accompli* that both classical and modern (deep-learning-based) computer vision system are built on. In this paper we ask: what if the raw "images" coming off an image sensor were instead loosely-grouped clusters with arbitrary shapes adapted to the scene being imaged? We call this hypothetical image sensor a "superpixelation camera" or SuperCam for short.

There is a long line of work on representing images as superpixels, groups of pixels that are usually clustered based on pixel intensity or other nuanced measures of pixel similarity. These superpixel segmentation algorithms start with the original high resolution image and generate a superpixel simplification representation from that high resolution image. A SuperCam is quite different; it is not merely a postprocessing algorithm that creates a parsimonious image representation from a high resolution image. We envision a sensor that avoids capturing the high resolution image altogether and directly outputs superpixels (Fig. 1(a-b)).

The motivation for this is partly academic and partly driven by the need for more resource-efficient image sensors. A superpixel approach more closely mimics how biological vision systems work—the human eye, for example, does not really form a high resolution image like a camera. Edge detection and grouping of perceptually similar scene regions happens as early as the retina itself, quite early in the visual processing chain. By contrast, image sensors contain hundreds of millions of pixels packed into a tiny area, consuming power and requiring large bandwidth only for most of it to be simply thrown away later[1].

We do not and cannot aim for raw image quality with SuperCam, so we eschew image quality metrics in favor of results showing direct applications. Our results demonstrate that with SuperCam images, it is still possible to perform many computer vision tasks, including semantic segmentation, object detection, and monocular depth estimation, with a significant reduction in memory requirements over using a conventional image (Fig. 1(c)). Our instantiation of SuperCam also comes with a "tuning knob" that can be adjusted to smoothly trade-off fidelity for memory requirements in these computer vision tasks as needed. We show results at different "memory settings" and show a graceful increase in performance (measured by the task-specific quality metrics) as the number of superpixels, and the on-sensor memory needed to store them, is increased.

2. Related Work

Because SuperCam is a novel superpixel algorithm built atop an unconventional camera using a sparse data stream, our related work explores both existing superpixel algorithms, methods for compressive sensing, and other unconventional vision sensors.

Superpixel algorithms. Superpixel segmentation, first introduced by Ren and Malik [49], refers to segmenting an image into smaller regions or clusters that are locally similar while preserving the overall coherence of the image. We refer the reader to [5, 49] for more detailed surveys of the field. Superpixel images can be very useful for computer vision tasks because they reduce the complexity of the input, simplifying the search space. Many superpixel algorithms have been proposed to increase efficiency and improve output quality for various computer vision tasks: semantic segmentation [3, 28, 46, 71, 73], 3D reconstruction [6, 30, 41, 49, 52], object detection [37, 45, 49, 50, 66], depth estimation [9, 10, 12, 26, 49], tracking [27, 49, 61, 62, 68] and optical flow [20, 22, 33, 49, 74].

Although there have been many recent technological advances in superpixel segmentation, many of these techniques have significant memory requirements that prohibit their use in lightweight edge-computing applications. This includes algorithms that rely on learned features [43, 63], unsupervised end-to-end networks [23, 75], and transformer-based solutions [34]. There are also a number of superpixel algorithms which can be light-weight but require the entire input image as input, such as graph-based [36, 58], clustering-based [1, 2] and energy-based [11, 32, 56] algorithms, as well as recent Bayesian techniques [17, 55]. By contrast, we do not have access to the entire image and would instead like to create a sensor whose raw data output is the superpixel information.

We therefore consider clustering-based algorithms such as SNIC [1] and SLIC [2] to be the most comparable methods to our work. In SNIC [1], segmentation is performed by adding and removing pixel elements from a queue until all pixels are labeled. The feature space used to create the queue is derived from a weighted combination of pixel distances and intensities. Similarly, SLIC [2] clusters pixels in a combined five-dimensional color and image plane space to efficiently generate compact, nearly uniform superpixels. Of these methods, we compare against SNIC [1] as it is widely used, provides the lowest error metrics among the non-learning based methods [24, 63, 64, 75], and meets all the necessary criteria for our input data stream.

Single-pixel and compressive image sensing. There is a rich line of work on single-pixel image sensing [19] starting with seminal work on compressed-sensing-based single-pixel camera that captures random linear projections and still recovers the original image [14]. Our motivation with SuperCam is related in the sense that we aim to capture parsimonious representations of the original scene. However, our end goal is different because we focus on performance metrics that have less to do with the image reconstruction

[1] Interestingly, the inventor of the pixel lamented his decision to make them square-shaped [15].

quality, but more to do with the downstream computer vision tasks.

Unconventional vision sensors. Image sensors with in-pixel and on-sensor compute capabilities can be used to implement intelligent vision sensing on edge devices. Such compute capabilities have been demonstrated with both traditional CMOS-type pixels through "pixel processor arrays" for realtime feature tracking [7, 70], single-photon-sensitive pixel arrays with in-pixel compute [4], and sensors with extremely low spatial resolution, coupled with a learning-based approach for specific vision tasks [29]. We envision future hardware implementations of SuperCam can leverage these advances in reconfigurable and reprogrammable image sensors. Event sensors reduce data rate by only transmitting intensity changes [16]. Our work is the spatial counterpart of event cameras—whereas event cameras transmit information when the *temporal* change is large, SuperCam reduces the output data volume by only storing superpixel information for *spatial* regions that are perceptually distinct in color and intensity information.

3. Superpixelation Camera: Imaging Model

In this section we present an abstract model for capturing scene information using a superpixelation camera. Since such a camera does not yet exist, next we present a practical method for simulating (purely on a computer) and emulating (using existing camera hardware) one. Instead of storing an image as a regular grid of pixel values, a SuperCam maintains an internal data structure of segment information in the form of segment boundaries and a single 8- or 24-bit intensity and color value associated with each segment. Previous works [1, 2, 24, 34, 60, 64] have used superpixels in the range of 1000 to 2000 for images from the BSD500[40] dataset, which are 321×481 in dimension. Unlike conventional images that require millions of pixels with 8- or 24-bit (intensity or color) information, the SuperCam data structure needs orders of magnitude lower memory.

Let $\mathcal{S} = (S_i, I_i)_{i=1}^{N}$ denote the set of N segments maintained by the SuperCam's internal, on-sensor data structure. Here S_i denotes the boundary information and I_i denotes the intensity information associated with the i^{th} superpixel. The image sensor acquires scene information by integrating photons (over a pre-specified exposure duration) at some arbitrarily chosen co-ordinates (x, y) on the sensor plane, obtaining a measurement ϕ of the incident photon flux. After each new measurement, the SuperCam data structure $\mathcal{S}$ is updated: all segments S_i such that $(x, y) \in S_i$, must update their intensity estimates, I_i, based on the new measurement $\widehat{\phi}$. This iterative update routine is shown in Fig. 2.

After all the initial superpixel boundaries and intensities are obtained, we fill in any holes in the image using nearest neighbor interpolation. We then apply a Gaussian blur kernel with a blur radius equal to the superpixel grid size in SuperCam. We use separable blur kernels with different blur radii for superpixel grids that are rectangular in shape. A derivation of the blur kernel size is given in the supplement.

Algorithm 1 The SuperCam Algorithm

1: $P \leftarrow$ number of superpixels
2: Divide the image into P number of equal rectangles.
3: **for** $i = 1$ to P **do**
4: $\quad (x_i, y_i) \leftarrow$ randomly chosen sensor-plane position
5: $\quad$ Expose (x_i, y_i) for a fixed exposure time τ.
6: $\quad$ Calculate the intensity estimate $\widehat{\phi}(x_i, y_i)$
7: $\quad$ Initialize segments with co-ordinates (x_i, y_i) and intensity estimates $\widehat{\phi}(x_i, y_i)$.
8: **end for**
9: Fill the holes in the image using the intensity value of the nearest segment.
10: Apply Gaussian blur.

Figure 2. **SuperCam pseudoscode.** SuperCam stores scene information as a collection of segment boundaries and their associated intensity values. Segment information is updated after a new measurement is available. We envision steps 4–7 of the algorithm to run on-sensor (in or near the camera pixels) and step 9–10 to be implemented off-sensor. The SuperCam algorithm is simple, does not require heavy compute and can be implemented with minimal hardware modifications. The nearest segment operation is parallelizable and can be implemented to run real-time off sensor.

SuperCam: Practical implementations. The abstract measurement model encompasses many existing image sensing techniques including a raster-scanning single-pixel sensor [19], position-sensitive photomultiplier tubes [31], and even a conventional pixelated sensor with row/column addressable readout [13]. For the remainder of this paper we focus on one specific implementation of the SuperCam idea that relies on a single-photon sensitive camera sensors made up of a high-resolution array single-photon avalanche diode (SPAD) pixels with a single readout that reports the (x, y) location of each new photon detection event. SPAD cameras sense scene information at the finest spatio-temporal granularity possible, with on-sensor compute that can mimic a wide range of image sensing modalities including conventional CMOS/CCD pixels and event cameras [51]. Hence they provide a suitable testbed for implementing and testing the performance limits of SuperCam.

We simulate the SuperCam as SPAD array passively collecting photons. We selectively expose the individual SPAD pixels to build the superpixel image in a single exposure. As a proof of concept, we implement a passive SPAD simulator that generates individual photon arrivals over a single exposure and generates an intensity image using the quanta burst model proposed in [38]. We also propose a passive SPAD simulator that simulates the given image with a mean photon per pixel setting that adaptively exposes the image to obtain an image that has minimum photon noise with a limited number of photon arrival simulation trials.

3.1. SuperCam Emulation using a SPAD Camera

Assuming a passive SPAD sensor array imaging a scene, the number of photons $Z(x,y)$ arriving at a sensor plane location (x,y) can be modeled as a Poisson random variable [67] $P\{Z=k\} = \frac{(\phi\tau\eta)^k e^{-\phi\tau\eta}}{k!}$ where $\phi(x,y)$ is the photon flux (photons/second) arriving at the pixel (x,y), τ is the exposure time, and η is the quantum efficiency. Each SPAD pixel can store at most 1 photon detection event, hence the pixel readout is a binary value $B(x,y)$ which follows a Bernoulli distribution: $P\{B=0\} = e^{-(\phi\tau\eta + r_q\tau)}$ and $P\{B=1\} = 1 - e^{-(\phi\tau\eta + r_q\tau)}$ where r_q is the dark count rate, which refers to spurious counts that are not due to incident photons.

Starting from RGB intensity images from publicly available datasets, we simulate the binary streams of photon detections using the following procedure. We choose a "mean photons per pixel" value of p, and adjust the exposure for each individual image to match this mean value. Assuming that the incident photon flux ϕ is proportional to the value of the intensity image I_i, the probability that a SPAD pixel registers a 1 is given by, $P\{B=1\} = 1 - e^{-cI_i}$, where the value of c is the exposure adjustment term. Next, assuming there are M pixels in the image and N binary image frames captured by the camera, we select a value of c to ensure that $\frac{1}{M}\sum_{i=1}^{M}(1 - e^{-cI_i}) = \frac{p}{N}$. Assuming the incident photon flux is low enough that $cI_i \ll 1$, we apply a Taylor series approximation $1 - e^{-cI_i} \approx cI_i$ which gives us a closed form expression for the per-image exposure scaling term: $c = \frac{p}{N I_{\text{avg}}}$ where I_{avg} is the average of all the pixel values in the ground truth image. Recovering the true pixel intensities from these binary-valued measurements involves summing the binary frames $S(x,y)$ and applying a log-compression [38]: $\hat{\phi}(x,y) = -\ln(1 - S(x,y)/M)/(c\eta) - r_q/\eta$.

3.2. Comparisons

Since there are no comparable baselines for cameras that generate superpixel images on the fly, we compare against modified versions of existing methods that have been restricted to use the same volume of on-sensor memory (subsequently referred to as "Restricted"). Here we illustrate our method for matching the memory footprint using SNIC [1].

For an original input image with M pixels and a target output number of P superpixels, our (unoptimized implementation of the) SuperCam data structure which stores the superpixel boundaries and intensity values has a memory footprint $\sim$*10P*. The off-the-shelf implementation of SNIC needs access to the raw RGB pixel values for the entire image so that they can be processed using a priority queue, which maintains a list of pixels most "similar" to the current pixel being processed. The memory footprint of SNIC therefore depends not only on P but also on M because the priority queue contains all pixel intensities. We produce a memory restricted implementation by first setting the memory budget (e.g., between 70-700KB), then adjusting the number of superpixels and image size such that the combined memory of the image and required data structure fit into the preset memory budget. In order to determine the ideal ratio between the number of superpixels and image size, we carried out coarsely sampled experiments for a range of settings and picked the value that produced the lowest Under Segmentation Error. For SNIC, the lowest error was obtained by allocating $5\times$ as much memory for the image data as for the number of superpixels.

Empirically, we found that applying Gaussian blur to the memory-constrained SNIC image using the same kernel size we applied to the SuperCam image improves the results of downstream computer applications for SNIC as well. We therefore include these blurred inputs for comparison, which have been labeled as "SNIC with blur" in subsequent figures.

So far in this section we have described how we simulate a SuperCam from scene information captured at the granularity of individual photons. In the next section, we will use simulated and emulated data to show SuperCam's capabilities for a wide range of computer vision tasks.

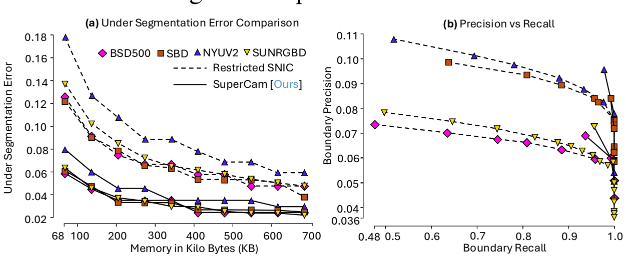

Figure 3. **Quantitative comparison with non-learning based methods.** We compare the superpixel segmentation quantitatively using the BSD500, NYUV2, SBD and SUNRGBD datasets. **(a) Under Segmentation Error** Under Segmentation Error comparison for SuperCam and SNIC for all datasets. SuperCam does at least twice as well as SNIC when using the same amount of memory. **(b) Superpixel Performance** Precision vs recall plot for the SuperCam and SNIC algorithms. SuperCam has better recall than SNIC on all datasets, although it has a lower precision than SNIC due to the higher quantity of superpixels used for the same amount of memory.

4. Results

We show results for four different computer vision tasks spanning low-, mid- and high-level vision. First, we show quantitative and qualitative evaluation of superpixel segmentation results, evaluating superpixel quality using the

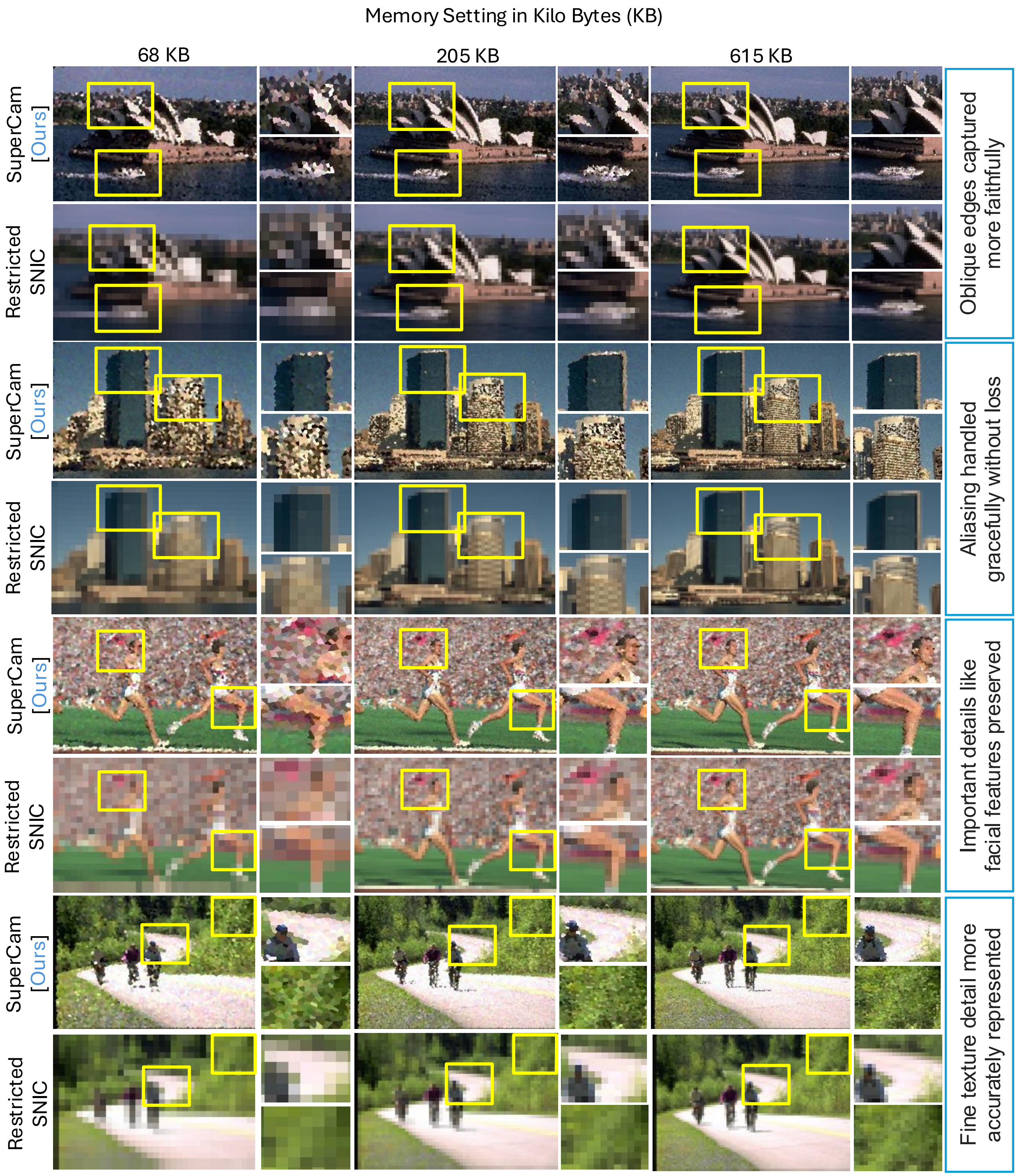

Figure 4. **Comparison of memory-constrained SNIC and SuperCam superpixel images.** Superpixel images generated from the BSD500 dataset for different memory settings in kilobytes (KB), shown here *without* Gaussian blur applied for either method. The SuperCam results show higher fidelity, better textural details, and less aliasing.

standard metrics of under segmentation error and precision vs. recall for those datasets where ground truth (hand-annotated) segmentation is available. Second, we show the task of image segmentation, where, using the superpixel image as input, a state-of-the-art transformer-based semantic segmentation model is used to generate segmentation

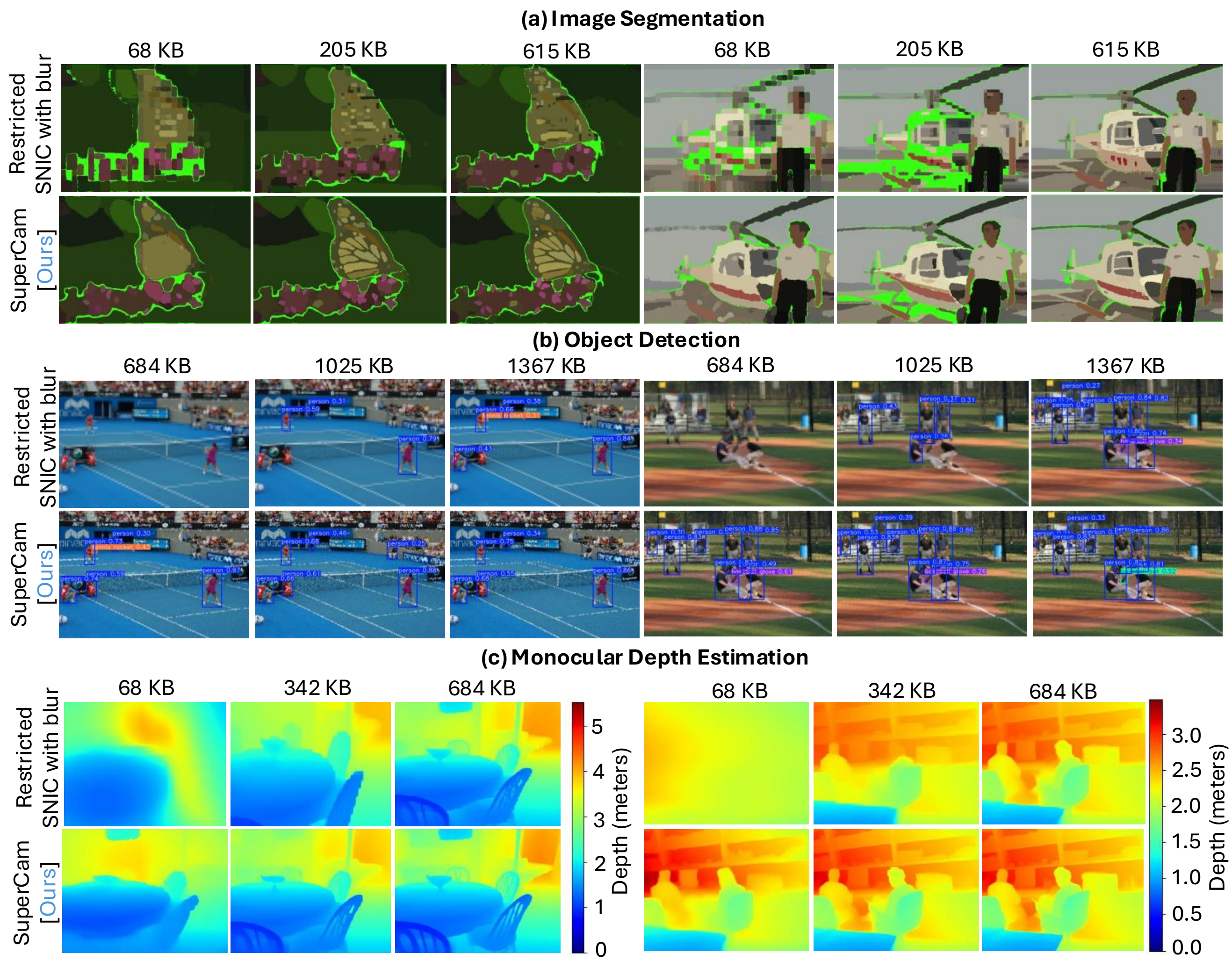

Figure 5. **Visual comparison of SuperCam and memory constrained SNIC for three different computer vision tasks.** **(a)** Image Segmentation results using SAM2 on the BSD500 dataset. The florescent green color refers to regions of the image that were not given any mask by the SAM2 model. **(b)** Object Detection results using the YOLOv12 model on the COCO dataset. **(c)** Monocular Depth Estimation results using the DepthAnythingV2 model on the NYUV2 dataset. Rows compare across algorithms and columns show different memory settings (in kilobytes (KB)). SuperCam produces visually better results than SNIC using the same amount of memory.

masks. Third, we show results for object detection and localization on superpixel images using standard benchmark datasets. Fourth, we show results for monocular depth estimation that assigns (relative) depth information at a per-pixel level. We chose these four tasks because they seem naturally suited for superpixelated input images where fine detail at the level of individual pixels is not available. Finally, we also show some qualitative results from real-world datasets of binary frames captured using a research-grade SPAD camera [39].

4.1. Superpixel Evaluation

Comparison with non-learning-based methods. We evaluate the quality of SuperCam's superpixel segmentation output using the under segmentation error [42] and boundary precision vs. recall plot. The precision and recall values are determined using the implementation given in [49]. Fig. 3 shows quantitative evaluation metrics on the BSD500 [40], NYUV2 [47], SBD [21] and SUNRGBD [25, 48, 65, 72] datasets. Observe that overall trends are similar across the four datasets used. The under segmentation error decreases monotonically as we increase the number of superpixels. For the same memory usage, SuperCam performs at least twice as well as SNIC and shows better recall for the same precision values as SNIC. We also provide a comparison with memory restricted SLIC [2] and ERS [36] in the supplement, both showing similar performance and error trends as those of SNIC. Visual comparison results shown in Fig. 4 indicate that for applications that are resource-constrained in terms of memory, SuperCam gives better edge preservation and a more complete superpixel image compared to a memory-constrained version of SNIC.

Comparison with learning based methods. In the supplement, we compare SuperCam against two recent deep-learning-based superpixel models SPAM [59] and LNS-Net [75]. The lowest superpixel/memory setting of SuperCam (68KB) is comparable with LNS-Net using 800 superpixels and $\sim$2 GB memory. We also compare quantized models with smaller memory footprints. SuperCam outperforms on Under Segmentation Error while consuming $> 1000\times$ less memory (kB vs GB).

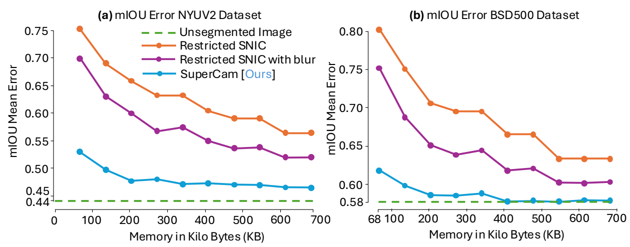

Figure 6. **Quantitative evaluation of image segmentation.** We compare the quality of image segmentation results produced by the SegmentAnythingV2 model using the mIOU metric on publicly available **(a)** NYUV2 and **(b)** BSD500 datasets. Observe that our proposed SuperCam method achieves a lower mIOU mean error than SNIC. The error approaches that of an unsegmented image as the number of superpixels is increased.

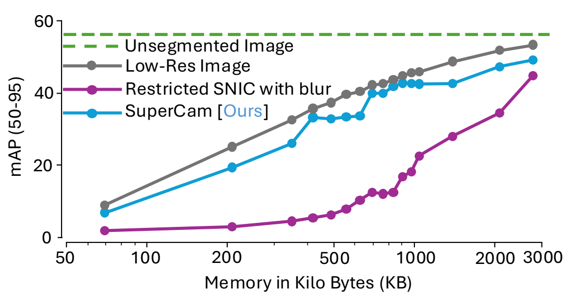

Figure 7. **Quantitative evaluation of object detection.** We compare object detection performance for the superpixel images of SNIC and SuperCam with downsampled images. We use mAP(50-95) values for the ground truth bounding boxes produced by the YOLOv12 on the COCO dataset. Observe that SuperCam detections are better than memory constrained SNIC and comparable to results for resolution-matched images. Missed detections in SuperCam are primarily smaller objects lost due to sampling limitations. Both SNIC and SuperCam converge toward the results produced by an unsegmented image as the available memory is increased.

4.2. Image Segmentation

We use Segment Anything Model 2 [44] to perform image segmentation on the BSD500 [40], NYUV2 [47] and SBD [21] datasets. As seen in Fig. 5(a), SuperCam segmentation results have overall fewer un-segmented regions

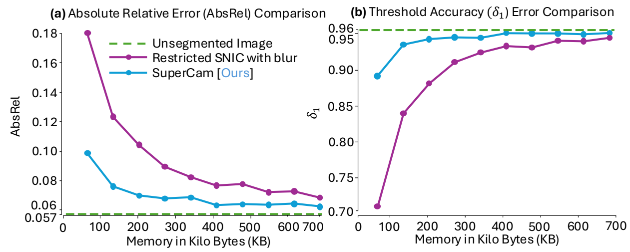

Figure 8. **Quantitative evaluation of monocular depth estimation.** We compare the **(a)** Absolute Relative Error (AbsRel) and **(b)** Threshold Accuracy (δ_1) error metrics for depth estimates produced by the DepthAnythingV2 model on the NYUV2 dataset. Observe that SuperCam error metrics are better when compared to SNIC using the same amount of memory. Both algorithms converge to the error values produced by the unsegmented image as memory increases.

(shown in fluorescent green) compared to SNIC. Moreover, our results also show more successful recovery of larger objects, such as the helicopter, even at lower memory levels. We provide a quantitative evaluation using the mIOU error metric in Fig. 6 for the NYUV2 [47] and BSD500 [40] datasets. Our method provides a consistently lower mIOU error than SNIC across both datasets. Moreover, the error metric shows a decreasing trend, which appears to converge to the error achieved if the original unsegmented image were used instead.

4.3. Object Detection

We ran an off-the-shelf object detection model, YOLOv12 [53], on superpixel outputs generated from SuperCam and a memory-constrained version of SNIC for comparison. We show results for the COCO dataset [35]. Observe that in Fig. 5(b) SuperCam successfully detects and localizes objects even at lower memory settings, while the same objects are completely missed when using SNIC. Both methods perform comparably at higher memory settings. A quantitative comparison of the precision of the detected bounding boxes is shown using the $mAP(50-95)$ score in Fig. 7. Our method consistently outperforms SNIC and gracefully approaches the mAP score of an unsegmented image as the number of superpixels (memory) is increased.

We note that superpixel images underperform compared to high-resolution images, for detecting extremely small objects that occupy very few pixels in the original image. This is a fundamental limitation of the superpixelation approach: applications that require reliable detection of extremely small objects will necessarily have to either trade off extra memory or re-run the superpixelation algorithm on an optically zoomed region of the scene.

4.4. Monocular Depth Estimation

Fig. 5(c) shows visual results for a state-of-the-art monocular depth estimation model, Depth Anything V2 [69], us-

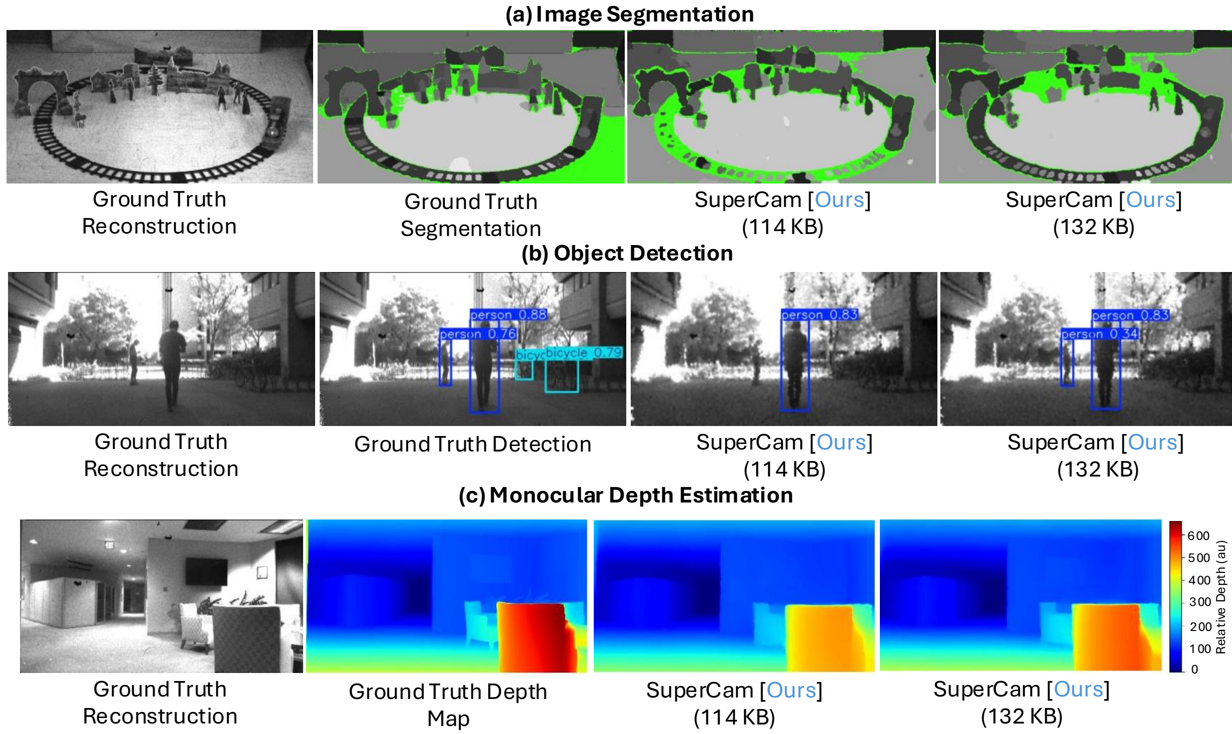

Figure 9. **Qualitative comparison of SPAD camera results.** We show experimental results of SuperCam on real world SPAD data from the SWISS SPAD sensor used in the Burst-Vision work. **(a)** Image Segmentation results using SAM2. The florescent green color refers to regions of the image that were not given any mask by SAM2 model. **(b)** Object Detection results using the YOLOv12 model. **(c)** Monocular Depth Estimation results using the DepthAnythingV2 model. Observe that results get better as we increase the memory used.

ing superpixelated images generated with SNIC and SuperCam. The predicted depth is compared to the ground truth for the KITTI [18], DIODE [57], NYUv2 [47], and Sintel [8] datasets. Observe that our results consistently outperform SNIC in terms of recovering depth edges for small objects and details. Some of these details are visible in the SuperCam output even at the lowest memory settings, while SNIC produces an unusable depth map. As seen in Fig. 8 SuperCam consistently outperforms SNIC on quantitative quality metrics such as the absolute relative error (AbsRel) and the threshold accuracy (δ_1) error metrics. Refer to the supplement for more results.

4.5. Hardware Results

Finally, we show SuperCam results on a publicly available real-world dataset, [39] captured using a SwissSPAD sensor [54]. These real-world captures consist of stacks of binary-valued photon detection frames, capturing indoor and outdoor settings over a wide dynamic range of illumination. We reconstruct the ground-truth intensity image by summing these binary frames and estimating scene intensity using log-compression as described in Sec. 3. Qualitative results on three of the image sequences are shown in Fig. 9. The reconstructed images are noisy due to Poisson noise and defective (dead and hot) pixels, and saturated in some regions due to challenging lighting conditions. Nevertheless, SuperCam performs comparable to ground truth results for all three computer vision tasks: segmentation, object detection, and monocular depth estimation.

5. Conclusion and Future Work

This paper introduces a new camera design that produces superpixel images on the fly without storing or capturing the entire image. Our results show that the information encoded in a SuperCam image is sufficient to perform image segmentation, object detection, and depth estimation. We hope that this work will serve as motivation for further research into the design of cameras that inherently capture parsimonious image representations and bypass the step of forming a high-resolution image altogether.

Future work for SuperCam would be the development of a hardware prototype for the camera. Although synthetic and experimental results look promising, SuperCam still remains a conceptual idea and releasing it in hardware will push the idea further. Additionally, it should be possible to reduce the memory required to implement SuperCam by another factor of two if we were to use a more lightweight data structure that does not include additional information necessary for debugging and testing algorithmic variations.

References

[1] Radhakrishna Achanta and Sabine Susstrunk. Superpixels and polygons using simple non-iterative clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4895–4904, 2017. 2, 3, 4, 1

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 34 (11):2274–2282, 2012. 2, 3, 6

[3] Iñigo Alonso and Ana C Murillo. Semantic segmentation from sparse labeling using multi-level superpixels. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5785–5792, 2018. 2

[4] Andrei Ardelean. Computational imaging SPAD cameras. *PhD thesis*, 2023. 3

[5] Isabela Borlido Barcelos, Felipe De Castro Belem, Leonardo De Melo Joao, Zenilton KG Do Patrocínio Jr, Alexandre Xavier Falcao, and Silvio Jamil Ferzoli Guimarães. A comprehensive review and new taxonomy on superpixel segmentation. In *ACM Computing Surveys*, pages 1–39, 2024. 2

[6] András Bódis-Szomorú, Hayko Riemenschneider, and Luc Van Gool. Superpixel meshes for fast edge-preserving surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2011–2020, 2015. 2

[7] Laurie Bose, Jianing Chen, and Piotr Dudek. Descriptor-in-pixel: Point-feature tracking for pixel processor arrays. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5392–5400, 2025. 3

[8] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Proceedings of the European conference on Computer Vision*, pages 611–625, 2012. 8, 11

[9] Ana B Cambra, Adolfo Muñoz, Ana C Murillo, José Jesús Guerrero, and Diego Gutierrez. Improving depth estimation using superpixels. In *CEIG*, pages 49–58, 2014. 2

[10] Aleksandra Chuchvara, Attila Barsi, and Atanas Gotchev. Fast and accurate depth estimation from sparse light fields. *IEEE Transactions on Image Processing*, 29:2492–2506, 2019. 2

[11] Christian Conrad, Matthias Mertz, and Rudolf Mester. Contour-relaxed superpixels. In *Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 280–293, 2013. 2

[12] Qiqin Dai, Fengqiang Li, Oliver Cossairt, and Aggelos K Katsaggelos. Adaptive illumination based depth sensing using deep superpixel and soft sampling approximation. *IEEE Transactions on Computational Imaging*, 8:224–235, 2022. 2

[13] Bart Dierickx, Danny Scheffer, Guy Meynants, Werner Ogiers, and Jan Vlummens. Random addressable active pixel image sensors. In *Advanced Focal Plane Arrays and Electronic Cameras*, pages 2–7. SPIE, 1996. 3

[14] Marco F Duarte, Mark A Davenport, Dharmpal Takhar, Jason N Laska, Ting Sun, Kevin F Kelly, and Richard G Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, 2008. 2

[15] Rachel Ehrenberg. Square Pixel Inventor Tries to Smooth Things Out, 2018. `https://www.wired.com/2010/06/smoothing-square-pixels/` (last accessed Nov 10, 2025). 2

[16] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 44(1):154–180, 2020. 3

[17] Kent Gauen and Stanley Chan. Bayesian-inspired space-time superpixels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5382–5391, 2025. 2

[18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 8, 11

[19] Graham M Gibson, Steven D Johnson, and Miles J Padgett. Single-pixel imaging 12 years on: a review. *Optics express*, 28(19):28190–28208, 2020. 2, 3

[20] Theodosios Gkamas and Christophoros Nikou. Guiding optical flow estimation using superpixels. In *Proceedings of the International Conference on Digital Signal Processing*, pages 1–6, 2011. 2

[21] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–8, 2009. 6, 7

[22] Yinlin Hu, Rui Song, Yunsong Li, Peng Rao, and Yangli Wang. Highly accurate optical flow estimation on superpixel tree. *Image and Vision Computing*, 52:167–177, 2016. 2

[23] Talha Ilyas, Abbas Khan, Muhammad Umraiz, and Hyongsuk Kim. Seek: A framework of superpixel learning with cnn features for unsupervised segmentation. *Electronics*, 9 (3):383, 2020. 2

[24] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *Proceedings of the European conference on Computer Vision*, pages 352–368, 2018. 2, 3

[25] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T. Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3-d object dataset: Putting the kinect to work. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1168–1174, 2011. 6

[26] Feng Jin and Xuefeng Li. A dense depth estimation method using superpixels. In *Proceedings of the International Computer Conference on Wavelet Active Media Technology and Information Processing*, pages 290–294, 2015. 2

[27] Liu Jingjing, Chen Ying, Zha Cheng, Yu Hua, and Zhao Li. Tracking using superpixel features. In *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation*, pages 878–881, 2016. 2

[28] Hoyoung Kim, Minhyeon Oh, Sehyun Hwang, Suha Kwak, and Jungseul Ok. Adaptive superpixel for active learning

in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 943–953, 2023. 2

[29] Jeremy Klotz and Shree K Nayar. Minimalist vision with freeform pixels. In *Proceedings of the European conference on Computer Vision*, pages 329–346, 2024. 3

[30] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Superpixel soup: Monocular dense 3d reconstruction of a complex dynamic scene. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 43(5):1705–1717, 2019. 2

[31] K Kuroda. Position sensitive photomultiplier. *Nuclear Instruments and Methods in Physics Research*, 196(1):187–197, 1982. 3

[32] Alex Levinshtein, Adrian Stere, Kiriakos N Kutulakos, David J Fleet, Sven J Dickinson, and Kaleem Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 31 (12):2290–2297, 2009. 2

[33] Xiuzhi Li and Chuanluo Xu. Moving object detection in dynamic scenes based on optical flow and superpixels. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pages 84–89, 2015. 2

[34] James Liang, Tianfei Zhou, Dongfang Liu, and Wenguan Wang. Clustseg: clustering for universal segmentation. In *Proceedings of the International Conference on Machine Learning*, 2023. 2, 3

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European conference on Computer Vision*, pages 740–755, 2014. 7, 3, 9

[36] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy rate superpixel segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2097–2104, 2011. 2, 6, 3

[37] Miriam Lopez-de-la Calleja, Takayuki Nagai, Muhammad Attamimi, Mariko Nakano-Miyatake, Hector Perez-Meana, et al. Object detection using surf and superpixels. *Journal of Software Engineering and Applications*, 6(09):511, 2013. 2

[38] Sizhuo Ma, Shantanu Gupta, Arin C Ulku, Claudio Bruschini, Edoardo Charbon, and Mohit Gupta. Quanta burst photography. *ACM Transactions on Graphics*, 39(4):79–1, 2020. 4

[39] Sizhuo Ma, Paul Mos, Edoardo Charbon, and Mohit Gupta. Burst vision using single-photon cameras. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5375–5385, 2023. 6, 8, 17

[40] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 416–423, 2001. 3, 6, 7

[41] Branislav Mičušík and Jana Košecká. Multi-view superpixel stereo in urban environments. *Internation Journal of Computer Vision*, 89(1):106–119, 2010. 2

[42] Peer Neubert and Peter Protzel. Superpixel benchmark and comparison. In *Proceedings Forum Bildverarbeitung*, pages 1–12, 2012. 6

[43] Hankui Peng, Angelica I Aviles-Rivero, and Carola-Bibiane Schönlieb. Hers superpixels: Deep affinity learning for hierarchical entropy rate segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 217–226, 2022. 2

[44] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 7, 3, 4

[45] Guang Shu, Afshin Dehghan, and Mubarak Shah. Improving an object detector and extracting regions using superpixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3721–3727, 2013. 2

[46] Yawar Siddiqui, Julien Valentin, and Matthias Nießner. Viewal: Active learning with viewpoint entropy for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9433–9443, 2020. 2

[47] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European conference on Computer Vision*, pages 746–760, 2012. 6, 7, 8, 11

[48] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European conference on Computer Vision*, pages 746–760, Berlin, Heidelberg, 2012. 6

[49] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An evaluation of the state-of-the-art. In *Computer Vision and Image Understanding*, pages 1–27, 2018. 2, 6, 4

[50] Waqas Sultani, Soroush Mokhtari, and Hae-Bum Yun. Automatic pavement object detection using superpixel segmentation combined with conditional random field. *IEEE Transactions on Intelligent Transportation Systems*, 19(7):2076–2085, 2017. 2

[51] Varun Sundar, Andrei Ardelean, Tristan Swedish, Claudio Bruschini, Edoardo Charbon, and Mohit Gupta. Sodacam: Software-defined cameras via single-photon imaging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8165–8176, 2023. 3

[52] Lucas Teixeira and Margarita Chli. Real-time local 3d reconstruction for aerial inspection using superpixel expansion. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4560–4567, 2017. 2

[53] Yunjie Tian, Qixiang Ye, and David Doermann. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025. 7, 3, 4, 9

[54] Arin Can Ulku, Claudio Bruschini, Xavier Michalet, Shimon Weiss, and EA Charbon. A 512×512 SPAD image sensor with built-in gating for phasor based real-time siFLIM. *Proceedings of the International Image Sensors Workshop*, pages 234–237, 2017. 8, 17

[55] Roy Uziel, Meitar Ronen, and Oren Freifeld. Bayesian adaptive superpixel segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8470–8479, 2019. 2

[56] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin De Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *Proceedings of the European conference on Computer Vision*, pages 13–26, 2012. 2

[57] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *CoRR*, abs/1908.00463, 2019. 8, 3, 11

[58] Olga Veksler, Yuri Boykov, and Paria Mehrani. Superpixels and supervoxels in an energy optimization framework. In *Proceedings of the European conference on Computer Vision*, pages 211–224, 2010. 2

[59] Julien Walther, Rémi Giraud, and Michaël Clément. Superpixel anything: A general object-based framework for accurate yet regular superpixel segmentation. In *Proceedings of the British Machine Vision Conference*, page 1035, 2025. 7, 3

[60] Kai Wang, Liang Li, and Jiawan Zhang. End-to-end trainable network for superpixel and image segmentation. *Pattern Recognition Letters*, 140:135–142, 2020. 3

[61] Lijun Wang, Huchuan Lu, and Ming-Hsuan Yang. Constrained superpixel tracking. *IEEE Transactions on Cybernetics*, 48(3):1030–1041, 2017. 2

[62] Shu Wang, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang. Superpixel tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1323–1330, 2011. 2

[63] Xuehui Wang, Qingyun Zhao, Lei Fan, Yuzhi Zhao, Tiantian Wang, Qiong Yan, and Long Chen. Semasuperpixel: A multi-channel probability-driven superpixel segmentation method. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1859–1863, 2021. 2

[64] Yaxiong Wang, Yunchao Wei, Xueming Qian, Li Zhu, and Yi Yang. Ainet: Association implantation for superpixel segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7078–7087, 2021. 2, 3

[65] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1625–1632, 2013. 6

[66] Junjie Yan, Yinan Yu, Xiangyu Zhu, Zhen Lei, and Stan Z Li. Object detection by labeling superpixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5107–5116, 2015. 2

[67] Feng Yang, Yue M Lu, Luciano Sbaiz, and Martin Vetterli. Bits from photons: Oversampled image acquisition using binary poisson statistics. *IEEE Transactions on Image Processing*, 21(4):1421–1436, 2011. 4

[68] Fan Yang, Huchuan Lu, and Ming-Hsuan Yang. Robust superpixel tracking. *IEEE Transactions on Image Processing*, 23(4):1639–1651, 2014. 2

[69] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Proceedings of the Advances in Neural Information Processing Systems*, 37:21875–21911, 2024. 7, 3, 4, 11

[70] Hongyi Zhang, Laurie Bose, Jianing Chen, Piotr Dudek, and Walterio Mayol-Cuevas. Focal Plane Visual Feature Generation and Matching on a Pixel Processor Array. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 29031–29039, 2025. 3

[71] Liang Zhang, Huan Li, Peiyi Shen, Guangming Zhu, Juan Song, Syed Afaq Ali Shah, Mohammed Bennamoun, and Li Zhang. Improving semantic image segmentation with a probabilistic superpixel-based dense conditional random field. In *IEEE Access*, pages 15297–15310, 2018. 2

[72] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Proceedings of the Advances in Neural Information Processing Systems*, 2014. 6

[73] Alex Zihao Zhu, Jieru Mei, Siyuan Qiao, Hang Yan, Yukun Zhu, Liang-Chieh Chen, and Henrik Kretzschmar. Superpixel transformers for efficient semantic segmentation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 7651–7658, 2023. 2

[74] Hao Zhu, Xiaoming Sun, Qi Zhang, Qing Wang, Antonio Robles-Kelly, Hongdong Li, and Shaodi You. Full view optical flow estimation leveraged from light field superpixel. *IEEE Transactions on Computational Imaging*, 6:12–23, 2019. 2

[75] Lei Zhu, Qi She, Bin Zhang, Yanye Lu, Zhilin Lu, Duo Li, and Jie Hu. Learning the superpixel in a non-iterative and lifelong manner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1225–1234, 2021. 2, 7, 3