MDPI

*Letter*

# Application-Oriented Retinal Image Models for Computer Vision

**Ewerton Silva [1], Ricardo da S. Torres [2]*, Allan Pinto [1], Lin Tzy Li [1], José Eduardo S. Vianna [1], Rodolfo Azevedo [1], and Siome Goldenstein [1]**

[1] University of Campinas; ewerton.silva@students.ic.unicamp.br, allan.pinto@ic.unicamp.br, lintzyli@gmail.com, jevianna@gmail.com, rodolfo@ic.unicamp.br, siome@ic.unicamp.br
[2] Norwegian University of Science and Technology; ricardo.torres@ntnu.no
* Correspondence: ricardo.torres@ntnu.no

Version April 2, 2020 submitted to Sensors

**Abstract:** Energy and storage restrictions are relevant variables software applications should be concerned about when running in low-power environments. Computer Vision (CV) applications, in particular, exemplify well that concern, since conventional uniform image sensors typically capture large amounts of data to be further handled by the appropriate CV algorithms. Moreover, much of the acquired data are often redundant and outside of the application's interest, which leads to unnecessary processing and energy spending. In the literature, techniques for sensing and re-sampling images in non-uniform fashions have emerged to cope with these problems. In this study, we propose Application-Oriented Retinal Image Models that define a space-variant configuration of uniform images and contemplate requirements of energy consumption and storage footprints for CV applications. We hypothesize that our models might decrease energy consumption in CV tasks. Moreover, we show how to create the models and validate their use in a face detection/recognition application, evidencing the compromise between storage, energy, and accuracy.

**Keywords:** Retinal image model; Space-variant computer vision; Foveation; Low-power; Energy consumption.

## 1. Introduction

By means of a conventional sensor, one can easily capture uniform high-resolution images and describe what is depicted. However, for computers, interpreting images is not trivial, demanding complex Computer Vision (CV) algorithms along with a proper management of the available resources, to allow the software applications to run efficiently in different hardware platforms. As a matter of fact, a computational burden might come into play due to real-time restrictions often imposed by the available hardware to process these high-resolution data [1]. In the mobile environment, for example, managing energy (i.e., battery life) is mandatory, as its negligence might prevent users from enjoying a satisfactory experience [2]. Whereas common strategies to save resources rely on uniform resolution reductions and frame-rate decreases, another one is to mimic the space-variant configuration of the human eye. Since some tasks as tracking and pattern recognition do not demand high resolution data across the whole image [1], it is reasonable to work with space-variant images.

The paradigm of capturing and processing uniform images co-exists with mechanisms to manage a biology-inspired image representation in the Space-Variant CV field. The overall insight comes from the nature of the human eye, where cones and rods – the photo-receptors responsible for detecting color and luminance, respectively – show a non-uniform spatial configuration that induces variable visual acuity levels across the retina [3]. The highest density of cones lies in the fovea, the central area of the retina, whereas the lowest one is found across the periphery. This provides a wide field of

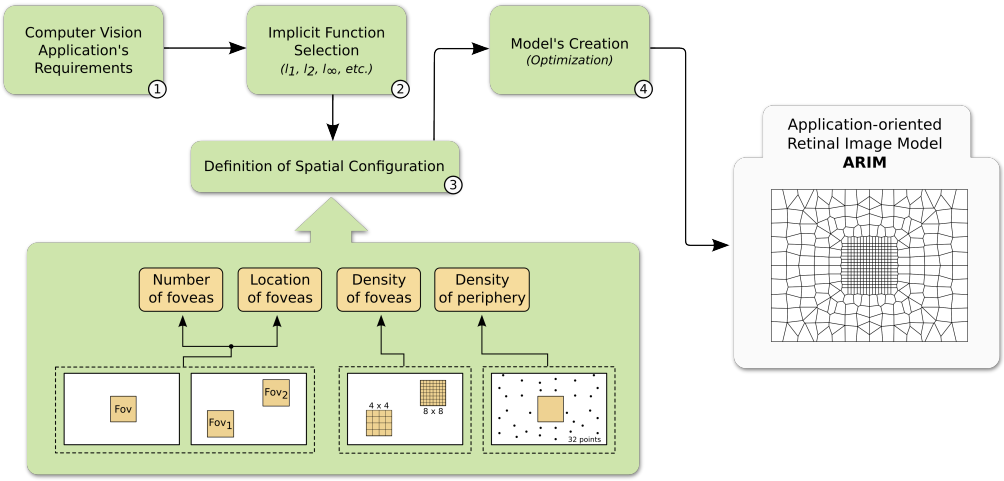


**Figure 1.** The proposed framework to generate application-oriented retinal image models. The workflow begins by defining the application's requirements regarding operation (e.g., objects' positioning, illumination) and efficiency (e.g., storage, accuracy). Then, a proper implicit function (e.g., $l_2$) and the spatial configuration of the retinal image model – comprising foveal and peripheral regions – are chosen. The next step is the generation of the model by means of an optimization procedure that considers the implicit function and the spatial configuration to resample points in the 2-d cartesian space. The final artifact is an application-oriented retinal model comprised by uniformly- and non-uniformly-sampled foveal and peripheral regions, respectively. This model is used to resample uniform images, taking them to a space-variant domain and potentially contemplating the requirements determined beforehand.

view and a high-resolution region that is used to *foveate* a point in a real scene, thereby reducing data processing to a dense, smaller region (fovea), or to a wider, sparse one (periphery) [3,4]. Both regions can also operate in synergy: the periphery examines coarse data to trigger a detailed analysis through foveation.

Concepts of the human visual system have already been explored from the hardware and software perspectives. On the hardware side, the problem has been dealt with, mainly, by two fronts: (i) the development of imaging sensors with specific non-uniform spatial configurations [5], and (ii) the use of an intermediary hardware layer to remap uniform images into variable-resolution ones. The first front allows the capture of topology-fixed foveated images at sensing time, whereas the second one provides more flexibility to change the mapping without relying on software routines. Specifically, some initiatives like [1] exploited the versatility of Field Programmable Gate Arrays (FPGA) to implement, at logical level, different space-variant mappings of uniform images, as with the case of a moving fovea that is dynamically adjusted according to the application's requirements. A similar study [6] integrated attention and segmentation mechanisms into a foveal vision system. The architecture of the solution comprised (i) a hardware layer responsible for mapping uniform cartesian images to space-variant ones and (ii) a software layer where segmentation and saliency estimation are done. In short, the salient regions from a frame might trigger a foveal shift to be performed by hardware when the next frame arrives.

Pure software-based approaches, in opposition, offer more flexibility to simulations, albeit with higher computational costs. In [7], a saccadic search strategy based on foveation for facial landmark detection and authentication is presented. The authors apply a log-polar mapping to some image points and extract Gabor filter responses at these locations, thus imitating the characteristics of the human retina. For training, SVM classifiers are used to discriminate between positive and negative classes of facial landmarks (eyes and mouth) represented by the collected Gabor responses. When testing, the saccadic search procedure evaluates several image points in the seek of candidate landmarks that are

further used to authenticate the depicted individual. A more complete review on space-variant imaging from the hardware and software perspectives using log-polar mappings is detailed in [8]. Furthermore, in [9], a foveated object detector is proposed. The detector operates on variable-resolution images obtained by resampling uniform ones with a simplified model of the human visual cortex. The results showed that the detector was capable of approximating the accuracy of a uniform-resolution-oriented one, thereby providing a satisfactory insight to evolutionary biology processes. In another work [10], image foveation is exploited along with a single-pixel camera architecture to induce a compromise between resolution and frame rate. The images are resampled by a space-variant model that is constantly reshaped to match the regions of interest detected in the image by a motion tracking procedure, thus effectively simulating a moving fovea that increasingly gathers high-resolution data across frames. To facilitate comparisons among different sensor arrangements, an appropriate method is described in [11]. The idea is to provide a common space for creating lattices of any kind. To demonstrate the viability of the method, the rectangular and hexagonal lattices are implemented and images built according to both arrangements are further compared.

Despite the progress in CV research fields in exploiting space-varying models, there is a lack of a single generic framework for handling seamlessly images generated by heterogeneous pixel sampling strategies. In this paper, we address this issue by proposing a framework for designing Application-Oriented Retinal Image Models (ARIMs) that establish a non-uniform sampling configuration of uniform images. We propose to define the appropriate model for an application on-demand, taking into account specific requirements of the target application. By exploiting such models, we hypothesize it might be possible to decrease the energy spent in computer vision tasks. We show how to create the models and validate their use in a face detection/recognition application, considering the compromise among storage rates, energy, and accuracy. We use a regular image sensor and perform the sampling procedures by means of a software layer, thus simulating the operation of a specific-purpose space-variant sensor and providing some flexibility. The overview of our framework is depicted in Figure 1.

## 2. Proposed Approach

In this section, we describe our methodology to generate ARIMs by detailing each step illustrated in Figure 1. The components of the proposed methodology will be presented in the context of a biometric application.

### *2.1. Definition of Application Requirements*

Instead of using a traditional image, coming from a general uniform sensor, we argue that the best approach is to examine the target application and investigate its requirements/demands. CV applications can comprise a very diverse set of requirements, ranging from efficiency-related ones, such as storage, speed, energy, and accuracy, to other very application-specific ones, such as the need for objects to move slowly or be positioned in specific locations in the scene, be situated in a minimum/maximum distance from the camera, be illuminated by a close light source, and so further. The application considered in this paper is concerned with user authentication based on his face: the individual enters and leaves the scene by any sides, placing himself in front of a camera that captures the scene in a wide field of view.

Although the authentication across a wide field of view is a good idea, since more faces are collected throughout the video, it is usual that the central part of the image be the protagonist of the process. In this vein, it is recommended that the individual stand or walk near the center of the image to proper positioning his/her face (e.g., to avoid severe rotations and perspective changes) for a more accurate authentication process. Thus, if one intends to reduce energy consumption, collecting faces only in a bounded central region (e.g., a square window) might be enough. On the other hand, restricting the image to its central part, albeit effective, might be seen as a very extreme decision, since other image areas may contribute with useful information for the authentication. In

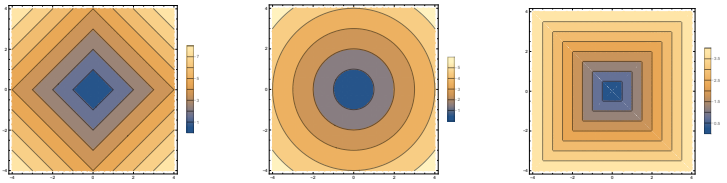

**Figure 2.** Examples of implicit functions. From left to right: $l_1$, $l_2$, and $l_\infty$.

this sense, retaining some pixel data in such areas, even in a sparse manner is also appropriate. Finally, another suitable strategy towards energy reduction is downsampling the image before performing face detection/recognition. This might reduce the energy spent in the whole authentication process, but at the cost of a drop in accuracy.

The issues discussed above illustrate examples of requirements to be defined by the analysis of an application's domain. In this paper, they were essential to guide the definition of a model for the biometric application.

### *2.2. Implicit Function Selection*

The design of the model starts with selecting a proper implicit function. The idea is that the function will act as a control mechanism to spread out the non-uniform sampled points over a desired image region. Figure 2 depicts examples of implicit functions we explored ($l_1$, $l_2$, and $l_\infty$).

### *2.3. Definition of Spatial Configuration*

This step is concerned with the spatial characteristics the model must obey. We developed hybrid space-variant models inspired on the human retina. In general, the models comprise two very distinct regions: the fovea and the periphery. The fovea is a fixed-size region of uniformly sampled pixels according to a predefined grid. For instance, a region of size $2^6 \times 2^6$ pixels can be uniformly sampled by a grid of size $2^5 \times 2^5$ pixels. Given these characteristics, we can apply conventional CV algorithms in the fovea. In opposition, the periphery is a fovea-surrounding region with a non-uniform pixel density that decreases with the distance from the fovea.

The following four parameters should be informed prior to the creation of the hybrid model:

- **Number of foveas**: Surely a human eye has only one fovea, but it is perfectly fine for a model to comprise more than one region of uniform sampling, depending on the application on hand. In our biometric application, we took into account only one fovea.
- **Location of foveas**: The foveas should be spatially organized adhering to the specific requirements of the application. In ours, the fovea is centralized in the image.
- **Density of foveas**: The foveas can be downsampled to simulate a uniform image resolution reduction. We tested different densities (grids) for our fovea.
- **Density of periphery**: The periphery is an important region that encompasses few sparse data in a non-uniform sampling configuration. As discussed previously, by retaining and wisely handling sparse peripheral information (e.g., detecting motion and coarse objects in such an area), the application's resource usage might be optimized.

### *2.4. Model Generation*

There are several ways to achieve a non-uniform point distribution. Our approach is inspired by the computer graphics literature and previous works [12,13]. Besides the implicit function, the number of peripheral (non-uniform) points and the aspect ratio of the sensor must be provided. We generate a points distribution via a local non-linear optimization procedure that, from an initial distribution, tries to minimize a global energy function defined in Equation 1, where $\vec{x}$ is a point in image space.

$$En\left(\{\vec{x}_i\}\right) = \sum_i \sum_{\vec{x}_j \leftrightarrow \vec{x}_i} \left(||\vec{x}_i - \vec{x}_j|| - \left(f(\vec{x}_i) + f(\vec{x}_j)\right)\right)^2 \quad (1)$$

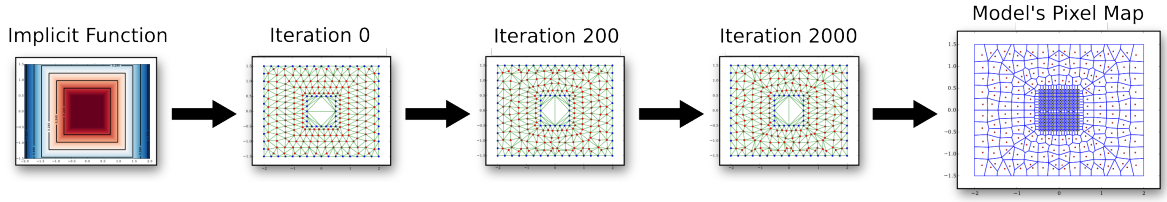


**Figure 3.** The evolution of an example of ARIM with 256 foveal (uniform), and 192 peripheral (non-uniform) pixels. The $l_\infty$ is the implicit function.

The optimum solution for Equation 1, i.e., when $En = 0$, would be a placement of every $\vec{x}_i$ such that the distance to its "neighbors" is the sum of the values of the implicit function at their locations. However, there is no closed-solution for this problem (the implicit function can be anything), nor any guarantees of a perfect solution for a scenario with an arbitrary number of points and implicit functions. Thus, we propose an approximation by means of a non-linear optimization procedure based on *Spring-Mass Models*. When doing so, each pair of points try to attract each other if they are too far, and try to repel each other when they are too close. We do not use Newton's physical model of forces from springs. Instead, we have a mass-free system, so springs generate "velocity forces."

The optimization process is very sensitive to its initial conditions. A uniform distribution of the initial positions over the valid domain coupled with a careful choice of the implicit function allows the system to converge under 2000 iterations. Figure 3 illustrates the generation of an ARIM where the optimization of uniform point distribution is carried out using the $l_\infty$ implicit function. Upon convergence, we obtain the full neighborhood map (Voronoi diagram) of the model.

## 3. Materials and Methods

In this section, we present the experimental setup necessary for simulating the usage the proposed models. The chosen dataset closely resembles one of a biometric application.

### *3.1. Dataset*

In our evaluations, we employed the Chokepoint Dataset [14] aimed at person identification/verification. The dataset comprises 48 sequences of images of $800 \times 600$ pixels resolution. Each sequence depicts several individuals entering or leaving a portal, one at a time. There are 25 and 29 individuals walking through portals 1 and 2, respectively. Moreover, each sequence is registered by three cameras placed above the portals to provide diverse sets of faces in different illumination and pose conditions. Due to the adopted settings, one of the cameras is able to capture image sequences of near-frontal faces. In short, the dataset is partitioned into the following four subsets:

- **P1E and P1L**: The subsets of frame sequences of people entering and leaving portal 1, respectively;
- **P2E and P2L**: The subsets of frame sequences of people entering and leaving portal 2, respectively;

A subset is comprised of four (4) frame sequences (S1, S2, S3, and S4), each of which is registered by three cameras (C1, C2, and C3). For instance, the frame sequence P1E_S2_C3 refers to the second sequence (S2) of people entering portal 1 (P1E) and captured by camera 3 (C3).

We used 34 image sequences (out of 48) from the dataset during our evaluations due to the following reasons:

1. One (1) of the sequences of individuals entering a portal (P1E_S1_C1) was used to train the face recognizer. Such sequence comes from camera 1, which obtains near frontal-face images. That sequence is also captured by cameras 2 and 3 at different angles, hence, to avoid biased evaluations, we ignored such sequences (P1E_S1_C2 and P1E_S1_C3), as both of these contain, essentially, the same faces of the former up to slight angle variations.

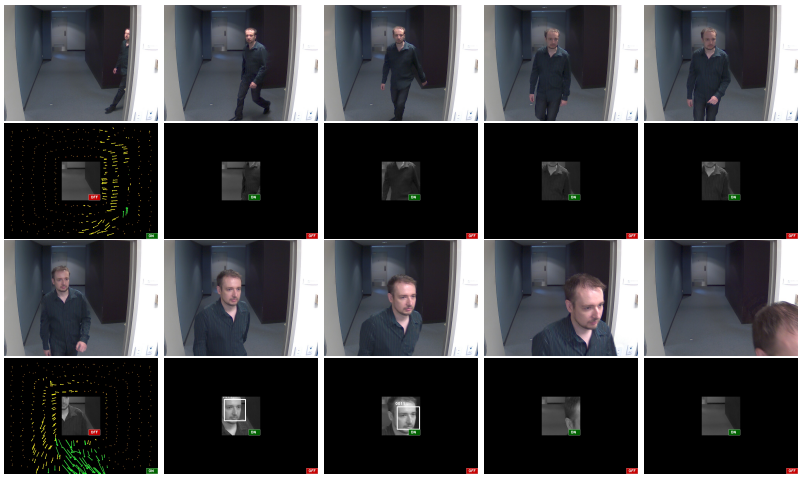

**Figure 4.** Example of a simulation using one of our ARIMs and a sample sequence from the dataset [14]. First and third rows: original frames; Second and fourth rows: reconstruction with a model that considers an optical flow peripheral representation. Green and yellow arrows indicate motion direction to the right and left sides, respectively, whereas the ON and OFF labels refer to the operational status of the foveal (face detection/recognition) and peripheral (optical flow) regions. Note that the motion analysis, besides triggering foveal analysis, is also able to restart conveniently, as long as faces are not detected in the fovea during a time interval of frames (left-most frame in the fourth row).

2. Eleven (11) sequences where no face is found in the fovea were ignored. This decision was taken because no face recognition accuracy evaluations (using our models) would apply to these sequences.

### *3.2. Application Implementation*

The biometric application uses the Viola-Jones [15] algorithm, which is a well-consolidated and widely used face detection method in the literature. As for recognizing faces, we used a descriptor based on a pretrained Deep Neural Network (DNN) model, which is essentially a ResNet network with 29 convolutional layers trained on a dataset containing approximately 3 million faces. The model is publicly available and integrates the Dlib C++ Library [16].

We simulated the operation of a specific-purpose sensor by re-sampling images according to our ARIMs. The idea is to generate images containing two regions: (i) the fovea, encompassing a small area where resolution is uniform, and (ii) the periphery, where pixels are arranged non-uniformly over a wider area. With such a configuration, we were able to perform experiments considering different foveal resolutions, while also taking advantage of the periphery according to the specific requirements of the application. In this vein, we adopted an optical flow representation (orientation and magnitude) for peripheral pixels. The motivation around that representation is that the detection/recognition in the fovea be triggered only when there is movement towards it coming from the periphery. Also, both the detection and recognition procedures turn off when no face is found under a predefined time interval. In this scenario, therefore, more energy can be saved. Figure 4 exemplifies image reconstructions with an ARIM, where we draw arrows representing the orientation and magnitude values of the identified motion in the periphery (bottom row).

The workflow of the simulation process is depicted in Figure 5, where we also discern between the software and hardware layers to illustrate an ideal hypothetical case where a specific-purpose (space-variant) sensor was available. Both layers are connected by a 1-D vector (named as bytestream) that stores the foveal and peripheral pixel values captured by the sensor (i.e., the sampled image), and are input to the application. We adopted bytestreams instead of a 2-D image representation in the software simulation to bring the process closer to the ideal conceived scenario. The simulator was implemented in C++ using the OpenCV 3.0.0 library.

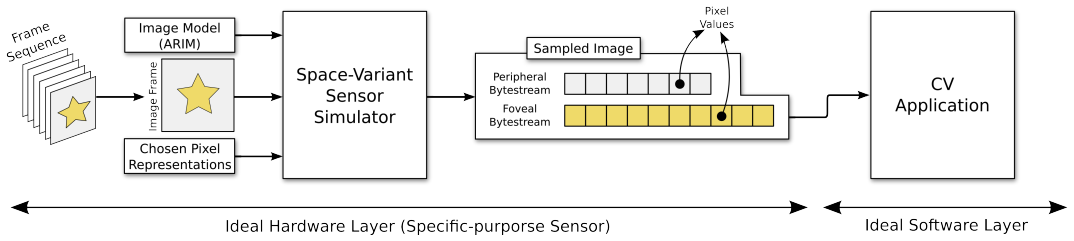


**Figure 5.** Implemented workflow for simulating the use of ARIMs in a specific CV application. In an ideal scenario, the ARIM, a captured image frame, and the chosen pixel representations for foveal and periphery areas are input to an hypothetical specific-purpose sensor that changes its configuration at run-time. Such a sensor would yield a stream (bytestream) of pixel data from each region of the captured image. The stream (not the 2-d image) would be forwarded to the CV application. For simulation purposes, however, this architecture is fully implemented by software.

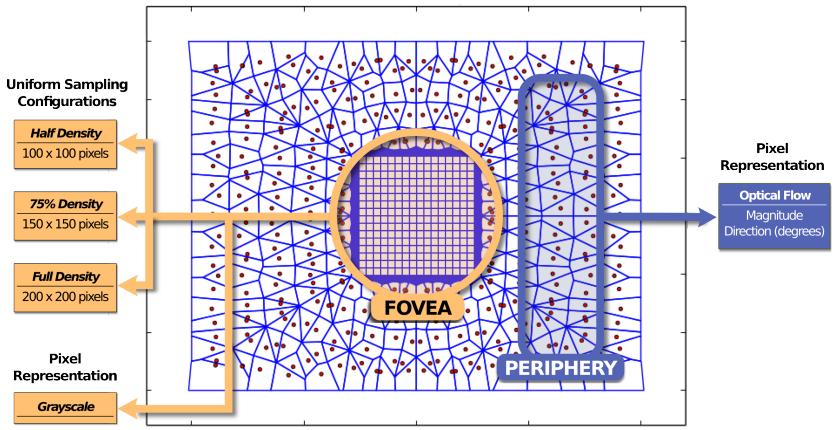


**Figure 6.** The pixel map of the evaluated ARIM and its configurations. The experimented foveal configurations comprised three uniform sampling setups: $100 \times 100$ (half density), $150 \times 150$, and $200 \times 200$ (full density) pixels. The pixel representations for the fovea and periphery were based on the grayscale and optical flow (magnitude and direction) values, respectively.

### *3.3. Evaluated models*

We evaluated three different ARIMs. Each model comprises 384 non-uniform peripheral points and a central foveal region of size $200 \times 200$ pixels. The models diverge from each other in the uniform-sampling configuration sizes adopted for their foveas, which are $100 \times 100$ (half density), $150 \times 150$ (75% density), and $200 \times 200$ (full density). Those settings allow us to simulate different foveal resolutions. For all models, optical flow peripheral information is used to trigger the face detection/recognition in the fovea. An illustration of the pixel map of these models and their configurations are shown in Figure 6.

### *3.4. Evaluation Criteria and Hardware Setup*

We compared the storage usage by computing the amount of bytes for storing the video, measured the energy spent (in Joules) in the biometric application for each evaluated model, and computed the mean recognition accuracy of each evaluated model considering all video frames. To measure energy, we used the Intel RAPL (Running Average Power Limit) interface [17], which is a set of internal registers from Intel processors called model specific registers (MSR). At the code level, we read these registers before and after a block of instructions, and calculate the difference between these values. More specifically, we read the MSR_RAPL_POWER_UNIT register to measure the energy spent in image readings, face detection/recognition procedures, and optical flow analysis (when using ARIMs).

**Table 1.** Number of pixels and data size reduction results for the evaluated models relative to the baseline.

| | Num. of pixels | Num. of pixels reduction | Bytes per region FOV | Bytes per region PER | Total bytes | Data size reduction |
|---|---|---|---|---|---|---|
| **Original** | 480000 | - | - | - | 1440000 | - |
| **Model_1** | 10384 | 97.83% | 30000 | 768 | 30768 | 97.86% |
| **Model_2** | 22884 | 95.23% | 67500 | 768 | 68268 | 95.25% |
| **Model_3** | 40384 | 91.58% | 120000 | 768 | 120768 | 91.61% |

The hardware setup to perform the experiments comprised an Intel Core i7-5500U, with 2.04GHz clock, 4MB cache, and 16MB RAM.

## 4. Results and Discussion

In this section, we present the experimental results regarding storage allocated, face recognition accuracy, and energy consumption induced by different ARIMs.

### *4.1. Storage reduction*

Quantifying reductions in numbers of pixels and image data sizes are essential for assessing the benefits of using different ARIMs in practical situations. Table 1 shows these measurements. We notice that the ARIMs reduced the number of pixels and the size of images in more than 91%.

### *4.2. Face recognition accuracy*

We defined accuracy as the number of true positives (i.e., correctly labeled faces) in the foveal region of a frame sequence, each of which has a benchmark for comparison. The ChokePoint Dataset informs all faces and their labels detected and recognized in each uniform image frame. However, for a fair accuracy comparison among the uniform images and the ones re-sampled by our models, we use as benchmark only the information regarding the foveal region, meaning that faces in the periphery are not considered.

Figure 7 shows an expected face recognition accuracy decreasing of our ARIM-resampled frame sequences compared to their correspondent benchmarks. The ARIMs rely on movement analysis to authenticate users, which creates a dependency between peripheral and the analysis of foveal information, some faces can be lost. Another variable influencing the accuracy rates is the foveal resolution of each tested ARIM. In fact, the accuracy rates increase with foveal resolution, and are not too low even under the 50% sampling degradation induced by Model_1, for example. In the case of Model_3, where foveal resolution matches that of the benchmark, the small loss in accuracy is justified by the quality of optical flow analysis, which seem to be acceptable for the tested application. Table 2 presents the minimum, mean, and maximum accuracy loss rates induced by each model in comparison to the benchmarks. Whereas the maximum obtained loss was 50% for Model_1 and the P2E dataset, very small loss rates (close to 0%) were registered in more than one scenario. Another interesting phenomenon is the high loss rates observed for the P2E and P2L datasets, possibly due to slight divergent conditions relative to the P1E and P1L datasets.

### *4.3. Energy consumption evaluation*

The experiments show lower energy consumption values for scenarios involving our models, as evidenced in Figure 8. The difference in energy values among our models and the baseline results directly from he data amount reduction caused by the combination of peripheral optical flow and the sampled foveal face detection/recognition. The robust and timely activation/deactivation of these latter algorithms, therefore, reduce the total energy spent in the whole authentication process, while

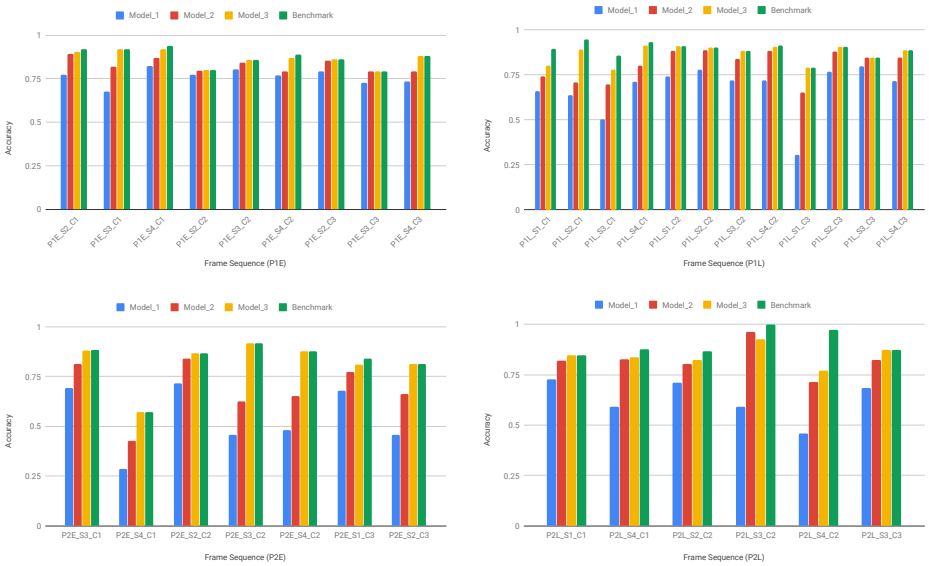


**Figure 7.** Mean face recognition accuracy regarding each evaluated model and the benchmark frame sequences from the P1E, P1L, P2E, and P1L datasets.

**Table 2.** Minimum, mean, and maximum accuracy loss rates induced by our ARIMs compared to the provided benchmarks.

| Dataset | Accuracy Loss | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Model 1 | | | Model 2 | | | Model 3 | | |
| | Min. | Mean | Max. | Min. | Mean | Max. | Min. | Mean | Max. |
| P1E | 0.032 | 0.123 | 0.264 | 0 | 0.050 | 0.108 | 0 | 0.006 | 0.021 |
| P1L | 0.060 | 0.248 | 0.613 | 0 | 0.094 | 0.255 | 0 | 0.023 | 0.103 |
| P2E | 0.174 | 0.353 | 0.500 | 0.032 | 0.172 | 0.318 | 0 | 0.006 | 0.037 |
| P2L | 0.143 | 0.300 | 0.529 | 0.033 | 0.086 | 0.265 | 0 | 0.063 | 0.206 |

keeping accuracy rates acceptable, as previously discussed. Table 3 presents the minimum, mean, and maximum energy reduction rates induced by each model relative to the benchmarks, i.e., the obtained energy savings. As expected, the reduction rates decrease with the increase in foveal resolution, because there is more data to process. This is verifiable by a quick comparison between the mean rates of Model_1 (half density) and Model_3 (full density), for example.

## 5. Conclusions

A crucial observation that led to the present study is that image data captured by uniform sensors is often dense and redundant, leading to computationally expensive solutions in terms of storage, processing, and energy consumption. We addressed this issue by exploiting a space-variant scheme which was inspired by mechanisms of biological vision, in particular, the way humans sense through the retina. We introduced a generic framework for designing application-oriented retinal image models. The models should be used to re-sample the input images prior to executing an specific CV task. We selected a biometric application to illustrate the conception and usefulness of appropriate models.

The experiments on the Chokepoint dataset and three different ARIMs demonstrate the flexibility of the proposed framework in devising models with different properties regarding storage requirements, energy consumption, and accuracy performance. We could observe, for example, that the use of different space-variant strategies may lead to a big reduction in terms of storage resources and energy consumption, whereas the accuracy loss rates were low in most cases. Such a trade-off

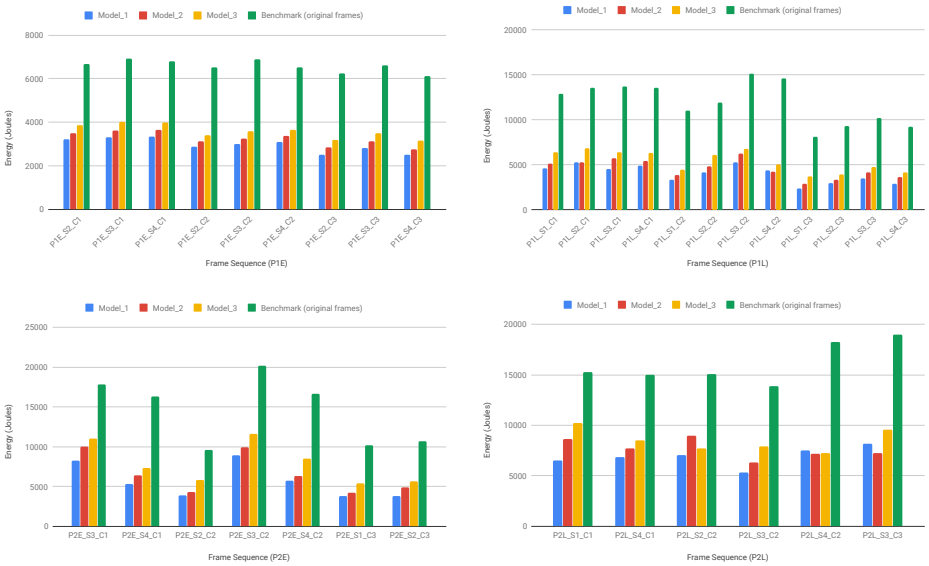


**Figure 8.** Total energy consumption regarding each evaluated model and the benchmark frame sequences from the P1E, P1L, P2E, and P1L datasets.

**Table 3.** Minimum, mean, and maximum energy reduction rates induced by our ARIMs compared to the provided benchmarks.

| Dataset | Energy Reduction | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Model 1 | | | Model 2 | | | Model 3 | | |
| | Min. | Mean | Max. | Min. | Mean | Max. | Min. | Mean | Max. |
| P1E | 0.505 | 0.551 | 0.598 | 0.463 | 0.508 | 0.550 | 0.414 | 0.456 | 0.489 |
| P1L | 0.612 | 0.667 | 0.711 | 0.582 | 0.619 | 0.710 | 0.490 | 0.548 | 0.657 |
| P2E | 0.536 | 0.610 | 0.672 | 0.439 | 0.549 | 0.619 | 0.381 | 0.454 | 0.551 |
| P2L | 0.533 | 0.571 | 0.618 | 0.406 | 0.516 | 0.620 | 0.332 | 0.464 | 0.603 |

evidences the viability of the proposed models and the conformity to our initial expectations regarding resources saving.

In future works, we intend to use our framework in other CV applications, such as surveillance and assembling line inspection. Another possibility is to represent the periphery of our models as super-pixel-like artifacts (voronoi cells) that could be filled with the grayscale pixel value at each cell's central point in the original image. The analysis of degraded peripheral regions represented in grayscale might be applied to the aforementioned application domains as well. Finally, we plan to integrate our approach into an FPGA, responsible for resampling uniform images according to some predefined or dynamic space-variant models. The models could be computed at the FPGA or by software, in which case an efficient communication mechanism between these layers should be implemented. Also, a more complex repertoire of variables would need to be considered, including the costs of computing the models and resampling in the FPGA, as well as the application's domain. Even with these variables in the field, we believe such an infrastructure could yield positive impacts in the energy saving.

**Author Contributions:** R.S.T., R.A., and S.G. conceptualized the study. E.S., S.G., and L.T.L. developed the proper software for simulations. E.S. conducted the majority of the writing, drafting, and production of data visualizations for the paper. Nevertheless, all authors contributed to the writing, review, and editing processes. E.S., R.S.T., A.P., and L.T.L. worked on the validation stages. J.E.S.V. and R.A. provided valuable contributions to the paper by discussing the benefits and implications of the idea regarding the hardware perspective. R.S.T., R.A.,

and S.G. supervised all stages of the study and were responsible for the funding acquisition. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by São Paulo Research Foundation – FAPESP (grants #2014/12236-1, #2016/50250-1, and #2017/20945-0) and the FAPESP-Microsoft Virtual Institute (grants #2013/50155-0 and #2014/50715-9). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Bailey, D.G.; Bouganis, C.S., Vision Sensor with an Active Digital Fovea. In *Recent Advances in Sensing Technology*; Mukhopadhyay, S.C.; Gupta, G.S.; Huang, R.Y.M., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2009; pp. 91–111. doi:10.1007/978-3-642-00578-7_6.
2. Bornholt, J.; Mytkowicz, T.; Mckinley, K.S. The model is not enough: Understanding energy consumption in mobile devices. *Power (watts)* **2012**, *1*, 3.
3. Wandell, B.A. *Foundations of Vision*; Sinauer Associates, Incorporated: United States, 1995.
4. Bolduc, M.; Levine, M.D. A Review of Biologically Motivated Space-Variant Data Reduction Models for Robotic Vision. *Computer Vision and Image Understanding* **1998**, *69*, 170–184.
5. Berton, F.; Sandini, G.; Metta, G., Anthropomorphic visual sensors. In *Encyclopedia of Sensors*; Grimes, C.; Dickey, E.; Pishko, M.V., Eds.; American Scientific Publishers, 2006; pp. 1–16.
6. González, M.; Sánchez-Pedraza, A.; Marfil, R.; Rodríguez, J.A.; Bandera, A. Data-Driven Multiresolution Camera Using the Foveal Adaptive Pyramid. *Sensors* **2016**, *16*. doi:10.3390/s16122003.
7. Smeraldi, F.; Bigun, J. Retinal vision applied to facial features detection and face authentication. *Pattern Recognition Letters* **2002**, *23*, 463 – 475. In Memory of Professor E.S. Gelsema, doi:https://doi.org/10.1016/S0167-8655(01)00178-7.
8. Traver, V.J.; Bernardino, A. A review of log-polar imaging for visual perception in robotics. *Robotics and Autonomous Systems* **2010**, *58*, 378–398.
9. Akbas, E.; Eckstein, M.P. Object detection through search with a foveated visual system. *PLOS Computational Biology* **2017**, *13*, 1–28. doi:10.1371/journal.pcbi.1005743.
10. Phillips, D.B.; Sun, M.J.; Taylor, J.M.; Edgar, M.P.; Barnett, S.M.; Gibson, G.M.; Padgett, M.J. Adaptive foveated single-pixel imaging with dynamic supersampling. *Science Advances* **2017**, *3*. doi:10.1126/sciadv.1601782.
11. Wen, W.; Kajínek, O.; Khatibi, S.; Chadzitaskos, G. A Common Assessment Space for Different Sensor Structures. *Sensors* **2019**, *19*. doi:10.3390/s19030568.
12. Goldenstein, S.; Vogler, C.; Velho, L. Adaptive Deformable Models for Graphics and Vision. *Computer Graphics Forum* **2005**, *24*, 729–741. doi:10.1111/j.1467-8659.2005.00898.x.
13. de Goes, F.; Goldenstein, S.; Velho, L. A Simple and Flexible Framework to Adapt Dynamic Meshes. *Computers & Graphics* **2008**, *32*, 141-148. doi:10.1016/j.cag.2008.01.009.
14. Wong, Y.; Chen, S.; Mau, S.; Sanderson, C.; Lovell, B.C. Patch-based Probabilistic Image Quality Assessment for Face Selection and Improved Video-based Face Recognition. IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops. IEEE, 2011, pp. 81–88.
15. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2001, Vol. 1, pp. I–511–I–518. doi:10.1109/CVPR.2001.990517.
16. King, D.E. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research* **2009**, *10*, 1755–1758.
17. Khan, K.N.; Hirki, M.; Niemi, T.; Nurminen, J.K.; Ou, Z. RAPL in Action: Experiences in Using RAPL for Power Measurements. *ACM Trans. Model. Perform. Eval. Comput. Syst.* **2018**, *3*. doi:10.1145/3177754.