Pupil Detection for Augmented and Virtual Reality based on Images with Reduced Bit Depths

Gernot Fiala
Institute of Technical Informatics
Graz University of Technology
Graz, Austria
ISS OSS SW
ams-OSRAM AG
Premstaetten, Austria
gernot.fiala@{tugraz.at, ams-osram.com}

Zhenyu Ye
ISS OSS SW
ams-OSRAM AG
Premstaetten, Austria
zhenyu.ye@ams-osram.com

Christian Steger
Institute of Technical Informatics
Graz University of Technology
Graz, Austria
steger@tugraz.at

***Abstract*—For future augmented reality (AR) and virtual reality (VR) applications, several different kinds of sensors will be used. These sensors, to give some examples, are used for gesture recognition, head pose tracking and pupil tracking. All these sensors send data to a host platform, where the data must be processed in real-time. This requires high processing power which leads to higher energy consumption. To lower the energy consumption, optimizations of the image processing system are necessary. This paper investigates pupil detection for AR/VR applications based on images with reduced bit depths. It shows that images with reduced bit depths even down to 3 or 2 bits can be used for pupil detection, with almost the same average detection rate. Reduced bit depths of an image reduces the memory foot-print, which allows to perform in-sensor processing for future image sensors and provides the foundation for future in-sensor processing architectures.**

***Index Terms*—pupil detection, smart image sensor, augmented reality, virtual reality, bit depth, in-sensor processing**

I. Introduction

Future AR/VR applications will contain several different sensors for gesture recognition, head pose tracking and pupil tracking. All these sensors send large amounts of the data to a host platform, where the data needs to be processed in real-time. Therefore, processing power and the corresponding energy consumption is an important factor by designing new AR/VR applications. As stated by Liu et al. [1] future image sensors will contain a sensing layer stacked with a processing layer. The first intelligent image sensor was developed by Sony in the year 2020 [2]–[4]. A neural network (Mobilenet V1) can be processed directly on the sensor for real-time object detection/tracking. This sensor used such a stacked configuration with a pixel chip and a stacked logic chip. The memory and the digital signal processor (DSP) for processing the neural network are parts of this logic chip. Another important factor is the communication between the different processing levels, described by Liu et al. in [5]. Future mobile AR/VR devices could replace today's smartphones. These future devices require different sensors and communications to different processing levels. Traditional image sensors and image processing systems can not meet the requirements of power consumption and performance for these next generation AR/VR devices [5]. Therefore, a combination of new image sensor technologies and optimized in-sensor processing are required. Since area and processing power for in-sensor processing is very limited, optimizations of the image processing system and of the processing architectures are necessary.

The main contributions of this work are:

- A small pupil detection dataset with 105 images with ground truth pupil center coordinates. The dataset was extended by images with reduced bit depths from 8 bits down to 1 bit.
- Demonstrating the fact that the bit depth for pupil detection can be reduced even down to 3 or 2 bits while maintaining almost the same average detection rate compared to 8 bits.
- Demonstrating that image sensors with reduced bit depths can be the foundation for novel hardware architectures. It enables in-sensor processing, which is of significant importance for future AR/VR devices.

This paper is organized as follows: Section II shows related work of smart image sensors in the field of AR/VR. Section III gives an overview of the dataset for pupil detection with reduced bit depths. Section IV shows the methodology of the pupil detection with reduced bits per pixel of the images and the evaluation metrics. Section V presents the results and Section VI concludes the paper.

II. Related Work

In the last few years the research of smart or intelligent image sensor is growing. Also the demand for smart image sensors is increasing due to upcoming AR/VR applications.

As stated by Liu et al. in [5] communications between the processing levels have a big impact on the power consumption. Therefore, research is trying to reduce the communication from lower levels to higher levels.

One approach is to reduce the transmitted data by data compression as described by Pinkham et. al in [6]. A neural network is used to compress the data into a transmission map.

This research was funded by ams-OSRAM AG. Thank you.

This saves energy of the communication on common wired and wireless interfaces, but the total overall energy stays similar.

A related approach is to use smart communication interfaces, which switch into a low-power mode when no transmission is required, as stated by Fiala et al. in [7]. In addition, sending regions-of-interest (ROI) data lowers the communication power consumption. Furthermore, with in-sensor processing a more energy efficient host can be used for the image processing system.

Another method is to split the image processing system into two parts, one for in-sensor processing and one for the host platform as described by Pinkham et. al [8]. Neural networks can be split up to lower the overall power consumption. This two-processor system approach is more suitable for smaller neural networks.

Optimizing an algorithm as described by Fiala et al. [9] to only integer operations, allows to remove the floating-point units (FPU) of system-on-chip (SoC) designs for smart image sensors. Since area and processing power are limiting factors for in-sensor processing, optimizing an algorithm is a key approach, which can also be applied to neural networks by using quantization of the neural networks. The most famous framework is TensorFlow-Lite [10] but also other frameworks are developed like Apache TVM [11]. They optimize the neural network by quantization of the 32-bit float values to 8 bit integers. With this optimization the memory footprint is reduced. Apache TVM is a machine learning compiler framework, which optimizes the code for specific hardware. Furthermore, neural networks can be compressed by using a sparse-structured matrix format, described by Mishra et. al shown in [12]. A 2:4 sparsity pattern is introduced, which increases the math throughput of dense matrix units with a factor of 2. All of these methods are very important for developing next generation AR/VR image sensors.

In this work, the required bit depth for pupil detection is investigated and how far the bit depth can be reduced while maintaining almost the same average pupil detection rate. A computer vision (CV) based pupil detection algorithm is used to detect the pupil center location. The pixel error is calculated and the average detection rate is evaluated. A small dataset with 105 images and ground truth data was generated and further a dataset with reduced bit depths from 8 bits to 1 bit was processed.

III. Pupil Detection Dataset

In this work, a synthetic dataset was generated from a 3D model. Synthetic eye data for pupil detection is used for state-of-the-art eye tracking algorithms. These synthetic eye data can be generated much faster from 3D models, which are based on 3D model frameworks such as Unity[1] [13], [14] or Blender[2] [19], [20]. There are several datasets for pupil detection, gaze tracking and eye segmentation [15]–[19], but we decided to generate our own dataset based on Blender to be more flexible for future research. In this work the base model was developed by Swirksi et al. [19], [20]. It was extended with 3 additional iris colors for the eye and up to 8 light reflections (glints) can be selected. For this dataset, only the left eye with an image resolution of 500x500 and 8 glints are used. Furthermore, a 3D to 2D mapping of the pupil center coordinates was implemented and the 2D pupil center coordinates are saved during the rendering process. Blender 2.78c [21] was used to render the dataset. Some images of the dataset are shown in Fig.1. The rendered images are gray scale images. The iris colors have different gray levels from nearly black to brighter gray values. The iris colors are labeled from left to right as nearly black, dark, dark brown, light and light brown. The label names are used in the following sections. Furthermore, the eyelid position, gaze direction, number of glints (up to 8) and pupil size can be selected.

Fig. 1: Example images from the rendered dataset with all five different iris colors, labeled as nearly black, dark, dark brown, light and light brown (left to right), different eyelid positions, gaze directions and pupil sizes with 8 glints.

The rendered dataset consists of 105 images with a bit depth of 8 bits. They were used to generate additional datasets (subsets) for 1, 2, 3, 4, 5, 6 and 7 bits. The bit values were normalized between 0 and 255. An example for each iris color with the reduced bit depths is shown in Fig. 2. The number of bits per pixel are reduced from 8 bits (left) down to 1 bit (right). The labeled color values are nearly black, dark, dark brown, light and light brown from top to bottom.

IV. Methodology

As a first investigation in the reduction of the bit depths, an open source implementation [22] of the pupil tracker algorithm from Swirksi et al. [23] was used. The algorithm is an edge-based pupil detection algorithm and was developed in 2012. This algorithm is sufficient to analyze the pupil detection rate based on images with reduced bit depths. It was modified to process whole directories with images and to save the calculated pupil center coordinates. These calculated pupil center coordinates were compared with ground truth data to derive the pixel error.

The processing pipeline is shown in Fig.3. First, the dataset is generated with the ground truth data. The images are adjusted to different bit depths, from 8 bits down to 1 bit, and normalized to values between 0 and 255. We processed the pupil center coordinates with the algorithm 10 times for all

[1]https://unity.com/

[2]https://www.blender.org/

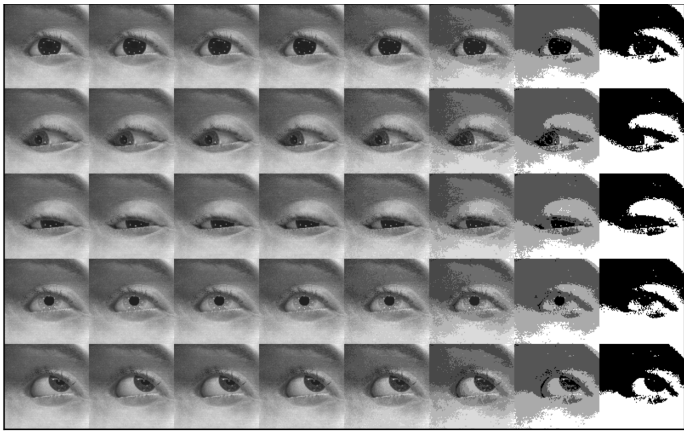

Fig. 2: Examples of the dataset with the five different iris colors and the corresponding images with the reduced bit depth. From left to right the bit depth is reduce from 8 bits to 1 bit.

images and all bit depths. Then, the pixel error was calculated based on the output of the algorithm and the ground truth data. The pixel error is the Euclidean distance from the ground truth pupil center coordinates to the calculated pupil center coordinates. Based on the pixel error, the average detection rate of the pupil detection algorithm was calculated. This was done for all bit depths. The average detection rate was used as the evaluation metric. It shows how often the pupil center was detected for the given pixel error value. Usually, pixel errors up to 5 are rated as correct detection.

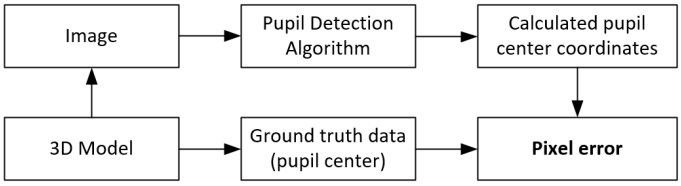

Fig. 3: Processing pipeline from dataset generation to pixel error calculation.

The algorithm calculates an integral image and uses Haar-like features for a given radius to find the strongest response for a possible pupil region. To approximate the pupil location, intensity-based segmentation is used. An image histogram is calculated and k-means clustering is used to segment the histogram into 2 clusters. The dark cluster is assumed to be the pupil and the other cluster represents the background. Then, a binary image is calculated to find connected components, where the largest represents the pupil. To perform ellipse fitting, some image pre-processing steps are required. A morphological 'open' operation is used to remove features like eyelashes and glints. Then, the edges between the pupil and the iris are calculated with a Canny edge detector. From the detected edge points, 5 are randomly taken to perform Random Sample Consensus (RANSAC) to fit an ellipse. The center position of the best ellipse fit is finally taken as pupil center position. A more detailed description of the algorithm can be found in [23].

Since some iris colors are very dark, the edges between the pupil and the iris are not detected and instead of the pupil, the iris region is found with the Haar-like features. Therefore, the minimum and maximum radius must be set accordingly. Three different settings for the radii were used, shown in Table I. Since the pupil tracker algorithm randomly takes edge points for the ellipse fitting step, all images were processed 10 times. The average pixel error and the average detection rate were calculated. The results are shown in the next section.

TABLE I: Parameters for Haar-like features with minimum and maximum radius applied to all images of the whole dataset.

Minimum radius	Maximum radius
17	38
17	85
55	85

V. Results

In this section, the average detection rate of the pupil detection algorithm for images with different bit depths is discussed. The average detection rate across the whole dataset for all three different radii settings is shown in Fig. 4. The average detection rate is for bit depths from 8 bits downto 4 bits very close to each other. For 1 bit depth, we expected the results to show a low detection rate, since the images are only black and white. For a smaller range of the radii, the algorithm has a very small drop in the average detection at 4 bits and a bigger drop at 3 and 2 bits, shown in Fig. 4a. For a bigger radii configuration the average detection rate is similar for 8 bits down to 2 bits, shown in Fig. 4a and Fig. 4c. With a radii configuration of 55/85, the best results are with 2 and 3 bits at a pixel error of 5.

Also, the results based on the iris colors are analyzed. The best detection results for all 5 different iris colors are shown in Fig. 5. For the nearly black iris, Fig. 5a, the radii settings are 55/85. This allows to fit around the iris edges, since the pupil edges are not detected. However, the algorithm nearly gives the same results for images with reduced bit depths down to 2 bits. For the dark brown iris color it is interesting to see, that for 2 and 3 bits the average detection rate is the highest at a pixel error of 5, shown in Fig. 5b. A similar behaviour can be observed in Fig. 5c. The detection rate for the dark iris is better for images with bit depths of 4 bits and higher, shown in Fig. 5d. However, for a different radii configuration, the average detection rate goes down. Interestingly, with a pixel error of 5, 2 bits also result in almost the same average detection rate, shown in Fig. 5e. The best average detection rate is for the light iris, because the edges between the pupil and the iris are very sharp. Also the detection rate for 8 bits down to 2 bits is very similar, shown in Fig. 5f. In total, the results are very similar across the images with the different bit depths. However, the brighter the iris, the better the detection rate (at least with this algorithm). This work shows that pupil

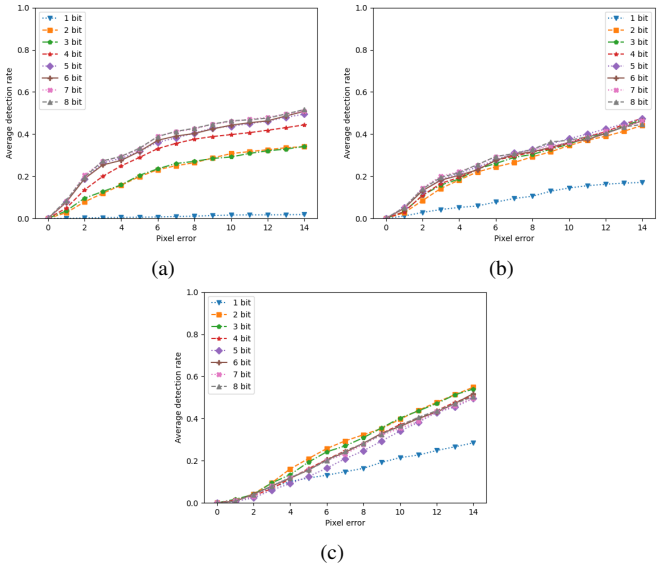

Fig. 4: Average detection rate for the given pixel error across the whole dataset with different bit depths. (a) shows the average detection rate for radii configuration 17/38. (b) shows the average detection rate for radii configuration 17/85 and (c) shows the average detection rate for radii configuration 55/85.

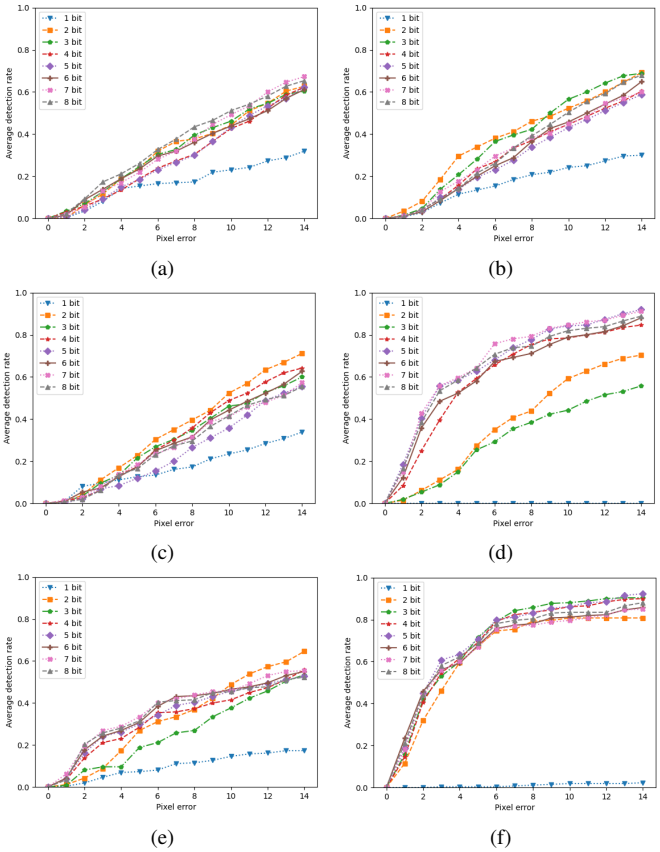

Fig. 5: Average detection rate across subsets of the dataset based on iris color. (a) shows the highest average detection rate for the nearly black iris color with a radii configuration 55/85. (b) shows the average detection rate for the dark brown iris color with radii configuration 55/85. (c) shows the average detection rate for the light brown iris with a radii configuration 55/85. (d) shows the average detection rate for the dark iris with a radii configuration 17/38. (e) shows again the average detection rate for the dark iris but with a radii configuration 17/85. (f) shows the average detection rate for the light iris with a radii configuration 17/38.

detection works based on images with reduced bit depths. The best reduced bit depths is 4 bits, but for different configurations bit depths even down to 2 bits can be used. This points to a promising direction to design and develop novel hardware architectures, which can be used for in-sensor processing in resource-constrained smart image sensors for pupil detection.

VI. Conclusion and Future Work

This paper introduced a new dataset for pupil detection with ground truth data and an extended dataset with images of different bit depths from 8 bits down to 1 bit.

It shows, based on an edge-based pupil detection algorithm, that pupil center coordinates can be detected in images with reduced bit depths. The experiments show, that a reduction of the bit depth is possible to 4 bits or in some cases even down to 3 or 2 bits while maintaining almost the same average detection rate compared to images with 8 bits.

The reduction of the bit depths allows to reduce the memory footprint for images and can be the foundation for developing novel hardware architectures, especially for in-sensor processing for resource-constrained image sensors. Smart image sensors specialized for pupil detection can be designed for future AR/VR devices.

Future steps are to show that the same approach can be applied to neural networks. Therefore, during writing this paper, we render a much bigger dataset for pupil detection, which we can use for training neural networks.

Acknowledgment

A big thank you goes to the colleagues from TUGraz and ams-OSRAM AG who supported this work with discussions and feedback.

References

[1] C. Liu, M. Hall, R. De Nardi, N. Trail, R. Newcombe, "Sensors for Future VR Applications", 2017 International Image Sensor Workshop (IISW), pp. 250–253, 2017.

[2] Sony Corporation, "Sony to Release World's First Intelligent Vision Sensors with AI Processing Functionality", https://www.sony.com/en/SonyInfo/News/Press/202005/20-037E/, 14 May 2020, [Online; accessed 22 July 2021].

[3] Sony Group Corporation, "Sony's Latest Image Sensors and the Technologies that Lies Behind Them", https://www.sony.com/en/SonyInfo/technology/stories/imagesensor7tech/, 15 October 2020, [Online; accessed 22 July 2021].

[4] Sony Semiconductor Solutions Corporation, "Imaging and Sensing Technology", https://www.sony-semicon.co.jp/e/technology/imaging-sensing/, 18 February 2021, [Online; accessed 22 July 2021].

[5] C. Liu, A. Berkovich, S. Chen, H. Reyserhove, S.S. Sarwar, T.-H. Tsai, "Intelligent Vision Systems – Bringing Human-Machine Interface to

AR/VR", 2019 IEEE International Electron Devices Meeting (IEDM), pp. 10.5.1–10.5.4, 2019, doi: 10.1109/IEDM19573.2019.8993566.
[6] R. Pinkham, T. Schmidt, A. Berkovich, "Algorithm-Aware Neural Network Based Image Compression for High-Speed Imaging", 2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), pp. 196–199, IEEE Computer Society, 2020, doi: 10.1109/AIVR50618.2020.00040.
[7] G. Fiala, J. Loinig, C. Steger, "Impact of image sensor output data on power consumption of the image processing system", Intelligent Systems Conference 2022. IntelliSys 2022, in press.
[8] R. Pinkham, A. Berkovich, Z. Zhang, "Near-Sensor Distributed DNN Processing for Augmented and Virtual Reality", IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 4, no. 4, pp. 663–676, 2021, doi: 10.1109/JETCAS.2021.3121259.
[9] G. Fiala, J. Loinig, C. Steger, "Evaluation of an Integer Optimized Shape Matching Algorithm", 2021 IEEE Sensors Applications Symposium (SAS), pp. 1–6, 2021, doi: 10.1109/SAS51076.2021.9530015.
[10] Google Brain Team, "TensorFlow-Lite", https://www.tensorflow.org/lite.
[11] Apache Software Foundation, "Apache TVM", https://tvm.apache.org/download.
[12] A. K. Mishra, J. Albericio Latorre, J. Pool, D. Stosic, D. Stosic, G. Venkatesh, C. Yu, P. Micikevicius, "Accelerating Sparse Deep Neural Networks", ArXiv, 2021, doi: 10.48550/ARXIV.2104.08378.
[13] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, A. Bulling, "Learning an Appearance-Based Gaze Estimator from One Million Synthesised Images", in Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, pp. 131–138, 2016.
[14] E. Wood, T. Baltrusaitis, X. Zhang, Y. Sugano, P. Robinson, A. Bulling, "Rendering of Eyes for Eye-Shape Registration and Gaze Estimation", in Proc. of the IEEE International Conference on Computer Vision (ICCV 2015), 2015.
[15] S. J. Garbin, O. Komogortsev, R. Cavin, G. Hughes, Y. Shen, I. Schuetz, S. S. Talathi, "Dataset for Eye Tracking on a Virtual Reality Platform". In ACM Symposium on Eye Tracking Research and Applications. pp. 1—10, 2020, doi: 10.1145/3379155.3391317.
[16] J. Kim, M. Stengel, A. Majercik, S. De Mello, D. Dunn, S. Laine, M. McGuire, D. Luebke, "NVGaze: An Anatomically-Informed Dataset for Low-Latency, Near-Eye Gaze Estimation", In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1–12, 2019, doi: 10.1145/3290605.3300780.
[17] M. Tonsen, X. Zhang, Y. Sugano, A. Bulling, "Labelled Pupils in the Wild: A Dataset for Studying Pupil Detection in Unconstrained Environments", In Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, pp. 139–142, 2016, doi: 10.1145/2857491.2857520.
[18] W. Fuhl, G. Kasneci, E. Kasneci, "TEyeD: Over 20 Million Real-World Eye Images with Pupil, Eyelid, and Iris 2D and 3D Segmentations, 2D and 3D Landmarks, 3D Eyeball, Gaze Vector, and Eye Movement Types", IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 367–375, 2021, doi 10.1109/ISMAR52148.2021.00053
[19] L. Swirski and N. Dodgson, "Rendering synthetic ground truth images for eye tracker evaluation", In Proceedings of Eye Tracking Research and Applications Symposium (ETRA), pp. 219–222, March 2014, doi: 10.1145/2578153.2578188.
[20] L. Swirski, "Eyemodel", https://github.com/LeszekSwirski/eyemodel.
[21] Blender Foundation, 2017. Blender 2.78c, https://download.blender.org/release/Blender2.78.
[22] L. Swirski, "Pupiltracker", https://github.com/LeszekSwirski/pupiltracker.
[23] L. Swirski, A. Bulling and N. Dodgson, "Robust real-time pupil tracking in highly off-axis images", In Proceedings of Eye Tracking Research and Applications Symposium (ETRA), pp. 173–176, March 2012, doi: 10.1145/2168556.2168585.