Analog Image Processing Using Gaussian Based Filtering for Motion Detection Applications

Mecid Alhamuy[1], Burcu Erkmen[2]

[1]Yildiz Technical University, Istanbul, Turkey
mecidalhamuy89@gmail.com
[2]Yildiz Technical University, Istanbul, Turkey
bkapan@yildiz.edu.tr

Abstract

A novel approach to motion detection is presented in our study, utilizing a 3×3 Gaussian image filtering technique. Inspiration is drawn from the human visual system, particularly the human retina, and a straightforward Complementary Metal Oxide-Semiconductor (CMOS) circuit design is employed. Gaussian image filtering and noise mitigation are achieved through convolution and masking operations. The challenge lies in translating these mathematical operations into hardware using basic electronic components like current mirror and adder circuits. Additionally, enhancements are made to the 2-D pixel sensor array, optimizing the commonly used pixel circuitry.

Our imaging technique conducts preliminary image processing before analog-to-digital conversion, supporting subsequent digital image processing stages in camera systems. We introduce an 8-MOSFET filtering cell array for retinomorphic analog image filtering. These cells collaborate to filter image signals received from the pixel circuit array. Our analysis is based on two moving scenes with a resolution of 150×150 pixels, and we compare the results with theoretical outcomes from digital image processing techniques. We utilize CMOS $0.18\mu m$ technology parameters for our proposed analog circuit structure.

1. Introduction

In this study, we aim to improve imaging systems by optimizing their output and reducing frame processing time. Common imaging systems predominantly process images digitally due to ease of implementation and data storage [1]. However, digital processing can introduce time delays [2]. Analog image processing, on the other hand, offers faster response times and shares similarities with the human eye's biological imaging system. There is growing interest in mimicking biological image processing and developing bio-inspired retinomorphic optical sensing technology to replicate the efficiency of the human retina. Researchers like Effrosyni et al. [3] have mathematically demonstrated retinal image processing using Gaussian Weighted Difference (WDoG) filters, while Sengupta et al. [4] have implemented a retinomorphic system in Matlab for transforming videos into sparse dynamic events using peak detection and selectable filters. Katic et al. [5] demonstrated a retinal-inspired edge detection imager capable of quantizing high-frequency images over multiple frequency bands, providing detailed image reconstructions. Yildirim et al. [6] developed an analog image processing chip utilizing current mirror circuits as a 2-D Laplace filter for edge detection and enhancement. There are also innovative approaches like Song et al.'s [7] Processing in Pixel (PIP) scheme, which applies convolution operations before transmitting the readout signal to improve image reading speed with lower power consumption. Our goal is to alleviate the burden on image processors by shifting a portion of the image processing from digital to the analog component of the system. Gaussian image filtering is a fundamental method that typically requires digital software due to the substantial input data involved. In our study, we delve into Gaussian image filtering without digital processing, harnessing an analog electronic circuit inspired by the biological retina's intricate interconnections between photoreceptors and neighboring cells. This novel approach not only achieves noise reduction but also enables effective edge detection. Our work paves the way for future projects, encouraging the development of innovative hardware implementations rooted in Gaussian filtering principles.

The structure of this article is as follows: Section 2 provides an overview of the Gaussian filter principle. Section 3 delves into the proposed system architecture. Section 4 analyzes and discusses the experimental results, and the concluding section summarizes the findings.

2. Gaussian Smoothing

Gaussian smoothing, also known as Gaussian blur, is a common image processing technique used to simplify images and reduce noise [8]. Its name comes from the use of the Gaussian function to smooth the image. The primary purpose of this Low Pass Filter (LPF) is to reduce the variation in high-frequency image data, which can be disruptive to certain image feature extraction algorithms [9]. Mathematically, Gaussian smoothing involves convolving the image with a Gaussian function. To construct the kernel for this convolution, we use Equation (1) for the 1-D Gaussian function:

$$G(x) = \frac{1}{\sqrt{2 * \pi * \sigma^2}} * e^{-\frac{x^2}{2*\sigma^2}} \tag{1}$$

Here, π represents the standard deviation. For image processing, we use the 2-D Gaussian function, as shown in Equation (2):

$$G(x, y) = \frac{1}{2 * \pi * \sigma^2} * e^{-\frac{x^2+y^2}{2*\sigma^2}} \tag{2}$$

Both Equations (1) and (2) indicate two critical parameters for constructing a Gaussian smoothing kernel: dimensions (x and y) and variance (σ^2). Properly setting these kernel parameters for the specific scene significantly affects the performance of Gaussian filtering. Incorrect settings can result in distorted images and hinder the intended purpose [10]. Inaccurate Gaussian filtering can lead to issues such as edge position displacement, disappearing edges, or phantom edges. Despite these challenges, Gaussian filtering remains effective for several reasons [5, 11–13]:

1. Gaussian filtering excels at removing noise that disrupts the continuity of edge lines, making it preferable for edge detection applications.
2. Gaussian smoothing is widely used in image processing studies because it uses a 2-D distribution as a "point-spread" function, simulating optical blurring in the human visual system.
3. The rotationally symmetric shape of the Gaussian function's kernel matrix helps negate mispositioning issues.
4. Gaussian blur plays a role in averaging pixel weights with their surroundings, making images suitable for downsampling. It's commonly used in data compression to reduce image sizes.
5. Adjusting the smoothing parameters for Gaussian blur is straightforward, mainly relying on σ, where a higher σ value results in more blur.

2.1. Digital Processing for Gaussian Smoothing

To translate Gaussian blur from mathematical operations into hardware circuitry, a comprehensive understanding of it's implementation in digital processing techniques is essential. A fundamental operation involved in this process is convolution, which combines two functions to create a modified function that blends their shapes. Convolution refers to both the result and the process itself. Mathematically, it's an integral operation between two functions: one called the input signal (F), representing the input image, and the other known as the mask, filter, or kernel (H). The result of convolution (G) can be used for tasks like edge enhancement, sharpening, or blurring in image applications. The mathematical equations for convolution are given in both frequency (3) and time domains (4):

$$G = F * H \qquad (* : \textit{Convolution Operator}) \tag{3}$$

$$g(t) = \int_{-\infty}^{\infty} f(t-\tau) * h(\tau)\, d\tau \tag{4}$$

Here, t represents time, and τ denotes the variable during convolution. Implementing convolution in hardware may seem complex due to the integral involving multiplication, addition, and

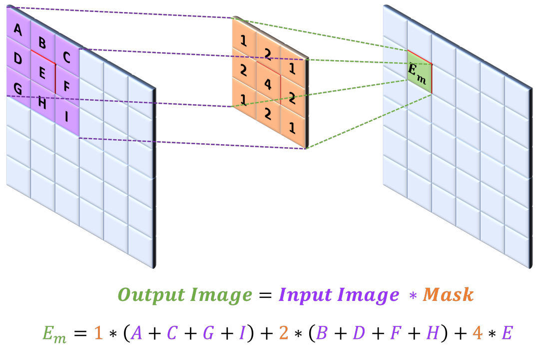

Fig. 1. The mathematical explanation of the 2-D image convolution operation

subtraction. However, it can be simplified as a sum of product operations, assuming that the pixel signals in the image are discrete time quantities (which they are in a single frame). In this context, convolution can be defined as shown in Equation (5):

$$g[x, y] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u, v] * f[x-u, y-v] \tag{5}$$

Where x and y are horizontal and vertical coordinates, and u and v represent variables during convolution. Fig. 1 illustrates the mathematical concept of 2-D image convolution, where (A, B, C, D, E, F, G, H) represent selected input image pixel signals multiplied by a 3×3 Gaussian kernel ($\sigma = 1$), chosen for this system. The result (E_m) represents the filtered pixel signal. The goal of digital filtering techniques is to enhance specific image characteristics, allowing the filtered image to be used in further processing algorithms. As mentioned earlier, Gaussian smoothing is advantageous for motion edge detection in noisy scenes. It prioritizes multiple moving pixels in small regions over isolated pixels scattered in the background, making it effective for highlighting moving object edges. In this work, we employ Gaussian filtering with a 3×3 masking matrix. The matrix coefficients follow the Gaussian function described in Equation (2), with the standard deviation playing a crucial role in edge contrast. A lower standard deviation emphasizes edges by giving higher coefficients to center pixels. Fig. 1 displays the used masking matrix in our circuit, featuring a standard deviation of 1 ($\sigma = 1$). We have chosen to explain Gaussian filtering using a small masking matrix to avoid complex circuitry associated with larger scales.

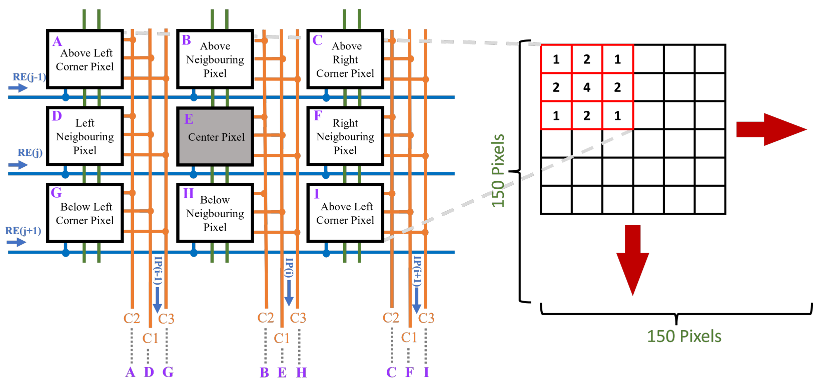

Fig. 2. Connections between center and neighboring pixels to be able to transmit the pixel data to the Gaussian filter

3. System Architecture

To implement Gaussian filtering within the analog circuit structure, we partition the process between the pixel array and the Gaussian filtering module.

3.1. CMOS Image Sensor Array

Fig. 2 illustrates a 3×3 block from the proposed pixel array module. This design differs from conventional CMOS image sensor arrays used in prior scientific studies [14] because its distinctive role lies in serving a large size of pixel signals to the Gaussian filter. In standard pixel arrays, a decoder selects a single pixel line for reading. However, Gaussian smoothing requires data from all 9 pixels in a 3×3 block. In this design, we have reconfigured the pixel array so that selecting one pixel row activates both the rows above and below it. This simultaneous activation is achieved through interconnections between pixel circuits in the array (see Fig. 3 for the pixel circuit and its symbol). When a pixel row is selected using the Read Enable $RE(j)$ signal, its corresponding subcircuit activates neighboring pixels both above and below. As a result, all three rows collectively participate in the image read-out process. This configuration draws inspiration from observed interactions among groups of visual neurons in the inner retinal layers, where a triggered neuron in the center interacts with neighboring neurons. Although our pixel array's interaction mechanism differs from the biological structure, it shares an activation relationship between the center and neighboring pixels. After the read-out process, the generated pixel signals traverse through the columns of the pixel array. Each column, denoted as $IP(i, j)$, accommodates three base data lines for the activated pixels in that column—left lines for the top row, right lines for the bottom row, and the center column for the center row. $M1$, $M2$, and $M3$ transistors serve as switches, controlling the flow of current from the pixel sensor input. The activation input received by the subcircuit determines the path for these data signals. They are then routed through the respective baselines and proceed through the Correlated Double Sampling (CDS) and amplifier circuits before reaching the Gaussian filtering module.

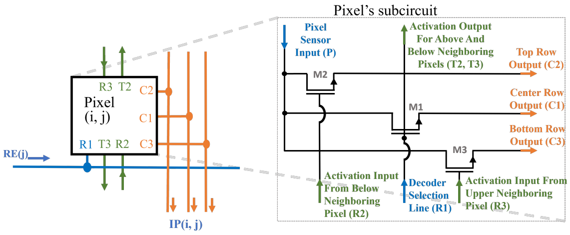

Fig. 3. The added subcircuit to the common pixel structure

3.2. Gaussian Filter Module

This module is responsible for applying Gaussian smoothing to selected pixels. It represents the analog circuitry that converts mathematical convolution operations into hardware. The design of the proposed module includes a row of masking cells, as illustrated in Fig. 4. Given that the read-out signals from the pixel array are transmitted row by row, the size of the Gaussian filter register aligns with the number of pixels in a row. Each masking cell is linked to a column from the pixel array and receives input from three pixels simultaneously. In accordance with Fig. 2, the baseline of the center pixel row is processed in the masking cell with a different multiplying coefficient compared to the top and bottom rows, which is why they are received through separate inputs. This 3-input feature in the masking cell also allows the processing of the column of three-pixel signals at once. Within the filtering register, each masking cell generates three outputs: one primary and two adjacent side outputs. The primary output shares a node with the two side outputs from neighboring cells on the left and right. The result of this interconnection node emerges from the module's terminal as the output of a filtered pixel. The internal

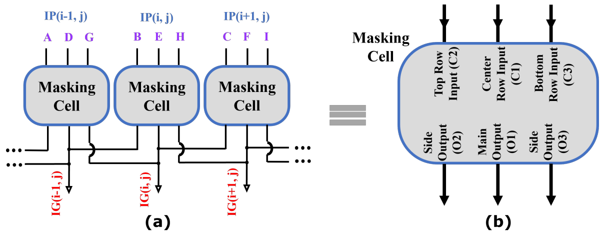

Fig. 4. (a) A 3 Masking cell block depicted from Gaussian filter's register (b) Masking cell's symbol

circuit of the masking cell is detailed in Fig. 5. Each cell's circuit comprises 4 NMOS and 4 PMOS transistors, forming three current mirror structures ($CM1$, $CM2$, $CM3$). It has been previously discussed that the convolution operation in image processing can be managed as outlined in Equation (5). However, executing the entire process within a single block is complex. In this design, we divide the convolution operation into multiplication and addition components. In the multiplication phase, the input pixel signals are multiplied by the coefficients assigned to their positions in the Gaussian kernel matrix, as depicted in Fig. 1. Current mirrors possess the capability to multiply current signals. Their circuit structures can be adjusted by altering the width-to-length ratio (W/L) of the MOSFET channels to align with their intended masking coefficients. Table 1 provides details on the components of the designed analog Gaussian filter circuit, including MOSFET transistor W/L ratios, input current value ranges, and other relevant values.

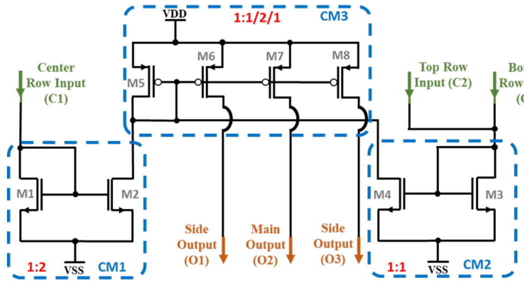

Fig. 5. Masking cell's internal circuit

Table 1. Masking cell circuit components and input current values list

V_{dd}		$1.8V$
V_{ss}		$0V$
W/L	M1, M3, M4	$18\mu m/1.8\mu m$
	M2	$36\mu m/1.8\mu m$
	M5, M6, M8	$140\mu m/1.8\mu m$
	M7	$280\mu m/1.8\mu m$
Input Current Range		$270 - 360\mu A$

The masking cell operates as follows: $CM1$ functions as a current amplifier, doubling the input current from the center row. $CM2$, on the other hand, maintains a product factor of 1 with summing the data currents from the top and bottom rows. Both $CM1$ and $CM2$ converge at $CM3$, which features three outputs, corresponding to $M6$, $M7$, and $M8$ transistors. These outputs have product factors of $1 : 2 : 1$, respectively. The outputs with a factor of 1 are shared side outputs with neighboring masking cells, while the output with a factor of 2 serves as the main output. Regarding the addition phase of the convolution operation, it is achieved by connecting MOSFET transistors in parallel. Since the outputs of $CM3$ are interconnected with those of their neighbors, every set of three adjacent cells generates a single filtered pixel data signal. This output results from the summation of the main output from the center cell and one side output from each neighboring cell.

4. Analysis Results

In this section, we discuss and analyze the results of our experiment using the proposed design. We applied the design to two different scenes, each containing dynamic properties from two consecutive frame images. Initially, we generated theoretical results for the 3×3 Gaussian filter using software algorithms and then compared them with our analysis results. After discussing this comparison and presenting mathematical calculations, we examined the differences between our calculations and those from related works in the literature.

The data transfer in the analog part of our camera circuit is based on current. We expect our design to handle input current values ranging from $270\mu A$ to $360\mu A$, corresponding to the common pixel intensity range of 0 to 255. In our experiment, we extracted moving edges from input frames using the proposed 3×3 Gaussian filter circuit, resulting in a binary image where pixels are categorized as moving (logic 1) or stationary (logic 0). The theoretical and analysis results for two scenes, one featuring a walking man and the other a boy playing with a ball, are presented in Fig. 6 and 7, respectively. These figures include the original grayscale input image, the used spatial kernel for filtering, Theoretical result, our analysis results and the result image without any filtering. Evaluating the new image is important to understand the costs and benefits of using the proposed design. Production time and image quality are key factors to consider in demonstrating the value of an image processing design. Fortunately, time measurement is irrelevant since the analog image processing design smooths the received pixel data instantly, which is a standard advantage of analog processing. Regarding the quality of the image factor, We assessed image quality by comparing our design's images with digitally filtered ones using the Structural Similarity Index Metric (SSIM) and Peak Signal to Noise Ratio (PSNR) metrics. SSIM is defined as follows:

$$SSIM = \frac{(\overline{xy} + C_1)(2\sigma_{xy} + C_2)}{(\sigma_x^2 + \sigma_y^2 + C_2)((\overline{x})^2(\overline{y})^2 + C_1)} \tag{6}$$

Where C_1 and C_2 are constants, x and y are datasets of theoretical and analytical results, and $\overline{x}$, $\overline{y}$, (σ_x^2), (σ_y^2), and σ_{xy} are defined as follows:

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{7}$$

$$\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{8}$$

$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2 \tag{9}$$

$$\sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \overline{y})^2 \tag{10}$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y}) \tag{11}$$

Mean Square Error (MSE) is a widely used metric for assessing image quality. However, in the case of binary images, MSE is not well-suited for our design due to certain limitations. MSE primarily focuses on pixel intensity differences, which may not align with the characteristics we aim to evaluate in binary images, where the primary information of interest often pertains to the spatial arrangement and distribution of 'white' and 'black' pixels. To address this, we turn to derivative metrics like Peak Signal to Noise Ratio (PSNR). PSNR measures the ratio between the

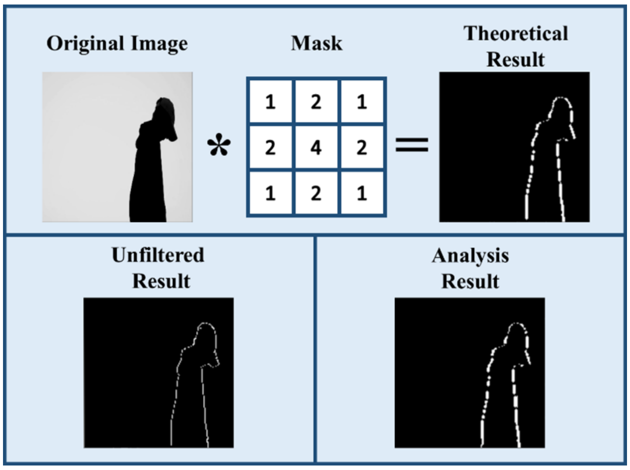

Fig. 6. Comparing the outcomes for a scene featuring a walking man

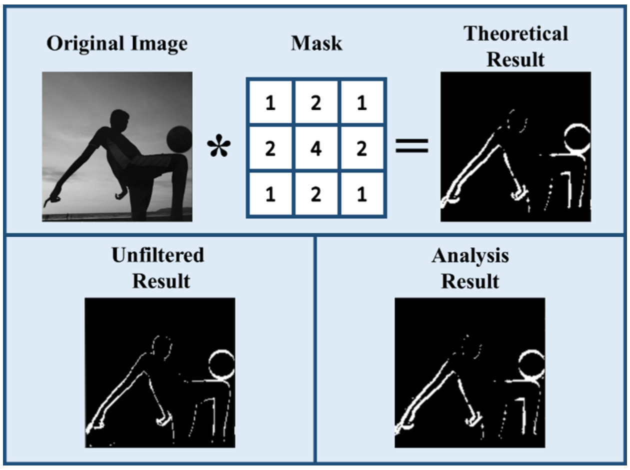

Fig. 7. Comparing the outcomes for a scene featuring child playing with a ball

maximum possible pixel value and the root mean square error, providing a measure of image fidelity.

$$PSNR = 10\log\frac{(2^8-1)^2}{\sqrt{MSE}} \tag{12}$$

Where MSE is defined as:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(x(i,j)-y(i,j))^2 \tag{13}$$

Here, i and j denote the spatial coordinates of the pixels, M and N represent the dimensions (width and height) of the image frame, and $x(i,j)$ and $y(i,j)$ signify the theoretical and analytical results of the image, respectively. The quality metrics used for comparing the analysis results with the theoretical results are shown in Table 2. We also collected additional results from other works in the literature concerning analog image processing techniques.

Table 2. Similarity metrics between theoretical results and analysis results for 2 different scene

When $x(i,j) = y(i,j)$	
SSIM	1
PSNR	∞
Walking Man	
SSIM	0.9865
PSNR	25.74031
Child playing with a ball	
SSIM	0.9657
PSNR	22.95278
Average Performance in [2]	
SSIM	0.7578
PSNR	24.6398

5. Conclusion

In this paper, we introduced a novel analog image processing technique inspired by Gaussian filtering. Our circuit design is versatile and can find applications in various scenarios, but our analysis here focused on its effectiveness in detecting moving edges. We achieved this technique using simple analog circuits, including current mirrors, and made some adjustments to the pixel array. We collected results from the analysis of binary images generated from real scenes and evaluated them using SSIM, PSNR metrics. Our experiments demonstrated that our straightforward analog circuit design produced results highly similar to those obtained from digital image processing software. This underscores the capability of our proposed design to execute Gaussian filtering rapidly, making it suitable for dynamic object detection applications. Performing early imaging processing before analog-to-digital conversion can enhance the support for digital imaging components within camera systems. The continued development of innovative image processing techniques remains pivotal for achieving high-performance outcomes in applications involving artificial intelligence and other high-speed imaging mechanisms.

6. References

[1] Dalavai, Lavanya, et al. "Computerised Image Processing and Pattern Recognition by Using Machine Algorithms." (2023).

[2] Zhou, You, et al. "Flat optics for image differentiation." *Nature Photonics* 14.5 (2020): 316-323.

[3] Doutsi, Effrosyni, et al. "Retina-inspired filter." *IEEE Transactions on Image Processing* 27.7 (2018): 3484-3499.

[4] Sengupta, Jonah, Susan Liu, and Andreas Andreou. "RetinoSim: an Event-based Data Synthesis Tool for Neuromorphic Vision Architecture Exploration." *Proceedings of the International Conference on Neuromorphic Systems 2022*. 2022.

[5] Katic, Nikola, Alexandre Schmid, and Yusuf Leblebici. "A retina-inspired robust on-focal-plane multi-band edge-detection scheme for CMOS image sensors." 2014 *IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2014.

[6] Yildirim, Melih, and Firat Kacar. "Adapting Laplacian based filtering in digital image processing to a retina-inspired analog image processing circuit." *Analog Integrated Circuits and Signal Processing* 100 (2019): 537-545.

[7] Song, Ruibing, et al. "A reconfigurable convolution-in-pixel cmos image sensor architecture." *IEEE Transactions on Circuits and Systems for Video Technology* 32.10 (2022): 7212-7225.

[8] Deng, Guang, and L. W. Cahill. "An adaptive Gaussian filter for noise reduction and edge detection." *1993 IEEE conference record nuclear science symposium and medical imaging conference*. IEEE, 1993.

[9] Jeong, Hong, and Chang-Ik Kim. "Adaptive determination of filter scales for edge detection." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 14.05 (1992): 579-585.

[10] Williams, Donna J., and Mubarak Shah. "Normalized edge detector." *[1990] Proceedings. 10th International Conference on Pattern Recognition.* Vol. 1. IEEE, 1990.

[11] Wang, Manyu, et al. "A new image denoising method based on Gaussian filter." *2014 International Conference on information science, electronics and electrical engineering.* Vol. 1. IEEE, 2014.

[12] Liu, Jian-Lei, and Da-Zheng Feng. "Two-dimensional multi-pixel anisotropic Gaussian filter for edge-line segment (ELS) detection." *Image and vision computing* 32.1 (2014): 37-53.

[13] Zhang, Zhengwu, Eric Klassen, and Anuj Srivastava. "Gaussian blurring-invariant comparison of signals and images." *IEEE Transactions on Image Processing* 22.8 (2013): 3145-3157.

[14] Lai, Junyang. "The Principle and Design of CMOS Active Pixel Sensor." *Highlights in Science, Engineering and Technology* 32 (2023): 143-149.