Adaptive block level bilateral filtering algorithm

Yuhao Zhang

Sichuan University, Xihangang Street, Shuangliu District, Chengdu, Sichuan, China

591256546@qq.com

Abstract. During the acquisition or transmission process, video images are subject to random signal interference and generate noise, which can hinder people's understanding of the image and subsequent processing work. Therefore, it is necessary to study video image denoising and filtering algorithms. Bilateral filter is one of many typical video image filtering algorithms. However, the traditional bilateral filter algorithm does not consider the differences in the contents of different regions of the image. It is difficult to obtain the optimal filtering effect by using a fixed filtering weight to filter the entire image, which leads to problems such as blurred image edges and inadequate details processing. This paper studies the influence of different filter block sizes on the bilateral filter effect, and proposes an algorithm to adaptively update the bilateral filter weight according to the block's variance. The experimental result shows that the performance of adaptive bilateral filter with different block sizes is expected to be better than that of traditional algorithms with fixed filter weights.

Keywords: video noise, video filtering, bilateral filtering, adaptive strength.

1. Introduction

Digital images are frequently contaminated by numerous noises during their creation, transmission, and recording as a result of flaws in the imaging system, transmission medium, and recording apparatus. Given this, an image filter is a crucial operation in picture preprocessing. Its goal is to reduce the target image's noise while retaining as much of the image's information as possible, including its details, textures, and edges. Binary, grayscale, color, false color, multispectral, stereo, and 3D images are all types of digital picture signals. Since color images include more information than grayscale photos, they are a common image type in today's image processing research. It consists of YUV, HSI, RGB, and YCbCr. This paper mainly deals with the processing of YUV type color video images.

Typical image filtering methods can be divided into linear filtering and nonlinear filtering. In detail, linear filtering contains box filter, mean filter, Gaussian filter, and nonlinear filtering includes median filtering and bilateral filtering. Median filter (MF) is a famous method for removing salt-and-pepper noise (SPN) [1]. MF uses a fixed-size window to process noises by assigning the median value of pixels in the window to the central pixel. But when removing medium-density and high-density SPN, MF works ineffectively [2-3]. For a more efficient way of smoothing images, the Gaussian filter (GF) is applied [4]. GF is an effective low-pass filter, especially for removing noise that obeys normal distribution, and it selects the weights according to the shape of Gaussian function [4]. However, Gaussian kernel only considers the distance from the pixel around the center pixel to the pixel in the process of filtering, and it will blur the images' details if not consider the size relationship of pixels.

Unlike linear filtering, the bilateral filter uses a range kernel along with a spatial kernel, where both kernels are usually Gaussian [5]. The weight of gray information is added to the bilateral filtering, that is, in the neighborhood, the weight of the points whose gray value is closer to the central point's gray value is greater, and the weight of the points whose gray value difference is greater is smaller. The weight is determined by the range Gaussian function.

After the bilateral filter is proposed, many scholars improve it. In [6], Bai Xiaodong et al. recalculate the noise standard deviation and filtering window by using the half-edge rotating window method to retain the edge information. However, the calculation amount of this algorithm will take a lot of time. Zhang B Y et al. add a compensation function to the traditional bilateral filtering model to sharpen the image and eliminate noise [7]. However, the experimental results show that the noise is sharpened while the image is sharpened. In [8], Shi K Q et al. adaptively adjust the gray level standard deviation in the bilateral filter according to the noise standard deviation in the filter window, but the calculation method has defects, resulting in poor image edge preservation effect. C. Xiong et al. propose a parameter adaptive bilateral filtering algorithm [9], and it also removes the pseudo-edges of the image, which is better than that achieved by traditional Canny edge detection algorithm [10-11].

This paper studies the optimization algorithm of typical bilateral filtering. At present, most bilateral filtering methods will maintain fixed filtering parameters in the whole image or video, and cannot achieve content-based adaptive filtering. In actuality, the content of different regions has different characteristics. So, using different filter sizes may help improve performance. It is anticipated that the experiment would examine the impact of various filter sizes on the effectiveness of bilateral filtering. We take the video image with noise as the original processing sample, and divide each frame of image into blocks with different granularity for bilateral filtering. Finally, we calculate the PSNR (Peak Signal Nosie Ratio) value of the images filtered with different block sizes, compare the PSNR value, and analyze the experimental results. In the experiment, the initial block-level size is set to 8*8 and the PSNR value gradually increases as the block size increases from 8*8 to 128*128. The result shows that processing images in blocks is better than processing in frames. To be more specific, the PSNR value is the largest when the block size is 128*128, meaning that the filtering result obtained in this case is the best. Also, we should ensure that the blocks have the appropriate size as much as possible (the size cannot be too small), in consideration of the experimental result. In conclusion, the processed images' details are better preserved compared with the video images processed by conventional bilateral filtering by means of using our proposed the method.

The reminder of this work is organized as follows. Section II introduces related work. Section III presents our innovative algorithm. Section IV gives the experiment results and conclusion is drawn in Section V.

2. Related work

A. Classical bilateral filtering algorithm

Bilateral filtering is a kind of nonlinear filter, which can achieve the effect of keeping edges and reducing noise [5]. The bilateral filtering adopts the combination of two Gaussian filters. One is responsible for calculating the weight of spatial proximity, which is the commonly used Gaussian filter principle. The other is responsible for calculating the weight of pixel value similarity. Under the simultaneous action of two Gaussian filters, bilateral filtering can be computed as,

$$BF(p) = {}^{1}/_{Wp} \sum_{q \in a} G\sigma_s\left(\left||p-q|\right|\right) G\sigma_r\left(I_p - I_q\right) I_q \tag{1}$$

where $BF(p)$ represents the image pixel values after bilateral filtering. The parameter p stands for the center pixel point in neighborhood. q represents a pixel point around p in neighborhood. a is the range of neighborhood. The purpose of a bilateral filter is to weighted average all adjacent points for each point in the image to achieve smoothing. For this, the bilateral filter adopts different weights at the point q, which has coordinate (x', y'), adjacent to p, which has coordinate (x, y), and the strategy includes two core functions, $G\sigma_s$ and $G\sigma_r$. $G\sigma_s$ is a Gaussian function related to spatial distance between pixels

DOI: 10.54254/2755-2721/17/20230917

and $G\sigma_r$is related to pixel value difference between pixels. Both of them use Euclidean distance of the parameter. $G\sigma_s$ and $G\sigma_r$ are defined as,

$$G\sigma_s(||p-q||) = exp\left(-{(x'-x)^2+(y'-y)^2}\Big/{2\sigma_s^2}\right) \tag{2}$$

$$G\sigma_r(I_p-I_q) = exp\left(-{((I(x',y')-I(x,y))^2}\Big/{2\sigma_r^2}\right) \tag{3}$$

where Wp is a normalization weight factor and I represents the pixel value in the current coordinate system. It is defined as,

$$Wp = \sum_{q\in a} G\sigma_s(||p-q||)G\sigma_r(I_p-I_q) \tag{4}$$

However, the image processed by conventional bilateral filter will appear edge blur, because the pixel value of edge points will affect the calculation results of $G\sigma_s$ and $G\sigma_r$. As a result, many academics are concentrating on improving its performance by putting forth numerous ideal methods to accomplish adaptable parameters for filtering strengths.

B. Improved bilateral filtering algorithms

In literature [9], C. Xiong et al. proposed a parameter adaptive bilateral filtering algorithm. The adaptive parameter selection method in the paper can automatically select the filtering parameters according to different images by giving their experimental result that when the image gray scale is [0,255], the bilateral filtering result will achieve better using the optimized parameters. The researchers give 3 values, which are n, σ_s and σ_r. n is a created default value (it is set to 11), where it can be utilized to adjust filtering strength. σ_s is the neighborhood size-related value, σ_r represents the surrounding neighborhood pixel value fluctuation-related value, they can be defined as,

$$n = 11,\ \sigma_s = 3*n,\ \sigma_r = {96}\big/{\Delta} \tag{5}$$

where Δ is a created value that reflect the neighborhood pixel value fluctuations, which is defined as,

$$\Delta = {\sum_{q\in a}\left|\overline{I_q}-I_q\right|}\Big/{m} \tag{6}$$

where m is the number of neighborhood pixel points, $\overline{I_q}$ is the average of pixel value in the neighborhood which is used to reflect the fluctuation of pixel value.

Based on the above theory, the first step of their parameter adaptive bilateral filtering algorithm is to determine whether there is available n and σ_s input or not. If not, $n = 11\ and\ \sigma_s = 3*n$. Then, initialize all of $BF(p)$ and Wp are set as 0. For each pixel (x,y) that has pixel value I(x,y), they firstly calculate the value of $\sigma_r$whose formula is deformed to

$$\sigma_r = {96m}\Big/{\sum_{q\in a}\left|\overline{I_q}-I_q\right|} \tag{7}$$

Then for each pixel point (x',y') around point (x,y) that has pixel value I(x',y'), they calculate the weight, and update the summation of weight and the final value of $BF(p)$. In the end, they normalize the result.

The final result shows that this algorithm can adaptively select the filtering parameters to obtain a better result than classical bilateral filtering does. In the Figure.1 (a), it is the original image and Figure.1 (b) is the image after bilateral filtering, while Figure.1 (c) is the experimental image which uses the adaptive parameter selection method proposed in the paper. It is obvious that the color contrast of Figure.1 (c) is more evident in different regions compared with Figure.1 (b).

(a) (b) (c)

Figure 1. (a) Original image. (b) Bilateral filtering result. (c)Parameter adaptive bilateral filtering result [9].

In literature [6], Bai Xiaodong et al. proposed an another improved bilateral filtering algorithm. They recalculate the noise standard deviation and filtering window by using the half-edge rotating window method to retain the edge information. First, in order to obtain more samples and ensure the clarity of the image, the calculation method of σ_s is updated as,

$$\sigma_s = {}^{(r-v)}/_{2} \tag{8}$$

where v is a constant. The filter window radius r is designed to be v larger than the empirical value, so that 95% of the spatial weight is concentrated in the central area of the square with a radius of $(r-v)$, which can prevent the spatial weight from being dispersed and the image from being blurred. The main function of the remaining $r^2-(r-v)^2$ pixels at the window edge is to participate in the calculation of the noise standard deviation. Ultimately, equation (9) can be substituted into equation (2).

When calculating the noise standard deviation of a point, it is necessary to determine in advance whether the filtering window of the point contains edges of the image. Then, they determine whether the filtering window of the point contains the edge of the image in advance. For most images, the area of the stationary region of the image is much larger than the area of the edge region, so the threshold method can be used to determine whether the filtering window of a point contains image edges. The threshold value is set to the median of the noise standard deviation of all pixel points in the entire image, and the specific formula is

$$E(x,y) = \begin{cases} 1, \sigma_n(x,y) > median(\Omega_{\sigma_n}) \\ 0, others \end{cases} \tag{9}$$

In the equation above, Ω_{σ_n} is the noise standard deviation set of all pixels in the whole image. Median $(\cdot)$ is the median sign; $E\ (x,y)$ is a matrix that stores the calculation results of each point. If the noise standard deviation of a point is greater than the median of the noise standard deviation of the entire image, it is set to 1, indicating that the window centered on the point contains the image edge. Otherwise, it is set to 0.

Based on the above discussion, if $E\ (x,y)$=1, it indicates that there is an image edge within the window centered at that point. Then, they need to obtain the true noise standard deviation and the optimal filtering window for this point. They put forward the method of half rotating window. Firstly, they take the left half window of a square window with a side length of (2r+1), and rotate the half window by K angles with the center of (x,y) point and ${}^{2\pi}/_{K}$ as the angle step. As the window rotates, the noise standard deviation changes constantly. When the noise standard deviation reaches the minimum, it is considered that the window does not contain the edge of the image. At this time, the minimum noise standard deviation of K angles is obtained $\sigma_{n(min)}$. It is defined as

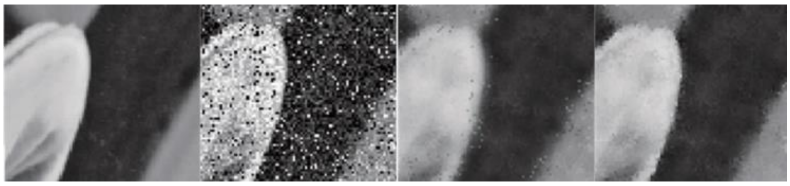
$$\sigma_{n(min)} = min_{k=0,1,\ldots,K-1}\left\{std\left[f\left(T^{2\pi k/_{K}}\right)\right]\right\} \tag{10}$$

where $T^{2\pi k/_{K}}$ stands for the coordinate set in the window. $min(.)$ is the symbol for taking the minimum value. Set the rotation angle at this point to k_0, and set the window with the angle k_0 as the filtering

DOI: 10.54254/2755-2721/17/20230917

window for the point. In equation (3), due to σ_r and σ_n are linearly related, it is advisable to take σ_r=2$\sigma_{n(min)}$ when calculating the final value of $BF(p)$ in equation (1).

By analyzing the results as given in Figure.2, the experimental result indicates that the algorithm has better edge preservation effect and strong noise removal effect.

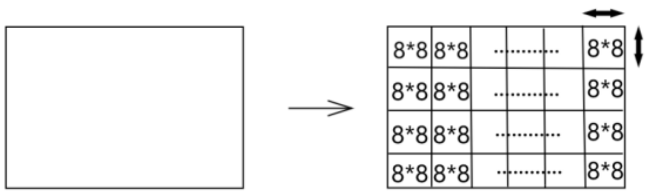


(b) (b) (c) (d)

Figure 2. (a) Original image; (b) Noise image; (c) The denoising results of conventional bilateral filtering algorithm; (d) The denoising results of the proposed algorithm [6].

Although the above algorithms have improved the performance of bilateral filtering, most of them adopt the same filtering parameters and filtering frame level size for the entire image, without considering the different contents of different regions of the image, that is, the impact of the region size on the bilateral filtering effect. Therefore, this paper will study the performance impact of bilateral filtering algorithm with different block sizes.

3. Granularity blocks oriented bilateral filter



(a) (b)

Figure 3. (a) Original video image (b) Video image is cut into blocks (8*8).

In this paper, we improve the conventional bilateral filtering algorithm by dividing each video image to be processed into different granularity blocks and then performing bilateral filtering. In details, we primarily take YUV video image as the original sample and add noise. YUV is a color coding method. It is often used in various video processing components [15-16]. Y indicates brightness (Luminance or Luma), also known as "gray scale value"; UV is expressed as chrominance or chroma, which describes color and saturation, and is used to specify the color of pixels. The YUV format is mainly divided into two types: planar and packed. Planar plane format means that the storage format is to store all Y components first, then U components, and then V components. The packed packing mode stands for that the Y, U and V components of each pixel are continuously and alternately stored. In this experiment, we use the planar plane format to store the YUV video images.

First, after adding noise to the original video images, we perform optimized bilateral filtering processing for each frame of image. We divide one YUV image in the video into blocks, then for each pixel point (x', y') around point (x, y) that has pixel value I(x', y'), we calculate their weights of Y, U and V separately and update the weight summation by substituting Equations (2) and (3) into Equation (4).

$$Wp' = \exp\left(- {((I(x',y') - I(x,y))^2} \Big/ {2\sigma_r^2} - {(x'-x)^2 + (y'-y)^2} \Big/ {2\sigma_s^2}\right) \quad (11)$$

$$Wp = \sum\nolimits_{q \in a} Wp' \tag{12}$$

where Wp' means the weight of the currently calculated point (x, y) and Wp represents the summation of Wp'. Parameter σ_s determine the spatial weight distribution and parameter σ_r reflect the gray scale similarity of points (x', y') and point (x, y). These two parameters are manually set according to specific image. In the algorithm, we set σ_s to 3 and σ_r is optimized to an adaptive parameter,

$$\sigma_r = max\left(0.15 * \sqrt{D(B)}, 20\right) \tag{13}$$

$$D(B) = {\sum\left(B(x,y) - \overline{B}\right)^2}\Big/{row * col} \tag{14}$$

where max represents selecting the one with the highest numerical value in parentheses. $D(B)$ means the variance of pixel values in each block B whose length and width are respectively row and col. $\overline{B}$ means the average of all pixels in a block, and $B(x, y)$ represents the pixel value of each point in the block. The index (x, y) is the coordinate of the point.

Then, we update the value of $BF(p)$. It should be noted that the direction of pixel matrix processed by the algorithm is from left to right and from top to bottom. When the weights of all the points (x, y) that can be processed in a block are calculated, the next block is processed.

Finally, we output the processed YUV video images which are stored in the computer, and compare the differences between images which are processed by classical bilateral filter algorithm and optimized algorithm.

The PSNR is employed here to evaluate the filtering performance, in which the calculation function aims to calculate the peak signal-to-noise ratio of the images. PSNR is the most common and widely used objective measurement method for evaluating image quality [17]. PSNR can be defined as,

$$PSNR = 10log10\left({MAX * MAX}\big/{MSE}\right) \tag{15}$$

where MAX is the maximum value of image pixels. It is a default value (here it is set to 255). And MSE represents mean square error. It is defined as,

$$MSE = {1}\big/{ab} \sum_{i=0}^{a-1} \sum_{j=0}^{b-1} [I(i,j) - K(i,j)]^2 \tag{16}$$

where a and b respectively represent the number of rows and columns of a image. I is an original image without noise, and $K(i, j)$ is the noise approximation of $I(i, j)$. To conclude, the higher the PSNR value, the less distortion.

Based on the above method, the algorithm pseudocode is given as the pseudocode for bilateral filtering optimization algorithms with different granularity block sizes as follows.

Input: YUV video image I, σ_s, σ_r, block-level size.
Output: Filtered image Ib
Initialize all of Ib and Wp to 0.
Process the blocks and points in a left-to-right, top-down direction as shown in Figure.3;
In one block, for each pixel point ((x', y') around point (x, y) that has pixel value I((x', y') then
3.1 Calculate the weight by equation (11);
3.2 Update the weight summation;
3.3 Update the $BF(p)$ by equation (1);
3.4 Jump to the next block;
4. Normalize the result;
5. Calculate the PSNR value of Y, U, V and the average value of them by equation (13).

4. Experiment results

The experimental file is in YUV format, so the PSNR has PSNR_Y, PSNR_U, PSNR_V three components of which we need to calculate the average value. The computer used in the experiment is based on an x64 operating system, Windows 10. Device name is LAPTOP-Q2D9ECLP. And the processor of the computer is Intel (R) Core (TM) i5-10210U CPU @ 1.60GHz 2.11 GHz. The on-board RAM is 8.0GB. The software implemented by the code used in the experiment is Microsoft Visual Studio Ultimate 2012. In the experiment, the YUV video images' sequence name is RaceHorses.yuv. The video image totally has 50 frames with resolution as 416*240.

The experimental result indicates that it can be an effective way to improve the improve noise reduction efficiency by splitting an image into blocks when performing bilateral filtering. In the experiment, we compare the average PSNR value of the images which are processed in frame with those which are split into blocks. In Table 1, we can distinctly observe the size changes of YUV images' average PSNR values. When images are processed in frame, the PSNR values of Y, U, V are respectively smaller than those of images which are divided into blocks. In the meanwhile, we should notice that the size of the block cannot be too large. In Table 1, when the block size increases to 128*128, the PSNR values of Y, U, V are the largest, indicating that it can reduce the noise reduction efficiency of bilateral filtering if the block size is larger. More specifically, after the image is divided into blocks, the pixel values of points between blocks will not affect each other. Considering that closer neighbor pixels are involved in the filtering window compared with the frame level method, and larger filtering strength can be provided according to its larger variance due to larger noise level, we can remove more noise adaptively. Thus for final filtering strength Wp according to equation (11) and the values of $G\sigma_s$ and $G\sigma_r$ in equation (2) and (3) can be calculated more precisely during bilateral filter, better preserving the image edges and their details.

Table 1. The average PSNR values in the experiment.

Filter Level	PSNR_Y	PSNR_U	PSNR_V
NoFilter	31.13 dB	31.12 dB	31.13 dB
Frame	32.43 dB	33.30 dB	33.28 dB
8*8	31.51 dB	31.82 dB	31.79 dB
16*16	32.12 dB	33.18 dB	33.11 dB
32*32	32.66 dB	34.33 dB	34.20 dB
64*64	32.95 dB	35.11 dB	34.93 dB
128*128	**33.03 dB**	**35.42 dB**	**35.18 dB**

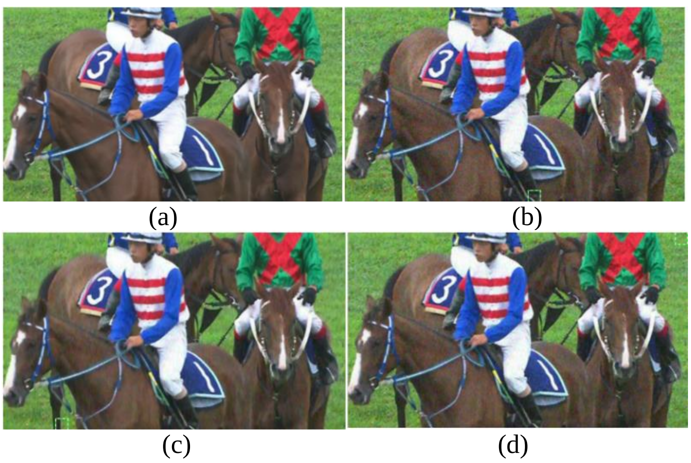


Figure 4. (a) Original image (b) Noise image (c) 128*128 block level filtered image (d) Frame level filtered image.

DOI: 10.54254/2755-2721/17/20230917

We can infer from the findings in the table above that filtering images in blocks can improve the performance of bilateral filters to some extent. It can be observed that the block size cannot be too small. The average PSNR values are lower when the block size is set to 8*8 or 16*16, indicating that the algorithm has no advantage over the frame-level bilateral filter algorithm in terms of optimization. Additionally, the PSNR demonstrates that the results of the bilateral filter are optimum when it is set to 64*64 and 128*128.

Furthermore, we can observe in Figures 4 and 5, for the entire image and segment details, that the subjective filtering result is superior when the image is filtered at the 128*128 block level as opposed to the frame level filtered image. The former is more detailed and has less noises than the original image.

(a) **(b)** **(c)** **(d)**

Figure 5. Details of (a) Original image (b) Noise image (c) 128*128 block level filtered image (d) Frame level filtered image.

To conclude, this method verifies that block level filtering is helpful to improve the performance of bilateral filter, and provides reference for the research of adaptive block level filtering algorithm.

5. Conclusion

This paper investigates how the size of the filter block affects the effect of the bilateral filter and proposes an algorithm to adaptively update the weight of the bilateral filter in accordance with the variance of the block, realizing the adaptive bilateral filter in accordance with the properties of various block contents. Experiments demonstrate that adaptive bilateral filters with varied block sizes can perform objectively and subjectively better than the results of using the fixed default filtering weights. This research will be helpful to provide reference for adaptive bilateral filter algorithm. In the future, we will continue to deeply study the bilateral filter algorithm based on the adaptive filter block size and weight of image content to obtain better performance.

References

[1] U. erkan, D. N. H. Thanh, L. M. Hieu and S. Engínoğlu, "An Iterative Mean Filter for Image Denoising," in IEEE Access, vol. 7, pp. 167847-167859, 2019, doi: 10.1109/ACCESS.2019.2953924.

[2] K. Panetta, L. Bao and S. Agaian, "A new unified impulse noise removal algorithm using a new reference sequence-to-sequence similarity detector", IEEE Access, vol. 6, pp. 37225-37236, 2018.

[3] S. Wang, " Dictionary learning based impulse noise removal via L1 - L1 minimization ", Signal Process., vol. 93, no. 9, pp. 2696-2708, 2013.

[4] M. Wang, S. Zheng, X. Li and X. Qin, "A new image denoising method based on Gaussian filter," 2014 International Conference on Information Science, Electronics and Electrical Engineering, Sapporo, Japan, 2014, pp. 163-167, doi: 10.1109/InfoSEEE.2014.6948089.

[5] R. G. Gavaskar and K. N. Chaudhury, "Fast Adaptive Bilateral Filtering," in IEEE Transactions on Image Processing, vol. 28, no. 2, pp. 779-790, Feb. 2019, doi: 10.1109/TIP.2018.2871597.

[6] Bai Xiaodong, Shu Qin, Du Xiaoyan, etc Improved adaptive bilateral filtering algorithm [J] Progress in Laser and Optoelectronics, 2020, 57 (4): 041003.

[7] Zhang B Y, Allebach J P. Adaptive bilateral filter for sharpness enhancement and noise removal[J]. IEEE Transactions on Image Processing, 2008, 17(5): 664-678.

[8] Shi K Q, Wei W G. Image denoising method of surface defect on cold rolled aluminum sheet by bilateral filtering[J]. Surface Technology, 2018, 47(9): 317-323.

[9] C. Xiong, L. Chen and Y. Pang, "An Adaptive Bilateral Filtering Algorithm and its Application in Edge Detection," 2010 International Conference on Measuring Technology and Mechatronics Automation, Changsha, China, 2010, pp. 440-443, doi: 10.1109/ICMTMA.2010.41.

[10] W. Rong, Z. Li, W. Zhang and L. Sun, "An improved Canny edge detection algorithm," 2014 IEEE International Conference on Mechatronics and Automation, Tianjin, China, 2014, pp. 577-582, doi: 10.1109/ICMA.2014.6885761.

[11] L. Xuan and Z. Hong, "An improved canny edge detection algorithm," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2017, pp. 275-278, doi: 10.1109/ICSESS.2017.8342913.

[12] V. Aurich and J. Weule, "Non-linear Gaussian filters performing edge preserving diffusion", Proceedings of the DAGM Symposium Proceed, vol. 17, pp. 538-545, 1995.

[13] S. Smith and J. Brady, "Susan—a new approach to low level image processing", International Journal of Computer Vision, vol. 23, no. 1, pp. 45-78, 1997.

[14] C Tomasi and R Manduchi, "Bilateral filtering for gray and color images", Proceedings of the Sixth International Conference on Computer Vision, pp. 839-846, 1998.

[15] Podpora M, Korbas G P, Kawala-Janik A. YUV vs RGB-Choosing a Color Space for Human-Machine Interaction[C]//FedCSIS (Position Papers). 2014: 29-34.

[16] Keith Jack. Video Demystified. ISBN 1-878707-09-4.

[17] S. Winkler and P. Mohandas, "The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics," in IEEE Transactions on Broadcasting, vol. 54, no. 3, pp. 660-668, Sept. 2008, doi: 10.1109/TBC.2008.2000733.