SIN-CHN SCIENTIFIC PRESS PTE.LTD.

Molecular & Cellular Biomechanics 2025, 22(3), 978.
https://doi.org/10.62617/mcb978

Article

Research on cellular and molecular biomechanics-inspired enhancement of visual communication in medical product design via graphic processing algorithm optimization

Junlin Li

College of Art and Design, Liaoning Petrochemical University, Fushun 113001, China; Lijunlin19962023@163.com

CITATION

Li J. Research on cellular and molecular biomechanics-inspired enhancement of visual communication in medical product design via graphic processing algorithm optimization. Molecular & Cellular Biomechanics. 2025; 22(3): 978.
https://doi.org/10.62617/mcb978

ARTICLE INFO

Received: 3 December 2024
Accepted: 12 December 2024
Available online: 13 February 2025

Abstract: The computational complexity of graphic processing algorithms is increasing under the continuous development of information technology. At the same time, the medical product design field puts forward higher and higher requirements on the visual communication effect of related images. In this study, the up-sampling optimization algorithm and the threshold filtering algorithm are proposed to optimize the Laplacian graphics processing algorithm. Notably, in the threshold filtering algorithm, the Triangle algorithm is employed to address the grayscale of images pertinent to medical product design. The perception of the grayscale and binarized images by the human visual system triggers neural signals that propagate and can influence intracellular processes. When observing medical product images, neurons fire, leading to the release of neurotransmitters like glutamate. These neurotransmitters bind to receptors on cells, initiating signaling cascades such as the MAPK pathway. This pathway can affect gene expression and protein synthesis, potentially modulating cellular functions related to perception and response to the medical product design. The results show that the optimized graphical processing algorithms in this paper outperform the comparison algorithms in the CLBLAS library, and the floating-point computational values of the Laplacian algorithm are much higher than the comparison algorithms in the face of the large-scale input parameters. The Laplacian algorithm is able to accurately stitch and process the captured 2D images of cellular microtubules related to the design of the medical products in a guaranteed high efficiency (31 min), and The Laplacian algorithm was able to achieve an average subjective score of 0.856 for the visual communication of medical product design images. This graphic processing algorithm can generate images of superior perceptual quality, which holds substantial significance for augmenting the visual communication effect of medical product design images and considering the underlying cellular molecular biomechanical responses.

Keywords: Laplacian; threshold filtering algorithm; triangle algorithm; cellular molecular biomechanics; neural signals

1. Introduction

Most modern people begin and end their lives in hospitals. Medical products, as an important part of the healthcare system, have a significant impact on human life. When it comes to the hospital environment, most people will immediately think of the heavy theme of life, old age, sickness and death and the neat and quiet environmental atmosphere, and when it comes to medical products, it will naturally be associated with simple and stereotypical forms and monotonous neutral colors [1]. Indeed, in order to safely and reliably realize the diagnostic and therapeutic functions, the design of medical products has always played a relatively silent role in the wave of commercialism, and its evolution is not like the ordinary consumer

products that constantly follow the tide of the wave, seeking change and innovation [2,3]. However, the impact of medical product design on the therapeutic effect does not only depend on the functionality of the product, but also the impact of the product on the patient's psychological feeling, the coordination and adaptability of the product itself with the environment, etc. have all become issues that need to be paid attention to [4–6].

The main focus of industrial design intervention in medical product research and development is the material form of the product, which is carefully planned and considered from the aspects of product form and color. Morphological studies have shown that product form can have an obvious impact on the user's psychology, so in medical product design, it is common to adopt a holistic modeling, using gentle curved surfaces, soft lines and corners and other modeling techniques, to eliminate the stereotypical form of the product and the sense of tension, and to increase the affinity [7–11]. The application of color in modern medical product design is relatively cautious and conservative, the cold colors with low purity can create a neat and calm environment, and the warmer colors such as goose yellow and ochre represent softness and calmness, while the warning signs, emergency stop switches, etc. according to the specifications must be in high contrast and high purity colors [12–16]. In the era of artificial intelligence, deep learning algorithms are integrated into medical product design, which is committed to ensuring the functionality of the product itself while realizing the attention to human physiological and psychological needs [17,18].

This paper proposes a Laplacian graphics processing algorithm consisting of horizontal filtering, vertical filtering, up sampling and down sampling computational processes. Four versions of up-sampling kernel algorithm are used to optimize the up-sampling process in the graphics processing algorithm, and conditional compilation judgment is used to improve the performance of the up-sampling algorithm. The threshold filtering algorithm is used to reduce the noise of the relevant graphics in the medical product design, and the grayscale histogram of the graphics is obtained by the Triangle algorithm. The gray level with the largest Triangle value is then selected as the optimal threshold, and the graph is binarized according to the optimal threshold. Subsequently, the Laplacian graphic processing algorithm is improved and optimized by combining the binarization processing in the threshold filtering algorithm and the serial method at the CPU side, so as to achieve the purpose of optimizing and enhancing the visual communication effect in the design of medical products by using the improved algorithm. After verifying and analyzing the performance of the graphic processing algorithm, this study utilizes the algorithm to process the images related to medical product design, and demonstrates the enhancement effect of the algorithm on the visual communication effect of the images related to medical product design in terms of the reconstruction quality of the images, the splicing quality, and the subjective effect of the visual communication.

2. Method

2.1. Graphics processing algorithm construction

The graphic processing algorithm proposed in this paper, Laplacian [19], which is applied to the design of medical products to improve the visual communication effect, consists of horizontal filtering, vertical filtering, down-sampling, up-sampling, and difference computation, and the specific implementation flow of the algorithm is shown in **Figure 1**. Horizontal filtering includes adding vertical boundary padding and horizontal filtering two parts, vertical filtering includes adding horizontal boundary padding and vertical filtering two parts.

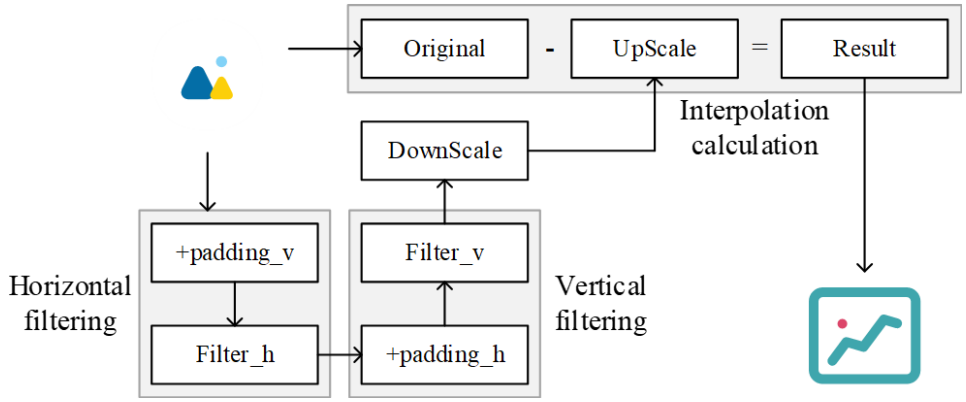

Figure 1. Process of the Laplacian algorithm.

(1) Horizontal Filtering

Two columns of padding are added to the beginning and the end of the source matrix, and the specific padding values of the padding are calculated from the values of the first four and the last four columns of the source matrix. That is, each row will add two elements at the beginning and the end, the two elements added at the beginning are calculated from the first four elements of the row, and the two elements added at the end are calculated from the last four elements at the end of the row. The two padding elements added to each row are calculated from the four elements of their neighboring peers, and the process is independent and unrelated, with good parallelism. After the padding is added, the calculation window of size 1 × 5 is shifted back by one in turn, and the five element values of a row are taken each time to calculate the target value corresponding to the center element of the window in the target matrix. Each part of the element calculation for each window size is relatively independent, so it has good parallelism.

(2) Vertical Filtering

Vertical filtering process and horizontal filtering is similar to the beginning and end of the source matrix to add two rows of padding, padding value also has its neighboring four values calculated, that is, the first column to add the two values calculated by the first four elements of the column. Calculation of padding process is independent and unrelated, each column of the two padding elements can be calculated in parallel at the same time. After adding the padding, the size of the 5 * 1 calculation window in order to move down one column at a time, each time to take a column of five elements of the value of the calculation of an element of the target matrix.

(3) Down-sampling

Takes values across rows and columns. The size becomes 1/2*1/2 of the original graphic size. Elements of the original graphic matrix that exist in both odd rows and odd columns are retained.

(4) Up-sampling

Up-sampling is divided into two parts: boundary calculation and body calculation, the boundary value is calculated from the value of its neighboring rows or columns. The rows are shifted back one bit sequentially in a window of size 1 × 2, and two elements of a row are obtained each time to compute two element values of the boundary rows of the target matrix, and in fact, the first element value is a direct padding, and the second value is computed from the two values obtained. If the number of rows is odd, you need to calculate the first row of boundary values, and the subsequent part of the body partially filled. If the number of rows is even, then in addition to the first row, you need to calculate the boundary value of the last row, the same method as the first row. Columns are shifted back one place in sequence in a window of size 2 × 1, 2 elements of a column are obtained each time, and the values of the two elements of the boundary column of the target matrix are calculated, in the order of the columns are calculated, the first value is filled in directly, and the second value is calculated from the two values obtained. If the number of columns is odd, only the last column value is calculated and the rest is filled by the main body part. The main body is shifted back one place by row and column in order in a window of size 2 × 2, and 4 elements of the source matrix are obtained each time, and the 4 element values of the target matrix are calculated.

2.2. Optimized design of graphics processing up-sampling algorithm

In order to improve the performance of the graphics processing algorithm on the graphics related to medical product design, so as to obtain a better visual communication effect, this paper optimizes the up-sampling algorithm in the graphics processing algorithm, and at the same time proposes the threshold filtering optimization algorithm to optimize the performance of the graphics processing algorithm.

Since the up-sampling algorithm in the graphics processing algorithm involves the boundary processing problem, whether its rows and columns are odd or even respectively will cause multiple judgments in the specific implementation of the algorithm. In order to reduce the increase of the conditional judgment branch here, a more direct way of implementation is to write four versions of the up-sampling kernel, according to the specific circumstances of the CPU side of the judgment to determine the final call kernel. four versions of the rows and columns of the rows and columns of the situation of the arrangement of combinations of rows odd + columns of odd, rows even + columns of even, rows even + columns odd, rows odd + columns of even. But this method, the code is cumbersome, the experimental results of the performance is not ideal, therefore, after a detailed analysis of the algorithm itself, the use of local judgment of the four-kernel fusion into a kernel.

The principle of the up-sampling algorithm is to take the window size of 2 × 2 from the source matrix in turn to calculate the target matrix 2 × 2 elements, and then the window is sequentially shifted back one until the source matrix is traversed in its

entirety, which means that the size of the target matrix obtained is $(width - 1) \times 2 \times (height - 1) \times 2$, that is, the lack of the boundary value, the lack of the boundary between the number of rows and columns of the original matrix rows and columns of the parity of the original matrix is related. But in fact, except for the last row and the last column, the other parts are calculated in the same way regardless of whether the number of original graphic rows and columns is odd or even, the first column needs to be processed separately, and then the main part is processed in the same way. For the original matrix with an even number of rows, an additional computation is needed for the last row. The same is true for the columns. Therefore, conditional compilation can be used here to add conditional compilation judgments to the original graphic matrix with an even number of rows and/or columns to avoid unnecessary conditional judgment overhead after determining the size of the input matrix. Thus, the algorithm as a whole will have one conditional judgment for one row, containing a conditional compilation for the last parity judgment. The last row will have a conditional compilation judgment, which contains a conditional compilation for the last parity judgment. The main part of the calculation, including a conditional compilation for the last parity judgment. After optimizing the code, the amount of code is not only reduced to 1/3 of the previous one, but also the performance is improved up to 24 times.

2.3. Threshold filtering based algorithm optimization approach

2.3.1. Principles of threshold filtering algorithm

Threshold filtering algorithm [20] is a noise reduction process for graphics that filters out or retains pixel points in a graphic whose gray value is below or above a set threshold, thus achieving the effect of reducing noise and enhancing graphic details. Threshold filtering algorithms have a wide range of applications and play a very important role in the field of digital graphics processing and computer graphics. It is commonly used in tasks such as graphic segmentation, target detection, edge detection, graphic enhancement, graphic denoising, and binary processing. The formula is shown below:

(1) The filtering method of signal reconstruction through signal mode maximum using signal singularity features, the basic principle is to use the original signal and noise signal in the wavelet transform domain of different features:

$$dst(x,y) = \begin{cases} maxval & if\ src(x,y) > thresh \\ 0 & otherwise \end{cases} \quad (1)$$

(2) Eliminate the mode maximum points generated by noise, and retain the mode maximum points generated by the original signal:

$$dst(x,y) = \begin{cases} 0 & if\ src(x,y) > thresh \\ maxval & otherwise \end{cases} \quad (2)$$

(3) The signal is reconstructed through the remaining maximum points:

$$dst(x,y) = \begin{cases} threshold & if\ src(x,y) > thresh \\ src(x,y) & otherwise \end{cases} \quad (3)$$

(4) Spatial filtering is realized by using the correlation between signal scales:

$$dst(x,y) = \begin{cases} src(x,y) & if\ src(x,y) > thresh \\ 0 & otherwise \end{cases} \quad (4)$$

(5) The signal is reconstructed and the filtered signal is obtained:

$$dst(x,y) = \begin{cases} 0 & if src(x,y) > thresh \\ src(x,y) & otherwise \end{cases} \quad (5)$$

Thresholding filtering algorithm performs a fixed thresholding operation on a single-channel graph to obtain a binarized graph, while removing graph noise and filtering out graph points with very small or very large pixel values.

Thresholding serves to divide the data into two parts and process the values of the two parts according to the type of operation. Thresholding is mainly done by Triangle algorithm [21]. Triangle algorithm is a graphic segmentation method which is used to segment a gray scale graphic into two sub-graphs with the same gray level for target region extraction or separation of the target region from the background region. The basic idea of the Triangle algorithm is to compute the histogram of the graphic's gray level and count the number of pixels in each gray level. The number of pixels in each gray level is counted. Calculate the cumulative probability based on the gray level histogram, i.e., the ratio of the number of pixels from gray level 0 to the current gray level to the total number of pixels. Calculate the sum of squared gray differences for each gray level, i.e., the sum of squared gray differences between the foreground and the background when thresholded at the current gray level. Based on the calculated sum of squared gray differences and the cumulative probability, a Triangle value for the current gray level is calculated, i.e., Triangle = sum of squared gray differences/ cumulative probability. The gray level with maximum Triangle value is selected as the optimal threshold value which maximizes the gray difference between foreground and background. The graph is binarized using the optimal threshold value to convert the grayscale graph into a binary graph, where pixels larger than the threshold value are set as foreground and pixels smaller than the threshold value are set as background.

2.3.2. Parallelism analysis and algorithm optimization

Threshold filtering is mainly divided into threshold acquisition and graph binarization. Threshold acquisition cannot be massively parallelized using GPUs, and in this paper, we use a serial approach on the CPU side for computation. The graph binarization process reduces the dependence on off-chip memory bandwidth by storing the thresholds into the intermediate cache constant memory and de-histogramming.

(1) Threshold acquisition optimization

The user can adaptively select the filtering method to get the desired filtering effect. This part can be realized serially at the CPU side. In this paper, we will divide the algorithm into two parts to deal with. One part is optimized and accelerated at the CPU side. One part is accelerated at the GPU side using massive parallelism.

On the CPU side, for the OTSU algorithm, the optimization focuses on the computation of the grayscale histogram of the graph, which can be optimized to improve the execution efficiency of the algorithm by optimizing the computation process of the grayscale histogram. In this paper, the optimized histogram

equalization algorithm introduced in the previous section is used to pre-process the graph rows and reduce the amount of histogram computation.

In the process of algorithm execution, it is necessary to calculate the half-mean gray level of the foreground and the background respectively, this paper combines these two processes to avoid the waiting of the program and reduce the redundancy of the code volume. The cumulative gray level and the number of pixels is calculated simultaneously during the computation process, thus reducing unnecessary calculations.

(2) Graphics binarization optimization

The main work carried out on the GPU side is the binarization of graphics [22], where the user can adaptively choose the type of binarization needed according to the actual needs. For graph binarization, the optimization focuses on optimizing memory usage and thread organization.

Since the threshold value is a constant, it is called at every execution during the execution of the algorithm. It is effective to reduce the dependence on off-chip memory bandwidth by using intermediate layer caching. CPU side in the binarization process, it is necessary to perform the histogram operation on the graph. The histogram is reconstructed according to the obtained threshold value and binarization type. The reconstructed histogram is divided into two parts according to the threshold value. The processed histogram is compared with the target pixel points and the target pixel values are replaced by finding the content in the index corresponding to the histogram through the values of the target pixel points. This process is very time-consuming for large graphics operations and can severely degrade the performance of this algorithm.

In GPUs, the threshold value is preloaded into the GPU's constant memory before binarization, and it takes a shorter clock cycle for the shader to read the data from the constant memory compared to texture memory and shared memory. It can speed up the GPU's memory access. Also storing the graphics in texture memory can shorten the clock cycle to fetch the target pixel.

For de-histogramming, the GPU side optimization focuses on binarization operations. In this paper, we use the de-histogramming method to perform binarization operation on the graph, directly comparing the target pixel points with the threshold value, and directly determining the binarized value according to the binarization type as well as the threshold value, which saves the memory consumption of opening the histogram and saves the time of accessing the memory. Reduces the comparison lookup process between the pixel points and the histogram, saving the time of traversing the histogram.

Improve thread utilization, binarization of threshold filtering is the last step of the algorithm and the binarization process has no data dependency. The data are independent of each other. Massive parallelism using multiple threads is the key to program performance improvement. In the computational shader, the graphics are processed using the relationship of one thread mapping one pixel, in this paper, the computation of each pixel point is relatively small, the organization size of the thread group can be as large as possible, the algorithm is processed using fine-grained parallelism, and the setup workgroups are loaded into the shader, and there will be

upper dry threads in the GPU performing the same binarization operation at the same time. The speed of binarization can be accelerated.

3. Results and discussion

3.1. Performance analysis of graphics processing algorithms

3.1.1. Experimental environment

The optimized graphics processing algorithm Laplacian designed in this paper is experimented and tested on an ARM Tengui 1200 processor. The processor consists of one large A78 core at 3.0 GHz, three medium A78 cores at 2.6 GHz, and four small A55 cores at 2.04 GHz. The processor also packs a Mali-G77MC9 GPU based on the ARM Mali-G77 architecture. Version 4.5.5 of the OpenCV library was used in the experiments for comparison, and version 3.2 of the OpenGL ES interface was used for development.

3.1.2. Comparative analysis of algorithm performance

There are a couple of things that deserve special note in the experiments in this paper. First, all the data of input graphs used in this paper are random numbers generated by the code program. By having random numbers of input graph data, the idiosyncrasies and biases associated with real graph data can be eliminated, thus allowing for a more accurate assessment of performance under the action of different algorithms and optimization techniques. At the same time, this approach also ensures the reproducibility of the experimental results, allowing other researchers to reproduce and compare them under the same experimental conditions. Second, the use of random numbers to generate the source graph data also makes the experiments more generalizable, which means that the experimental results in this paper are not only applicable to a particular type of graphs or a particular medical product design domain, but are general and widely adaptable to medical product design data. This is extremely important for the promotion and application of the optimization algorithm of this paper in medical product design. In addition, in the experimental part of this paper, some relevant graphics processing algorithms from the OpenCV library are called to compare the performance and accuracy with the optimization algorithm (threshold filtering algorithm) implemented in this paper. It should be noted that the default parameters provided by OpenCV are used in calling these algorithms unless otherwise stated, which is also to ensure the fairness of the comparison experiments and the consistency of the experiments.

(1) Threshold filtering algorithm performance comparison analysis

The graphics processing threshold filtering algorithm implemented in this paper is compared with the OpenCV algorithm in 8UC3 (three-channel 8-bit unsigned integer) under the graphics data, and the running time of the threshold filtering algorithm in the graphics processing algorithm under different sizes of graphs is shown in **Figure 2** (3 × 3, 5 × 5, and 7 × 7 indicate the size of the filtering window). The 1–16 in the horizontal coordinates of the figure represent graphic pixels 512 × 512, 512 × 1024, 512 × 2048, 512 × 2160, 1024 × 1024, 1024 × 2048, 1024 × 2160, 2048 × 512, 2048 × 1024, 2048 × 2048, 2048 × 2160, 4096 × 512, 4096 × 1024 × 4096 × 2048, and 4096 × 2160 pixel points. It should be noted that the execution

count parameter of both algorithms is set to 1 here so that the most intuitive performance difference graphs can be obtained. The average running time of the threshold filtering optimization algorithm in this paper is 0.505 ms, 0.533 ms, and 0.606 ms under the 3 × 3, 5 × 5, and 7 × 7 windows, respectively, in different sizes of graphs. Observing the performance comparison graphs, it can be found that in the OpenCV algorithms, the time consumption of the threshold filtering algorithms under the same computational scale increases to different degrees as the filtering window becomes larger, which leads to the overall performance of the graphics processing algorithm to decrease. The up-sampling optimization algorithm implemented in this paper, on the other hand, does not show a significant performance difference in the same situation. This is because when the window is changed from 3 × 3 to 7 × 7, the number of data that need to be involved in the operation to compute a single target pixel value increases from 9 (3 × 3) to 49 (7 × 7), which means that 40 additional data are required for the computation of each target pixel value. In OpenCV's algorithm, the time-consuming computation of all the pixel values of the whole graph accumulates, which leads to an overall performance degradation when the algorithm changes from a 3 × 3 window (7.688 ms) to a 7 × 7 window (8.526 ms). In contrast, the optimization algorithm in this paper is able to distribute the load caused by increasing the filtering window to each core of the GPU for parallel execution in a balanced manner, instead of accumulating computational time on a single computational core.

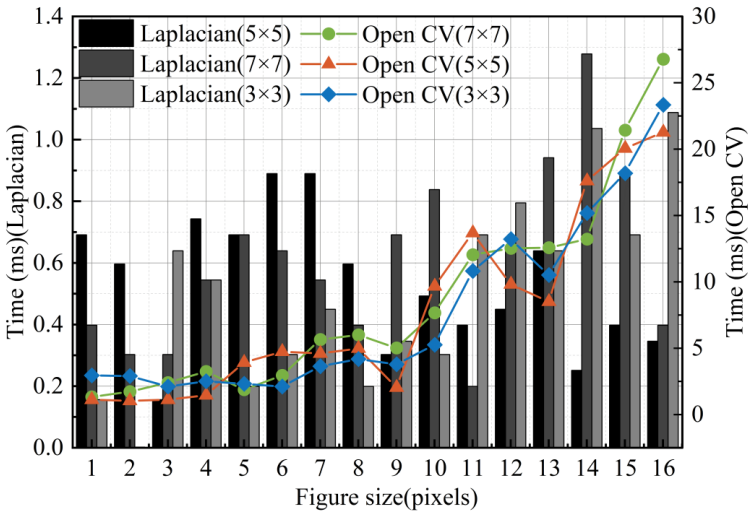

Figure 2. The performance comparison analysis of threshold filter algorithm.

(2) Open-source database simulation experiment analysis

After the above analysis, it is proved that the threshold filter optimization algorithm proposed in this paper can improve the overall graphics processing performance of Laplacian algorithm. After that, this paper will realize the performance of the algorithm and CLBLAS open-source library for algorithm accuracy and performance comparison, in the algorithm accuracy to maintain the same premise, compared with the corresponding GEMM algorithm in different data sizes of the number of floating-point operations. CLBLAS library is a cross-heterogeneous platforms written using OpenCL open-source BLAS library, the use

of CLBLAS library can be migrated on all platforms that support OpenCL. OpenCL-enabled platforms. It is on this basis that this paper chooses to perform an all-round performance comparison with the corresponding algorithms in the CLBLAS library in the performance comparison section. In this section, this paper compares the performance of commonly used GEMM algorithms in the context of graphics processing, in addition to performance comparisons on large-scale dense input matrix data. The performance of the graph processing optimization algorithm proposed in this paper is compared with the corresponding algorithm in CLBLAS under large-scale dense matrices, and the comparison results are shown in **Figure 3**. The performance of the graph processing optimization algorithm in this paper outperforms the corresponding algorithm in the CLBLAS library at all data sizes. At an input size of 128, the performance of the Laplacian algorithm (100.2 GFLOPS) is about twice that of CLBLAS (50.23 GFLOPS). Unlike CLBLAS the optimized algorithm in this paper still shows a relatively superior performance (9937.85 GFLOPS) with a relatively large input parameter (16384), whereas CLBLAS shows a more drastic performance degradation (8875.74 GFLOPS) in the same case. This point also proves that the Laplacian algorithm implemented in this paper is more adaptable to the scale of the input data, and the performance is more stable and reliable in the large-scale dense matrix computation related to medical product design.

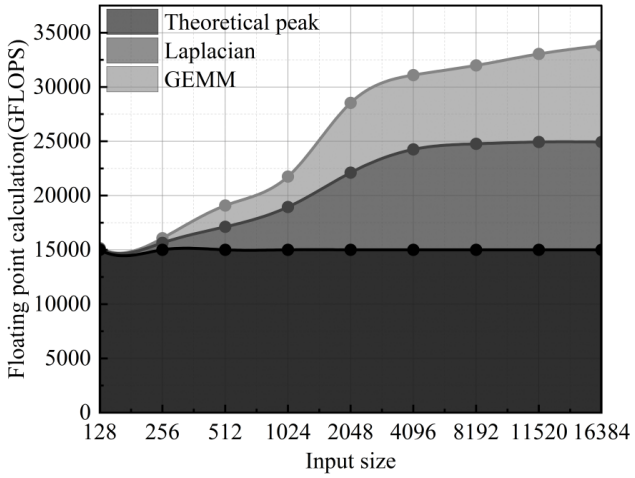

Figure 3. Performance comparison with the CLBLAS open-source library.

3.2. Optimization analysis of visual communication effect in medical product design

3.2.1. Quality analysis of medical high-density graphic reconstruction

In medical product design, high-density excitation two-dimensional imaging images such as cell microtubules and molecular substances are often involved, so this paper uses an optimized graphic processing algorithm to optimize the high-density excitation two-dimensional imaging images in medical product design, to improve the reconstruction quality of the relevant images in medical product design, and to enhance its visual communication effect. The high-density excitation 2D imaging data of 2000 1024 × 1024 pixels (pixel size of 108.3 nm, corresponding to 110 μm × 110 μm field of view) U-2 OS medical cell microtubules in medical

product design were acquired. The Laplacian algorithm localized 7.24 × 10^3 fluorescent molecules per medical image, and the localization density was estimated to be 3.26 μm^{-2} based on the number of molecules divided by the structural area. In addition, the percentage of sparse molecules localized in this data is only about 26.34%, further confirming that this data was acquired at high excitation density. Based on a subregion of 256 × 256 pixels, the results of the characterization resolution analysis of the three graphical processing algorithms localized in this paper are calculated and shown in **Figure 4**. The Laplacian algorithm localized a total of 6.1 × 10^5 molecules, which is lower than the 7.25 × 10^5 molecules of the ThunderSTORM algorithm and the 6.52× 10^5 molecules of the WindSTORM algorithm. However, the normalized intensity of the Laplacian algorithm is better than the other two algorithms, and the normalized intensity between −100 nm~−70 nm is at 1.0. This is mainly because the Laplacian algorithm filters out the molecules with low localization accuracy. In the region of high fluorescence background, fluorescent molecules tend to have lower localization accuracy, and it can be clearly found in the details of the super-resolution maps reconstructed by the three software, and the results processed by the Laplacian algorithm have significantly fewer spurious points (See **Figure 4**).

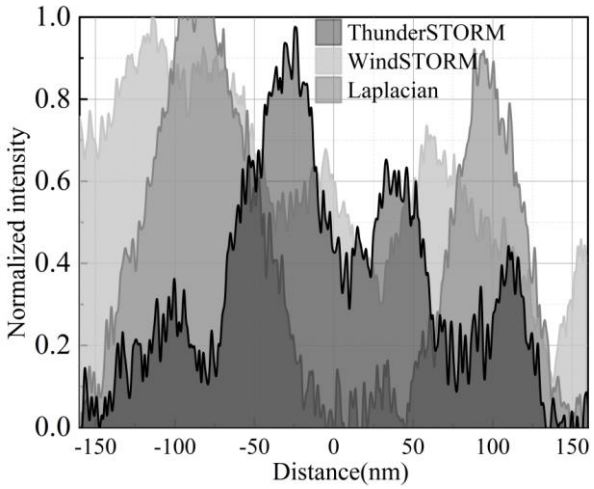

Figure 4. Analysis of high-density location of medical image.

3.2.2. High-density imaging graphic stitching effects

At present, some medical product designers try to extend localization imaging to high-throughput imaging in order to improve the visual communication effect, but due to the low graphic processing speed of the existing methods, the current high-throughput localization imaging is limited to low-density and small-field-of-view imaging, which leads to poor visual communication effect of medical product design images. Based on the graphic real-time processing algorithm proposed in this paper, a larger field of view and higher excitation density can be used for high-throughput localization imaging, so as to efficiently acquire spliced imaging with a large field of view and improve the visual communication effect of medical product design images. Based on a 110 μm × 110 μm flat-field illumination field of view and 10 ms exposure time, this paper continuously imaged 10 × 10, i.e., 100 microtubule 2D

imaging data of U-2 OS medical cells. The Laplacian algorithm processes the raw maps in real time while imaging, and is able to complete localization and super-resolution graph reconstruction immediately after imaging is completed. In this paper, a 5% field-of-view overlap of neighboring fields-of-view is used for inter-correlation-based graph splicing at a later stage. Ultimately, a 1 mm × 1 mm large-area field-of-view spliced super-resolution map is obtained. Compared to a single imaging field of view, this can conveniently provide a variety of different cellular morphologies and interactions to observe, thus facilitating the enhancement of visual communication in medical product design. A spatial resolution of about 50 nm can be obtained by acquiring 100 raw maps per field of view, which is verified by the microtubule profile intensity distribution maps, the results of which are shown in **Figure 5a,b** representing the analyzed results of microtubule profile regions 1 and 2, respectively. The algorithm in this paper is able to accurately splice and acquire 2D imaging of microtubules, and the acquisition time of 100 field-of-view graphs plus the additional time spent on axial drift correction and field-of-view shifting totaled 31 min. Compared to existing high-throughput imaging methods that took 7.8 h to image 95 22.5 μm × 22.5 μm fields of view, this paper improves the throughput by several orders of magnitude.

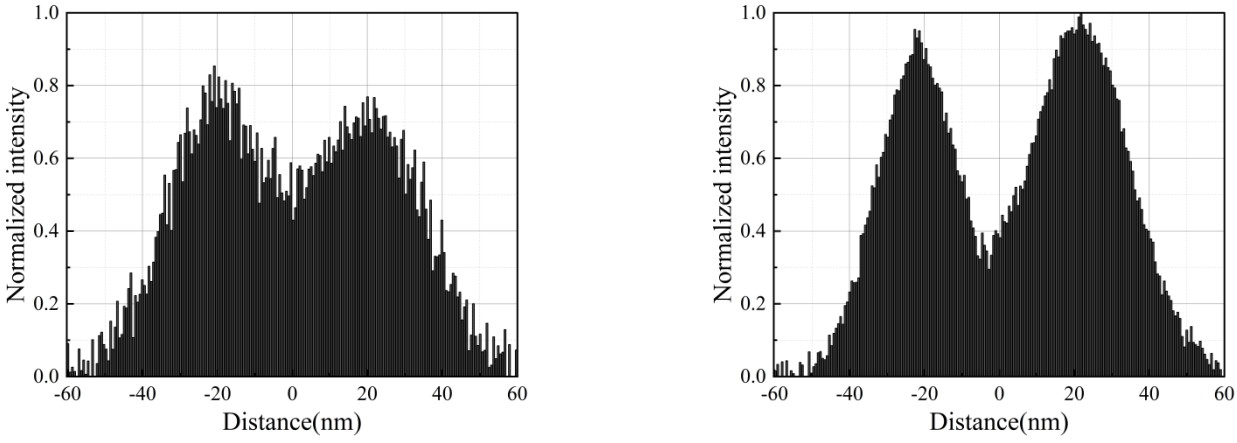

Figure 5. Micropipe profile intensity profile **(a)** Subregion 1; **(b)** Subregion 2.

3.2.3. Results of subjective analysis of visual communication effects

In order to more intuitively show the optimization of the visual communication effect of the graphical processing algorithm of this paper on the medical product design. In this paper, two groups of real medical product images (10 in each group) were selected from the NTIRE 2022 medical product database to conduct subjective perception test of visual communication effect, which was used to prove the effect and applicability of the algorithm in practical application.

The images were processed using the graphic processing algorithm proposed in this paper, and the subjective effect of visual communication of the processed medical product design images was analyzed using human subjective perception MOS values. Subjective quality evaluation is based on the evaluator's subjective feeling and experience of the quality of the image. The principle is that subjects are asked to rate the images one by one according to their visual perception in a specific

experimental environment based on pre-set rating criteria. This process ensures the objectivity and accuracy of the evaluation and provides strong data support for the evaluation of the visual communication effect of images. The results of subjective evaluation are often used to evaluate and optimize image processing techniques, providing an important basis for algorithm improvement. In order to ensure that the image quality evaluation can more accurately express the visual perception of different groups of people, the researchers will increase the number of subjective evaluators and average all the subjective evaluation results in order to arrive at a more objective subjective evaluation score. This is the most convenient and direct evaluation method in image quality evaluation. The formula for subjective quality evaluation is shown below:

$$MOS = \frac{1}{P}\sum_{i=1}^{P} S_i \tag{6}$$

where P is the total number of observers and S_i represents the image quality rating given by each observer.

The results of subjective evaluation of visual communication effect of 20 images related to medical product design processed using the algorithm of this paper are shown in **Table 1**. From the results, it can be seen that the average value of human subjective perception MOS score is 0.856, which indicates that the images related to medical product design processed by the algorithm proposed in this paper have a high score of visual communication effect, which proves that the algorithm has strong superiority in optimizing and improving the visual communication effect of medical product design.

Table 1. Visual communication effect subjective evaluation analysis.

Image number	Subjective evaluation score	Image number	Subjective evaluation score
1	0.900	11	0.756
2	0.849	12	0.951
3	0.835	13	0.878
4	0.868	14	0.942
5	0.796	15	0.793
6	0.873	16	0.732
7	0.975	17	0.943
8	0.743	18	0.752
9	0.900	19	0.895
10	0.792	20	0.956
Average			0.856

4. Conclusion

In this paper, the optimized Laplacian graphic processing algorithm using up-sampling algorithm and threshold filtering algorithm is optimized to enhance the visual communication effect of relevant images in medical product design, and the

performance of the graphic processing algorithm and the processing effect of the visual communication of the images are analyzed, and the results show that:

(1) The threshold filtering optimization algorithm proposed in this paper is able to distribute the load caused by increasing the filtering window to each core of the GPU for parallel execution in a balanced manner, and the algorithm's running time averages 0.505 ms, 0.533 ms, and 0.606 ms in the 3 × 3, 5 × 5, and 7 × 7 filtering windows, respectively. In addition, the performance of the graphical processing optimization algorithm in this paper outperforms the corresponding algorithm in the CLBLAS library for different input data sizes. The floating-point computation values of this paper's algorithm and the CLBLAS comparison algorithm are 9937.85 and 8875.74, respectively, for an input parameter of 16,384.

(2) In the two-dimensional imaging processing of high-density excitation in medical product design, the Laplacian algorithm proposed in this paper is able to filter out molecules with low localization accuracy, and the image processing results have significantly fewer stray points. And the algorithm can accurately splice and acquire microtubule 2D imaging, 100 field of view graphic acquisition time plus axial drift correction and field of view movement and other additional time, a total of 31min, much lower than other algorithms. It is also found that the average human subjective perception MOS score of medical product design images processed by the algorithm in this paper is 0.856, which proves that the algorithm has strong superiority in optimizing and enhancing the visual communication effect of medical product design.

The graphic processing algorithm designed in this paper can ensure the uniformity of the imaging quality of the relevant images in the medical product design, which lays a strong foundation for the improvement of its visual communication effect.

In conclusion, this method is applied to the design side of medical products, which makes the visual communication effect significantly improved. However, the method in this paper also has some limitations, including the high detail and complexity of the generated visual materials for medical products, which makes its aesthetic threshold and application scenarios always limited. In addition to the application of technical research, there are also further studies on the inward subdivision of visual system, the quantitative expression of design rules, and the collaborative layout of design strategies. In the future, the collaborative help of artificial intelligence can be added to optimize the above problems from the thinking mode and design process strategy.

Ethical approval: Not applicable.

Conflict of interest: The author declares no conflict of interest.

References

1. Mang, B., Oh, Y., Bonilla, C., & Orth, J. (2023). A medical equipment lifecycle framework to improve healthcare policy and sustainability. Challenges, 14(2), 21.
2. Hennein, R., Goddard, E., & Sherman, J. D. (2022). Stakeholder perspectives on scaling up medical device reprocessing: A qualitative study. Plos one, 17(12), e0279808.

3. Elabed, S., Belal, A., & Shamayleh, A. (2019, October). Sustainability of medical equipment in the healthcare industry: An overview. In 2019 Fifth International Conference on Advances in Biomedical Engineering (ICABME) (pp. 1-4). IEEE.
4. Chen, J., Mo, R., Wu, L., & Yu, S. (2016, August). A method of collaborative task allocation for cloud service platform of industrial design. In 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC) (Vol. 1, pp. 484-487). IEEE.
5. Arefin, A. M., Khatri, N. R., Kulkarni, N., & Egan, P. F. (2021). Polymer 3D printing review: Materials, process, and design strategies for medical applications. Polymers, 13(9), 1499.
6. Lo, H. W., Gul, M., Yucesan, M., & Liaw, C. F. (2024). A risk assessment model with dependent failure modes for the manufacturing and design processes of medical equipment. International Journal on Interactive Design and Manufacturing (IJIDeM), 1-16.
7. Javaid, M., & Haleem, A. (2018). Additive manufacturing applications in medical cases: A literature based review. Alexandria Journal of Medicine, 54(4), 411-422.
8. Zhang, J., Hu, Q., Wang, S., Tao, J., & Gou, M. (2019). Digital light processing based three-dimensional printing for medical applications. International journal of bioprinting, 6(1), 242.
9. Zhang, Z., & Rao, W. (2021). Key risks and development strategies for china's high-end medical equipment innovations. Risk Management and Healthcare Policy, 3037-3056.
10. Bouget, D., Allan, M., Stoyanov, D., & Jannin, P. (2017). Vision-based and marker-less surgical tool detection and tracking: a review of the literature. Medical image analysis, 35, 633-654.
11. Sastri, V. R. (2021). Plastics in medical devices: properties, requirements, and applications. William Andrew.
12. Laursen, J., Danielsen, A., & Rosenberg, J. (2014). Effects of environmental design on patient outcome: a systematic review. HERD: Health Environments Research & Design Journal, 7(4), 108-119.
13. Calori, C., & Vanden-Eynden, D. (2015). Signage and wayfinding design: a complete guide to creating environmental graphic design systems. John Wiley & Sons.
14. Pati, D., Harvey Jr, T. E., Willis, D. A., & Pati, S. (2015). Identifying elements of the health care environment that contribute to wayfinding. HERD: Health Environments Research & Design Journal, 8(3), 44-67.
15. Goyanes, A., Madla, C. M., Umerji, A., Piñeiro, G. D., Montero, J. M. G., Diaz, M. J. L., ... & Basit, A. W. (2019). Automated therapy preparation of isoleucine formulations using 3D printing for the treatment of MSUD: First single-centre, prospective, crossover study in patients. International Journal of Pharmaceutics, 567, 118497.
16. Kellert, S., & Calabrese, E. (2015). The practice of biophilic design. London: Terrapin Bright LLC, 3(21).
17. Javaid, M., & Haleem, A. (2019). Industry 4.0 applications in medical field: A brief review. Current Medicine Research and Practice, 9(3), 102-109.
18. Berg, L. P., & Vance, J. M. (2017). Industry use of virtual reality in product design and manufacturing: a survey. Virtual reality, 21, 1-17.
19. Guantao Chen & Fernando C. Tura. (2024). A linear algorithm for obtaining the Laplacian eigenvalues of a cograph. Special Matrices(1).
20. Ruikai Xin,Jiarui Lin,Shendong Shi,Rao Zhang,Jianlong Zhang & Jigui Zhu. (2024). Research on dual-threshold detection based adaptive fault-tolerant kalman filtering algorithm for dynamic 6-DOF measurement. Mechanical Systems and Signal Processing111190-.
21. Yun JIANG,Huiyang LIU,Xiaopeng JIAO,Ji WANG & Qiaoqiao XIA. (2024). Triangle Projection Algorithm in ADMM-LP Decoding of LDPC Codes:Regular Section. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences(8),1364-1368.
22. Fei Tang. (2018). Application of vector graphics binaryzation and refining treatment in two-dimensional animation modeling. Journal of Discrete Mathematical Sciences and Cryptography(5),1167-1175.