ALGORITHM FOR FINDING CORRECTION COEFFICIENTS OF INTELLECTUAL SYSTEM

Abdullayeva Ozoda Safibullayevna
Phd, professor, Namangan Engeneering-Construction Institute

https://doi.org/10.5281/zenodo.12579119

Abstract. *This article presents an algorithm for determining reference brightness correction coefficients to improve image quality. The algorithm utilizes a combination of statistical analysis and image processing techniques to identify and correct brightness discrepancies in digital images. By establishing a robust and efficient method for calculating these coefficients, the algorithm aims to enhance the overall visual fidelity of images across various applications, such as photography, medical imaging, and remote sensing. The proposed algorithm demonstrates its effectiveness through experimental results and comparisons with existing methods, highlighting its potential for practical implementation in image processing workflows. In the world, the filtering of digital images by the convolution method with a pulse characteristic in the spectral region scientific research is being conducted to improve the quality level of digital television images, methods for modeling filtration processes and highly efficient control systems in a number of priority areas, including: on the formation of mathematical models of filtration processes, improving the methods of wavelet, Fourier, Haar, Walsh-Hadamard, Karhunen-Loev in increasing the clarity and brightness of images based on linear and nonlinear differential equations; creation of methods for eliminating additive, pulsed and adaptive-Gaussian types of noise in images using additive and adaptive filtering; methods of algorithms and software for introducing intra-frame and inter-frame image transformations; methods of adaptive brightness system control using the Chebyshev matrix series; methods of gradient, static and Laplace methods for image segmentation and dividing it into contours; formation of criteria and conditions for evaluating image quality. Conducting scientific research in the above research areas confirms the relevance of the topic of this article.*

Keywords: *adaptation of the brightness control mechanism, processing dynamic image, sequences, video codecs.*

INTRODUCTION

Images captured or displayed through digital means often suffer from variations in brightness, leading to reduced visual quality. Correcting these brightness inconsistencies is crucial for applications spanning photography, medical imaging, and remote sensing. This article introduces an algorithm designed to ascertain reference brightness correction coefficients for improving image quality. By leveraging statistical analysis and image processing techniques, the algorithm aims to accurately identify and rectify brightness variations, thereby enhancing visual fidelity and enabling a more accurate depiction of the original scene. The proposed algorithm seeks to address the challenges associated with brightness correction and offers a promising solution for optimizing image quality across diverse domains. One of the most urgent tasks in the field of audio-video data processing is the improvement of audio-video data compression methods, taking into account the elimination of temporary redundancy of TV images and audio accompaniment. This problem is very relevant in conditions of limited frequency resources. In addition, it becomes possible to significantly reduce the preparation time for television reports to be broadcast directly from the event sites by transmitting signals from TV cameras directly to the installation hardware of television centers over cellular networks. At the same time, there is no need to use expensive and not always available broadband communication channels [1].

More scientists and their students P.A. Bakut, A.I. Galushkin, Yu.V. Gulyaev, Yu.I. Zhuravlev, N.G. Zagoruiko, A.A. Potapov, V.A. Snetkov, V.A. Soyfer, Ya.A. Furman made a special contribution to the development of methods and means of digital processing of television images and pattern recognition [1], as well as such domestic and foreign authors of computational methods, models and algorithms, including tracking objects and subjects on dynamic television images, as B.A. Alpatov, P.V. Babayan, Yu.V. Visilter, V.I. Vasiliev, U. Grenander, R. Duda, A.E Kononyuk., A.L Priorov., Y. Fisher, P. Khart, B.D. Lucas, D. Marr, T. Kanade, W.K. Pratt, L.G. Roberts, I.E. Sobel [1]. For example, the batch method of comparing dynamic television images and recorded images has its own background [6, 7] and in a certain sense, it is based on the results of a number of scientific works on the study of textures. Thus, when studying the image texture, a method was applied based on the analysis of a family of images obtained with a combination of distortions of the analyzed image using a variety of filters, while the combination of filters was determined intuitively by visual analysis of images (stripes, spots, etc.). The tasks that were solved in these works were not related to face recognition, but allowed us to draw practical significant conclusions: firstly, filters should not be complicated, it is sufficient in some cases to use linear combinations of Gaussian filters; secondly, if possible, it is necessary to use such filtering methods that generate visually understandable images, that is, intuition should not be abandoned.

METHODOLOGY

The algorithm for finding reference brightness correction coefficients can be summarized as follows:

Image Acquisition:

Acquire a reference image using the imaging system to be standardized. This image should represent the typical conditions and features of the objects or samples to be imaged.

Image Analysis:

Divide the reference image into smaller regions of interest (ROIs) that capture a range of brightness levels and features.

Calculate Average Intensity:

Compute the average intensity value for each ROI to obtain a representative measure of the brightness variation within the image.

Standardization:

Identify a subset of ROIs that exhibit consistent and desirable brightness characteristics. These ROIs will serve as the reference areas for standardization.

Calculate Correction Coefficients:

Determine correction coefficients by comparing the average intensity values of the selected reference ROIs to a target value or a standard reference image. This comparison helps derive the coefficients needed to adjust the brightness levels of the entire image.

Apply Correction Coefficients:

Utilize the calculated correction coefficients to adjust the brightness of subsequent images captured by the same imaging system, ensuring uniformity and accuracy across different acquisitions.

It's important to iterate and refine this process as needed, considering factors such as the specific imaging modalities, environmental conditions, and the requirements of the imaging application. Additionally, validation using standardized reference images or phantoms can help ensure the accuracy and effectiveness of the correction coefficients.

Algorithm for finding reference brightness correction coefficients consist of folowing steps:

- Collect a set of reference images with different levels of brightness and a reference brightness value for each image.
- Convert the images to a common color space, such as RGB or YCbCr.
- For each image, calculate the average brightness value using the reference brightness value as a reference point.
- Create a scatter plot of the reference brightness values against the average brightness values for all reference images.
- Fit a linear regression model to the scatter plot to find the best-fit line.
- Calculate the slope and intercept of the best-fit line to obtain the brightness correction coefficients.
- The slope represents the correction factor for brightness, and the intercept represents the offset.
- Apply the brightness correction coefficients to new images to adjust their brightness levels accordingly.
- The methodology for the algorithm designed to find reference brightness correction coefficients can be outlined as follows:
- Data Collection: Gather a dataset of images exhibiting variations in brightness. These may include images captured under different lighting conditions or displayed on screens with varying brightness levels.
- Pre-processing: Normalize the images to ensure consistent color balance and remove any artifacts or noise that may affect the brightness analysis.
- Statistical Analysis: Utilize statistical measures such as mean, median, and standard deviation to quantify the brightness levels of each image. This step aims to identify the extent of brightness variations within the dataset.
- Reference Image Selection: Choose a reference image from the dataset that is representative of the desired brightness level. This can be achieved through visual inspection or automated selection based on predefined criteria.
- Coefficient Calculation: Develop an algorithm to calculate the correction coefficients by comparing the brightness of each image to that of the reference image. This may involve mathematical transformations or image processing techniques to derive the necessary adjustments.
- Validation and Optimization: Validate the algorithm using a separate set of images with known brightness variations. Fine-tune the algorithm parameters to optimize its performance across a wide range of scenarios.
- Implementation: Integrate the algorithm into a software framework or platform that enables users to automatically apply the computed correction coefficients to their images.

Evaluation: Assess the effectiveness of the algorithm by comparing the corrected images with the original ones, using metrics such as peak signal-to-noise ratio (PSNR) or subjective user evaluations to gauge the visual improvements.

The brightness control algorithm is designed to correct image defects by applying the formula (1) [1].

$$z_n = U_{\theta-1}^{-1}\left(\frac{1}{2}C\right)U_{\theta-n-1}\left(\frac{1}{2}C\right)\left[z_0+\sum_{k=1}^{n-1}U_{k-1}\left(\frac{1}{2}C\right)(u_k-v_k)\right]$$
$$+U_{\theta-1}^{-1}\left(\frac{1}{2}C\right)U_{n-1}\left(\frac{1}{2}C\right)\left[z_\theta+\sum_{k=n}^{\theta-1}U_{\theta-k-1}\left(\frac{1}{2}C\right)(u_k-v_k)\right]$$

where: z_0 and z_θ - rows of source data for calculating correction coefficients, they are set directly from the image (these are the first and last rows of the area under study);

$U_{\theta-1}^{-1}$- the inverse matrix of the Chebyshev matrix polynomial in degree $\theta - 1$; $U_{\theta-n-1}^{-1}$- matrix Chebyshev polynomial in degree $\theta - n - 1$; $C -$ the square Jacobiev matrix is a tridiagonal matrix, an argument for calculating Chebyshev polynomials U_n;

U_{k-1} matrix Chebyshev polynomial in degree$k - 1$; u_k and v_k-control parameters in the form of vectors; $n -$ row number in the correction factor matrix for the brightness matrix; $k -$ the current index of the summed array; $\theta -$ a parameter indicating the dimension of all the matrix (brightness, correction) control vectors.

As a result of processing, the required brightness is achieved, which allows you to adjust the brightness of pixels in the selected area of the image due to control coefficients.

In addition, as a result of the algorithm, two results are obtained:

1) The area of the image where the defect was is corrected in accordance with its environment.

2) The control coefficients u and v are calculated for the entire section where the correction is performed. They can be applied to subsequent images with the same defect and its surroundings.

This algorithm uses two basic formulas:

$U_{n+2}(x) = 2 * U_{n+2}(x) - U_n(x), n \geq 0, \quad U_0(x) = 1, U_1(x) = 2x,$

where: U_{n+2} - Chebyshev matrix polynomial calculated at the next step ($n + 2$ – degree of the polynomial); U_{n+1} Chebyshev matrix polynomial, which was calculated in the previous step ($n + 2$ – the previous degree of the polynomial); U_n Chebyshev's matrix polynomial, which was calculated two steps before the current one; $x -$ the input parameter on the basis of which the polynomial is calculated $U_{-2}(X) = E$, $U_{-1}(X) = \theta$, $U_0(X) = E$, $U_1(X) = 2X$, where E – single, and θ – zero matrix.

Thus, by searching for control coefficients u, v, using Chebyshev polynomials of the second kind (1) and (2), it is possible to adjust the brightness in accordance with a given range $(\beta - \varepsilon, \beta + \varepsilon)$, where ($\beta$ - minimum brightness, $\varepsilon -$ the difference between maximum and minimum brightness, $z_{i,j}$- the brightness of this point) within the workspace. As a result of the measurement β, ε the brightness range is being set $(\beta - \varepsilon, \beta + \varepsilon)$, which should include all points from the workspace after correction. In this case, two additional matrices are created with the size of the selected image area, which are filled with values U and V. Next, the algorithm looks through all the points of the image and calculates the values for each one by iteration U and V, corresponding to the condition, and the values found are then written to the corresponding elements of additional matrices. After all the points are viewed, the process of summing their current values with the calculated ones takes place U and V, i. e.

$Z''_{i,j} = z_{i,j} + Z'_{i,j}$,

where Z and Z' – matrix of brightness values before and after correction, U and V - matrices of control values. As a result of this step, the sharpness and clarity of the image increase.

RESULTS

This algorithm uses the following processing steps:

1) The square workspace is highlighted in the image since the use of only square matrices is due to the presence in formula (1) of an inverse matrix that exists only for square matrices. Next, the area under study is determined by adding one row of pixels of the current image, the width of the workspace, from above and from below to the workspace.

2) The search is made for reference brightness correction coefficients, which is carried out according to the following rules:

if $z_{i,j} \in [\beta - \varepsilon, \beta + \varepsilon] \rightarrow K_{i,j} = 0$,

if $z_{i,j} < (\beta - \varepsilon) \rightarrow K_{i,j} = (\beta - \varepsilon) - z_{i,j}$,

if $z_{i,j} > (\beta + \varepsilon) \rightarrow K_{i,j} = (\beta + \varepsilon) - z_{i,j}$.

The brief description of the algorithm (functional blocks of the algorithm are indicated in parentheses): the search begins (1) by opening the width loop (2) and then opening the height loop (3), in which checks occur: if the brightness of the current point is in the range $[\beta - \varepsilon, \beta + \varepsilon]$ (4), then the coefficient is zero (5), otherwise, if it is smaller $[\beta - \varepsilon]$ (6), then the coefficient is equal to $(\beta - \varepsilon) - z_{i,j}$ (7), otherwise, if it is smaller $[\beta - \varepsilon]$ (8), then the coefficient is equal to $[\beta + \varepsilon] - z_{i,j}$ (9). The algorithm is completed (10).

3) Preparation of N matrix Chebyshev polynomials is performed (from U_0 to $U_{\theta-1}$). Formula (2) is used to calculate the coefficients, we substitute it into formula (1). In this case, the coefficients are expressed as Chebyshev matrix polynomials to the degree from 0 to θ-1.

Description of auxiliary functions (getMatrixE, multipleMatrix, getNextU and others) is given at the end of the section. The brief description of the algorithm (functional blocks of the algorithm are indicated in parentheses): the algorithm begins (1) calculating values for U_0 and U_1 (2).

4) Correction coefficients are calculated for n the line of the investigated area, $n\epsilon[1, \theta - 1]$. Moreover, for each row (except the first and last, they are used as reference) from the study area, the control coefficients are calculated using the formula (1) u, v and the brightness correction coefficients are calculated from them.

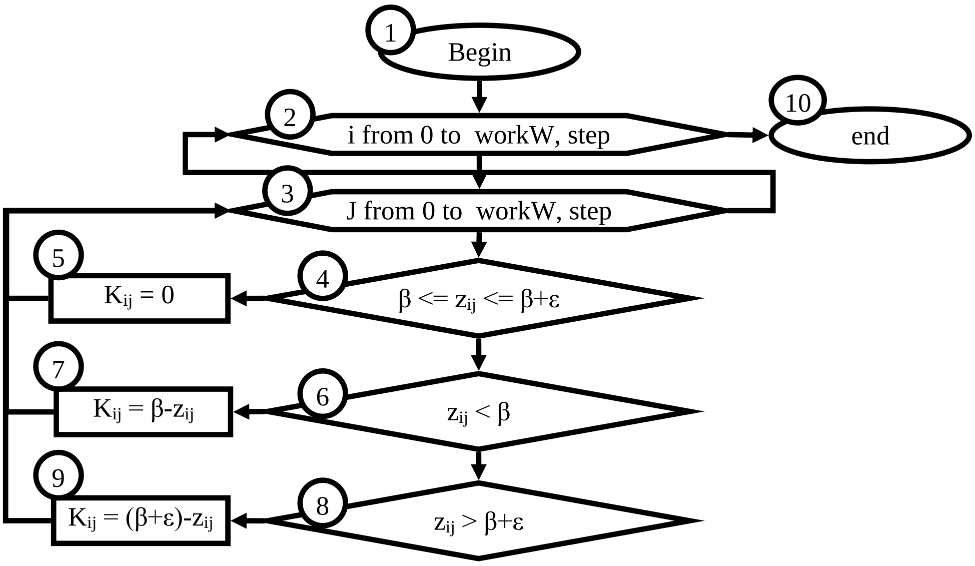
1
Begin
2
i from 0 to workW, step
10
end
3
J from 0 to workW, step
5
$K_{ij} = 0$
4
$\beta <= z_{ij} <= \beta+\varepsilon$
7
$K_{ij} = \beta - z_{ij}$
6
$z_{ij} < \beta$
9
$K_{ij} = (\beta+\varepsilon) - z_{ij}$
8
$z_{ij} > \beta+\varepsilon$

FIGURE 1. Algorithm for finding reference brightness correction coefficients.

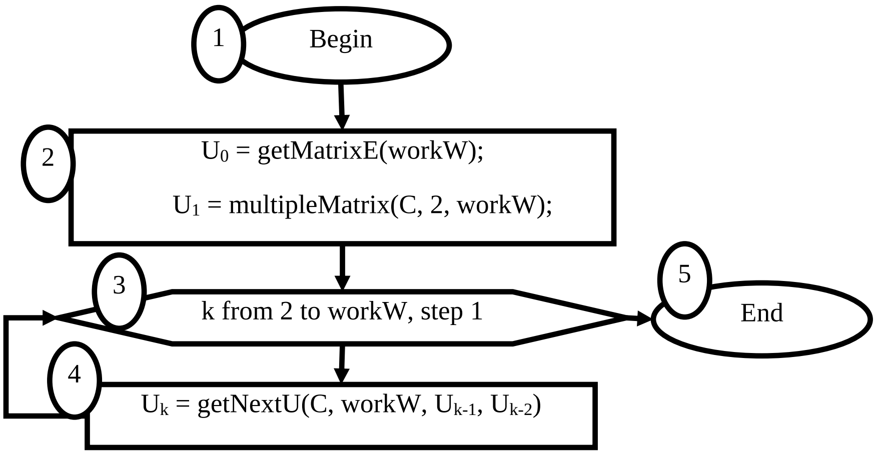
1
Begin
2
U_0 = getMatrixE(workW);
U_1 = multipleMatrix(C, 2, workW);
3
k from 2 to workW, step 1
5
End
4
U_k = getNextU(C, workW, U_{k-1}, U_{k-2})

FIGURE 2. Algorithm for calculating Chebyshev matrix polynomials.

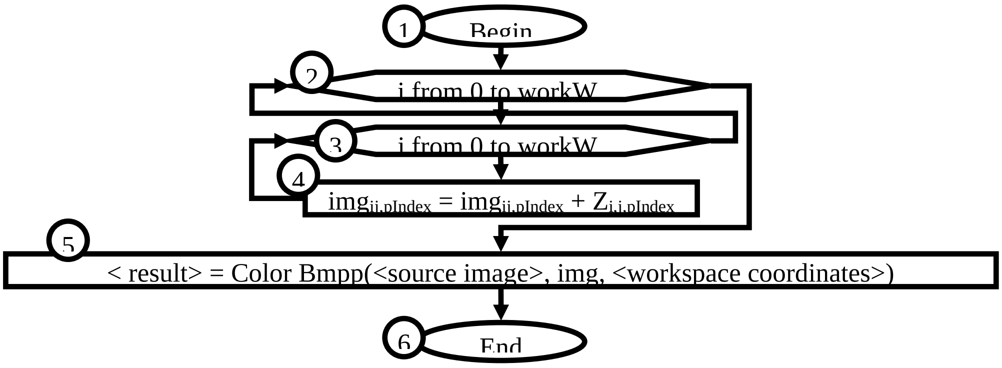
1
Begin
2
i from 0 to workW
3
i from 0 to workW
4
$img_{ij,pIndex} = img_{ij,pIndex} + Z_{i,j,pIndex}$
5
< result> = Color Bmpp(<source image>, img, <workspace coordinates>)
6
End

FIGURE 3. Algorithm of correction coefficients calculation.

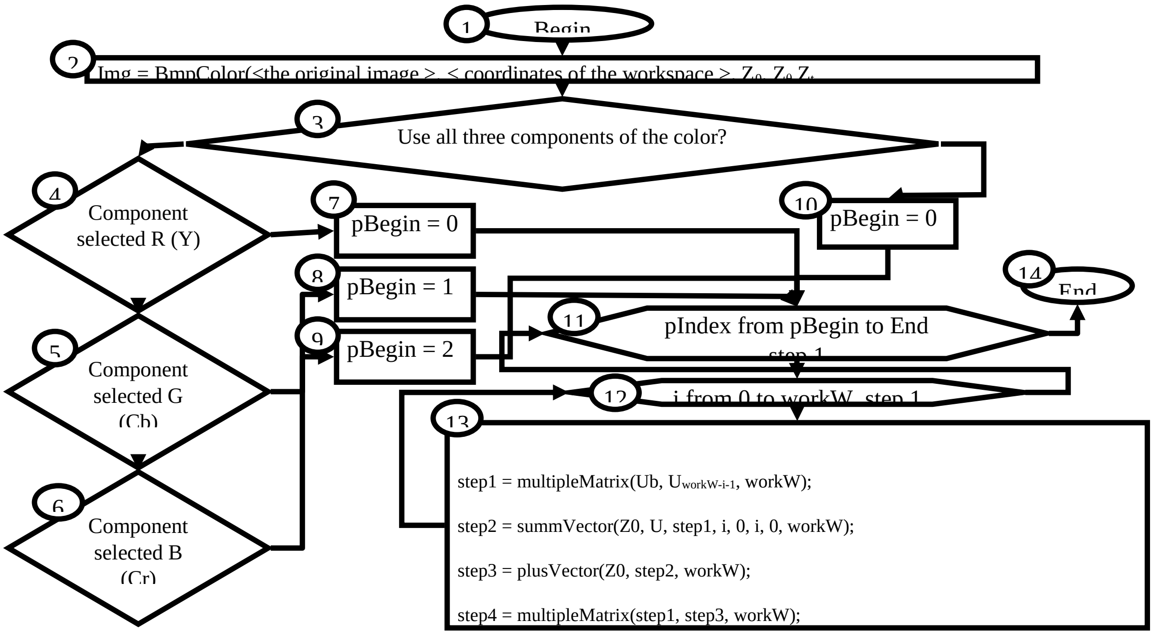
1
Begin
2
Img = BmpColor(<the original image >, < coordinates of the workspace >, Z0, Zθ Zt
3
Use all three components of the color?
4
Component selected R (Y)
7
pBegin = 0
10
pBegin = 0
8
pBegin = 1
14
End
9
pBegin = 2
11
pIndex from pBegin to End step 1
5
Component selected G (Cb)
12
i from 0 to workW, step 1
13
step1 = multipleMatrix(Ub, UworkW-i-1, workW);
step2 = summVector(Z0, U, step1, i, 0, i, 0, workW);
step3 = plusVector(Z0, step2, workW);
step4 = multipleMatrix(step1, step3, workW);
6
Component selected B (Cr)

FIGURE 4. Algorithm for determining correction coefficients and control coefficients.

The brief description of the algorithm (the functional blocks of the algorithm are indicated in parentheses): the algorithm begins (1) by loading the selected area into the img variable (2), then it is checked whether all three color components need to be processed (3): if so, the pBegin and pEnd variables are set to 0 and 2, respectively (10), otherwise it is checked for which color component processing is required - for red/brightness (4), in this case the variables pBegin and pEnd are set to 0 (7); for green/chromatic blue (5), in this case the variables pBegin and pEnd are set to 1 (8); for blue/chromatic red (6), in this case the variables pBegin and pEnd are set to 2 (9). Next a cycle opens from begin to pEnd inclusive with a step of 1 (11) and inside it a height cycle with a step of 1 (12), in which the formula (2) (13) is applied to the current row. The algorithm is completed (14).

5) The correction coefficients found are used for image reconstruction (Fig. 4).

Brief description of the algorithm (in parentheses are the functional units of the algorithm): the algorithm starts (1) discovery cycle on height with a step of 1 (2), and then the opening of the cycle width in increments of 1 (3), within which the current value of the color component is formed with the appropriate correction factor and the result of addition is written as the current value of the color component, erasing the previous value (4).

Description helper functions.

1) BmpColor - returns a description of the workspace in the form of a matrix of three-digit elements (R, G, B), as well as vectors of the original values z0 and zθ in the form of output parameters. If necessary, converts the palette to YCbCr.

Input parameters: output image, selection coordinates.

2) ColorBmp - returns a description of the workspace in the form of an "image" object. Input parameters: source image, matrix of three-digit elements (R, G, B), selection coordinates. getMatrixC - returns a matrix with a parameter for calculating Chebyshev matrix polynomials.

Input parameters: matrix size, divisor (default is 1).

3) getMatrixE - returns the unit matrix. Input parameters: matrix size.

4) getBackMatrix - returns the matrix inverse of the given one. The method of elementary transformations is used. Input parameters: the original matrix, the size of the matrix.

5) multipleMatrix - returns the product of: a. two matrices, input parameters: first matrix, second matrix, matrix size; b. matrices and vectors, input parameters: matrix, vector, matrix size; c. matrices and numbers, input parameters: matrix, number, matrix size.

6) plusMatrix - returns the sum of two matrices. Input parameters: first matrix, second matrix, matrix size.

7) minusMatrix- returns the difference of two matrices. Input parameters: first matrix, second matrix, matrix size.

8) plusVector - returns the sum of two vectors. Input parameters: first vector, second vector, vector size.

9) getVectorW - returns a vector of control coefficients values for the processed string. Input parameters: vector of initial values (z0 or zθ), the matrix is the result of calculating the first (fifth) step of the formula, the number of the processed row, the Chebyshev matrix polynomial, the divisor, the size of the matrix.

2) summVector - returns the sum of the products of Chebyshev matrix polynomials on the control vector.

Input parameters: a vector of initial values (z0 or zθ), an array of Chebyshev matrix polynomials, a matrix-the result of calculating the first (fifth) step of formula (1), the number of the processed row, the initial summation index, the final summation index, the size of the matrix.

3) getNextU - returns the Chebyshev matrix polynomial. Input parameters: matrix - the value of the polynomial, the size of the matrix, the polynomial from the previous step, the polynomial from the pre-previous step.

DISCUSSION

In one case, when determining the spectral coefficients $|M_c(u,v)|$. The discrete cosine transform (DCT) was used, and the discrete Fourier transform (DFT) was used in the other. It can be seen from the figure that for two different bases, the discrete spectra differ greatly from each other. Therefore, the definition $K(u,v)$ it should be carried out as follows. First, by means of a two-dimensional integral Fourier transform based on a known analog frequency transfer function $K(\omega_x, \omega_y)$ the corresponding impulse response is found $h(x,y)$ which is subjected to spatial sampling at the same interval as the filtered image. After that, the discrete impulse response is normalized so that the sum of all its samples is equal to one. And only then is the discrete frequency transfer function calculated $K(u,v)$ by applying to $h(k,n)$ a two-dimensional orthogonal transformation using the same basic functions as when calculating spectral coefficients.

CONCLUSION

Threshold transformations occupy a central place in applied problems of image segmentation due to intuitive properties, ease of implementation, and computational speed. In this section, a more formal definition of the threshold transformation is given and on its basis, much more general methods of threshold processing are constructed than are described so far.

Suppose that shown in Fig. 2. the histogram corresponds to some image $f(x,y)$, containing bright objects against a dark background so that the brightness of the pixels of the object and background are concentrated near the two prevailing values. The obvious way to distinguish objects from the surrounding background is to select a threshold value T, delimiting brightness distribution modes. Then any point (x, y), wherein $f(x, y) > T$, called the point of the object, otherwise, the point of the background. In other words, a segmented image $g(x, y)$ is defined by the relation

$$g(x,y) = \begin{cases} 1, & if\ f(x,y) > T \\ 0, & if\ f(x,y) \leq T \end{cases}.$$

If the value T since there is a constant applied to the entire image, this formula describes the so-called global threshold transformation. If the threshold value is not constant in the image, one speaks of a transformation with a variable threshold. Speaking of a threshold transformation, the terms “local” or “in the neighborhood” are sometimes used to indicate that a variable threshold T anywhere in the image (x, y) depends on the properties of the neighborhood (x, y) (for example, from the average brightness of the pixels in this neighborhood). If the threshold T depends on the characteristics of the processed image and changes with spatial coordinates x and y, then this variable threshold transformation is often called dynamic or adaptive.

In conclusion, the algorithm outlined in this article provides a systematic approach for finding reference brightness correction coefficients. By following the steps outlined, researchers and technicians can effectively determine the coefficients needed to correct for variations in brightness between different imaging systems. This method can be applied across various fields, from medical imaging to quality control in manufacturing, to ensure accurate and consistent image analysis. Implementing this algorithm can lead to improved reliability and reproducibility of results, ultimately benefiting scientific research and industrial applications. Further research and experimentation may be necessary to refine and validate this algorithm for specific applications, but it serves as a valuable tool for standardizing brightness correction in imaging systems.

REFERENCES

[1] V. M. Artyushenk, O. I. Shelukhin, and M. Yu. Afonin, Digital compression of video information and sound (Dashkov and K, Moscow, 2003).

[2] D. Vatolin, A. Ratushnyak, M. Smirnov, and V. Yukin, Data compression methods (Dialog-MIFI, Moscow, 2003).

[3] D. Salomon, A guide to data compression methods (Springer Science & Business Media, New York, 2002).

[4] V. I. Vorobyov, V. G. Gribunin. Theory and practice of the wavelet transform (VUS, St. Petersburg, 1999)

[5] N. G. Zagoruiko, I. A. Borisova, V. V. Dyubanov, and O. A. Kutnenko, Pattern recognition and image analysis 18, 1-6 (2008).

[6] K. Onthriar, K. K. Loo, and Z. Xue, “Performance comparison of emerging Dirac video codec with H. 264/AV,” in International Conference on Digital Telecommunications (ICDT'06) (IEEE, Cote d'Azur, 2006) pp. 22-25.

[7] S. S. Beknazarova, A. S. Mukhamadiyev, and M. K. Jaumitbayeva, “Processing Color Images, Brightness and Color Conversion,” in 2019 International Conference on Information Science and Communications Technologies (IEEE, Tashkent, 2019), pp. 1-5.

[8] N. Sedova, R. Bazhenov, V. Sedov, A. Karavka, and S. Beknazarova, “Automated Stationary Obstacle Avoidance When Navigating a Marine Craft,” in 2019 International Multi-

Conference on Engineering, Computer and Information Sciences (SIBIRCON) (IEEE, Novosibirsk, 2019), pp. 0491-0495.

[9] D. Salomon, Data compression (Springer, New York, 2007).

[10] K. Sayood, Introduction to Data Compression (Wiley-Interscience, New York, 2006).

[11] C. Krintz and S. Sucu, IEEE Transactions on Parallel and Distributed Systems 17, 15-24 (2006).

[12] Y. Pan, X. Guo, H. Lin, and S. Luo, Proceedings of 2020 ACM International Conference Proceeding Series (Association for Computing Machinery, Ottawa, 2020), pp. 130-133.

[13] E. A. Lomtev, M. G. Myasnikova, N. V. Myasnikova, and B. V. Tsypin, Measurement techniques 58, 250-255 (2015).

[14] Beknazarova S., Abdullayeva O., Abdullayev Z. METHODS OF PROCESSING INFORMATION RESOURCES IN MULTIMEDIA SYSTEMS //Конференции. – 2021.