CHAPTER 1

INTRODUCTION

1.1 IMAGE PROCESSING

Images play a vital role in visual perception. Image is usually represented by collection of values called pixels and it may be two- dimensional (binary image) or three-dimensional (color image). In Digital Image Processing, the image can be processed by low-level, mid-level or high-level processing methods. Low-level processing is used for reducing noise and for sharpening and enhancement of images. Mid-level processing is used for any particular task in analysis. High-level processing is for recognized objects. Digital image processing advances over analog by avoiding signal distortion during processing. Digital image processing provides a practical technology for texture analysis, classification, feature extraction, Pattern recognition, Projection and Multi-scale signal analysis.

1.2 FEATURES OF AN IMAGE

Feature of an image includes the color, shape, edge and texture. A color model is a system that uses three primary colors to create a larger range of colors. There are different kinds of color models used for different purposes, and each has different components of colors they can produce. Color space is a mathematical model which allows representation, visualization and reproduction of colors. Edges are defined by their sudden change of discontinuity in pixels and also they are the boundaries of an object in an image. Texture may be said as a repeated pattern which may be regular, near regular, irregular, stochastic or near-stochastic. It is also stated as linearly varying pixel intensity values.

1.3 COLOR MODEL

A color model is an abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values or color components. There are different types of color models. They are RGB, YUV, CMY, HSV, YCbCr etc. In this proposed method we used both RGB YCbCr color model. In YCbCr, Y represents the luminance (light) component and C_B and C_R are the blue-difference and red-difference chroma components. The whole range of colors that a specific type of color model produces is called a color space.

1.4 TEXTURE ANALYSIS

An image texture is a set of metrics calculated in image processing and it is designed to quantify the perceived texture of an image. Image texture gives us information about the spatial arrangement of color or intensities in an image or selected region of an image. Texture analysis can be used widely when an object is characterized using its texture than its intensity values. Simply, it is the characterization of regions of an image by their texture content. Texture analysis involves identifying the texture component and extraction of it.

1.5 ILLUMINATION INVARIANCY

Illumination refers to the light distribution or the luminance of an image. Illumination invariancy is referred when a light is highly focused on a particular region in an image which causes uneven distribution of light. Illumination is an important factor to affect image quality in computer vision. In order to solve illumination many algorithms are used under spatial domain like tone mapping, SVD etc. In this proposed work, we identify the illuminated region by extracting the luminance value in YCbCr color model.

1.6 GOAL

In this proposed work, we try to detect the illuminated region by luminance calculation, and the bright spot of the image is calculated by applying average filter. Highly illuminated regions are extracted by bit slicing algorithm and then normalized, with box-filter.

1.7 MOTIVATION

Illumination of an image can vary according to the nature of the lightning conditions present in the surroundings and this may result in uneven distribution of light. This phenomenon may result in ruining the quality of the data present in the image. In order to solve this problem, we propose a novel methodology for illumination correction by extracting and normalizing the luminance component present in the image.

1.8 ORGANIZATION OF THE REPORT

The report is organized as follows

- Literature Survey
- Architectural Design
- Proposed illumination detection and correction method
- Experiments and Results
- conclusion

In chapter 2, the related literatures to the proposed work are collected and presented. In chapter 3, the architectural design of the proposed illumination correction algorithm is presented. In chapter 4, the proposed method is explained and demonstrated. In chapter 5, experiments and results of the proposed system is given. Finally, chapter 6 explicates the conclusion for the proposed system.

CHAPTER 2
LITERATURE SURVEY

2.1 INTRODUCTION

Illumination invariancy refers to uneven distribution of light in an image. The illumination incident on a scene always reduces the performance reliability of many computer vision systems and creates complexity in detection of low level contents present in these images. This phenomenon is corrected by altering the luminance values in the highly illuminated image.

2.2 LITERATURE REVIEW

In the method reported in [1] , YCbCr color model is used for skin color modeling . Based on the skin color model, face detection is performed. Detection of skin color pixels in image is based on histogram estimation. This shows mechanism for face detection and also gives a solution to gain face detection efficiency for the faces with skin color types with lower luminance values. A single value decomposition scheme [3] is used in spatial domain to solve the illumination. In this paper, the Lambertian model is explained, which defines the pixel value as a product of reflectance and illumination components and then the illuminated region is detected and normalized using distribution of local intensities and corresponding coefficients.

In [13], an enhanced gradient descent method is given which is used for face recognition under varying illumination condition. Gradient angle is used as the input feature. The gradient angle is calculated using differential equation which preserves the detail of the image. In [14] under illumination, various features of illuminated faces are extracted and is used for face recognition. [15] presents a method for skin color detection under rapidly changing illumination conditions. Skin colors are modeled under the Bayesian decision framework. Face detection is employed to online sample skin colors and a dynamic thresholding technique is used to update the skin color model. In [16], an unsupervised method for segmenting the illuminated regions and estimating the illumination power spectrum from a single image of a scene lit by multiple light sources, is reported.

In [4], discrete cosine transform (DCT) is employed to compensate illumination variations in the logarithmic domain. Illumination variations mainly lie in the low-frequency band from which an appropriate number of DCT coefficients are truncated to minimize variations under different lighting conditions. Discarding low-frequency DCT coefficients in the logarithm domain is equivalent to compensation of illumination variations which solves illumination problem in an image. The obtained results are in gray scale. In [5], local contrast correction algorithm is explained in which lowlight regions and highlight regions adjusted, based on multi– scales Gaussian are filtering. Linear color restoration method is used to minimize the change in color between the input and the output image that is used for illumination compensation.

In [6], YCbCr color space is used to separate luminance and chrominance. Y component deals with the luminance part. In this paper, fire flame pixels are extracted by comparing the adjacent pixel values. It also separates high temperature fire pixels by computing the statistical parameters of fire image in YCbCr color space using mean and standard deviation. We use this mechanism in luminance range to extract the high luminance area present in an image. This helps in localizing the highly illuminated area. In [7], two pixel-wise algorithms are used to achieve relative color constancy by working under the spectral domain in which each pixel is mapped to the reflectance ratio of objects appeared in the scene and perform illumination correction in spectral domain. A camera calibration technique is used to calculate the characteristics of a camera without the need of a standard reference.

In [8], illumination and reflection components from an image are extracted by the logarithmic transformation. The high-frequency part is enhanced and low frequency component of the image is reduced by smooth filter to smoothen the image. Gamma correction is done on illumination component to achieve illumination compensation following the procedure reported in [2]. In [9], an adaptive weighted total variation model is developed to optimize illumination field. First step is to separates the illumination component from reflectance. Then cluster is formed for illuminated region. At last correction takes place by clustering moment matching method. In [10], Iterative process of noisy pixel detection using noise profiles and replacement of these pixels using the average values of neighbouring noise free pixels is reported. This averaging technique is implemented to normalize the high illumination value present in the image. Correction of uneven illumination has been reported in many applications. Leong et al [11] designed a scheme for illumination correction with shading method. It is based on intrinsic properties of the image under analysis, which are revealed through Gaussian smoothening. It is applied to digital microscopy images with empty fields as a correction filter. Use of matrix-masking to balance non uniform in the images captured through optical micrographs is reported in [12].

2.3 CONCLUSION

In this chapter, literatures related to normalization of non-uniform illumination in digital images were reported. In the next chapter architectural design of proposed normalization of illumination invariancy is presented.

CHAPTER 3

ARCHITECTURAL DESIGN

3.1 ARCHITECTURE

In this section, the architectural diagram for our proposed system is presented. The stages involved in the design of proposed illumination uniforming scheme is in Figure 3.1.

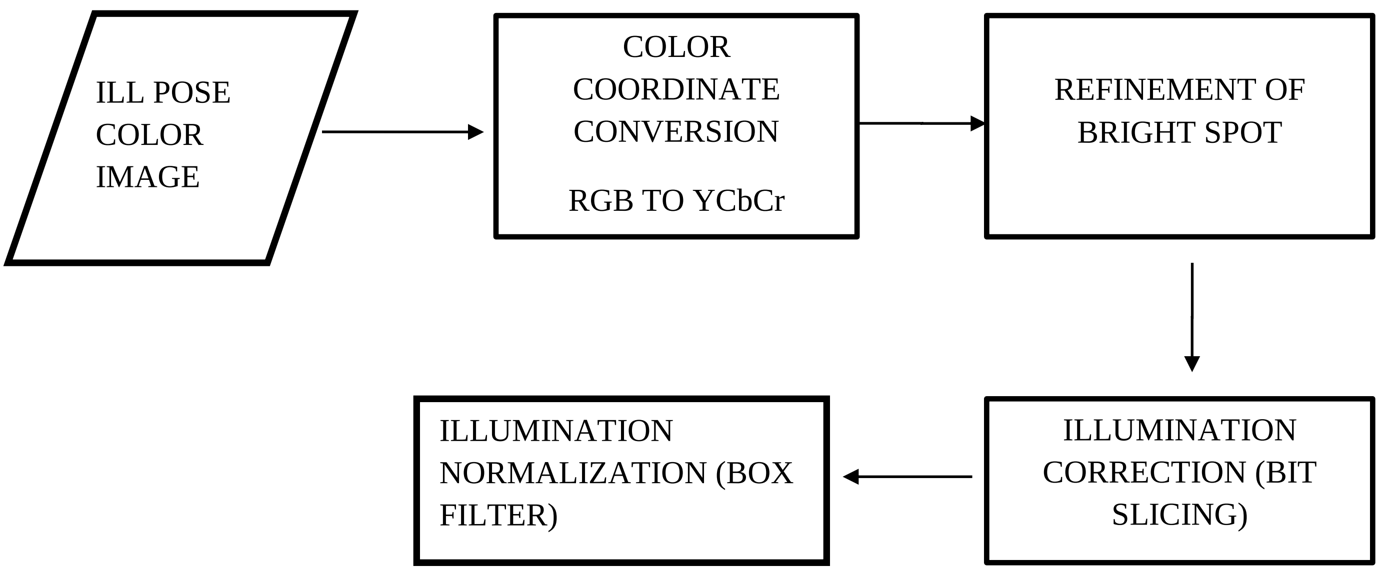

Figure 3.1 Architecture diagram for illumination correction method

The blocks in the architecture diagram represent various stages of the system. Output from each stage is fed as input to the next stage in both the systems. Standard images with illumination effects from the dataset are fed as input.

3.2 ILLUMINATED IMAGE

Illuminated image represents the luminance of an image. A highly illuminated image forms a white tone on the top layer of the image. This reduces the quality of the layer present beneath the Illuminated part of the image. In medical field a small range of data plays a significant role. Thus error due to high illumination may lead to big miscalculations. Our novel approach proposes a method to nullify this error. The various steps involved is explained below. In our method an Illuminated image is fed as the input.

3.3 LUMINANCE CALCULATION

The raw input image is converted into pixel values. The input is in RGB color space. It is difficult to calculate the luminance in this color space. So, the image represented in RGB color space is converted to YCbCr color space. Then, the luminance value is extracted from the pixel value of the image. In YCbCr color model the Y component represents the luminance value whereas the Cb and Cr represent the chrominance value. Thus, the RGB image is converted into YCbCr model and the luminance value is extracted.

3.4 ISOLATION OF MAXIMUM ILLUMINATED AREA

In order to find the maximum Illuminated area, the average of the luminance values is calculated and the values above the calculated average are separated and proceeded for further processing. The values below the average value are discarded because the lower values correspond to the darker regions in the image. This can be further illustrated using histogram equalization. This is repeated until the high illuminated area is obtained.

3.5 AVERAGE FILTER

The output of the previous step results in hard edges around the high illuminated region. This forms an artifact, surrounding the edges. In order to reduce this, average filter is applied for each 3x3 luminance value of the image. This produces smoothed edges around the artifact region and then the illumination is detected so that it can be rectified in the subsequent processes.

3.6 LUMINANCE CORRECTION

The luminance value is corrected using bit slicing operation and box filtering algorithm. Here the luminance value is converted into binary format and the most significant bit is altered or simply reduced to zero. This will reduce the intensity of the light present in that pixel. This mechanism is applied to the entire region where the luminance value is high. Thus the high luminance value is partially reduced and thus luminance is corrected.

3.7 SOFTWARE AND HARDWARE ENVIRONMENT

3.7.1 HARDWARE ENVIRONMENT

Processor : Any processor above 2.5 GHz

RAM : 4 GB and above

Hard Disk : 10 GB

GPU : Integrated GPU or above

Input Device : Standard Keyboard, Mouse

Output Device : VGA Monitor

3.7.2 SOFTWARE ENVIRONMENT

- Net beans :IDE
- Front end : Java

CHAPTER 4

PROPOSED DETECTION AND CORRECTION METHOD

4.1 PRE-PROCESSING

Initially a color image of size (256 x 256) is taken as input. Images with various types of illumination effects form the input for the system. This illumination effect will be reflected as a white color tone on the image. Initially the pixel values are extracted from the image and then the red(r), green (g) and blue (b) color components are segregated from the extracted pixel values. Up to this phase, the image is represented as RGB color model. For the purpose of working on illumination-vary, the RGB image is converted into YCbCr model. Here Y represents the luminance of the image and Cr ana Cb refer to color difference in red and blue respectively. Luminance and chrominance components are separated so the color representation is indifferent to the varying luminous conditions. This Color space YCbCr offers the same advantage as HSV and furthermore is widely used in many video broadcasting standards. YCbCr colour space effectively separates luminance from chrominance compared to other colour spaces.

4.2 LUMINANCE CALCULATION

Luminance is a measure for the amount of light emitted from a surface. After the pre-processing stage, the luma component of the image at every point is calculated. There is a dependency between luminance component value of face detection results.

Dependency between RGB and YCbCrcolor space is described by the equation (4.1)

$$Y = 16 + (65.481R' + 128.553G' + 24.966B') \quad \text{...... (4.1)}$$

where R’, G’, B’ refers to the red, green and blue color component values. After a large study of the constants involved here, it is estimated that the above constants prune to the correct identification of all the components in the model.

The luminance component is thus calculated and it is represented as a grayscale image. The darker regions become black and the illuminated portion alone is exhibited in the image. The darker regions represent the non-illuminated regions and so it is black in color. The illuminated regions are represented as a grayscale image. Thus only the illuminated region gets exposed. This image is used further to identify the most illuminated part in the image.

4.3 REFINING THE MOST ILLUMINATED PORTION

After the luminance extraction from the ill-pose image, the most illuminated region should be identified in order to rectify it. Illumination is very important part in visible light images. The acquired image is modeled as product of reflectivity and illumination as the illumination change, the image pixel intensity will change. For this repeated averages of the luminance values are taken and the area below the average are discarded. Again the average is taken for the remaining luminance values and this process is repeated until the average value gets saturated at one point. As the average of all the luminance points change, the region becomes smaller and smaller. As a result, the refined illumination portion is obtained which means the maximum brightness present in the image. Although there are many algorithms for image enhancement, the existing techniques often produce defective results with respect to the portions of the image with intense or normal illumination, and such techniques also inevitably degrade certain visual artifacts of the image. Using these values the illumination effect is corrected using the subsequent steps.

4.4 BIT-PLANE SLICING

The gray level of each pixel in a digital image is stored as one or more bytes in a computer. For an 8-bit image, 0 is encoded as 00000000 and 255 is encoded as 11111111. Any number between 0 to 255 is encoded as one byte. The bit in the far left side is 0000000000000 referred as the most significant bit (MSB) because a change in that bit would significantly change the value encoded by the byte. Here the value in the most significant bit is altered so that the value of the respected position decreases with respect to the change.

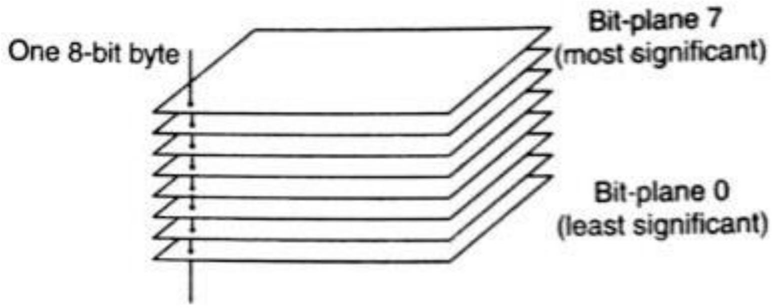

Figure 4.1 bit-plane slicing

4.5 ALGORITHM FOR ILLUMINATION DETECTION

Input: A standard image of size (256 x 256)

Output: Illumination detected image

BEGIN

Step 1: Input a color image from the data set.

Step 2: Extract the pixel values and segregate into r, g, b components

Step 3: Convert the image into YCbCr model

Step 4: calculate the luma component

Step 5: Obtain the luminance image

Step 6: Find the repeated average for the luminance values obtained

Step 7: Refine the bright spot i.e. illuminated area

END

4.6 ILLUMINATION NORMALIZATION

Illumination normalization is done with the help of box filters. Box filtering involves replacing each pixel of an image with the average in a box. Box filtering is basically an average-of-surrounding-pixel kind of image filtering. It is actually a convolution filter which is a commonly used mathematical operation for image filtering. A convolution filter provides a method of multiplying two arrays to produce a third one. In box filtering, image sample and the filter kernel are multiplied to get the filtering result. The filter kernel is like a description of how the filtering is going to happen, it actually defines the type of filtering.

A box blur (also known as a box linear filter) is a spatial domain linear filter in which each pixel in the resulting image has a value equal to the average value of its neighboring pixels in the input image. It is a form of low-pass ("blurring") filter. A 3 by 3 box blur can be written in matrix notation as

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Initially, the image is divided into 3x3 blocks and then convolution is performed with the blur matrix so that the result of this operation yields a normalized image. This is done for the whole image obtained as a result of bit plane slicing. Thus normalization is done and illumination is corrected for the ill-pose image.

4.7 ALGORITHM FOR ILLUMINATION NORMALIZATION

Input: Illumination detected image

Output: Illumination normalized image

BEGIN

Step 1: Input the illuminated detected image from the previous step

Step 2: Divide the image into 3x3 blocks

Step 3: Obtain the box blur operator

Step 4: Perform convolution operation for a single 3x3 block

Step 5: Apply convolution operation for the whole image

Step 6: Extract the normalized image

END

CHAPTER 5

EXPERIMENTS AND RESULTS

The proposed illumination detection and correction scheme was implemented based on various techniques. A number of color images from the standard database decribed under [3] were taken so that illumination is detected and corrected. For this proposed work, we considered many images. Out of all the experimented images, we have shared the experiments and results obtained for a particular sample image is presented in the following sections.

5.1 LUMINANCE CALCULATION

The proposed luminance calculation was done for a set of 100 images from a standard database. The results from this part are subjected to illumination portion identification. One of the input images taken is illustrated in figure 5.1.

Initially, the image of size (256 x 256) is pre-processed .If the image is not by the desired size then it is resized to the appropriate size. In pre-processing stage, the pixel values in an image are initially extracted and then the red, green, blue components are segregated so that the image in RGB color model is converted to YCbCr color model. This process is done because the luminance can be calculated using luma component. Y in YCbCr model thus corresponds to luminance in the image. The luminance is calculated as given in equation (4.1) of section 4.2.The new values are calculated with the constants involved within the formula.

Figure 5.1 Input image(containing non-uniform illumination)

Figure 5.2 Result of identification of luminance component corresponding to the original image shown in figure 5.1

5.2 REFINING THE ILLUMINATED PART IN THE IMAGE

With the luminance image obtained as a result of luminance calculation, the output of luminance calculation phase is set as input to identify the illuminated part. For the illuminated part to be identified, initially the average of maximum number of pixels was taken so that the portions below the average are discarded. Only the bright portions were considered and then the process is repeated further until the illuminated portions are refined to its finest. At one point the average of the illumination values gets saturated and this forms the illuminated portion in the image. This illuminated region is then rectified and then combined with other regions to produce a normalized image. The result of refined illumination spot corresponding to the original image shown in Figure 5.1 is presented in Figure 5.3.

Figure 5.3 Result of refined illumination spot corresponding to the original image shown in figure 5.1

5.3 BIT-PLANE SLICING

The gray level of each pixel in a digital image is stored as one or more bytes in a computer. The illuminated portions are bit sliced so that the illumination is reduced to a particular extend. Initially the image is converted into the binary format and then alteration is done to the most significant bit of the digits. To convert the luminance values from decimal format to binary format in an image, processing is done on each of these bit planes and their results are combined to produce the binary output of the input sample image. This processing includes calculation of inverted image of each of the bit planes produced, done by including a bias value. The most significant bit plane is then taken. These inverted images would be mostly constituted of background and noise pixels. After this, the difference of the grayscale image and the image obtained through bit plane slicing is calculated. Together with the image difference calculated so, it is required to have another image difference which is theoretically the inverse of the image just obtained. Thus bit plane slicing can be applied according to the illuminated region obtained. If the region is bigger, then appropriate results would be obtained. Here, bit plane slicing is applied to both smaller and bigger illumination detected regions in the image. The results are represented as figure 5.4 and figure 5.5.

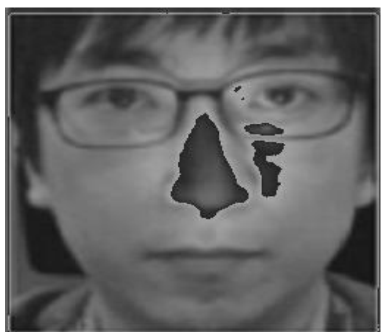

Figure 5.4 Result of Bit plane slicing applied for a smaller region

Figure 5.5 Result of Bit plane slicing applied for a larger region

5.4 ILLUMINATION NORMALIZATION

In this stage, the output of the bit plane slicing is further normalized using box filter. Figure 5.5 is initially divided into 3x3 blocks and each 3x3 block is performed a convolution operation with the box blur matrix. Doing this repeatedly, normalized image is obtained which is shown in figure 5.6.

Figure 5.6 Result of illumination normalization

Thus illumination is detected and corrected using the above mentioned algorithms

CHAPTER 6

CONCLUSION

6.1 CONCLUSIONS

Illumination is one of the main components of an image. Both high and low illumination reduces the quality of the data associated in that region. Our novel methodology overcomes this problem by proving an optimal solution to normalize the illumination value on high illuminated area. In previous methods tone mapping is used to adjust the brightness of the image. In our, approach we obtain the luminance value for extraction of highly illuminated region of an image. Later further processing as discussed in chapter 4 is done on the highly illuminated region to normalize the illumination effect all over the image. The next step is to normalize the high illumination by implemented using box filtering. The bright spot formed due to nature of the lightning condition (which is different in different scenarios) are identified and normalized. This novel method helps researchers to obtain high quality information from images of varying lightning conditions. The normalization method we implemented doesn't result in any loss of data. It only enriches the quality of it by eliminating the high illumination.

6.2 FUTURE WORKS

This work can be further developed by processing all these steps in transform domain rather than using the spatial domain. As transform domain has highly uncorrelated value, the processing of the images can be done easier and any change of a single value doesn't affect any other values.

REFERENCES

[1] M. Beniak and J. Pavlovicova, “Luminance component in skin color detection,” 2011 18th International Conference on Systems, Signals and Image Processing, Sarajevo, pp. 1-4, 2011.

[2] Y. Shi, J. Yang and R. Wu, "Reducing Illumination Based on Nonlinear Gamma Correction," 2007 IEEE International Conference on Image Processing, San Antonio, TX, pp. I - 529-I - 532, 2007.

[3] W. Kim, S. Suh, W. Hwang and J. Han, “SVD Face: Illumination-Invariant Face Representation,” in IEEE Signal Processing Letters, vol. 21, no. 11, pp. 1336-1340, Nov. 2014.

[4] W. Chen, MengJooEr and Shiqian Wu, “Illumination compensation and normalization for robust face recognition using discrete cosine transform logarithm domain,” in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 36, no. 2, pp. 458-466, April 2006.

[5] S. Lee, B. Kwak, H. Han, W. Choi and B. Kang, “Dynamic range compression algorithm for mobile display devices using average luminance values,” Proceedings of the 2009 12th International Symposium on Integrated Circuits, Singapore, pp. 340-343, 2009.

[6] C. E. Premal and S. S. Vinsley, “Image processing based forest fire detection using YCbCr colour model,” 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014], pp. 1229-1237, 2014.

[7] Y. Zhao, C. Elliott, H. Zhou and K. Rafferty, “Pixel-wise Illumination Correction Algorithms for Relative Color Constancy Under the Spectral Domain,” 2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 638-643, 2018.

[8] X. Huang, Z. Jiang, L. Lu, C. Tan and J. Jiao, “The study of illumination compensation correction algorithm,” 2011 International Conference on Electronics, Communications and Control (ICECC), 2011, pp. 2967-2970.

[9] S. Luo, H. Li and H. Shen, “Shadow removal based on clustering correction of illumination field for urban aerial remote sensing images,”2017 IEEE International Conference on Image Processing (ICIP), Beijing, pp. 485-489, 2017.

[10] A. Konieczka, J. Balcerek and A. Dąbrowski, “Method of adaptive pixel averaging for impulse noise reduction in digital images,” 2018 Baltic URSI

Symposium (URSI), Poznan, pp. 221-224, 2018.

[11] FJWN. Leong, MBrady and JOD McGee, “Correction of uneven illumination in digital microscopy images”, Journal of clinical pathology, vol.56, No.8, PP 619 – 621,2003.

[12] P.Nordenfelt, J.M. Cooper and A. Hochstetter, “Matrix – masking to balance nonuniform illumination in microscopy”, optics express, vol. 26, No.13, PP 17279 – 17288, 2018.

[13] T. Zhang, Y. Y. Tang, B. Fang, Z. Shang, and X. Liu, “Face recognition under varying illumination using gradientfaces,” IEEE Trans. Image Process., vol. 18, no. 11, pp. 2599–2606, Nov. 2009.

[14] P. Belhumeur, A. Georghiades, and D. Kriegman, “From few to many:Illumination cone models for face recognition under variable lightingand pose,” IEEE Trans. Patt. Anal. Mach. Intell., vol. 23, no. 6, pp.643–660, Jun. 2001.

[15] L. Liu, N. Sang, S. Yang and R. Huang, “Real-time skin color detection Under rapidly changing illumination conditions,” in IEEE Transactions on Consumer Electronics, vol. 57, no. 3, pp. 1295-1302, August 2011.

[16] L. Gu, C. P. Huynh and A. Robles-Kelly, "Segmentation and Estimation of Spatially Varying Illumination," in IEEE Transactions on Image Processing, vol. 23, no. 8, pp. 3478-3489, Aug. 2014.