

## Article

# A Regional Brightness Control Method for a Beam Projector to Avoid Human Glare

Hyeong-Gi Jeon<sup>1</sup> and Kyoung-Hee Lee<sup>2,\*</sup> <sup>1</sup> Smart ICT Convergence HRD Center, Pai-Chai University, Daejeon 35345, Republic of Korea; jeonhk@pcu.ac.kr<sup>2</sup> Department of Software Engineering, Pai-Chai University, Daejeon 35345, Republic of Korea

\* Correspondence: leekhe@pcu.ac.kr; Tel.: +82-042-520-5341

**Abstract:** In this study, we proposed a system to reduce the speaker's suffering from the strong light of a beam projector by applying regional brightness control over the screen. Since the original image and the projected one on the screen are quite different in area, brightness, and color, the proposed system first transforms them so that they have the same area and similar color tone. Then, to accurately determine the difference between those images, we have introduced a SSIM map, which is a perception-based method of measuring image similarity. Accordingly, an image segmentation model is used to determine the speaker's silhouette from the SSIM map. We applied a couple of well-trained segmentation models, such as Selfie and DeepLab-v3, provided with MediaPipe. The experimental results showed the operability of the proposed system and that it determines most of a lecturer's body area on the screen. To closely evaluate the system's effectiveness, we have measured error rates consisting of false-positive and false-negative errors in the confusion matrix. With the measured results, the error rates appeared so insignificant and stable that the proposed system provides a practical effect for the speakers, especially in the case of applying DeepLab-v3. With the results, it is implied that an accurate segmentation model can considerably elevate the effectiveness of the system.

**Keywords:** multimedia system; beam projector; brightness control; SSIM; segmentation



**Citation:** Jeon, H.-G.; Lee, K.-H. A Regional Brightness Control Method for a Beam Projector to Avoid Human Glare. *Appl. Sci.* **2024**, *14*, 1335. <https://doi.org/10.3390/app14041335>

Academic Editor: Luis Javier Garcia Villalba

Received: 21 December 2023

Revised: 2 February 2024

Accepted: 4 February 2024

Published: 6 February 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

One of the old beam projectors developed in the 1990s, the “Eidophor”, could project an analog video signal into a space the size of a movie theater. At the time, its effectiveness as a video output device for educational purposes was tested, and positive results were reported [1]. Thereafter, a beam projector could have been considered an effective device for implementing a large screen at a low cost.

In recent years, beam projectors have been improving in many ways, such as in their brightness, resolution, miniaturization, and lightweight designs, thanks to advances in hardware. These improvements have led to their widespread use in education, homes, businesses, and more. Advancements in light sources are especially encouraging from the audience's perspective, as they allow for sharper images to be projected onto the screen. However, a speaker facing the beam projector may suffer from more glare due to much stronger light [2]. The intense glare can make the speaker's communication with the audience uncomfortable and even cause the speaker's vision loss from prolonged exposure. Nevertheless, the ability to present audiovisual material on a large screen for a large audience is an advantage of beam projectors, which cannot be easily achieved with other display devices. Therefore, speakers are willing to tolerate this inconvenience in many situations.

If a beam projector could selectively adjust the area of light projection, it would prevent intense light from troubling a speaker's eyes, providing a better presentation experience while maintaining the benefits of the projector. Moreover, such a system is needed not

only by the instructors but also by the students who use projectors for their learning activities [3]. Some related research shows that beam projectors can be used to design systems that can interact with learners [4,5]. Lin et al. proposed the basic workings of a beam projection system while considering the brightness control to protect a speaker from strong light [5]. Their research attempted to recognize faces through a camera and cover the image, illuminating the face with a black circle. To perform so, they have introduced a facial recognition algorithm based on skin color extraction to find a speaker's area over the screen.

Nowadays, for the purpose of such human detection, a machine learning-based face recognition model [6,7] can be a good solution. However, there are some considerations that make it difficult to directly apply typical face recognition models to the proposed system. First, the light from the beam projector is projected over the face of a speaker. This may make the recognition results quite inaccurate. Second, if an original image fed to the beam projector contains a human face, the recognition model may also detect it as a speaker. Third, if the speaker's facial area has been completely darkened due to brightness control, the recognition model will not be able to detect the face in the next stage.

This study aims to continuously recognize human figures to find adequate areas to adjust the luminance while strong light interferes with the recognition process. For this purpose, we focus on the similarity of the images, wherein we designed a system that compares an input image of the beam projector with a captured image of the screen to find the different regions. The segmentation of the human body is performed within these different regions. In Section 3, we describe the proposed system's structure and each module's role in detail, including finding image differences, human body segmentation, and adjusting the luminance in specific areas. Section 4 presents implementation details and results by discussing some behaviors and experimental measurements.

## 2. Related Research

### 2.1. Structural Similarity Index Measure

Image quality assessment (IQA) is a research topic that studies the degree of distortion in an image due to its compression, movement, rotation, etc. Methods to evaluate image quality distortion are generally classified as Full Reference (FR), No Reference (NR), and Reduced Reference (RR) methods, depending on how much of the original image can be referred to during the evaluation process [8]. Our study can apply the FR method since an original presentation image and the captured image from a camera are both available. If the captured image has relatively low quality compared to the original one, it may be caused by an obstacle appearing between the screen and the camera.

In traditional FR methods, Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) are popular techniques that can evaluate the difference between an original image (or video) and a distorted one with simple calculations. Unfortunately, they are known to not be so useful when the distortion effects overlap or multi-distorted images are given for evaluation [9–11].

To address these limitations, Chandler and Hemami (2007) argued that it is necessary to combine techniques based on the human visual system, such as luminance, contrast, and texture, to evaluate the image quality efficiently [12]. There has been some research on IQA models motivated by the human visual system. The Structural Similarity Index Measure (SSIM) is a representative one of those models, which is based on the hypothesis that the human visual system is mainly affected by the structural information of images [8]. SSIM measures the similarity of two images by extracting luminance, contrast, and structural features from those images. While traditional IQA models, including MSE and PSNR, produce numerical evaluation results that are not so understandable by humans, SSIM has an advantage in that it enables the analysis of the differences in images from more understandable results [13].

Though the SSIM measured the similarity of images globally [8], there was an argument that the similarity of the images could be used as an indicator to find non-similar regions of an image when applied regionally. A. Aki et al. showed that it was possible to measure dissimilarity and observe changes in the ground using satellite images of the same area at different times as an example [14]. In their approach, though the exact formula to calculate SSIM was not applied, they introduced an idea to measure the similarity of images regionally. This suggests SSIM can also be regionally used to find the more probable area for human detection in an image by excluding the other areas with higher similarity to the original one.

In summary, a review of the literature on SSIM suggests the following implications for this research: First, if one specifies a window and measures the SSIM at each point [15], the SSIM can detect changes in different parts of the video. If there is a significant change between what the projector is trying to show and what one sees, it can be assumed that something is distorting the image. Second, two compared images must represent the same area to obtain an accurate SSIM. The screen will have a defined projection area due to a keystone or lens shift of the projector. When a camera observes the screen, the projection area in a captured image would appear as a rectangle (e.g., a trapezoid) [16]. Therefore, the geometric transformation process would be necessary to match the image with the original one. Finally, color correction is required before the SSIM is calculated. Suppose the projected light reflects discoloration due to the characteristics of the light source, lens, or sensor. Thus, the luminance, which is a fundamental factor in calculating the SSIM, will be influenced by those characteristics.

## 2.2. Semantic Segmentation

Image segmentation is generally carried out to obtain information regarding where the objects are located in an image, what those objects look like, and which pixels correspond to which objects [17]. If a picture is segmented and each pixel in the image is labeled according to the segmentation, pixels with the same label would share the specific properties. Image segmentation tasks are divided into semantic segmentation and instance segmentation based on their purposes. If the purpose is to label the image's pixels according to which object they are included in, then only the semantic segmentation task is enough to achieve the goal. Instance segmentation is necessary when each pixel should be categorized into individual instances in the image. Since segmentation is typically expensive and not mandatory for our study, we focused on segmenting the human body rather than on segmenting each person [18].

Modern segmentation models are based on convolutional neural networks (CNNs), which are similar to object classification models [19]. However, the structure of the segmentation model requires more consideration. According to the explanation of Long et al. (2014), as a fully convolutional network goes through convolutional and pooling layers, the resolution becomes progressively lower, and the image details are lost [20]. Semantic segmentation models usually take the form of downsampling and upsampling to compensate for the shortcomings of fully convolutional networks. The module executing the downsampling is called the encoder, and the module executing the upsampling is called the decoder.

Figure 1 shows an example model configured to utilize MobileNet V3 for image segmentation [21]. The first half was presented with MobileNet V3 as the backbone. The second half was configured with R-ASPP Lite as the split head. MobileNet V3 performs the downsampling role, and R-ASPP Lite performs the upsampling role.

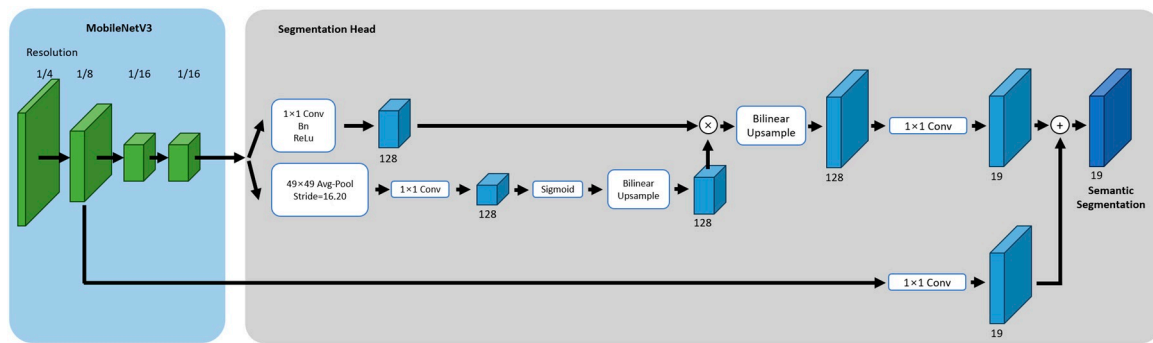


Figure 1. Segmentation architecture with MobileNet V3 [21].

Google offers some segmentation models with a similar structure as solutions in MediaPipe, which include SelfieSegmenter and DeepLab-v3 [22]. SelfieSegmenter is based on MobileNet V3 and is divided into square and landscape versions depending on the input shape. Another well-known image segmentation model, DeepLab-v3, has a similar structure to MobileNet V3. It is reported that DeepLab-v3 has a similar performance to other models by employing ‘atrous convolution’ while performing the upsampling [23]. The atrous convolution method is to extend the window size without increasing the number of weights by setting some values in the convolution kernel to zero.

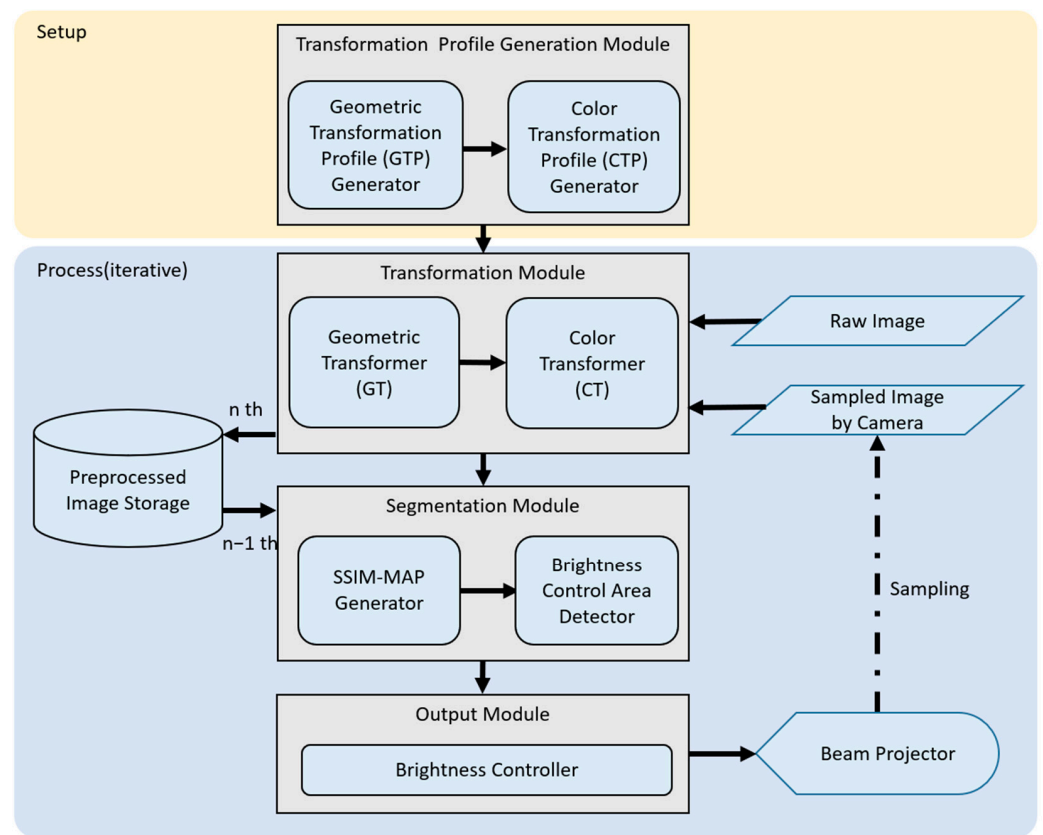
Table 1 summarizes the characteristics and performance of each segmentation model in MediaPipe. Performance in the table was measured using the CPU and GPU of Google’s Pixel 6. From those models, our study utilized SelfieSegmenter and DeepLab-V3 with a square input size. Both models are semantic segmentation models and have similar input sizes. However, the precision of the quantization type is different, and the CPU and GPU latencies are higher in DeepLab-V3. The main reason for this performance difference is believed to be due to differences in the precision of the quantization types.

Table 1. Benchmarks of MediaPipe segmentation models [22].

Model Name	Input Shape	Quantization Type	CPU Latency	GPU Latency
SelfieSegmenter (square)	$256 \times 256$	Float 16	33.46 ms	35.15 ms
SelfieSegmenter (landscape)	$144 \times 256$	Float 16	34.19 ms	33.55 ms
HairSegmenter	$512 \times 512$	None (float 32)	57.90 ms	52.14 ms
SelfieMulticlass	$256 \times 256$	None (float 32)	217.76 ms	71.24 ms
DeepLab-V3	$257 \times 257$	None (float 32)	123.93 ms	103.30 ms

### 3. Proposed System

The structure of the proposed system consists of four modules: the Transformation Profile Generation Module, the Transform Module, the Segmentation Module, and the Output Module. Figure 2 is a schematic representation of the overall processes and behaviors that occur for the system to work. The Transformation Profile Generation Module concerns image calibration according to the environment in which the system is installed. Thus, this module involves only a one-time process when the installation environment is determined. The other three modules are repeated until the system stops working.

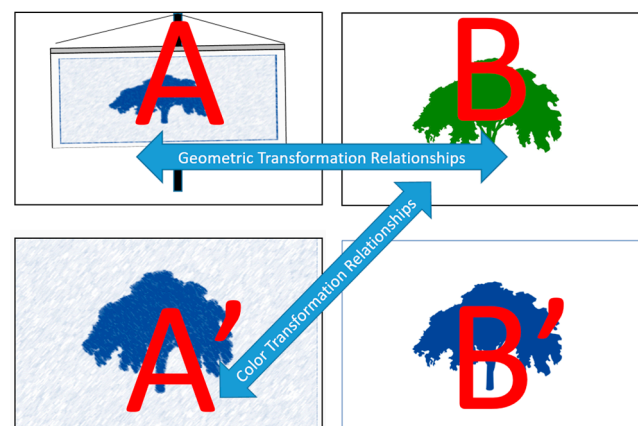


**Figure 2.** Overall functional architecture of the proposed system.

### 3.1. Transformation Profile Generation Module

In the proposed system, the transformation profile refers to a geometric transformation model and a color relationship transformation model to compare the similarity of the images with considerations of such two-type transformation relationships caused by the screen's position and the characteristics of a camera observing the screen and the beam projector.

Figure 3 shows the transformation relationships between the images. In Figure 3, image *A* is sampled by the camera, and image *B* is the original image. Image *A* is transformed into image *A'* by tearing off a part of image *A* and adjusting its rotation angle. To perform this, the geometric transformation relationship between image *A* and image *B* is referred to. This rule for geometric transformation from image *A* to image *A'* forms the Geometric Transformation Profile (GTP).



**Figure 3.** Relationships between images for conversion.

Image  $B$  can be transformed to image  $B'$  by referring to the color transformation relationship between image  $B$  and image  $A'$ . Accordingly, this rule for the color transformation relationship from image  $B$  to image  $B'$  forms the Color Transformation Profile (CTP).

### 3.1.1. GTP Generator

The geometric transformation matches the area of the screen observed by the camera to the area of the original image to be projected. The screen view that the camera is watching is largely distorted by the camera's angle of view and its position relative to the beam projector. To compensate for this, we applied a perspective shift operation. The features of the images for perspective translation and a transformation matrix to make those two images have similar regions should be found. Equation (1) is a matrix that converts a feature point  $(x_1, y_1)$  on the camera screen to a corresponding feature point  $(x_2, y_2)$  in the original image [24].

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & b_0 \\ a_{10} & a_{11} & b_1 \\ a_{20} & a_{21} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \quad (1)$$

This step needs to find the eight unknowns in the matrix by finding at least four points that correspond between images  $A$  and  $B$ . To perform this, various algorithms can be used for feature detection, such as corner detection, SIFT, and HOG features [25–27]. If these features are unambiguously extracted, they can be used as a reference point for calculating the transformation matrix.

### 3.1.2. CTP Generator

If image  $A'$  in Figure 3 has been obtained by geometric transformation, then a color transformation process is performed to get image  $B'$  by correcting the color distortion that occurred during the projection of image  $B$  and capturing it with a camera. The color transformation model can be obtained by machine learning. It uses the coordinates and color values of each pixel of image  $B$  as the input data and then produces the outputs, which are the color values, to construct image  $B'$ . The required model can be expressed as Equation (2).

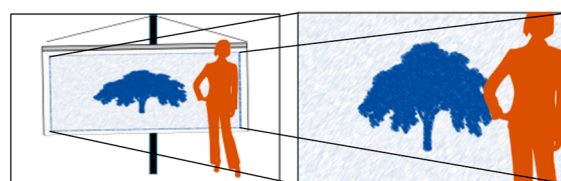
$$f(\text{pos } X, \text{pos } Y, R, G, B) = R', G', B' \quad (2)$$

Equation (2) converts the colors  $R$ ,  $G$ , and  $B$  of the coordinates  $(x, y)$  of the original image (i.e., image  $B$ ) to the intentionally distorted colors  $R'$ ,  $G'$  and  $B'$  of the same coordinates of the output image (i.e., image  $B'$ ). It is not easy to get the perfect color-matching results from this transformation. However, it can reduce the considerable color differences between those two images.

## 3.2. Transformation Module

The two transformation profiles generated by the Transformation Profile Generation Module, called GTP and CTP, are applied to the two input channels of the Transformation Module as follows:

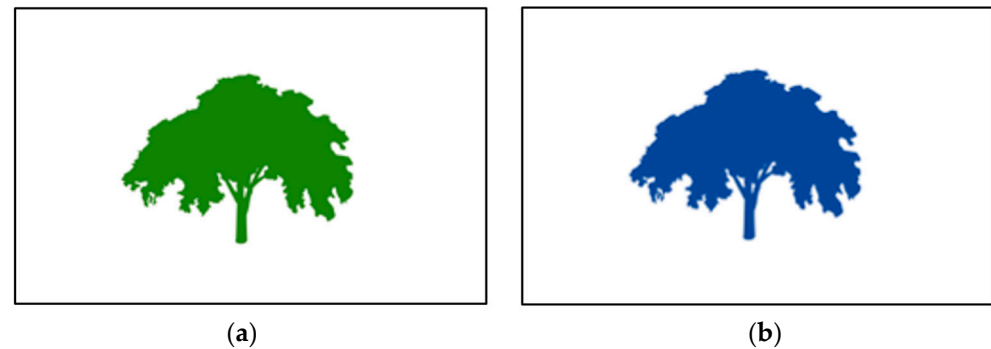
First, the Geometric Transformer converts the image sampled by the camera into a region of interest (ROI) with the GTP. The effect of the geometric transformation is to cut out unnecessary areas and leave only those matching the original image. To perform this, the camera should be watching the speaker and the projection screen at the same time. Figure 4 shows an example of extracting a ROI from a captured image.



**Figure 4.** Transforming screen footage with GTP.



Second, the CTP is used by the Color Transformer to convert the original image. This color transformation reduces the color difference between the original image and the output of the Geometric Transformer (i.e., the ROI image). As shown in an example of Figure 5, the green color in the original image has been converted to the blue color to increase the color similarity between the two images compared by the Segmentation Module.

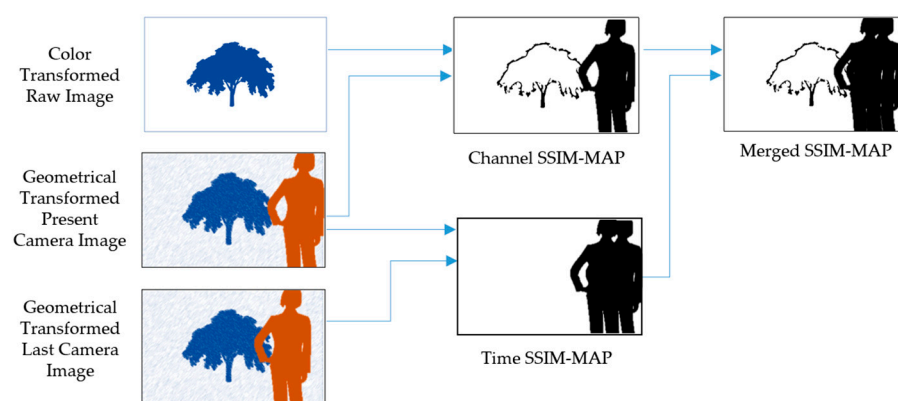


**Figure 5.** Original image (a) and color-transformed image (b) with CTP.

### 3.3. Segmentation Module

The Segmentation Module process is performed in two steps: SSIM-MAP generation and human segmentation. The SSIM-MAP represents the similarity for each point having the same coordinates in two images, with higher similarity displayed by a bright color and lower similarity displayed by a dark color.

The proposed system measures the SSIM values in two directions. The first direction is to compare the geometrically transformed image of Figure 4 with the color-transformed image of Figure 5b. This measurement produces an output defined as the channel SSIM-MAP, as shown in Figure 6. The second direction is to examine the similarity between the past and the current images, which are taken by the camera with a short interval, to catch the differences caused by the speaker's movement. We have defined the output of this measurement as the time SSIM-MAP, as illustrated in Figure 6. Finally, those two SSIM-MAPs are merged into a single SSIM-MAP containing all the areas that should be considered as the dimming areas controlled by the Output Module.

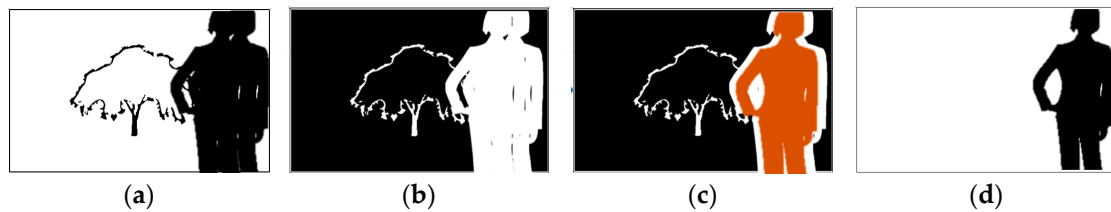


**Figure 6.** SSIM-MAP generation process.

The Channel SSIM-MAP enables the detection of the appearance of an unexpected object. If an object that does not exist in the original image appears in the camera image, the object will be displayed as an area with low SSIM values. This means that if an obstacle in front of the screen (such as a person) impedes light travel, the channel SSIM-MAP will show that obstacle. SSIM-MAP is used to detect moving objects. Since the image on the wallpaper is stationary, areas of low similarity could be where the speaker's movement has occurred when comparing the previous and current camera images. Finally, those channel

SSIM-MAP and time SSIM-MAP images are merged into a single SSIM-MAP so that it can be used to determine the areas for regional brightness control.

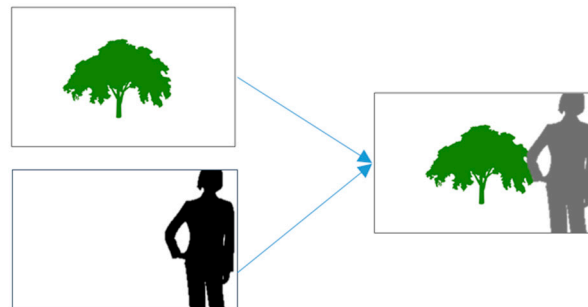
As shown in Figure 7, to extract a human body area for the brightness control, the merged SSIM-MAP image (Figure 7a) is first inverted into grayscale to be used as a mask to filter non-human areas in the camera-captured image. Since the mask (Figure 7b) indicates a transparent area when it overlaps with the captured image, only the human body area keeps the original color while the other areas are covered with the black color (Figure 7c). Finally, human body segmentation is performed on the composite image to remove all non-human areas (Figure 7d).



**Figure 7.** Brightness control process: (a) binarized merged SSIM-MAP; (b) inverted SSIM-MAP as a mask; (c) masked camera image; (d) segmented human body.

### 3.4. Output Module

The Output Module creates a lower luminance setting for the segmented area. It generates a composite image of the next frame of video and the refined segmented region. Accordingly, the composite image has a dark color in the segmented region, as shown in Figure 8.



**Figure 8.** Generating the brightness control area.

If the luminance of the refined segmentation region is perfectly zero, the refined segmentation process will fail in the next iteration of the loop since the speaker may disappear into darkness. Therefore, the luminance of the segmentation region should be set to a low level that the Segmentation Module can recognize.

## 4. Implementation Details and Results

### 4.1. Hardware Configuration and Settings

The hardware of the proposed system consists of a webcam camera device, a beam projector, and a desktop PC with a CPU based on the AMD64 architecture. Table 2 shows the detailed specifications of the system hardware. The GPU was utilized for machine learning in the Transformation Profile Generation Module, while image processing was mainly conducted with the CPU.



**Table 2.** System hardware specifications.

Component	Specifications
CPU	13th Gen Intel(R) Core(TM) i7-13700K
GPU	Nvidia RTX4090
RAM	DDR5 32 GB
Webcam	C930 e 1080 p, 30 fps
Beam Projector	Wanbo T2 Max 1080 p, LED Light source

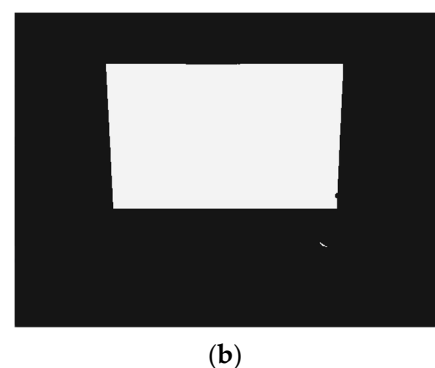
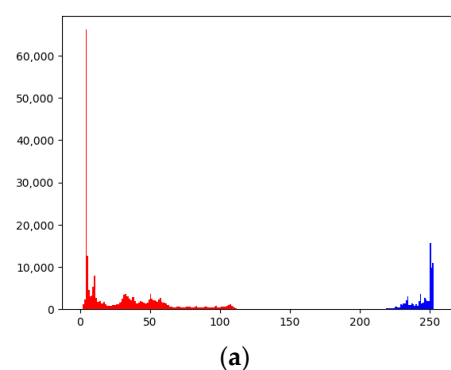
#### 4.2. Transformation Profile Generation Module Implementation

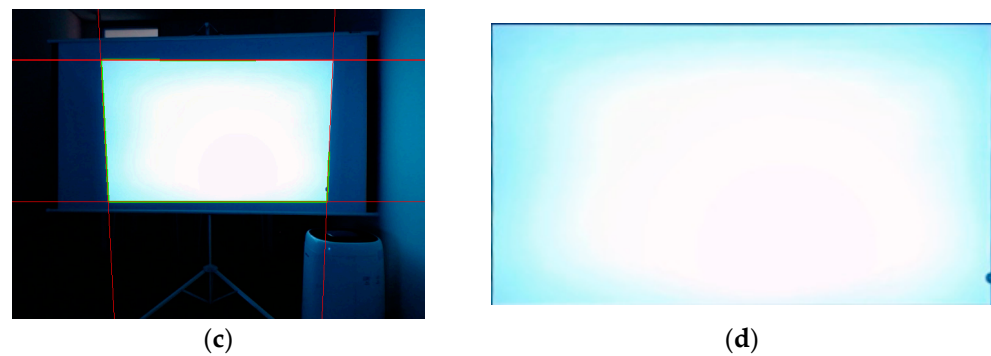
##### 4.2.1. GTP Generation

Geometric Transformation Profile (GTP) is a transformation matrix to extract the screen area from the image captured by a camera. To get this matrix, the screen with the highest brightness is first captured and inverted into a grayscale. Then, the projection and non-projection areas are separated into 0 and 1 values by the appropriate threshold. The process for determining the threshold is as follows:

1. Performs K-means clustering on the brightness of the pixels to find two clusters: black and white [28,29].
2. Finds the maximum and minimum brightness values of the two clusters.
3. Sorts the values and taking the average of the second- and third-ranked values as the threshold.

The threshold set by the above procedure will be in the range where brightness occurs least frequently in the original image. Figure 9a is a histogram of brightness values, and if you follow the procedure, the threshold can be determined automatically. Figure 9b is a result of binarization by automated thresholding. As a result of this binarization, all pixels in the screen area form the shape of a white square (generally, a trapezoid). To find four vertices of the square, the Hough transform [30,31] is applied to extract the line segments of the square and calculate their intersection points. Figure 9c shows the edges of the screen area detected by the Hough transform. Finally, the transformation matrix is derived by mapping those vertices into a rectangular area of the size of  $320 \times 180$ . Figure 9d shows the automated result of the above process.

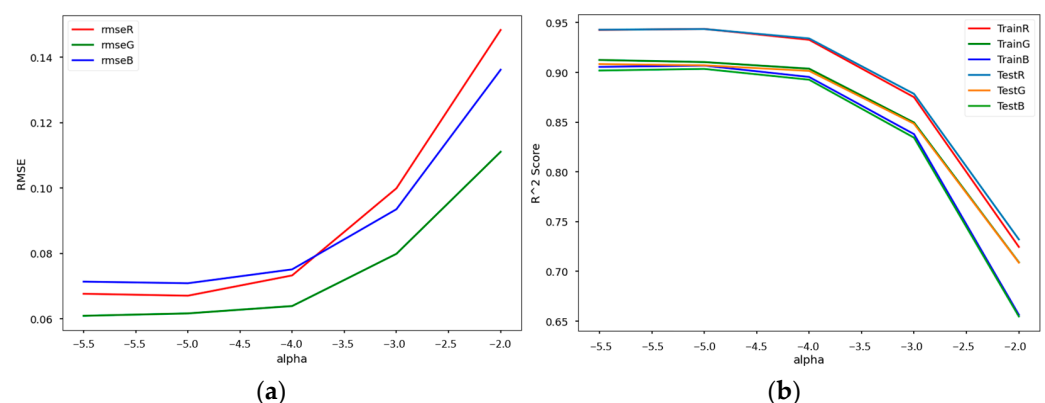
**Figure 9.** Cont.



**Figure 9.** Process of GTP Generation: (a) histogram of the K-means clustering result, red bars are low brightness pixels, blue bars are high brightness pixels; (b) quantized image by threshold. (c) camera image with detected intersections of straight lines; (d) transformed image using a GTP calculation based on intersections.

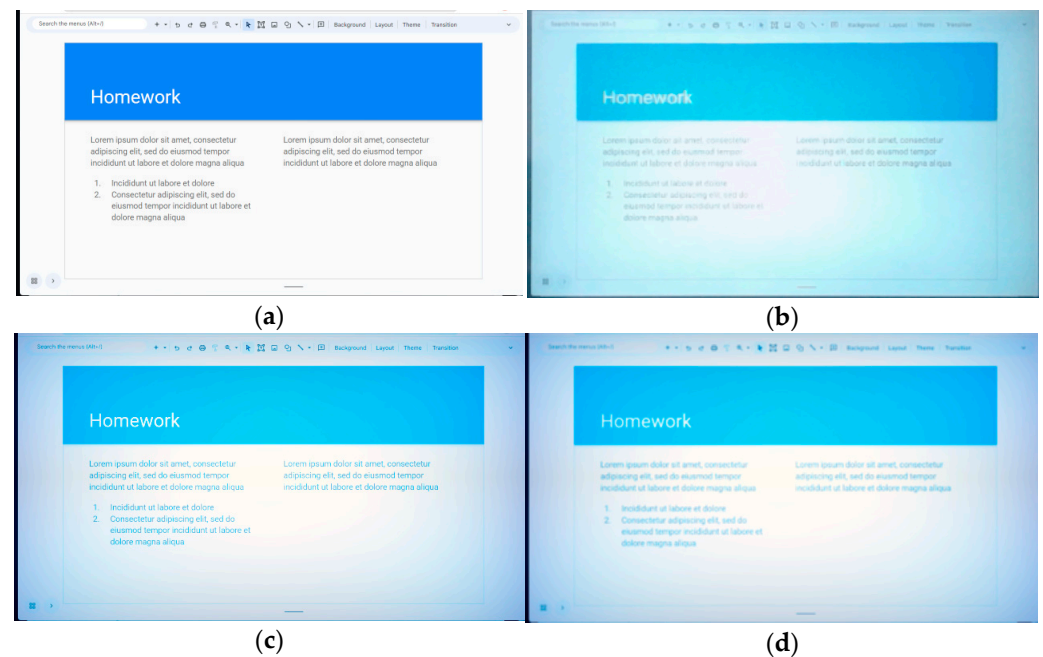
#### 4.2.2. CTP Generation

The Color Transformation Profile (CTP) is used to reduce color differences between the original image and the geometrically transformed captured image. These differences are mainly due to the color distortion during image projection and capture with the camera. In the captured image, the edges of the screen are generally darker than the center and may have some glares. In our study, to determine the color distortion, a polynomial linear regression was performed, and LASSO regularization was applied to eliminate the computation with little relevance [32,33]. For the training data set, 500 images have been generated with random colors. Each pixel has been sampled from random coordinates in those images. Figure 10 shows the polynomial linear regression training results with LASSO regularization. In LASSO regularization, the alpha value represents the strength of the regularization. Hence, a low strength indicates that the regression model will include many coefficients that are less effective, increasing the amount of computation in the system. Based on the RMSE and R-square calculated by training, we used the correlation coefficient of alpha value  $-4.0$  for this study.



**Figure 10.** Result of the polynomial linear regression with LASSO regularization: (a) RMSE; (b) R-square score by alpha value.

The correlation coefficients inferred from the polynomial linear regression are used as a filter to correct the color distortion based on the location. To make similar effects with the glare on the edge, the Gaussian blur method is applied [34]. Figure 11 shows the raw image, camera image, and affected image by filter and blur.



**Figure 11.** Color transformation result images by effects: (a) raw image; (b) geometrically transformed result image from camera image; (c) color-transformed result image from the raw image; (d) blurred color-transformed result image from the raw image.

Table 3 summarizes the means and standard deviations of the SSIM values improved by those color filtering and blurring effects. As shown in the table, the blur effect can improve the similarity of two images (the original and captured ones) more than color filtering.

**Table 3.** SSIM measurement results from image effect.

Effects	Mean	Std
Original	0.6575	0.1271
Filtered	0.6899	0.1251
Blurred	0.7508	0.0925
Filtered + Blurred	0.7837	0.0911

### 4.3. Segmentation Module Implementation

#### 4.3.1. SSIM-MAP Generator

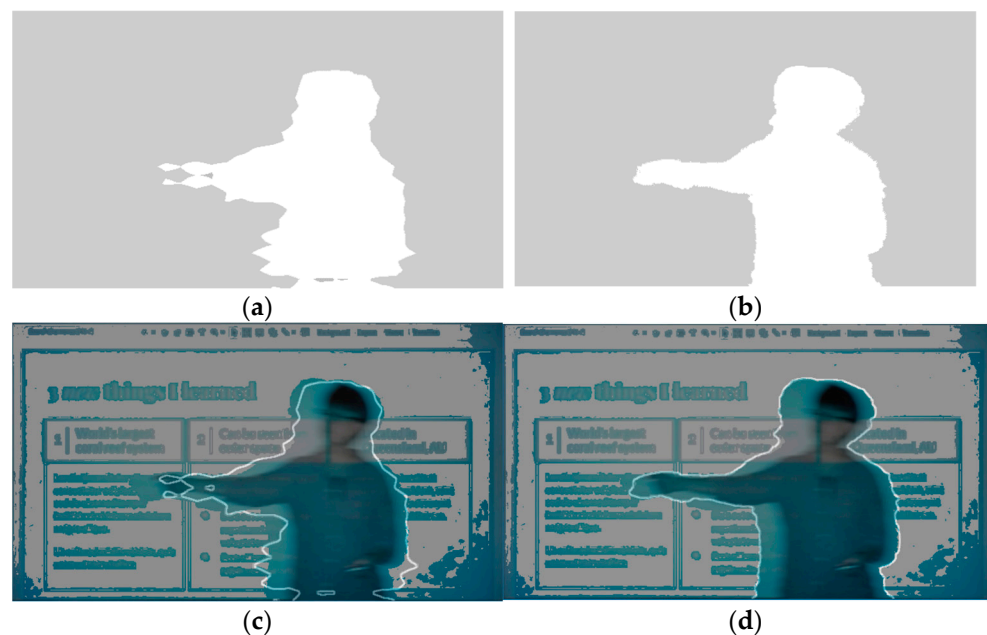
After applying transformations with the GTP and the CTP, the SSIM-MAPs are generated to detect whether there are any objects in front of the screen that block the light coming from the projector. In this study, the SSIM-MAP is made by binary-quantizing with a  $3 \times 3$  window and the appropriate threshold. In this study, we applied a level of 0.5 as the threshold, which corresponds to the third quartile of the range of SSIM values from  $-1.0$  to  $1.0$ . Since the size of the window is the number of surrounding pixels being used to measure a pixel's SSIM value, it affects the sensitivity of the measurements. The smaller window size would derive the more sensitive SSIM value. However, it should be at least  $3 \times 3$  [15]. If the window is too large, there may not be any difference in SSIM values based on the location of the image [15]. As mentioned before, the proposed system generates two different SSIM-MAPs, called the Channel SSIM-MAP and the Time SSIM-MAP, and then merges them into a single SSIM-MAP to be used for human area segmentation by the Brightness Control Area Refiner.

#### 4.3.2. Brightness Control Area Detector

In this study, we have employed two well-known segmentation models, DeepLab-v3 and Selfie [35], to remove non-human areas from the masked camera image. Both segmenta-

tion models are included in MediaPipe 0.10.0, which is an open-source AI library developed by Google. Since those models are provided with sufficient training, no additional training has been performed for our study. To check the results of applying those two segmentation models, we have detected a brightness control area with each model for a single image in which a speaker is giving a lecture.

Figure 12 shows the example results of brightness control area detection performed by the proposed system. Figure 12a,b represent segmentation results using the DeepLab-v3 and Selfie segmentation models, respectively. These results show that the Selfie model has found a smoother segmentation line than that of the DeepLab-v3 model. Figure 12c,d show the segmentation line of each model for the area masked by SSIM-MAP. They illustrate that most non-human areas have been successfully removed to detect the brightness control area.



**Figure 12.** Brightness control area detection results: (a) segmentation result of DeepLab-v3; (b) segmentation result of Selfie; (c) DeepLab-v3 segmentation outline on masked image; (d) Selfie segmentation outline on masked image.

#### 4.4. Performance Evaluation

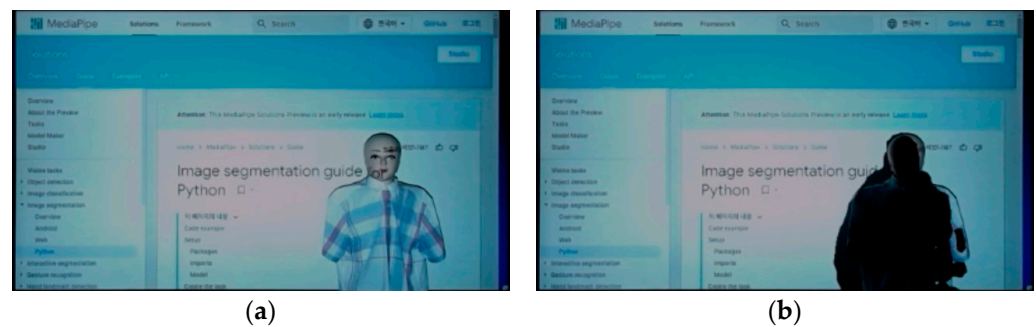
If the body area has been segmented and the bright control is applied, there will be less light in the human body. A reduction in the projected light will lead to the camera obtaining a darker area in the picture, which would also affect the next segmentation due to the difference in the incoming image. We sequentially adjusted the dimming brightness level in 50 increments to measure this effect.

##### 4.4.1. Evaluation Methods

In this study, the dimming area is determined by two segmentation models and the regional brightness control of the projector. In this section, we explain how to evaluate the proposed system's effectiveness and present some measured results from our implementation.

To evaluate the system's effectiveness, a mannequin that could be recognized as a person was placed in front of the beam projector's screen. The measurements have been performed for two images resulting from two segmentation models: Selfie and DeepLab-v3. Each frame has been processed based on a size of  $640 \times 360$ . In our experimental environments (shown in Table 2), the total execution time with the Selfie segmentation model appeared to be about 200 ms to process one frame. With DeepLab-v3, the total execution time for one frame appeared to be about 220 ms. In the installation process, our

implementation took 135 ms to generate the SSIM-MAP. Figure 13 shows a webcam image of the screen with a mannequin and the proposed system's output with DeepLab-v3.



**Figure 13.** Result of regional brightness control: (a) projection of an original image with a mannequin; (b) projection image including a brightness control area.

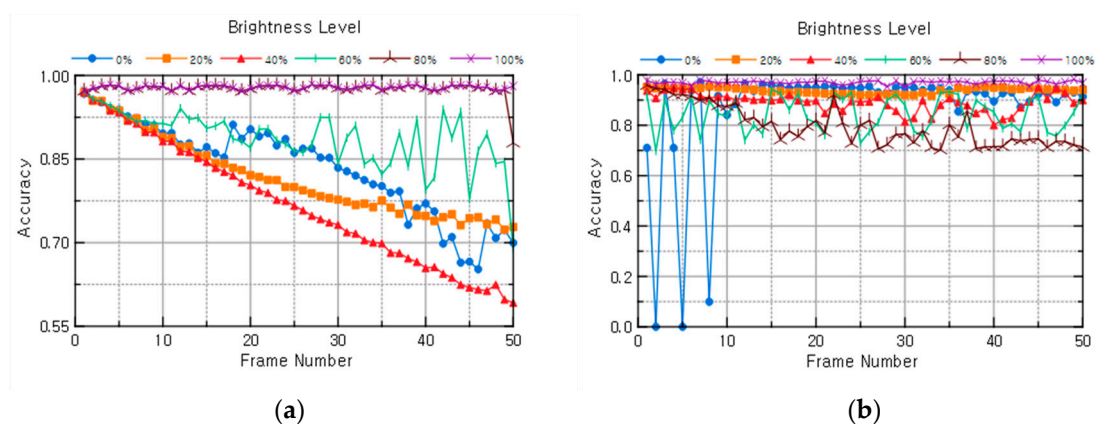
Segmentation is the task of labeling which classification each pixel belongs to. Usually, the performance evaluation of a classification model is carried out based on the confusion matrix. With the confusion matrix, the error rate of a system can be calculated by the following Equation (3) [36].

$$\text{Error Rate} = \frac{FP + FN}{TP + TN + FP + FN} \quad (3)$$

In the formula,  $TP$ ,  $TN$ ,  $FP$ , and  $FN$  represent true-positive, true-negative, false-positive, and false-negative, respectively. In our study, the  $FP$  value indicates the number of error pixels that are not actually included in the human body, but the segmentation model determined they are. On the other hand, the  $FN$  value indicates the number of error pixels that are actually included in the human body, but the segmentation model determined they are not. The error rate of the system can be calculated using the rate of total error pixels, including both  $FP$  and  $FN$  pixels. Then, the system's accuracy can also be easily calculated by subtracting the error rate from 1.0.

#### 4.4.2. Implementation Results

Figure 14 is a frame-by-frame graph of the accuracy rate when our implementation is operated using the Selfie and DeepLab-v3 segmentation models, respectively.



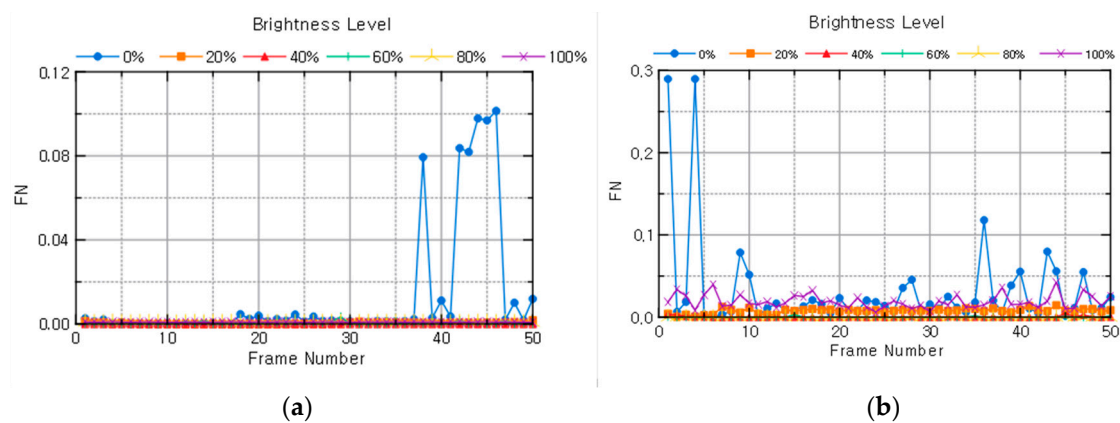
**Figure 14.** The proposed system's accuracy was measured with our implementation: (a) Selfie model case; (b) DeepLab-v3 model case.

In Figure 14, the numbers on the x-axis represent the order in the series of input frames, and the values on the y-axis show the accuracy rates for those input frames, which have been measured according to the methods in Section 4.4.1. The brightness levels in the index



mean that our measurements have been carried out by increasing the brightness from 0% to 100% in 20% intervals. In Figure 14a, the accuracy rates with the selfie model show a gradually decreasing trend as the number of frames increases, except in the case of the brightness level of 100%. In Figure 14b, the accuracy rates with the DeepLab-v3 model appear to be over 70% in most cases, except for the brightness level of 0%. The results in Figure 14 illustrate that the accuracy rates are more stable with the Deeplab-v3 model compared to the case of the Selfie model.

We have also measured the *FN* rates and the *FP* rates to analyze the differences between the cases further using those two segmentation models. Figure 15 shows a frame-by-frame graph of the *FN* rates measured with our implementation. In Figure 15a, the *FN* rates of the Selfie model case generally appear to be small values (around 0) and do not have many changes. When the brightness level is set to 0%, the *FN* rates considerably increase since the segmentation model does not work appropriately with much lower brightness. As shown in Figure 15b, the *FN* rates of the DeepLab-v3 model case also appear stable below 0.05 at most brightness levels except for level 0%. This implies that the accuracy rate differences of both segmentation models in our implementation are not significantly affected by the *FN* errors.

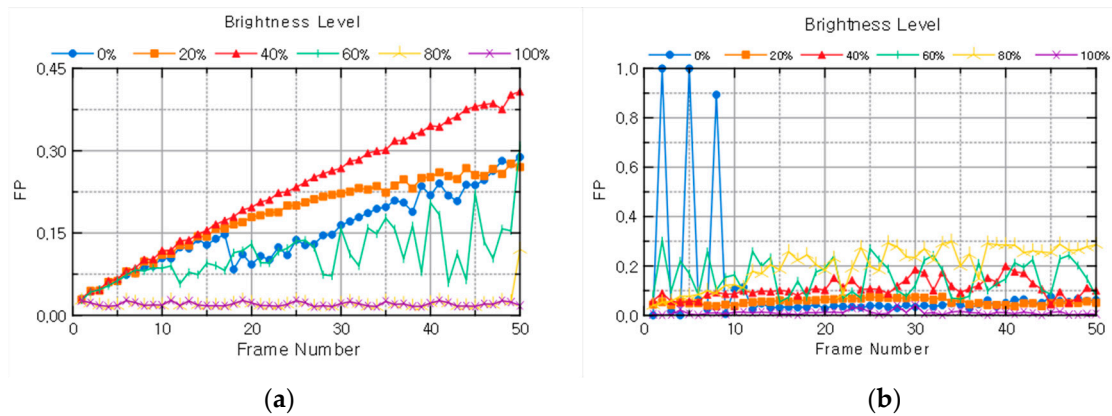


**Figure 15.** FN rates of the proposed system: (a) Selfie model case; (b) DeepLab-v3 model case.

Figure 16 is a frame-by-frame graph of the *FP* rates when our implementation is operated using the Selfie and DeepLab-v3 segmentation models. In Figure 16a, the *FP* rates of the Selfie model case continuously increase as the frame progresses. This may be caused by the fact that the dark area generated by the regional brightness control is provided with a feed-back for processing the human body segmentation of the next frame. When the brightness level is high, 80% through 100%, the *FP* rate plots are quite stable under 0.05. At a brightness level of 60%, the *FP* rate plots become unstable between 0.0 and 0.3. As the brightness level decreases, the *FP* error rates appear significant, so the proposed system does not work properly. As shown in Figure 16b, the *FP* rates of the DeepLab-v3 model case appeared stable, with values lower than 0.3 in most cases of brightness levels. Likewise, in this case, the changes in *FP* rates become more active as the brightness level decreases.

When reviewing the operational results of the proposed system, several implications can emerge. The measured results of our implementation seemed highly dependent on the ability of the segmentation model. The Selfie model of MediaPipe did not work well without sufficient light. With brightness levels below 60%, it could not properly distinguish the human body and blurred areas as well. Otherwise, the DeepLab-v3 model did not suffer much from low light levels. It is important that more experiments with various segmentation models and experimental scenarios be conducted to increase the proposed system's practicality. This is because the results presented in this paper depend quite a bit on several factors, including the brightness level and features of the previous frame.





**Figure 16.** FP rates of the proposed system: (a) Selfie model case; (b) DeepLab-v3 model case.

It is also expected that the performance of the proposed system could be greatly enhanced if the segmentation model is newly trained using a dataset appropriate for the purpose of this study. Because the existing well-known segmentation models, including Selfie and DeepLab-v3, have already been trained with clear images with normal brightness levels, they are not suitable for our experiments to be directly applied. Training models with appropriate data sets for our study could be an important item for our future work.

## 5. Conclusions

The hardware technology associated with beam projectors is continuously evolving. As higher-performance light sources for beam projectors are developed and applied, the negative effects on human glare are also increasing. This study proposes a regional brightness control method for beam projector systems to protect the eyesight of people who perform activities such as delivering lectures in front of a beam projector.

The proposed system detects the human body area on the screen and regionally reduces the light intensity in the detected area. To achieve these functions, the proposed system first generates two profiles for geometric and color transformations to eliminate the differences between the original source image and the projected image sampled by the camera. After applying transformations with those two profiles, the SSIM is used to generate a SSIM-MAP for the corresponding locations in the compared two images. Areas with lower SSIM values would indicate people or unintended objects in front of the screen. Then, those areas are segmented to extract only the actual human body. Finally, the system works by controlling the area's output brightness to reduce human glare.

One of the important features or implications of our study is that we have applied SSIM-MAP to the segmentation model so that the proposed system can identify the human body area with a method based on image differences. It enables the proposed system to escape confusion between human figures in the projected image and the real human body. The most important contribution of this research is proposing a new way to utilize the SSIM for beam projectors. The SSIM was developed as a metric to evaluate the quality of an image, but by reducing the window size for calculation, it can be used to identify where the changes have occurred.

Another point is that we have proposed and implemented blurring and color conversion methods to increase the similarity between the original image and the projected image captured by a camera. Though blurring is a common method to degrade image quality by reproducing the effect of light smearing [8], it is quite suitable for the proposed system to simulate image degradation due to projection and capture with a camera. Color conversion also achieved additional similarity improvements by correcting color values based on the position of each pixel. Since the two methods have different causes of the phenomenon they solve, we propose that applying both methods to this problem is an appropriate way to improve the similarity.

For future work, more experiments with various scenarios and configurations would be carried out to enhance the proposed system's performance and energy efficiency. To perform this, additional segmentation models need to be applied or newly trained to optimize our purpose. Some image quality assessment methods, such as PSNR and MSE, can be considered in comparison with SSIM. For real-time processing, you need to consider process optimization, dynamic brightness control, and energy efficiency to improve performance. During the installation process, some additional preprocessing tasks, such as brightness equalization and camera calibration, may considerably affect the system's performance. Finally, referring to user experience would improve the ease of use of your proposed system.

**Author Contributions:** Conceptualization, H.-G.J. and K.-H.L.; formal analysis, H.-G.J.; investigation, H.-G.J.; methodology, H.-G.J. and K.-H.L.; writing—original draft, H.-G.J.; writing—review and editing, K.-H.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2024-RS-2022-00156334).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. This data can be found here: <https://github.com/profjeon/dataset> (accessed on 1 February 2024).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Greenhill, L.P. The Educational Effectiveness, Acceptability, and Feasibility of the Eidophor Large-Screen Television Projector. 1962. Available online: <https://eric.ed.gov/?id=ED030306> (accessed on 3 February 2024).
- Berman, S.M.; Bullimore, M.A.; Jacobs, R.J.; Bailey, I.L.; Gandhi, N. An Objective Measure of Discomfort Glare. *J. Illum. Eng. Soc.* **1994**, *23*, 40–49. [\[CrossRef\]](#)
- Dave, A.; Kang, M.; Hwang, J.; Lorenzo, M.; Oh, P. Towards Smart Classroom: Affordable and Simple Approach to Dynamic Projection Mapping for Education. In Proceedings of the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 6–8 January 2020; IEEE: New York, NY, USA, 2020; pp. 0942–0947.
- Hu, Y.; Li, Q.; Hsu, S. Interactive Visual Computer Vision Analysis Based on Artificial Intelligence Technology in Intelligent Education. *Neural. Comput. Applic.* **2022**, *34*, 9315–9333. [\[CrossRef\]](#)
- Lin, C.-Y.; Chang, W.-W.; Chen, Y.-H. Intelligent Projector System Based on Computer Vision. In Proceedings of the 2011 Fifth International Conference on Genetic and Evolutionary Computing, Kitakyushu, Japan, 29 August–1 September 2011; IEEE: New York, NY, USA, 2011; pp. 176–179.
- Wang, M.; Deng, W. Deep Face Recognition: A Survey. *Neurocomputing* **2021**, *429*, 215–244. [\[CrossRef\]](#)
- Goel, R.; Mehmood, I.; Ugail, H. A Study of Deep Learning-Based Face Recognition Models for Sibling Identification. *Sensors* **2021**, *21*, 5068. [\[CrossRef\]](#) [\[PubMed\]](#)
- Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [\[CrossRef\]](#) [\[PubMed\]](#)
- Laouamer, L.; AlShaikh, M.; Nana, L.; Pascu, A.C. Robust Watermarking Scheme and Tamper Detection Based on Threshold versus Intensity. *J. Innov. Digit. Ecosyst.* **2015**, *2*, 1–12. [\[CrossRef\]](#)
- Wang, Z.; Bovik, A.C. Mean Squared Error: Love It or Leave It? A New Look at Signal Fidelity Measures. *IEEE Signal Process. Mag.* **2009**, *26*, 98–117. [\[CrossRef\]](#)
- Zhang, L.; Zhang, L.; Mou, X.; Zhang, D. FSIM: A Feature Similarity Index for Image Quality Assessment. *IEEE Trans. Image Process.* **2011**, *20*, 2378–2386. [\[CrossRef\]](#) [\[PubMed\]](#)
- Chandler, D.M.; Hemami, S.S. VSNR: A Wavelet-Based Visual Signal-to-Noise Ratio for Natural Images. *IEEE Trans. Image Process.* **2007**, *16*, 2284–2298. [\[CrossRef\]](#) [\[PubMed\]](#)
- Wang, Z.; Simoncelli, E.P.; Bovik, A.C. Multiscale Structural Similarity for Image Quality Assessment. In Proceedings of the Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, Pacific Grove, CA, USA, 9–12 November 2003; IEEE: New York, NY, USA, 2004; pp. 1398–1402.
- Akl, A.; Yaacoub, C. Image Analysis by Structural Dissimilarity Estimation. In Proceedings of the 2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA), Istanbul, Turkey, 6–9 November 2019; IEEE: New York, NY, USA, 2019; pp. 1–4.

15. Golestani, H.B.; Ghanbari, M. Window Size Influence on SSIM Fidelity. In Proceedings of the 7<sup>th</sup> International Symposium on Telecommunications (IST'2014), Tehran, Iran, 9–11 September 2014; IEEE: New York, NY, USA, 2015; pp. 355–360.
16. Sukthankar, R.; Mullin, M.D. Automatic Keystone Correction for Camera-Assisted Presentation Interfaces. In *Advances in Multimodal Interfaces—ICMI 2000*; Tan, T., Shi, Y., Gao, W., Eds.; Springer: Berlin/Heidelberg, Germany, 2000; pp. 607–614.
17. Haralick, R.M.; Shapiro, L.G. Image Segmentation Techniques. *Comput. Vis. Graph. Image Process.* **1985**, *29*, 100–132. [CrossRef]
18. Khoreva, A.; Benenson, R.; Hosang, J.; Hein, M.; Schiele, B. Simple Does It: Weakly Supervised Instance and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 26 July 2017; pp. 876–885.
19. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional Neural Networks: An Overview and Application in Radiology. *Insights Imaging* **2018**, *9*, 611–629. [CrossRef] [PubMed]
20. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 28 June 2014. [CrossRef]
21. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019. [CrossRef]
22. MediaPipe Solutions Guide | Google for Developers. Available online: <https://developers.google.com/mediapipe/solutions/guide> (accessed on 12 July 2023).
23. Chen, L.-C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587. [CrossRef]
24. OpenCV: Geometric Image Transformations. Available online: [https://docs.opencv.org/4.x/da/d54/group\\_imgproc\\_transform.html](https://docs.opencv.org/4.x/da/d54/group_imgproc_transform.html) (accessed on 12 July 2023).
25. Harris, C.G.; Stephens, M. A Combined Corner and Edge Detector. Available online: <https://www.semanticscholar.org/paper/A-Combined-Corner-and-Edge-Detector-Harris-Stephens/6818668fb895d95861a2eb9673ddc3a41e27b3b3> (accessed on 12 July 2023).
26. Lowe, D.G. Object Recognition from Local Scale-Invariant Features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; IEEE: New York, NY, USA, 2002; Volume 2, pp. 1150–1157.
27. Architectural Study of HOG Feature Extraction Processor for Real-Time Object Detection. Available online: <https://ieeexplore.ieee.org/abstract/document/6363206/> (accessed on 12 July 2023).
28. Bryson, M. K-Means Clustering Using Localized Histogram Analysis. 2008. Available online: <https://cse.sc.edu/~songwang/CourseProj/proj2007/report/bryson-report.pdf> (accessed on 6 February 2023).
29. Gonzales-Barron, U.; Butler, F. A Comparison of Seven Thresholding Techniques with the K-Means Clustering Algorithm for Measurement of Bread-Crumb Features by Digital Image Analysis. *J. Food Eng.* **2006**, *74*, 268–278. [CrossRef]
30. Hart, P. How the Hough Transform Was Invented [DSP History]. *IEEE Signal Process. Mag.* **2009**, *26*, 18–22. [CrossRef]
31. OpenCV: Hough Line Transform. Available online: [https://docs.opencv.org/3.4/d9/db0/tutorial\\_hough\\_lines.html](https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html) (accessed on 12 July 2023).
32. Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *J. R. Stat. Soc.* **1996**, *58*, 267–288. [CrossRef]
33. Sklearn.Linear\_model.Lasso. Available online: [https://scikit-learn.org/stable/modules/generated/sklearn.linear\\_model.Lasso.html](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html) (accessed on 12 July 2023).
34. Damon, J. Generic Structure of Two-Dimensional Images Under Gaussian Blurring. *SIAM J. Appl. Math.* **1998**, *59*, 97–138. [CrossRef]
35. Image Segmentation Guide | MediaPipe | Google for Developers. Available online: [https://developers.google.com/mediapipe/solutions/vision/image\\_segmenter](https://developers.google.com/mediapipe/solutions/vision/image_segmenter) (accessed on 13 December 2023).
36. Visa, S.; Ramsay, B.; Ralescu, A.; VanDerKnaap, E. Confusion Matrix-Based Feature Selection. *All Fac. Artic.* **2011**, *710*, 120–127.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.