Video Quality Assessment System using Deep Optical Flow and Fourier Property

Donggoo Kang  
Chung-Ang University

Yeongjoon Ki  
Chung-Ang University

Sunkyu Kwon  
Chung-Ang University

Hyunchol Kim  
4by4 Inc

Jinah Kim  
4by4 Inc

Joonki Paik (✉️ paikj@cau.ac.kr)  
Chung-Ang University

Article

Keywords:

Posted Date: March 24th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2685872/v1

License: 😊 This work is licensed under a Creative Commons Attribution 4.0 International License.  
Read Full License

Additional Declarations: No competing interests reported.
Video Quality Assessment System using Deep Optical Flow and Fourier Property

Donggoo Kang¹,†, Yeongjoon Kim²,†, Sunkyu Kwon², Hyunchol Kim³, Jinah Kim³, and Joonki Paik¹,²,*

¹Department of Image, Chung-Ang University, 84 Heukseok-ro, Seoul, 06974, Korea.
²Department of Artificial Intelligence, Chung-Ang University, 84 Heukseok-ro, Seoul, 06974, Korea.
†All authors contributed equally
*paikj@cau.ac.kr

ABSTRACT

Providing high-quality video content is critical in various fields such as VFX film, media facade, digital signage, product promotion, and interactive content, as it enhances the viewer’s visual experience. Quantifying the visual quality of videos is crucial in determining their pricing. Subjective quality is the most crucial feature in video quality assessment, although other factors also play a significant role. While fully automated video assessment is convenient, there are instances where engaging with an editorial specialist is necessary to assess subjective quality. Our focus was on the most common scenarios that affect video quality, namely irregular camera movements and incomplete focusing. To quantify irregular camera movement, we employed a deep learning-based optical flow estimation method, and to measure incomplete focusing, we used a blur detection algorithm based on Fast Fourier Transform (FFT). We also proposed an adaptive threshold that utilizes statistical techniques to classify good and bad scenes. In our experiments, we applied this system to videos from various domains, and it successfully quantified video quality using a reasonable threshold.

1 Introduction

Video quality assessment (VQA) is the process of evaluating the perceived visual quality of a video. It is a multidisciplinary field that combines techniques from image processing, computer vision, human vision, and statistics. The primary objective of VQA is to measure the quality of a video in a manner that is perceptually relevant to humans. With the increasing availability of high-definition video content and the growing demand for video streaming services, the need for accurate and efficient VQA methods has become increasingly important.

Video quality assessment involves evaluating various factors such as resolution, frame rate, compression, and color accuracy. However, the most crucial factor in determining video quality is subjective quality, which is based on the viewer’s perception of how the video appears and feels. This encompasses elements such as sharpness, noise, and overall visual appeal. There are several works that design hand-crafted features in view of these factors. These works focused on the specific problem, such as undesired blur, JPEG artifact, ringing artifact, frame freezing caused by camera error, and noise. A more comprehensive explanation of hand-crafted feature-based VQA can be found in.

Recently, deep learning-based VQA methods are presented to estimate quality score. Learning-based methods are more effective and robust than a hand-crafted feature on the given dataset distribution. However, there are certain limitations to using datasets collected by specific groups, as they may not represent the entire distribution of videos found in the world. Additionally, the process of labeling videos can be both expensive and time-consuming.

Subjective quality is typically assessed through subjective tests, where a group of viewers is presented with the same video and asked to rate it on a scale of 1 to 5 or 1 to 10. This approach provides a more precise assessment of video quality since it considers the viewer’s individual preferences and biases. Subjective quality is a critical factor in video quality assessment because it ultimately determines the level of enjoyment and engagement experienced by the viewer.

However, these subjective quality assessment is time-consuming and expensive, as it requires a large number of viewers to participate in the testing. It can also be affected by factors such as lighting and viewing conditions, making it difficult to compare results between different videos or devices. As a result, objective quality assessment methods have also been developed to complement subjective testing and provide a more comprehensive evaluation of video quality.

There are several representative objective quality assessment metrics, such as peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). It’s worth noting that, objective quality assessment methods are useful for providing a
Figure 1. An example of actual video clips available for purchase on a commercial site is KEYCUTstock. The clips offered on this site are captured from various domains and are predominantly edited as short clips.

quick and efficient evaluation of video quality, but they may not always align with human perception.

Table 1. Bad clips collected during the actual editing process.

<table>
<thead>
<tr>
<th>Undesired Camera Movement</th>
<th>Incomplete Focus</th>
<th>ETC(camera error)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clips</td>
<td>2,587</td>
<td>1,019</td>
</tr>
<tr>
<td>Ratio</td>
<td>69.94%</td>
<td>27.55%</td>
</tr>
</tbody>
</table>

To address these issues, we consider an actual video editing process and proposed a novel VQA system that involves collaboration with an editing specialist. Table 1 and Figure 1 presents the collection of video clips obtained through the editing process of high-resolution videos captured by photographers from around the world. These video clips are classified into three categories of imperfections: undesired camera motion, incomplete focus, and camera errors. Since most of the videos are recorded by a hand-held camera, the major problem that degrades video quality is undesired camera motion. Also, incomplete focus caused by the expert manual mode of camera is the second problem.

Based on this observation, we designed a novel VQA system with a specific focus on detecting and addressing undesired camera motion and incomplete focus, based on this observation. The proposed VQA system consists of two-parts: i) estimation of video quality and ii) computation of an adaptive threshold using an iterative search algorithm. Our approach focuses on quantifying video quality by exclusively analyzing camera motion and blurriness. To achieve this, we employ a deep optical-flow network to estimate the camera motion, and utilize the the Fourier transform property to estimate the level of blurriness present in the video. Through the estimated video quality, we use an iterative search algorithm to compute the video adaptive threshold, which allows us to classify good and bad frames.

The contributions of this paper are summarized as follows:

- A VQA system that accurately quantifies video quality using a deep optical-flow network and Fourier property.
- An iterative search algorithm that calculates a video adaptive threshold, which helps to overcome issues caused by a constant threshold approach.
- Extensive experiments that demonstrate the effectiveness of the proposed VQA system with both subjective and objective quality measures showing favorable result.

2 Related Works

2.1 Camera Motion

In computer vision algorithms, camera motion estimation involves determining the precise location of specific pixels in the next frame relative to their location in the current frame. By analyzing the changes in pixel position, velocity, and acceleration over
Figure 2. Optical flow in the motion field. In the $t$-th frame of a video, denoted by $I_t$, a pixel located at coordinates $(x, y)$ undergoes a displacement of $(dx, dy)$, resulting in the new position of $(x + dx, y + dy)$ in the next frame $I_{t+1}$. A collection of displacement of pixels across consecutive frames is referred to as the motion field.

Optical flow is a technique that uses the motion of pixels in consecutive frames of a video to estimate the movement of the camera. It can be used to measure the speed and direction of camera movement and the amount of rotation. Optical-flow can be applied to the video stabilization property, and this method enables camera movement estimation. Optical-flow is the motion of the pixel between consecutive frames, caused by the camera movement or object movement.

As shown in Fig. 2, A collection of displacement of pixels across consecutive frames is referred to as the motion field. An arrow in the motion field is the corresponding optical flow vector, which can estimate the information of the moving pixels between two consecutive frames as:

$$I_t = I(x,y,t), \quad F_t = (dx,dy),$$

where $(x,y)$ represents the coordinate of a pixel in the image $I_t$, $t$ the frame number, and the vector $F_t$ the displacement of the pixel across consecutive frames. From (1), the next frame is formulated as

$$I_{t+1} = I(x + dx, y + dy, t + dt).$$

Representative works to estimate optical flow include Lucas-Kanade methods. They assumed that the flow is essentially constant in a local neighborhood of the pixel, and solved using the least-squares criterion for all the pixels. While these methods show reasonable results under certain constraints, such as small and approximately constant displacement of pixels within a neighborhood, they can be computationally expensive due to the process of finding the best match between corresponding points in adjacent frames. To address this issue, there are feature tracking methods that track sparse pixel to reduce computational cost, and deep learning-based methods that perform vector operations parallel using GPU memory.

Deep-learning based optical-flow estimation methods are faster and more accurate than the conventional method but need large datasets to optimize millions of network parameters. Collecting optical flow data is challenging as ground-truth motion fields would require annotations for every pixel in a video clip. To overcome this problem, most datasets synthesize motion fields using virtual tools.

Feature tracking is a technique that uses distinctive features in a video, such as corners or edges, to track the camera’s movement. It can be used to determine the camera’s position and orientation of the camera in each frame of the video. Representative future detection and tracking methods include Harris Corner Detector, Scale Invariant Feature Transform (SIFT), and Speeded-Up Robust Features (SURF). These methods detect invariant feature points and track feature points at consecutive frames. Based on this measurement, we can compute the affine transformation matrix and use it to stabilize the video. Feature tracking methods are fast and memory efficient. However, if feature points are not detected due to the motion blur and camera distortion, the predicted consecutive transformation matrix may become unstable.
2.1.3 Stabilization
Camera stabilization is a technique used to remove unwanted camera movement from a video, and it involves measuring the amount and type of camera movement present in the video. Most of works use a combination of feature tracking and optical-flow methods for camera stabilization\textsuperscript{36–38}. Deep learning-based methods also use these methods as prior knowledge and constraints.

2.2 Blur Detection
Blur detection is a technique used to determine whether an image or video is blurred or not. Blurry images or videos have edges and details that are not sharp and clear, and the overall image appears to be out of focus. Blur detection is crucial in a variety of applications, including image and video processing, computer vision, and computational photography.

Fourier transform-based blur detection is a method for detecting blur in an image or video by analyzing the frequency content of the image\textsuperscript{2–7}. The Fourier transform is a mathematical technique that decomposes a signal into its constituent frequencies, which can then be analyzed to determine the frequency content of the signal. In the case of blur detection, the Fourier transform is applied to the image to convert it into the frequency domain. Once in the frequency domain, the image is analyzed to determine the frequency content and the distribution of the energy across different frequencies.

Blur detection is also a challenging task because the blur can be caused by different factors, such as camera shake, low light, compression artifacts, and user intentional out of focus. Actually, the last mentioned factor is the most difficult case in that it should consider photographer’s intention. Tang et al. proposed a deep learning-based segmentation method, DefocusNet, for detecting out-of-focus regions\textsuperscript{39}. This method requires a segmentation mask and generates a blur on the background region. While it shows favorable results, it only optimizes background region blur and may not work well for images with different distributions than those in the training datasets.

3 Proposed Method
3.1 System Overview
The proposed VQA system consists of 3-steps: i) blurriness estimation using the Fourier property, ii) camera movement estimation using the video stabilization method, and iii) calculation of statistical adaptive threshold. Figure 3 shows an overview of the proposed VQA system. To estimate blurriness, we adopt Fourier transform and calculate their high-frequency portion. We also estimate camera motion from optical flow obtained using a deep learning-based neural network to estimate camera motion.

Based on the estimated blurriness and camera motion, we compute the video adaptive threshold using iterative search and statistical algorithms. In the following subsections, we will provide a step-by-step detailed explanation of the three steps.

3.2 Blurriness Estimation Using the Fourier Property
The two-dimensional (2D) Fourier transform is a mathematical technique that is used to decompose an image into its constituent frequencies. It is used to analyze the frequency content of an image, which can be useful for a variety of image processing tasks, such as image enhancement, filtering, and compression. The 2-D Fourier transform is applied to an image by converting the image into the frequency domain, which represents the image in terms of its constituent frequencies rather than its spatial coordinates.

As a theoretical basis, the one-dimensional (1D) Fourier transform is calculated using the following formula:

\[
F(u) = \int_{-\infty}^{\infty} f(x)e^{-2j\pi ux}dx,
\]

where \( F(u) \) represents the transformed signal in the frequency domain, \( f(x) \) the input 1D signal, and \( u \) the frequency coordinates. Applying the Fourier transform to 2D image input is also possible using the following formula:

\[
F(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)e^{-2j\pi(ux+vy)}dx dy,
\]

where \( F(u,v) \) is the transformed image in the frequency domain, \( f(x,y) \) the original image in the spatial domain, and \( (u,v) \) the frequency coordinates.

The result of the 2D Fourier transform is a complex-valued image, where the magnitude, \(|F(u,v)|\), represents the strength of the frequency and the phase represents the position of the frequency. The magnitude of the transformed image is often used to analyze the frequency content of the image, as it represents the strength of the different frequencies present in the image. The
Figure 3. (From top to bottom) Optical flow and the corresponding video frames, blurriness and camera motion score graph with an adaptive threshold, a collection of video frames in a stable sequence, and their optical flows. We first estimate blurriness using the Fourier property and camera motion based on optical flow. After that, we calculate the video adaptive threshold based on the statistical adaptive threshold algorithm.

2D Fourier transform can be used to analyze the frequency content of an image in both the horizontal and vertical directions. This is useful for image processing tasks such as image enhancement, filtering, and compression.

We used Richard’s method to estimate the degree of blurriness, as described in [8]. This method employs the Fourier transform to measure no-reference noise and blur. Initially, the input image is transformed into the frequency domain using the Fourier transform. To evaluate the energy of a specific frequency in all directions, we create \( n \) equi-spaced circles with radius \( r_i \) for \( i = 1, \ldots, n \) in the 2D Fourier transform domain, identified by the \((u,v)\) coordinate system as shown in Figure 4(a). Subsequently, we calculate the normalized power within the \( i \)-th ring, which is the area between adjacent circles with radii \( r_i \) and \( r_{i-1} \), where \( r_0 = 0 \).

\[
p_i = \frac{1}{P} \left\{ \sum_{u^2+v^2 \leq r_i} F(u,v) - \sum_{u^2+v^2 \leq r_{i-1}} F(u,v) \right\}, \quad \text{for} \quad i = 1, \ldots, n, \tag{5}
\]

where \( P = p_1 \). The normalized power in each ring is shown in Figure 4(b).

Figure 5 shows sets of image, Fourier transform magnitude, and the corresponding normalized power. As shown in Figure 5(a), the natural scene has a linear and balanced shape on the histogram. On the other hand, the noisy scene has relatively more high-frequency components, and the blurry scene lacks both low- and high-frequency components as shown in Figures 5(b) and 5(c), respectively.

In Figure 4(b), the blurriness of an image is measured using the \( \ell_1 \) distance between a diagonal line \( L \) and its pixels \( p_i \).

\[
d_i = p_i - \ell_i. \tag{6}
\]

The overall blurriness distance \( D' \) is computed as
Figure 4. (a) Circles in the Fourier domain with \((u, v)\) coordinates and (b) the normalized power within each ring.

Figure 5. Visualization of (a) natural, (b) noisy, and (c) blurry images, with their corresponding Fourier transform magnitudes and histograms.

\[ D' = \sum_{i=0}^{n} d_i. \]  

(7)

A positive value of \(D'\) indicates a noisy image, while a negative value indicates a blurred image. However, as only blurriness is being considered in this work, the negative value of \(D'\) is used to estimate the blurriness of the image, denoted by \(B'\).
\(B' = \begin{cases} |D|, & \text{if } D < 0 \\ 0, & \text{otherwise} \end{cases}\) (8)

3.3 Camera Movement Estimation Using Video Stabilization Property

To estimate optical flow, we used FlowNet 2.0 as our backbone, which is a deep learning-based method that can jointly deal with both small and large displacement.

Figure 6 shows the architecture of FlowNet 2.0. The network consists of two paths: a large displacement path and a small displacement path. The former is constructed by stacking FlowNetC and FlowNetS modules, which respectively compute explicit correlation of input images or feature maps and warp the previous image \(I_{t-1} = (x, y)\) based on intermediate optical flow \(F_t = (\delta x_{t-1}, \delta y_{t-1})\). The resulting warped image \(\tilde{I}_{t-1}\) is then used to calculate the error.

\[e_t = \|\tilde{I}_{t-1} - I_t\|.\] (9)

The computed error is fed as input to optimize the network. This proposed warping operation significantly improves the results, allowing for the use of a shallow layer compared to previous methods. To handle small displacement, the network employs a smaller stride at the beginning and convolutions between up-convolutions in the FlowNetS architecture. The resulting flows from both paths are fused using the small fusion network to provide the final estimation.

Figure 7 demonstrates that the original FlowNet yields noisy results when dealing with small movements and details. In contrast, the FlowNet 2.0 with a two-path architecture produces optical flow estimates that are robust in handling small displacements.

![Figure 6. Structure of the FlowNet 2.0 model](image)

![Figure 7. Visual comparison between the original FlowNet and FlowNet2.0 using our proposed system.](image)
We estimated camera movement $M_t$ by utilizing the magnitude of optical flow $F_t$. The algorithm for estimating the camera movement is illustrated in Figure 8. The video sequence pair $I_t$ and $I_{t-1}$ are fed into the FlowNet 2.0 network to compute optical flow $F_t$. We then apply principal component analysis (PCA) to $F_t$, assuming that the main principal components correspond to camera movements as they represent global motion, while the movements of objects within the scene represent local motion.

$$M_t = \arg \max \text{PCA}(f_t),$$

(10)

where $f_t$ is each optical flow value on the corresponding pixel.

Finally, we computed the magnitude of camera motion $C_t$ by summing over the camera motion vectors.

$$C_t = \sum_i m_i,$$

(11)

where $m_i$ is each magnitude of the major principal components.

![Figure 8. Camera movement estimation network](image)

### 3.4 Statistical Adaptive Threshold

The proposed method enabled us to estimate blurriness ($B$) and camera movement ($C$) of a video. Based on these results, we calculate an adaptive threshold using statistical and iterative search algorithms. However, it is challenging to distinguish stable images from videos captured in various domains. For example, while a particular input video may remain stable within a predetermined range, videos of the sea or mountain areas are significantly impacted by waves or the shaking of leaves. Furthermore, different domains have distinct motion patterns, which makes it impractical to use a fixed threshold. Additionally, differentiating intentional motion from actual unstable motion in an input video is also challenging. As a result, it is crucial to discover a threshold that can adapt to the input video.

To address this problem, we propose the Statistical Adaptive Threshold (SAT), which separates the stable sequences from the videos. The proposed SAT method uses FlowNet2.0, a deep learning method for motion estimation in videos, as a backbone to extract stable images. Optical flow-based camera motion estimation is the foundation for image stabilization, making it a crucial component of SAT, and providing the improved performance. In addition, we use the blurriness value derived from the Fourier property. SAT is unique compared to other methods as it incorporates statistical techniques such as mean, standard deviation, and variance to evaluate the motion of an object in an input image. Additionally, SAT is distinct in the following ways:

- Firstly, SAT employs stabilization variables to identify stable ranges that enable the evaluation of object motion across various domains.
- Secondly, SAT removes outliers of object motion from the input image, ensuring that the optimal image is not affected by the separation process.
- Thirdly, SAT proportionally adds a margin value for human visual judgment to the threshold used for separating the optimal image.

To accurately determine the camera motion between frames and blurriness in various domains, the SAT algorithm is computed by the following formula.
\[ V'_n = \sum_{i=1}^{N} \left[ \sqrt{\left| f'_c(I_i, I_{i-1}) \right|^2 + f'_b(I_i)} \right], \]  

(12)

where \( I \) represents an image frame of input video, and values of video quality is stored as a list form. We also utilize a stabilization range variable to identify which frame is stable. A pseudo code for calculating stabilization range variables is shown in Algorithm 1. Let the object camera motion computed by FlowNet2.0 be \( f'_c \), and the blurriness using the Fourier property \( f'_b \), the amount of video quality \( C' \) is determined as follows:

**Algorithm 1 SV-Setup: Stabilizing Variable Setup in SAT**

**Input:** \( D \): input video, \( \alpha \): initial stabilization variable, \( \beta \): step size  
**Output:** \( \theta \): final stabilization variable(SV)

for each frame \( I_i \) in input video \( D \) do

Compute the video quality \( V^* \) using camera motion \( C' \) and the blurriness \( B' \)

\[
C'_n = [\sqrt{|C'_1|^2}, ..., \sqrt{|C'_i|^2}] \\
B'_n = \text{CDF of Fourier rings } |D| \\
V^* = C' + B'
\]

end for

\( S'_n = S_{\text{scaler}}(V^*) \) with outlier removal and normalization

\( \alpha = 0.85, \beta = 0.05 \)

for step \( i = \alpha \) to \( \beta \) do

if \( S'_n \leq \alpha \) then

\[
S'_{\text{min}} = \min(S'_n), S'_{\text{max}} = \max(S'_n) \\
S'_\text{median} = \frac{1}{2}(S'_{\text{max}} + S'_{\text{min}}) \\
S'_\text{mean} = \frac{1}{n} \sum S'_n \\
\]

if \( \overline{S}\text{'mean} \leq S'_\text{median} \) then

\( S'_{\text{ascending}} = [S'_1, ..., S'_n] \) is an ascending list

\( \alpha'_{\text{ascending}} = [\alpha'_1, ..., \alpha'_n] \) is an ascending list

\( \theta = \alpha'_{\text{ascending}}(\min(S'_{\text{ascending}})) \)

return \( \theta \)

else

\( \theta = \alpha \)

return \( \theta \)

end if

Update \( \alpha \leftarrow \alpha - \beta \)

end if

end for

If we set the threshold as a constant value, it may result in missclassifying intended small camera motions as bad clips. On the other hand, if we set the threshold without a constant and only rely on statistical methods, it may also missclassify stable videos as bad clips since the threshold must be between the values of stable videos. Therefore, we need to use both the stabilization variable and the stabilization constant to overcome this issue. To create a stabilization variable, we compute the value of \( S'_{\text{min}} \) and sort it based on the average of \( C'_n \) using \( S' \) as

\[
S' = \frac{\min(V'_n) + \max(V'_n)}{2}.
\]

(13)

In the the version of (14) when outliers are moved, we identify the value that minimizes the difference between the mean of \( C'_n \) and \( S' \) as
\[ S_{\text{min}}^* = \arg \min_{V_i \in V_n} \sum_{i=1}^{N} \frac{V_i' - Q_{2, \text{median}}(V_n')}{Q_3(V_n') - Q_1(V_n')}, \tag{14} \]

where \( Q_{2, \text{median}} \) is the median of \( C_n' \) in (12). \( Q_3 \) represents the third quartile (75%), and \( Q_1 \) represents the first quartile (25%). The distance from the center to zero is measured as a quartile. \( C_i' \) denotes the size of each object movement. Therefore, robust normalization mitigates the effect of the outliers. The stabilization range variable is denoted as \( \theta \), which can be formulated as

\[ \theta = f_{\alpha^*}(S_{\text{min}}^*). \tag{15} \]

The size of camera motion computed by (14), the maximum-minimum value of motion with outliers removed by (13), and \( \alpha^* \) at which \( S_{\text{min}}^* \) and \( S' \) are minimized in each equation becomes the optimal stabilization variable \( \theta \). After computing the stabilization variable \( \theta \) as described earlier, it is utilized to estimate the threshold using equations (16). To estimate the threshold for separating stable sequences, we use a statistical method commonly used in a normal distribution, such as mean and standard deviation. We assumed that computed values \( C_n' \) follow a normal distribution. In statistics of normal distribution, 2\( \sigma \) covers their 97.9% distribution. Based on our observations, we found that most of the problems occurred in a few seconds. Taking into account this observation and statistical considerations, we compute average, standard deviation, and margin values as follows:

\[ T_{\text{mean}} = \frac{1}{n} \sum_{k=0}^{\theta} V_n'(k), \quad T_{\text{std}} = \frac{\sum_{k=0}^{\theta} (V_n'(k) - T_{\text{mean}})}{n}, \quad \text{and} \quad T_{\text{margin}} = 1 + \frac{Q_3(V_n') - Q_1(V_n')}{\theta}. \tag{16} \]

The computed margin \( T_{\text{margin}} \) is initially set to 1, but in some domains, the constant value 1 can have a relatively large effect on the total threshold. To address this issue, we multiply stabilization variable \( \theta \) to the initial margin in (16). The final adaptive threshold is the sum of \( T_{\text{mean}}, T_{\text{std}}, \) and \( T_{\text{margin}} \), after taking into account the stabilization variable.

\[ SAT = T_{\text{mean}} + T_{\text{std}} + T_{\text{margin}}. \tag{17} \]

### 4 Experimental Results

#### 4.1 Dataset
We evaluated the proposed method on a dataset collected from the actual editing process. Figure 9 shows a collection of video data from KEYCUTstock website. It provides footage for Premiere in 4K, 8K, and higher resolutions. The collection includes various domains, such as cityscape, nature, timelapse, etc.

Table 1 shows the ratio of bad clips detected by an editing specialist in the collected dataset. The bad clips are mainly caused by unwanted camera movements, which can result from shaking hands while filming with a handheld camera, or the effects of wind or panning on a drone. Another significant issue is incomplete focus due to being out-of-focus. These issues were observed to occur in a few seconds of a long clip.

#### 4.2 Evaluation Metrics
We evaluated the proposed method with various metrics that are widely used for image quality assessment. Peak Signal-to-Noise Ratio (PSNR), Mean-Squared Error (MSE), Structural Similarity (SSIM), and Visual Information Fidelity (VIF) are used as metrics to evaluate the proposed method. To evaluate the quality of an input image, the following list is generally considered:

- Loss of image quality between frames of the same image.
- Similarity between frames by considering luminance, contrast, and structure of the image.
- Comparison of the amount of information present in the image with the of a reference image.

To assess the effectiveness of the proposed method, the initial image is compared with the image that has been processed using the proposed method, as described above. The performance of the proposed method is measured by the difference between the current and previous frames of both the original and the processed images.
4.2.1 PSNR
PSNR represents the power of noise relative to the maximum power that a signal can have. It is used to evaluate information loss and image quality on images that are contaminated by distortion. It is computed as the following formula:

\[
PSNR = 20 \times \log_{10} \left( \frac{MAX_I}{\sqrt{MSE}} \right)
\]

\[
= 20 \times \log_{10}(MAX_I) - 10 \times \log_{10}(MSE),
\]

where, \(MAX_I\) is the maximum value of the image. For 8-bit grayscale images, it ranges from 0 to 255.

4.2.2 MSE
MSE is a commonly used metric to evaluate the similarity between two images. The purpose of using MSE is to calculate the average squared difference between the pixel values of two images. When comparing two images using MSE, the pixel values of the images are compared, and the differences between the pixel values are squared. The squared differences are then averaged to produce a single value that represents the overall difference between the two images. The lower the MSE value, the more similar the two images are considered to be. The function of MSE is as follows:

\[
MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left| I(i, j) - I_{n-1}(i, j) \right|^2
\]

(20)

where \(I\) is an image of size \(m \times n\), and \(I_{n-1}\) is a distorted image by additive noise to \(I\). Here, \(I\) represents the current frame of the input video, and \(I_{n-1}\) represents the previous frame. The smaller the value of Mean Squared Error (MSE), as shown in Equation (20), the higher the value of Peak Signal-to-Noise Ratio (PSNR) because it appears in the denominator. Therefore, lower MSE and higher PSNR indicate better image quality. However, PSNR has certain limitations in terms of human-perceived image quality, such as inability to detect information loss and blurriness in the image.

4.2.3 SSIM
SSIM is used to compensate for the limitation of PSNR. SSIM evaluates the similarity to the original image as a method for evaluating image quality. It is an indicator to overcome the limitation of PSNR that it does not properly reflect human perception quality. It is a method of obtaining the similarity of images considering luminance, contrast, and structure. It has a value between 0 and 1, and the closer to 1, the higher the similarity. Therefore, SSIM equation supplemented with PSNR is as follows:

\[
SSIM(I, I_{n-1}) = L(I, I_{n-1}) \times C(I, I_{n-1}) \times S(I, I_{n-1}),
\]

(21)
where $L(I, I_{n-1})$ compares the average luminance of two images, $C(I, I_{n-1})$ is the contrast function, and $S(I, I_{n-1})$ is the structure function formulated as

$$
L(I, I_{n-1}) = \frac{2\mu_I\mu_{I_{n-1}} + c_1}{\mu_I^2 + \mu_{I_{n-1}}^2}, \quad C(I, I_{n-1}) = \frac{2\sigma_I\sigma_{I_{n-1}} + c_2}{\sigma_I^2 + \sigma_{I_{n-1}}^2}, \quad \text{and} \quad S(I, I_{n-1}) = \frac{\sigma_{I,I_{n-1}} + c_3}{\sigma_I\sigma_{I_{n-1}} + c_3},
$$

(22)

where $\mu$, $\sigma$, and $\sigma_{I,I_{n-1}}$ respectively represent the mean, variance, and covariance of $\{I, I_{n-1}\}$. $c_1, c_2, c_3$ are stabilizing variables that compensates for the problem of the denominator. $c_1 = (k_1 \ast L)^2$, $c_2 = (k_2 \ast L)^2$, and $c_3 = c_2/2$. We usually set $k_1 = 0.01, k_2 = 0.03$.

### 4.2.4 VIF

VIF is a popular image quality assessment algorithms along with SSIM. It is an index for measuring the fidelity of an image and is an overall reference image quality evaluation index based on image information. The quality is evaluated by comparing the amount of information present in the original image with the amount of information present in the image to be compared. As shown in Figure 10, The original image is a clean image signal without distortion and is represented by $C$.

![Figure 10. Visualization of VIF.](image)

The human visual system (HVS) perceives a signal represented by $E$, while the image signal affected by the distortion is represented by $D$ through the channel $C$. The signal $F$ recognized by the HVS throug the channel $C$ can be expressed as

$$
VIF = \frac{\sum_{j \in \text{subbands}} I_{n-1}(C^{N,j}, F^{N,j} | S^{N,j} = S^{N,j})}{\sum_{j \in \text{subbands}} I(C^{N,j}, E^{N,j} | S^{N,j} = S^{N,j})},
$$

(23)

where $I_{n-1}(C;F)$ and $I(C;E)$ are the amount of mutual information. In VIF, the amount of mutual information between $C$ and $E$, i.e., the entropy they share, is calculated. Similarly, the amount of mutual information between $D$ and $F$ is calculated in (23). The ratio of the two is then used to predict the quality of the image based on the amount of information. A value closer to 1 indicates better quality, while a value closer to 0 indicates lower quality.

### 4.2.5 Precision, Recall, and Dice score

Precision is the percentage of the actual Ground Truth value in the predicted positive value, which can be formulated as

$$
\text{Precision} = \frac{TP}{TP + FP} = \frac{TP}{\#\text{ground truth}}.
$$

(24)

Recall, also called sensitivity, is the percentage of the predicted positive range for the positive range of GT, which can be formulated as

$$
\text{Recall} = \frac{TP}{TP + FN} = \frac{TP}{\#\text{prediction}}.
$$

(25)

Dice score is a well known as F1 score as the harmonic average of Recall and precision. As shown in (27), F1 score achieves the optimal value at 1. Also, it refers to the best precision and Recall score, which can be formulated as

$$
\text{Dice} = \frac{2 \times TP}{(TP + FP) + (TP + FN)} = \frac{2(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})^2}.
$$

(27)
4.3 Quantitative results
We evaluated the structural quality of the processed videos using PSNR, SSIM, and VIF metrics. The thresholds for detecting stable clips were calculated using different methods, including average, average + standard deviation, average + bottom 30% of standard deviation, and the proposed SAT algorithm, on the actual videos. We then compared the stable clips obtained using each threshold and the original video. The results showed that the proposed threshold outperformed the other methods in terms of video stability. Moreover, the stable clips obtained using the proposed threshold exhibited higher quality compared to the original video, as evaluated objectively using the chosen metrics. The results are summarized in Tables 2 and 3.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SSIM</th>
<th>VIF</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>61.88</td>
<td>30.56</td>
<td>90.98</td>
<td>61.14</td>
</tr>
<tr>
<td>Average</td>
<td>69.66</td>
<td>32.38</td>
<td>94.95</td>
<td>65.66</td>
</tr>
<tr>
<td>Average+Std</td>
<td>65.06</td>
<td>31.38</td>
<td>93.10</td>
<td>63.18</td>
</tr>
<tr>
<td>Avg+Std(30%)</td>
<td>63.97</td>
<td>31.15</td>
<td>92.56</td>
<td>62.56</td>
</tr>
<tr>
<td><strong>Proposed method</strong></td>
<td><strong>70.11</strong></td>
<td><strong>32.41</strong></td>
<td><strong>94.99</strong></td>
<td><strong>65.84</strong></td>
</tr>
</tbody>
</table>

Table 3. Experimental results of Video 2 (PSNR, SSIM, VIF score).

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SSIM</th>
<th>VIF</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>28.93</td>
<td>23.51</td>
<td>83.33</td>
<td>45.26</td>
</tr>
<tr>
<td>Average</td>
<td>42.94</td>
<td>26.08</td>
<td>86.73</td>
<td>51.92</td>
</tr>
<tr>
<td>Average+Std</td>
<td>35.68</td>
<td>25.98</td>
<td>86.01</td>
<td>49.22</td>
</tr>
<tr>
<td>Avg+Std(30%)</td>
<td>42.06</td>
<td>25.85</td>
<td>86.48</td>
<td>51.46</td>
</tr>
<tr>
<td><strong>Proposed method</strong></td>
<td><strong>46.46</strong></td>
<td><strong>27.19</strong></td>
<td><strong>88.29</strong></td>
<td><strong>53.98</strong></td>
</tr>
</tbody>
</table>

Next, we measure the overall video quality using Dice, Precision, and Recall. We carry out this process by comparing ground-truth clips edited by the editorial specialist with clips edited by computed threshold. As shown in Tables 4 and 5, the proposed method outputs most similar clips except for recall. The reason for the low recall is that the proposed method divides clips slightly more narrowly than ground truth clips. However, the overall quality of dice, precision, and recall is higher than others.

<table>
<thead>
<tr>
<th>Method</th>
<th>Dice</th>
<th>Precision</th>
<th>Recall</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>97.41</td>
<td>97.02</td>
<td>97.80</td>
<td>97.41</td>
</tr>
<tr>
<td>Average+Std</td>
<td>92.66</td>
<td>87.39</td>
<td>98.60</td>
<td>92.88</td>
</tr>
<tr>
<td>Avg+Std(30%)</td>
<td>93.71</td>
<td>89.29</td>
<td>98.60</td>
<td>93.87</td>
</tr>
<tr>
<td><strong>Proposed method</strong></td>
<td><strong>98.06</strong></td>
<td><strong>100.0</strong></td>
<td><strong>96.19</strong></td>
<td><strong>98.08</strong></td>
</tr>
</tbody>
</table>

4.4 Qualitative results
We conducted a qualitative experiment using graphs for each threshold, as illustrated in Figures 11, 12, and 13.

Despite the presence of various types of input images in various domains, the proposed method shows the most similar threshold to the ground truth threshold. Even though there are various types of input images in various domains, our proposed
Table 5. Experimental results of Video 2 (Dice, Precision, Recall score).

<table>
<thead>
<tr>
<th>Method</th>
<th>Dice</th>
<th>Precision</th>
<th>Recall</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>84.09</td>
<td>88.10</td>
<td>96.35</td>
<td>89.51</td>
</tr>
<tr>
<td>Average+Std</td>
<td>73.41</td>
<td>59.29</td>
<td>96.35</td>
<td>76.35</td>
</tr>
<tr>
<td>Avg+Std(30%)</td>
<td>89.15</td>
<td>82.96</td>
<td>95.35</td>
<td>89.49</td>
</tr>
<tr>
<td><strong>Proposed method (SAT)</strong></td>
<td><strong>91.66</strong></td>
<td><strong>98.21</strong></td>
<td><strong>85.94</strong></td>
<td><strong>91.94</strong></td>
</tr>
</tbody>
</table>

Figure 11. Estimated graph and thresholds of video 1. Ground truth is manually set by an editorial specialist.

Figure 12. Estimated graph and thresholds of video 2.

Method crops stable images. Figure 11 is the case that the last few seconds have extremely large movements. Due to this problem, statistical methods are highly affected by outliers. However, the proposed method accurately finds the optimal threshold than others. Figure 12 is the case that has some movements in nature. In this case, some movements are acceptable but it is ambiguous. The editorial specialist judged that only a few seconds of the middle and a few seconds of the last are stable clips. The other methods have too high a threshold than ground truth, but the proposed method has a similar threshold than others. Figure 13 is a video that has tiny movements and large movements in the last few seconds. Tiny movements are acceptable motions but the first and the last movements are not acceptable. The other methods have a relatively high threshold than the ground truth, but the proposed method has a similar threshold to the ground truth threshold. Overall, the proposed method shows a lower threshold than the others and is similar to the ground truth threshold. Therefore, experiments show that the proposed method is similar to the editorial specialist.
5 Limitation and Future work

The proposed VQA system only focused on the problems that unwanted camera motion and blurriness. Experiments with various objective metrics and visual comparisons show reasonable performance. However, there are limitations to the proposed system. First, when the motion of an object is larger than the motion of the camera, the PCA operation misclassifies an object’s motion as a camera’s motion. Second, when the video was recorded by domains that lack high-frequency components, such as the sky and underwater, the frequency domain transformed by the Fourier transform cannot capture high-frequency. As a result, the blur estimation method misclassifies the sky which has low-frequency as a blur. Third, even if we use the statistical adaptive threshold algorithm, intentional camera movement cannot judge accurately. The first and second problems are difficult to solve because there are problems with the image processing nature. However, by conducting additional research into techniques for dividing an object’s area into distinct regions, such as using object segmentation methods, it is anticipated that the initial issue will be mitigated. Of course, to adopt this method, we need to train the segmentation network using a fine annotated dataset.

6 Conclusion

This paper proposes a system for assessing the quality of videos affected by camera movement and out-of-focus blur. The system uses a deep-learning based optical-flow estimation method to estimate camera movement and the Fourier transform-based method to compute blurriness. By analyzing the CDF of the frequency domain, blurriness is estimated based on its proportion. The proposed method then computes a video adaptive threshold using statistical methods and an iterative search algorithm. Experimental results demonstrate that the proposed method outperforms existing methods in both objective and subjective quality evaluations.

References


25. Stock footage, 4k stock footage, 8k stock footage - high picture quality stock, keycutstock.


Data availability

All data generated or analyzed during this study are included in this published article.

Acknowledgements

This work was supported partly by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by Korea government (MSIT) [2021-0-01341, Artificial Intelligent Graduate School Program(Chung-Ang University)], and partly by the Institute of Civil-Military Technology Cooperation Program funded by the Defense Acquisition Program Administration and Ministry of Trade, Industry and Energy of Korean government under grant No. UM20311RD3.

Author information

Authors and Affiliations

Department of Image, Chung-Ang University, 84 Heukseok-ro, Seoul, 06974, Korea.
Donggoo Kang, Joonki Paik

Department of Artificial Intelligence, Chung-Ang University, 84 Heukseok-ro, Seoul, 06974, Korea.
Yeongjoon Kim, Sunkyu Kwon, Joonki Paik

Hyunchol Kim, Jinah Kim

Contributions

D.K. formulated the problem and designed the system. Y.K. developed the system. S.K. performed experiments. H.K. and J.K. collected the dataset and manage the project. J.P. guided the project and wrote the original draft. All authors reviewed the manuscript.

Corresponding author

Correspondence and requests for materials should be addressed to J.P
Competing interests
The authors declare no competing interests.

Additional information

Publisher's note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.