Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 8 Issue 9, September - 2021

Implementation Of Bezel Compensation Algorithms On Raspberry Pi Microcomputer And Evaluation Of Video Impairments Using Video Quality Metric

Paul, N. B.
Department of Computer Engineering,
Kaduna Polytechnic, Kaduna.
paulnaanman@kadunapolytechnic.edu.ng

Okereke, O.U.
Department of Electrical and Electronics Engineering,
Abubakar Tafawa Balewa University, Bauchi.
ouokereke@atbu.edu.ng

Omizegba, E.E.
Department of Electrical and Electronics Engineering,
Abubakar Tafawa Balewa University, Bauchi.
eeomizegba@atbu.edu.ng

Anene, E.C.
Department of Electrical and Electronics Engineering,
Abubakar Tafawa Balewa University, Bauchi.
eanene@atbu.edu.ng

***Abstract*—Video distortions such as Noise, structural deformation and video impairments are major problems associated with video wall implantation. It is difficult to make a definite conclusion on the performance of the overlay and the offset algorithms when used for bezel compensation in video walls. In this paper, these two algorithms are implemented on Raspberry pi (R- pi) microcomputer as a sever-client model connected to Liquid crystal displays (LCDs) while, video quality metric (VQM) was used for evaluation of video impairments. Four (4) videos at varying frame rates (10fps, 25fps, 30fps, and 60fps) and resolutions of 360p, 480p, 720p and 1080p were selected, processed and recorded with a remotely controlled smartphone camera (64 Megapixel), as an image acquisition device (in an air-conditioned, darkroom). Results analyzed using one-way analysis of variance (ANOVA) at 95% confidence level (α=0.05), revealed significant differences for variation in resolution; $F_{1,6} = 16.92625$, p= 0.006257 with overlay having better mean performance of $\bar{x} = 13.7$, $\sigma^2 = 0.01875$, while offset had $\bar{x} = 14.3375$, $\sigma^2 = 0.077292$. Further results also showed significant differences for frame rate variation with the overlay outperforming the offset algorithm with up to 15.25%, 4.90%, 0.15% and 0.66% for 10fps, 25 fps, 30 fps and 60 fps respectively. These results revealed the overlay has less impairments than the offset at lower frame rates and that VQM is useful for evaluating video impairments.**

Keywords—Video wall; Raspberry pi (R-pi); bezel; offset; overlay; impairments; Video Quality Metric (VQM).

1.0 INTRODUCTION

1.1 Background

The configuration of multiple display units to form a larger screen is referred to as video wall. These type of displays are required to increase pixel density per unit cost, suit available space or customize layouts, orientation, and lower cost and display multiple information simultaneously [1-5].

Liquid crystal displays (LCDs) are predominantly used to save power, space and cost. Unfortunately they are limited by type of processing hardware, algorithms and screen bezels used, thereby leading to visual discontinuity, image distortion and user distractions. [4, 6, 7].

Screen bezels are the border areas surrounding the LCDs limiting the display of image and how close LCDs monitors can be tile together to form a video wall (Fig 1).

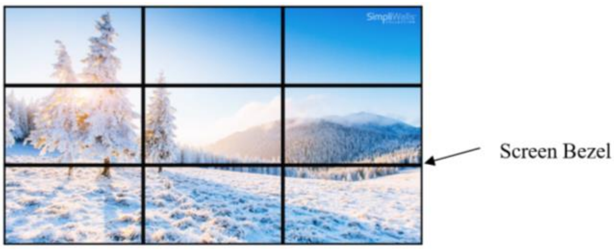


Fig 1: Bezel on LCD Based Video Wall

Bezel effects are largely tackled by physically removing plastic covers [8, 9] and using displays with smaller interior bezels [10, 11]. These approaches resulted in aesthetically pleasing displays, but limited by availability and cost of small interior bezel displays.

The alternative approach is to use bezel compensation techniques; thereby offering lower cost, increasing flexibility to use available displays (with varying bezel sizes), but with the tradeoff of visually image distortion [12].

Several researches reported varying findings associated to bezel effects, including no differences between bezel and bezel-less conditions [13-16], both positive and negative effects [17] and positive effects [8]. The inconsistency or inconclusiveness of these researches resulted in the called for a more controlled investigations [7].

With the introduction of bezel compensation (Offset and Overlay algorithms), studies such as [6, 7, 18] used available LCDs with varying bezels and applied these algorithms to explore aspects of image distortion (user distraction), when bezels are present.

Based on subjective assessment (mean opinion score (MOS)), [19] found out that the offset is better than the overlay, while [6, 20] demonstrated that the overlay is better than the offset. However [18] found that the performance of algorithm will depend on bezel size. Authors [18] concluded that these techniques, be investigated as there is no clear conclusion on the comparison of the two algorithms as the overlay approach performed poorly at a bezel of 1 cm instead of 4 cm.

In video wall several types of distortions are responsible for user distractions, some of these distortions include; noise, structural distortions and video impairment. Video impairments typically is made up of blurring [21], jerkiness and jitter [22], global noise, block distortion, and color distortion [23] and are associated to performances of displays, connections, algorithms and processing system. Particularly, when fast moving and high resolution scene are displayed.

[24] Simulated the two bezel compensation algorithms and evaluated with three objective metrics. VQM showed differences of up to 10% video impairments and recommended further investigation during development of video wall.

Based on literature, it has been observed that previous researches in this area used personal computers and subjective assessments leading to high cost and inconsistent reports. In addition, the impact of other possible causes of user distraction such as video impairment; blurring, jerkiness, global noise, block distortion, and color distortion in video wall has not been exclusively investigated.

A cost-effective, approach towards implementation of video wall is the use of microcomputers [2]. While, a consistent and control alternative that is comparable with subjective testing and useful for real-time measurement of video quality is objective quality assessment [25]. The use of objective assessment in a controlled condition has remain unexplored making it difficult to estimate the level of video impairment.

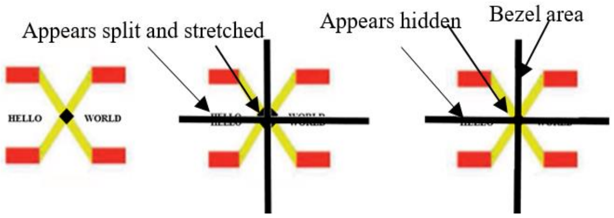
This paper therefore present a study of video impairments for fixed bezel (2 cm), 3-by-3 video wall, as a result of input video variation and bezel compensation algorithms. The study uses server client model with Raspberry pi (R-pi) microcomputers, piwall codec, 15" LCDs and 10/100Mbits switch to implement the two bezel compensation algorithms. Camera approach was used for data acquisition in a controlled environment while VQM and ANOVA for analysis. Two experiments were conducted to investigate video impairments, associated with frame rate and resolution variation when using the two algorithms.

1.2 Bezel Compensation Techniques

Bezel compensation techniques are software approaches aimed at reducing bezel effects in video wall as a result of video splitting. The two methods traditionally used to handle image distortions despite the presence of bezels includes;

(a) The offset approach (Fig 2(b)); which simply ignores the bezels and their effect on the continuousness of a scene. With this approach images are split to required forms and scaled to fit the desired displays but, image appear to be stretched.

(b) The overlay approach (Fig 2(c)); tries to compensate for the bezel problem by eliminating the images that would fall under the bezel areas. This result is an overall continuous image but with potentially important information "hidden" by the bezels [6].



(a) Original Image (b) Offset (c) Overlay.
Fig 2: Bezel Compensation in LCD Walls

1.3 R- pi (R-pi) Microcomputer

R-pi is a small size, low-cost microcomputer developed by the R-pi Foundation in the UK, the device is supported by the Raspbian operating system with variety of interfaces and multimedia codes. Some supported interfaces are as labeled in Fig 3.

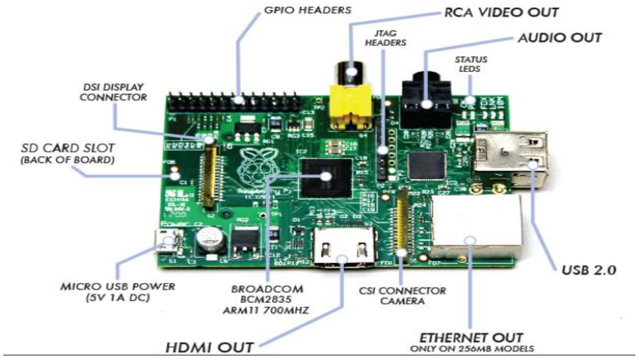


Fig 3: R-pi Microcomputer Board

R-pi also supports several programming languages ranging from C, C++, Python, Java, Ruby and Lisp. Among these, Python programming is preferred for the following reasons [26, 27]: Simpler, Portable, Fewer lines of codes, Low memory usages compare to others, Graphical User Interface (GUI), Network and internet programming.

R-pi are useful for video streaming with multimedia players such as GStreamer, FFmpeg, VLC and OMXPlayer. It has been demonstrated that it support a wide range of programming languages, has been used to demonstrate the implementation of video wall [2].

1.4 Video Quality Assessment (VQA)

Video quality assessment (VQA) is a measure of the goodness of a processed video compared to the original or similar processed video. This can be either subjective or objective assessment; the former are assumed to be consistent with actual video quality but dependent on human nature. While the latter, are based on mathematical algorithms developed to mimic the human judgment and to be consistent. Objective metrics are verifiable and applicable in applications such as monitoring, optimizing algorithms, adjusting video quality, parameter settings and benchmarking of video processing systems [28].

Video Quality Metric (VQM) is an objective metric for evaluating image quality similar to the degree of distortion seen by humans. It is useful for evaluating video impairments such as blurring, jerkiness, global noise, block distortion, and color distortion. VQM algorithm developed, has been tested and compared to subjective metrics by Video Quality Experts Group (VQEG), values found to correlate with subjective viewer ratings up to 0.9 (90%) [29 - 31].

The algorithm performs operations on the discrete cosine transform (DCT) coefficients (local contrast calculations and comparison with the contrast perception function). Values: 0 means no difference (best quality), the higher the value, the greater the difference (worst quality). This metric is based on simplified human spatial-temporal contrast sensitivity model. The model calculates distortion of a processed video in four steps:

1. For every frame the model performs DCT for 8 x 8 pixels blocks $b_i(x,y,t)$ of the original video frame p(x,y,t), (equation 1) and for blocks $b_i'(x,y,t)$ of the processed video frame p'(x,y,t) (equation 2).

$$DCTb_i(u,v,t) = DCT(b_i(x,y,t)) \quad (1)$$

$$DCTb_i'(u,v,t) = DCT(b_i'(x,y,t)) \quad (2)$$

2. The model converts DCT coefficients to Local Contrast values LC_i (u,v,t) by using DC component of each block (DC_i).

$$LC_i(u,v,t)) = \frac{DCTb_i(u,v,t).\left(\frac{DC_i}{1024}\right)^{0.65}}{DC_i} \quad (3)$$

$$DC_i = DCTb_i(0,0,t) \quad (4)$$

Similarly, $LC_i'(u,v,t)$ of the processed video is obtained.

3. The model converts $LC_i(u,v,t)$ and $LC_i'(u,v,t)$ to Just Noticeable Difference Values, $JND_i(u,v,t)$ and $JND_i(u,v,t)$, by using static and dynamic spatial contrast sensitivity function (CSF).
4. The JND coefficients of original and processed sequences are subtracted to produce a difference values $Diff_i(t)$. This model incorporates contrast masking into simple maximum ($Dist_{Max}$) operation and then weights it with the pooling mean distortion ($Dist_{Mean}$). Final VQM score is obtained by:

$$Dist_{Mean} = 1000.mean\left(mean\left(Diff_i(t)\right)\right) \quad (5)$$

$$Dist_{Max} = 1000.max\left(max\left(Diff_i(t)\right)\right) \quad (6)$$

$$VQM = Dist_{Mean} + 0.005.Dist_{Max} \quad (7)$$

1.5 Analysis of Variance (ANOVA)

Analysis of variance (ANOVA) is a statistical technique that used to verify if there are significant difference between the means of two or more groups ($\bar{x}_1, \bar{x}_2 \dots \bar{x}_k$). This is done by comparing the impact of one or more factors particularly the means of different samples for null hypothesis (H_0) or the alternative hypothesis (H_1) [32]. The null and alternative hypothesis are given as;

$$H_0 : \bar{x}_1 = \bar{x}_2 = \cdots = \bar{x}_k \ \ and$$

$$H_1 : \exists 1 \le i\ 1 \le k: \bar{x}_1 \ne \bar{x}_2$$

The null hypotheses is the condition by which all means are equal, for the alternative hypotheses there is at least one pair with unequal means. Where, $\bar{x}_i$is the mean of the group i; n_i is the number of observations of the group i; $\bar{x}$ is the overall mean; k is the number of groups; $x_{ij}$is the j[th] observational value of group i; and N is the number of all observational values.

The mean samples ($\bar{x}_i$), and grand mean of the data set ($\bar{x}$) are expressed as in equations 8 and 9.

$$\bar{x}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} x_{ij} \quad (8)$$

$$\bar{x} = \frac{1}{\mathrm{N}}\sum_{i=1}^{k}\sum_{j=1}^{n_i} x_{ij} \quad (9)$$

The sample variance (s_i^2) is given as;

$$s_i^2 = \frac{1}{(n_i - 1)}\sum_{j=1}^{n_i}(x_{ij} - \bar{x}_i)^2 \quad (10)$$

MSE is the estimate of the variance (σ^2) common to all k population. The comparison of the variation between groups (levels) and the variation within samples is carried out by analyzing their variances. Given that the sum of squares for error (or within groups) SSE, and the sum of squares for treatments (or between groups) SSC:

$$SSE = \sum_{i=1}^{k}\sum_{j=1}^{n_i}(x_{ij} - \bar{x}_i)^2 = \sum_{i=1}^{k}(n_i - 1)s_i^2 \quad (11)$$

$$SSC = \sum_{i=1}^{k}\sum_{j=1}^{n_i}(\bar{x}_i - \bar{x})^2 = \sum_{i=1}^{k} n_i(\bar{x}_i - \bar{x})^2 \quad (12)$$

The Fisher's distribution (F) statistic is the ratio of intergroup variance to intragroup variance given as;

$$F = \frac{Intergroup\ Variance}{Intragroup\ Variance} = \frac{\frac{SSC}{k-1}}{\frac{SSE}{N-k}} \quad (13)$$

$$F = \frac{\sum_{i=1}^{k} n_i(\bar{x}_i - \bar{x})^2 /(k-1)}{\sum_{i=1}^{k}(n_i - 1)s_i^2/(N-k)} \quad (14)$$

$$F = \frac{\sum_{i=1}^{k} n_i(\bar{x}_i - \bar{x})^2 /(k-1)}{\sum_{ij=1}^{n}(x_{ij} - \bar{x}_i)^2/N-k} \quad (15)$$

Under H_0 this statistic has $F(k-1, N-k)$ holds for the test criteria. $F > F_{1-\propto,\ K-1,\ N-k}$.

Where, $F_{1-\propto,\ K-1,\ N-k}$ is $(1{-}\propto)$ quantile of F-distribution with $K-1$, and $N-k$ degrees of freedom, then hypothesis H0 is rejected on significance level α [33].

In addition, the p-value shows the probability of rejection of the null hypothesis in case the null hypothesis holds. In case $p < \propto$, where α is chosen significance level, is the null hypothesis rejected with probability greater than $(1-\propto)100\%$ probability. [18, 33] used ANOVA with alpha-value of 0.05.

2.0 METHODOLOGY

2.1 Test Bed Development

To implement the 3-by-3 video wall testbed, R-pi microcomputers (model 3B+)as the server and the clients with 10/100Mbps switch and 15" computer monitors used as display1-9 as in Fig. 4.

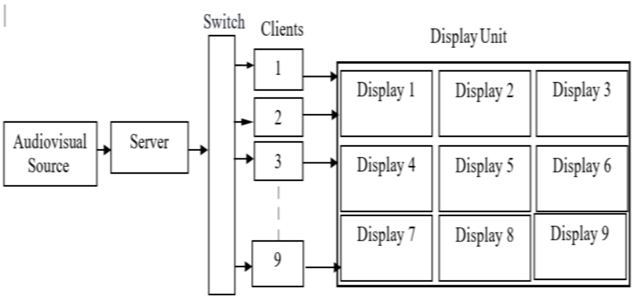


Fig. 4: Server Client Model

The computer monitors used in the work to setup the tile display unit has the properties shown in Table 1.

Table 1: Monitor Properties

Parameters	Abbreviations	Value
Height (mm)	h	230
Width (mm)	w	305
Bezel (mm)	b	10

Using Fig. 5, video splitting, bezel compensation and coordinate points for the video wall considering he two algorithms were obtained as in equations 16-23;

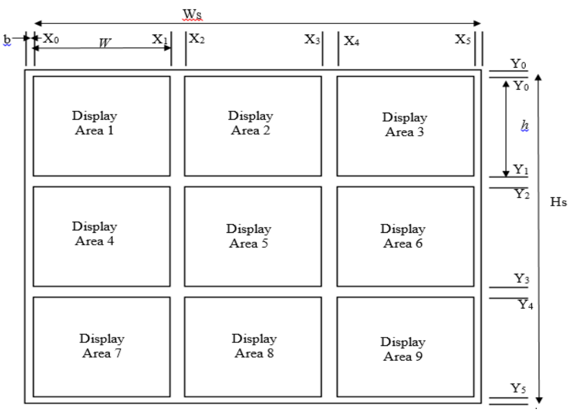


Fig. 5: 3 by 3 Video Wall Display Layout

The wall definition and the individual display definition, defined by coordinates based on the model in Fig 5 can be expressed using equations 16 and 17:

$$D_n = f[X_n, Y_n, w, h] \qquad (16)$$

$$S_W = f[X_0, Y_0, W_s, H_s] \qquad (17)$$

Where, D_n, is display definition, X_n, Y_n represent the top left *x* and *y* coordinates of the display while, w and h represent the width and height of the selected monitor respectively. *Sw*, is the wall size definition, X_0, Y_0 represent the initial *x* and *y* coordinate of the wall, while W_s and H_s the width and height of the predicted wall size respectively.

1. *For offset model*
   The video is configured to fit the wall size using Equations 18 and 19, then split (cropped to remove bezel area) using coordinate points obtained from Equation 20 and programed to operate as in (Fig 6 (a)).

$$W_s = Nw \qquad (18)$$

$$H_s = Nh \qquad (19)$$

$$f(X_n, Y_n) = \begin{cases} \frac{n}{2}w \ , \ n \ is \ even \ Nos \\ \frac{n}{2}h \ , n \ is \ even \ Nos \end{cases} \qquad (20)$$

2. *For overlay model*
   The video is configured to fit the wall size using equations 21 and 22, then split (cropped to remove bezel area) using coordinate points obtained from equation 23 and programed to operate as in Fig. 6 (b).

$$W_s = b + N(w + b) \qquad (21)$$

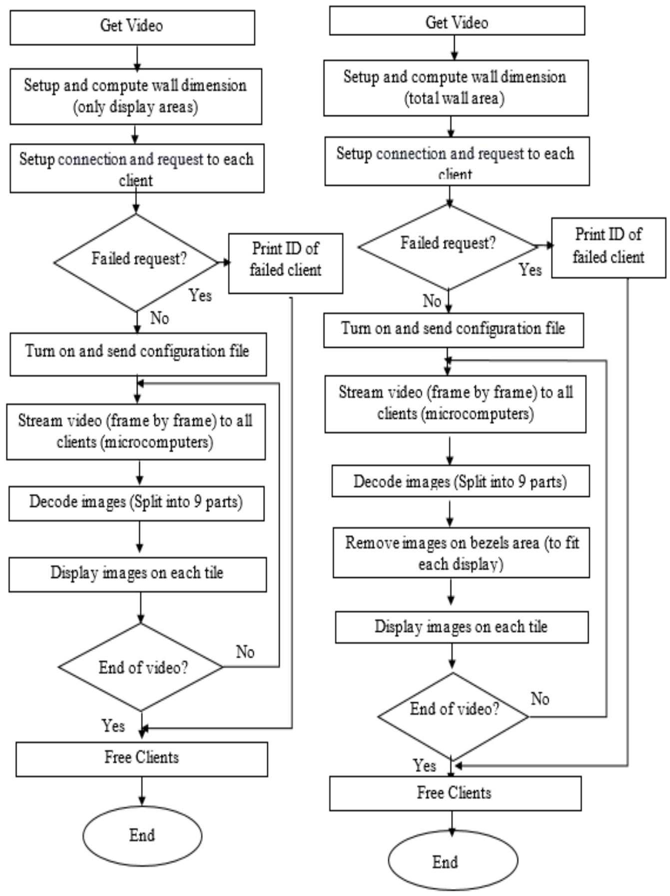
$$H_s = b + N(h + b) \qquad (22)$$

$$f(X_n, Y_n) = \begin{cases} n\left(\frac{w}{2} + b\right), n \ is \ even \\ n\left(\frac{h}{2} + b\right), n \ is \ even \end{cases} \qquad (23)$$

Where, W_s and H_s stand for width and height of entire wall, *w* or *h* for width and height of individual display area, while, *N* is the number of displays on rows or columns (3). X_n *and* Y_n represent *x* and *y* coordinates for rows and column of display from top left, and *n* is the number of coordinates point while *b* is the bezel size.

2.2 Hardware Implementation

To reduce effect of having variation on the displays as a result of aging. The 3-by-3 tile display was setup, using 15", 10 mm bezel monitors obtained from same source, with same manufacturer specifications and same batch numbers. Similarly, to reduce bezel variation during installation, all monitors were tightly aligned while, to reduce structural distortion such as contrast, luminance and texture, all parameters of the monitors were set as in Table 2.



(a) Offset (b) Overlay

Fig. 6: Bezel Compensation Algorithms models

Table 2: Monitor Parameter Settings

Parameters	Value
Brightness (%)	100
H. Position (%)	64
V. Position (%)	55
Clock	0
Phase (%)	25
Contrast	50

The tile screen configured with R- pi microcomputers based video wall was setup in an air-conditioned room 380 cm by 400 cm, and recording done under a 0 Lux (Dark room) condition, light intensity measured with a light meter at a fixed position perpendicular to the screen. Whenever the condition was switched, there was always at least 10 minutes idle period before experiment sections, this is to maximize the illuminant temporal stability. A 64 Megapixel camera was set in front of the screen to record the videos and transfer same to a Core i7 Laptop used for processing as shown in Fig. 7.

Fig. 7: Experimental Testbed and Data Acquisition setup

2.3 Software Implementation

Algorithms implemented based on equations and layout in Fig. 5 were implemented using Python codes. Piwall codecs, ffmpeg, and pwomxplayer multimedia were installed. Four (4) YouTube videos [34-37] at varying frame rates (10fps, 25fps, 30fps, and 60fps) and resolutions (360p, 480p, 720p and 1080p) downloaded using source site [38] and processed. These test videos are referred to as video 1, video 2, video 3, and video 4 as in Table 3.

Table 3: Video Parameters

Parameter	Video 1	Video 2	Video 3	Video 4
Name	TV colour bars	Profile of ATBU	Tom and Jerry _ No Way Out	60fps 4K-Official Blender
Frame Rate (fps)	10	25	30	60
Duration	00.00.10	00.08.09	00.04.59	00.10.34

Videos is to reflect wide sources of videos such as 360p; smart phones, and mobile devices. 480p used for DVD, laptop and smaller TVs. 720p; True high-definition (HD) television channels broadcast. 1080p; (Full HD) used in television stations and shared on social media, for showing on larger screens and TVs.

For comparison purpose, a similar sized television was used to display content of all bezel-less videos. Both videos with bezel and bezel-less videos were recorded with a High definition smart phone camera (64 Mega pixel) under dark room condition.

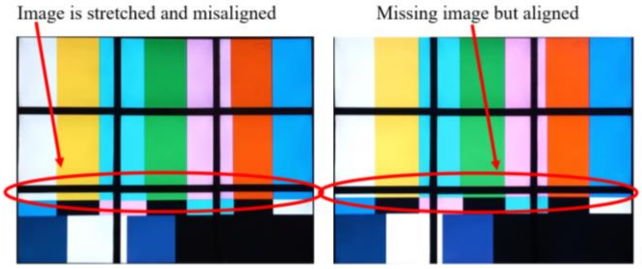
To reduce the influence of surrounding or background on the estimated video quality, all recorded videos were processed with python scripts trimming unwanted background and frames to generate a cropped version. The cropped version contains only the actual image content without surround or background, and frames recorded before or after the end of video.

Finally, resulting videos (bezels and bezel-less) were compared (evaluated) for video impairment using Moscow State University (MSU) VQMT [39] and analyzed with ANOVA.

3 RESULTS

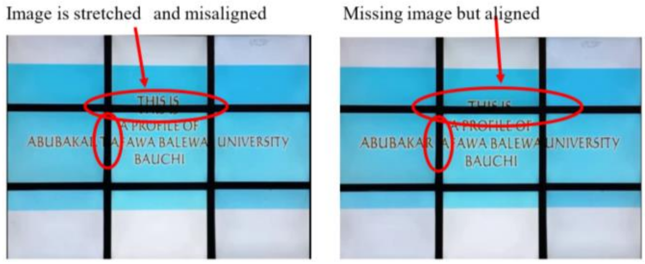
3.1 Implementation

The screenshots of processed video using the two approaches are as shown in Figs 8 to 11.



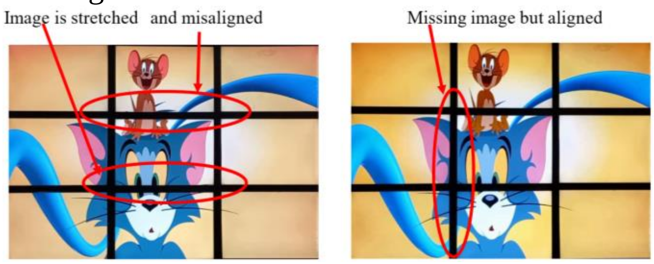
(a) Offset (b) Overlay

Fig. 8: Screenshots of Processed Video 1



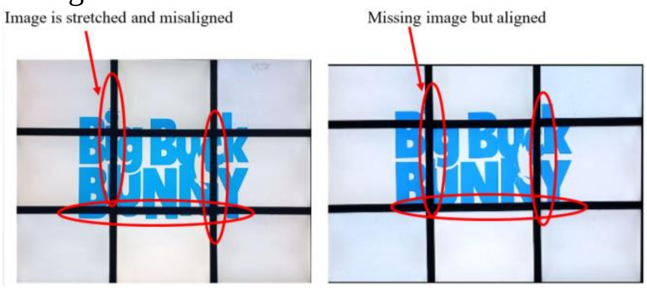
(a) Offset (b) Overlay

Fig. 9: Screenshots of Processed Video 2



(a) Offset (b) Overlay

Fig/ 10: Screenshots of Processed Video 3



(a) Offset (b) Overlay

Fig. 11: Screenshots of Processed Video 4

3.2 Video Impairment Evaluation

The MSU software was used to measure VQM values in a video sequence. Average metric values (Avg) for videos with varying frame rate and resolution were recorded and plot as shown in Figs 12, to 17. For the two algorithms, the four videos with fixed frame rate but varying resolutions are here referred to as “10 fps, 25 fps, 30 fps and 60 fps” (Figs 12 to 14) while varying frame rate the videos are here referred to as “360p, 480p, 720p and 1080p” (Figs 15 to 17). For comparison, all processed videos are recorded using same file format (mp 4). For interpretation of each parameter measured, the referenced values (best quality) of the metrics VQM = 0 [30].

Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 8 Issue 9, September - 2021

3.2.1 Effect of Resolution Variation

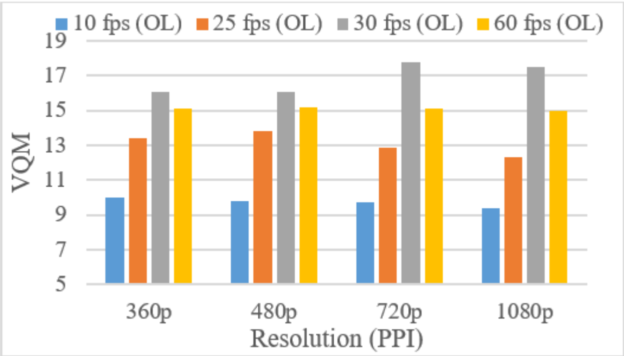


Fig. 12: Resolution Impairment with Overlay

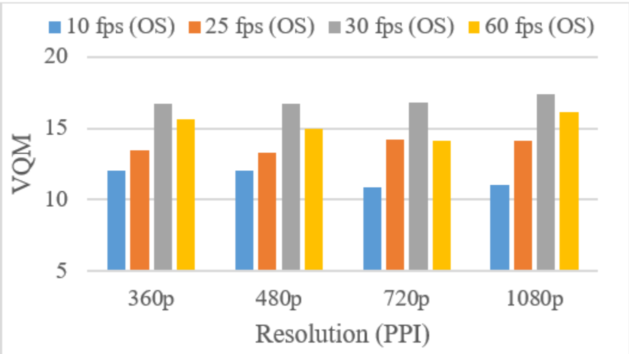


Fig. 13: Resolution Impairment with Offset

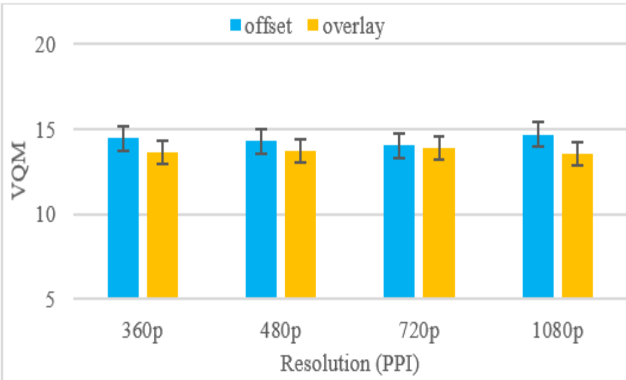


Fig. 14: Mean Resolution Impairment of Videos.

3.2.2 Effect of Frame Rate Variation

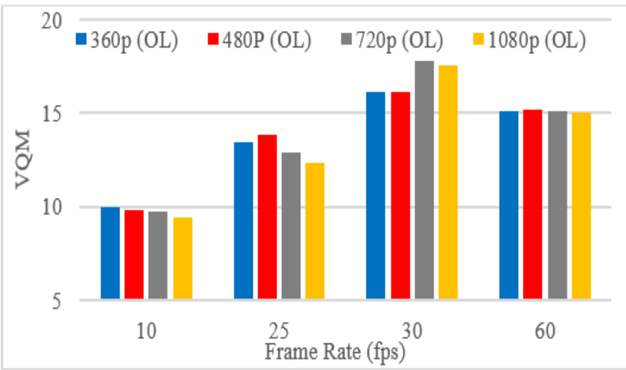


Fig. 15: Frame Rate Impairment with Overlay

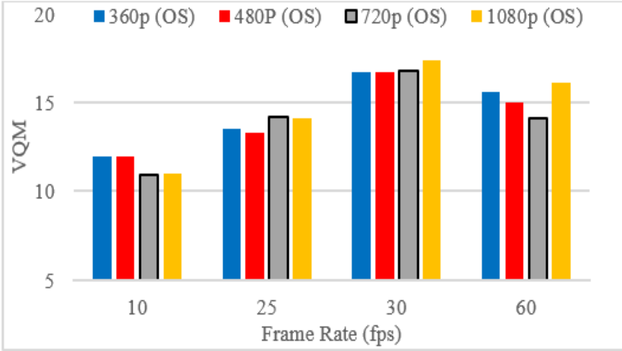


Fig. 16: Frame Rate Impairment with Offset

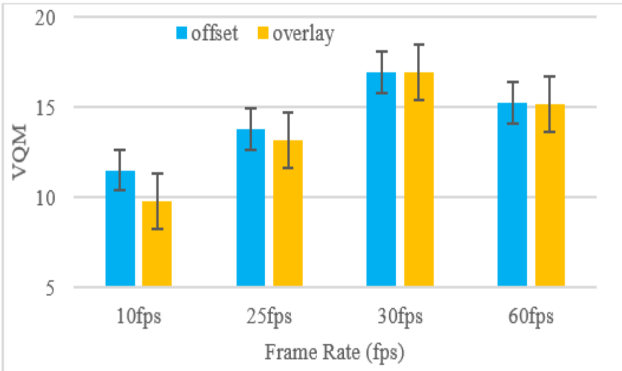


Fig 17: Mean Frame Rate Impairment of Videos.

4 DISCUSSION OF RESULTS

Figs 8 to 11 show how videos are split into nine segments using algorithms, Figs 8(a) to 11(a) show the offset approach, these show that when a video is spitted into nine (9) different non-overlapping segment and displayed on the tiled display, the image will be stretched and or misaligned, while the offset approach in Figs 8(b) to 11(b) show that images are aligned and not stretched but information are hidden.

Effects of variation in resolution Fig. 12 shows that as the resolution changes the video impairments changes is small, In using ANOVA for Fig. 12, analysis of VQM metrics reveals F < F_{crit} and p > α ($F_{3, 12}$ = 0.007726, p= 0.999013, F_{crit} = 3.490295) hence, insignificant differences in video impairment; irrespective of resolution; with best VQM ($\bar{x}$ = 13.55, σ^2 = 12.16333), at 1080p, worst of ($\bar{x}$ = 13.875, σ^2 =11.7625) at 720p for overlay. Similarly, for offset Fig. 13 shows F < F_{crit} and p > α ($F_{3, 12}$ =0.055697, p=0.981884, F_{crit} = 3.490295) with best VQM ($\bar{x}$ = 14, σ^2 = 5.833333), at 720p, worst of $\bar{x}$ = 14.65, σ^2 = 7.763333 at 1080p.

Comparing the two algorithms for variation in resolution Fig. 14 shows $F_{1, 6}$ =

16.92625, p= 0.006257, F_{crit} = 5.987378 (F > F_{crit} and p < α) hence, there is significant differences for use of algorithms, with overlay having better performance of $\bar{x}$ = 13.7, σ^2 = 0.01875, while offset had $\bar{x}$ = 14.3375, σ^2 = 0.077292.

Effect of frame rate variation Fig. 15 shows F > F_{crit} and p < α (where, α =0.05; 95% confidence level) hence, there is significant effect for frame rate variation with VQM; for overlay $F_{3, 12}$ =115.2542, p=4.12E-09, F_{crit} = 3.490295 having 10 fps video VQM ($\bar{x}$ = 9.725, σ^2 = 0.0625), and 25 fps video ($\bar{x}$ = 13.1, σ^2 =0.42) while, the 30 fps and 60 fps videos have VQM ($\bar{x}$ = 16.875, σ^2 =0.815833), and ($\bar{x}$ = 15.1, σ^2=0.006667) respectively. For offset Fig. 16 shows $F_{3, 12}$ = 59.49001, p= 1.78E-07, F_{crit} = 3.490295 with 10 fps video having best VQM ($\bar{x}$ = 11.475, σ^2 = 0.369167), while 25 fps, 30 fps, and 60fps videos had ($\bar{x}$ = 13.775, σ^2 =0.195833), ($\bar{x}$ = 16.9, σ^2 =0.113333) and ($\bar{x}$ = 15.2, σ^2= 0.74) conditions.

Comparing the two algorithms for variation in frame rate Fig. 17 shows $F_{1, 6}$ = 0.110783, p= 0.750576, F_{crit} = 5.987378 (F < F_{crit} and p > α) hence, there is insignificant differences for use of algorithms, though the overlay has better performance ($\bar{x}$ = 13.7, σ^2 = 9.400417) while offset had $\bar{x}$ = 14.3375, σ^2 = 5.273542.

5 CONCLUSIONS

In this paper a 3-by-3 video wall with overlay and offset bezel compensation algorithms has been implemented on R- pi microcomputers and video impairments evaluated with VQM while ANOVA has been used to analyze the evaluated results. Results showed, the offset introduces video distortion such as misalignments and stretching while, the overlay approach introduces missing contents. However, the use of VQM showed the overlay outperformed the offset algorithm with differences of up to 15.25%, 4.90%, 0.15% and 0.66% for 10 fps, 25 fps, 30 fps and 60 fps videos respectively. Also, with up to 5.54%, 3.68%, 0.89%, and 7.51% differences for 360p, 480p, 720p and 1080p videos respectively.

This showed less video impairments (less user distractions) when overlay is used while resolution changes. In addition, the use of any compensation technique alone showed significant difference in use of video with different frame rate. However comparing the two algorithms based on frame rate changes, there are less differences in video impairments. This paper has shown that the proposed approach of using VQM to evaluate the video impairment with bezel compensation algorithms can be a useful alternative to the use of subjective assessment for video wall development.

REFERENCES

[1] K. Ogawa and R. Mitsuhashi (2011). Development of a Video Wall Display System Using Ultrathin-Bezel LCD Panels. Imaging and Recognition Solutions, General Papers, *Mitsuhashi Renichi Nec Technical Journal. 6 (3)*: 84-88.

[2] G.G. Ramón and C.V. Rodríguez (2014). *Implementation of a Low Cost Video Wall using R-pi devices*, Master Thesis, Master of Science in Telecommunication Engineering and Management, Universitat Politechnica De Calalunya.

[3] R. Bundulis and G. Arnicans (2014). Concept of Virtual Machine Based High Resolution Display Wall, *Information, Electronic and Electrical Engineering*, pp. 1-6, Nov 2014.

[4] C. Papadopoulos., K. Petkov, A. E. Kaufman, and K. Mueller (2014). The Reality Deck--an Immersive GigGUIxel Display. *IEEE computer graphics and* applications, 35 (1): 33-45.

[5] S.D. Kim and S.E. Lee (2015). Architecture for a Zero Client Based Scalable Large Format Display System, *International Journal of Computer and Information Technology.04 (03)*

[6] A.Ebert, T .Sebastian, O. Peter-Scott, M. Joerg and H. Hans (2010). Tiled++: An Enhanced Tiled Hi-Res Display Wall, *IEEE Transactions on Visualization and Computer Graphics. 16 (1)*: 120-132

[7] X. Bi, S.-H. Bae and R. Balakrishnan, (2010). *Effects of interior bezels of tiled-monitor large displays on visual search, tunnel steering, and target selection*. In Proc. CHI 2010, CHI '10, ACM (2010), 65–74.

[8] R. Ball, M.Varghese, B. Carstensen, E. D. Cox., C. Fierer, M. Peterson. and C. North (2005). *Evaluating the Benefits of Tiled Displays for Navigating Maps*," Proc. Int'l

Assoc. Science and Technology for Development (IASTED) Int'l Conf. Human-Computer Interaction. pp 66–71.

[9] T. A. DeFanti., J. Leigh, L. Renambot, B. Jeong, A. Verlo, L. Long,, … and L. Smarr (2009). *The OptIPortal, a scalable visualization, storage, and computing interface device for the OptiPuter. Future Generation Computer Systems, 25(2), 114–123.* doi:10.1016/j.future.2008.06.016

[10] Leyard and Planar (2018). ClarityMatrix G3 LCD Video Wall System, *clarity-matrix-g3-lcd-video-wall-system-brochure_5-18,* Leyard Optoelectronics Co., Ltd. and Planar Systems, Inc.

[11] M. Saleem, H. E.Valle, S. Brown, V.I. Winters, and A. Mahmood (2018). The Hiperwall tiled-display wall system for Big-Data research. *Journal of Big Data*. 5(41) https://doi.org/10.1186/s40537-018-0150-7

[12] A. Bezerianos, and P. Isenberg, (2012). Perception of visual variables on tiled wall-sized displays for information visualization applications. *IEEE Transactions on Visualization and Computer Graphics 18(12)*.

[13] D. S. Tan, and M. Czerwinski, (2003). Effects of visual separation and physical discontinuities when distributing information across multiple displays. Human-Computer Interaction -- Interact'03, M. Rauterberg et al. (Eds.). IOS Press, (c) IFIP, 2003, pp. 252-255

[14] X.-D. Yang,., E. Mak, D. McCallum, P. Irani, X. Cao, and S. Izadi (2010). Lensmouse: augmenting the mouse with an interactive touch display. In Proc. CHI 2010, 2431–2440.

[15] C. S. Campbell, and P. P. Maglio, (2003). Segmentation of display space interferes with multitasking. In Proc Human Computer Interaction –Interact'03, Rauterberg et al. Eds., IOS Press Publ., IFIP 575–582.

[16] A. M. McNamara, F. Parke, and M. Sanford, (2011). Evaluating performance in tiled displays: navigation and wayfinding. In Proc. VRCAI 2011, 483–490.

[17] G. Robertson, M. Czerwinski, P. Baudisch, B. Meyers, D. Robbins, G. Smith, and D. Tan, (2005). The large-display user experience. Computer Graphics and Applications, IEEE 25, 4 (2005), 44–51.

[18] G. Smith, R. J.Teather, J. Lass, and J. Carette, (2015). *Effects of interior bezel size and configuration on gaming performance with large tiled displays*, 2015 IEEE Games Entertainment Media Conference (GEM), Toronto, ON, 2015, pp. 1-8.

[19] R. A.de Almeida, C. Pillias, E. Pietriga, and P. Cubaud, (2012). Looking behind bezels: french windows for wall displays. In Proc AVI 2012, 124–131.

[20] S. Lee, H. Kim, Y.-k Lee, M. Sim, and K.-p. Lee, (2011). Designing of an effective monitor partitioning system with adjustable virtual bezel. In Human Centered Design, M. Kurosu, Ed., vol. 6776. Springer Berlin Heidelberg, 2011, 537–546.

[21] Chan, S. H., and Nguyen, T. Q. (2011). *LCD Motion Blur: Modeling, Analysis, and Algorithm. IEEE Transactions on Image Processing, 20(8), 2352–2365.* doi:10.1109/tip.2011.2109728

[22] Q. Huynh-Thu, and M. Ghanbari (2006). Impact of Jitter and Jerkiness on Perceived Video Quality. Proc. Workshop on Video Processing and Quality Metrics, 2006. https://www.researchgate.net/publication/236882981

[23] X. Yanhui, W. Zhengyou, W. Wan, W. Jin, and W. Zheng (2013). Color Distortion of Digital Image and its Detection. Telkomnika, Vol. 11, No. 8, August 2013, pp. 4565~4571 e-ISSN: 2087-278X

[24] N. B. Paul, O. U Okereke, E. E. Omizegba, and E. C. Anene, (2021). Comparative Study of Overlay and Offset Algorithms in 3-By-3 Video Wall using Objective Metrics. Journal of Multidisciplinary Engineering Science and Technology (JMEST) ISSN: 2458-9403 Vol. 8 Issue 7, July – 2021

[25] E. Alkhowaiter, I. Alsukayti, and M. Alreshoodi (2021). Developing a Quality Prediction Model for Wireless Video Streaming Using Machine Learning Techniques. IJCSNS International Journal of Computer Science and Network Security, VOL.21 No.3, March, 2021. https://doi.org/10.22937/IJCSNS.2021.21.3.31

[26] K. R. Srinath (2017). Python – The Fastest Growing Programming Language. International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056

Volume: 04 Issue: 12 | Dec-2017 www.irjet.net p-ISSN: 2395-0072.

[27] M. Rowe (2018). IEEE Survey Ranks Programming Languages. IEEETimes. DesignLines Test and Measurement Designline. https://www.eetimes.com/document.asp?doc

[28] S. Karim, H. He, A. R. Junejo and M. Sattar (2020). Measurement of Objective Video Quality in Social Cloud Based on Reference Metric, *Hindawi Wireless Communications and Mobile Computing Volume 2020.*

[29] M. Vranješ, S. Rimac-Drlje, and D. Žagar (2007). Objective Video Quality Metrics, *ELMAR 2007, Zadar, 2007, pp. 45-49, doi:10.1109/ELMAR.2007.4418797 https://ieeexplore.ieee.org/document/4418797*

[30] M.O. Martínez-Rach, P. Piñol, O. M. López, M. P. Malumbres, J. Oliver, and C.T. Calafate (2014). On the Performance of Video Quality Assessment Metrics under Different Compression and Packet Loss Scenarios. *Hindawi Publishing Corporation* *□e Scientific World Journal Volume 2014, Article ID 743604, 18 pages http://dx.doi.org/10.1155/2014/743604*

[31] S. Akramullah (2014) Video Quality Metrics. In: Digital Video Concepts, Methods, and Metrics. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4302-6713-3_4

[32] E. Ostertagova, and O. Ostertag, (2013). Methodology and Application of One-way ANOVA. American Journal of Mechanical Engineering. 1. 256-261. 10.12691/ajme-1-7-21.

[33] D.C. Montgomery, and G.C. Runger, (2003). Applied Statistics and Probability for Engineers**, John Wiley and Sons, 2003.**

[34] YouTube (2017).Tom and Jerry | No Way Out | Boomerang Africa - Sunday Morning Shake Up. https://www.youtube.com/watch?v=bShpwwJR3Tg. Downloaded 29th may 2021

[35] YouTube (2017).SMPTE Color bar. https://www.youtube.com/watch?v=A3_88dy1SRk. Downloaded 29th may 2021

[36] YouTube (2017). PROFILE OF ATBU https://www.youtube.com/watch?v=TYoKzq8kUZk, Downloaded 29th may 2021.

[37] YouTube (2014). Big Buck Bunny 60fps 4K - Official Blender Foundation Short Film https://www.youtube.com/watch?v=aqz-KE-bpKQ. Downloaded 29th may 2021

[38]Tube offline (2020). https://www.tubeoffline.com/download-youtube-videos.php

[39] MSU Quality Measurement Tool: Download Page (2020)