Flickers Forecasting In CRT Using Stochastic Analysis

A. A. KHAN, S. SOOMRO*, A. G. MEMON**

Institute of Business Management, Pakistan
Email: safeullah.soomro@biztekian.com; ghafoor@usindh.edu.pk

Received 18th January 2012 and Revised 11th June 2012

Abstract: Videos are composed of sequence of interrelated frames. There is a minute difference among frames. Flicker is an error which is found in every video. It is like a checker box in a video, there are several reasons behind flickers generation, one of the main reasons is refresh rate of the monitor and second reason is number of frames per second in a video. The main objective of this study is to propose and develop a framework that identifies flicker location and minimizes the flickers rate. Analysis shows that flickers can be minimize by adjusting the persistence of pixel and higher refresh rate of CRT monitor. Further we have compared different isotopes of phosphorous pixels and generate its graphs. This paper highlighted the cause of flicker and its avoidance. Statistical research proves that proposed algorithm improves the video quality and reduce flickers ratio up to 90%.

Keywords: Flicker, CRT, Stochastic

1. INTRODUCTION

Series of individual images in a movie clip called frames. When these images are shown rapidly in succession, a viewer has the illusion that motion is occurring. The viewer cannot observe the flickering in a movie clip due frame continuation is known as persistence of vision. It is due to eye retains on a visual image for a fraction of a second after the source has been removed. While watching a movie observer noticed unwanted checker boxes when a movie scene changed this can be called as Flickers. These flickers are due to abrupt variation of intensities in the phosphorous pixels Yoon, (2011). Flicker forecasting and avoidance can improve the video quality and provides flicker free environment. It is difficult to calculate abruption in pixels but can be forecast by stochastic method. This process contains family of random variables which is indexed by the parameter “t” and defined on a common probability space \((\Omega, F, P)\) whereas \(t, w, \Omega, Xt (w)\) is a random variable. For each \(w, T, Xt (w)\) called a sample function or realization of the process. The result shows that intensity barrier is one of the main causes behind this error, by comparing two pixels of two separate frames shows the intensity gap. The intensity gap among two pixels can easily determine by probability of selected pixels. Hence by comparing the pixels intensities it is easy to calculate flickers. The analysis shows the flicker will emerge in a clip if probability among pixels is above 0.05.

In order to reduce flicker we add a new pixel among higher probability pixels. This additional pixel contains the mean pixel values of consecutive pixels that can reduce the luminosity of a flicker. On fast motion scenes, a variable bit rate uses more bits than it does on slow motion scenes of similar duration yet achieves a consistent visual quality. High definition video transmission, low bandwidth can also produce flicker or checkers. Human Eye cannot notice it due to persistence of vision and beta movement (Farrell, et al., 1987)

On the other hand, the Flat-panel liquid crystal display (LCD) monitors do not suffer from flicker even if its refresh rate is 60 Hz or even lower. This is because LCD pixels open to allow a continuous stream of light to pass through until instructed by the video signal to produce a darker color. In contrary monitors are composed of different types of phosphorous isotopes mainly P31, BARCO D65 P4, DP104 or RGB P22 etc. Further by comparing these pixel’s types it will be easy to get the best flicker regression element. In other words it can be stated as flicker rate is equal to flicker regression of the phosphorous pixels with pixel luminance energy decay time (Najim, et al., 2006).
Flicker rate = \frac{\text{Flicker regression of the phosphorous pixel}}{\text{Pixel luminance decay time}}

2. MATERIAL AND METHODS

System Design Platform:
Our aim is to decrease screen luminance energy until perceived flicker disappeared then increase screen luminance energy until perceived flicker reappears.

INITIALIZE:
LOAD: Import a scene
SET: Frame[a] = 0;
LOOP: If F[a] ≤ F[n], F[a] ;then Increment F[b] = F[a] + 1
COMPARE: F[a] with F[b]
CONVERSION: F[a]_image → F[a]_matrix ; F[b]_image → F[b]_matrix
CALCULATE: Probability := Prob{F[a] (m*n)_j, F[b] (m*n)_i} \ldots i=pixel location
LOOP: Probability ≥ 0.05
SPLIT F[a]_Red → F[a]_Red | F[a]_Green | F[a]_Blue
SPLIT F[b]_Red → F[b]_Red | F[b]_Green | F[b]_Blue
Mean (F[a], F[b])_Red, Mean (F[a], F[b])_Green, Mean (F[a], F[b])_Blue.
New Pixel color := Add (Mean (F[a], F[b])_Red, Mean (F[a], F[b])_Green, Mean (F[a], F[b])_Blue.)
INSERT (New Pixel color) := (F[a]_i, F[b]_j)
END

Fig. 1. Proposed Algorithm

There are two main loops in this algorithm one is for increment in frames and second one is for probability calculation (see Table 3 to 8). Software determines the need of mean pixel insertion. It segregates image into three colors then calculate mean value of pixels and insert it if it is required.

1. The System Design Tools:

Noise is a universal entity which is found everywhere it can easily been observed in videos as well. Unfortunately when flicker or unwanted noise appears between two frames, it is because of abrupt changes in luminance; refresh frequencies, screen luminance in terms of wave length, refresh rate of screen phosphors, and the difference between viewer and Cathode Ray Tube CRT. Consequently flicker emerges in a film; in this context we have calculated the probability of two pixels of the two different frames. Hence research shows that if the probability is greater than 0.05 than flicker emerges. Therefore proposed algorithm uses stochastic theorem to forecast the flicker and thus can reduce the flicker luminosity. Software calculates the pixels intensities and updates the flicker pixel with it. There are total 8 different images or 4 different pairs will be used for this experiment. The name of software is “flicker” which is developed in Matlab version seven releases 14, this “m” file calculates flickers and show results.

1.1 Flickers reduction Electronically

Image in a video contains series of luminance variation with exponentially decaying persistence of any pixel. If we can compute the amplitude coefficient of the fundamental frequency of the time-varying screen luminance (as per Oppenheim & Willsky, 1983) has mentioned. Amplitude coefficient is the formula that can compare the existing Phosphorous pixels. Formula of amplitude coefficients is as follows.

\[ \text{AmpCoeff}(F) = \frac{2}{\left[1 + (\infty \omega)^2\right]^{1/2}} \]

where as \( \alpha \) is the decaying or persisting time of a phosphor pixel ‘\( \omega \)’ is ’2\( \Pi \)” and “\( f \)” is the refresh rate of any monitor. Willsky further explain the degree of visual of angles which is 2\( \tan^{-1} \left( \frac{D}{2V} \right) \), whereas D is related with the display area measured in millimeters and V = 1.5
feet or 500 mm. Forth coming graphs provides a good summarized version of this paper (Donovan 2011).

1.2 The Flickers identification

White light is composed of seven colors. It can be segregated into seven different colors. There are total eight colors in Matlab programming list as follows. [6] as shown in (Table 1).

Table 1. Example of color computerized codes

| Color Name | Symbol | Computer Code |
|------------|--------|---------------|
| Yellow     | Y      | [1 1 0]       |
| Magenta    | M      | [1 0 1]       |
| Cyan       | C      | [0 1 1]       |
| Red        | R      | [1 0 0]       |
| Green      | G      | [0 1 0]       |
| Blue       | B      | [0 0 1]       |
| White      | W      | [1 1 1]       |
| Black      | K      | [0 0 0]       |

There are two frames from a movie that contains flickers (see Fig. 2). After image enhancement flickers are more visible to the human eye (see Fig. 3 and 4).

2. The Development of Proposed Framework:

Matlab is the main tool for programming and simulations. The following is the data flow diagram that elaborates steps by step working.

Fig. 4. Data flow diagram of proposed frame work

This framework helps in the identification process of flickers and its reduction. Software imports a video clip; convert each frame into separate red, green, blue signals. It determines the probability and test results. If probability of selected pixel value is above 0.05 it updates it with new one.

1.3 The Statistical Model:

Software identifies those pixels whose probability value is greater than 0.05. A real scenario is considered for step-wise development of proposed frame work. This real case information is presented in following (see Fig. 5 to 11).
| Actual | Relation between colors(R[a] with R[b]) |   |   |   |   |   |   |
|--------|---------------------------------------|---|---|---|---|---|---|
|        | Black | Blue | Green | Cyan | Red | Magenta | Yellow | White |
| Black  | 0     | 0.5  | 1     | 1.5  | 2   | 2.5     | 3      | 3.5   |
| Blue   | 0.5   | 0    | 1.5   | 2    | 2.5 | 3       | 3.5    | 4     |
| Green  | 1     | 1.5  | 0     | 2.5  | 3   | 3.5     | 4      | 4.5   |
| Cyan   | 1.5   | 2    | 2.5   | 0    | 3.5 | 4       | 4.5    | 5     |
| Red    | 2     | 2.5  | 3     | 3.5  | 0   | 4.5     | 5      | 5.5   |
| Magenta| 2.5   | 3    | 3.5   | 4    | 4.5 | 0       | 5.5    | 6     |
| Yellow | 3     | 3.5  | 4     | 4.5  | 5   | 5.5     | 0      | 6.5   |
| White  | 3.5   | 4    | 4.5   | 5    | 5.5 | 0       | 6.5    | 0     |

Fig. 5. Relationship among pixels.

| Stocknetic Matrix(Colom wise) | Probability see paper |
|------------------------------|-----------------------|
| Black | Blue | Green | Cyan | Red | Magenta | Yellow | White |
| Black | 0    | 0.029412 | 0.05 | 0.065217 | 0.076923 | 0.086207 | 0.09375 | 0.1  |
| Blue  | 0.035714 | 0    | 0.075 | 0.086857 | 0.096154 | 0.103448 | 0.109375 | 0.114286 |
| Green | 0.071429 | 0.088235 | 0    | 0.108969 | 0.115385 | 0.12069 | 0.125 | 0.128571 |
| Cyan  | 0.107143 | 0.117647 | 0.125 | 0    | 0.134615 | 0.137931 | 0.140625 | 0.142557 |
| Red   | 0.142857 | 0.147059 | 0.15 | 0.152174 | 0    | 0.152172 | 0.15625 | 0.157143 |
| Magenta | 0.178571 | 0.17947 | 0.175 | 0.173913 | 0.173077 | 0    | 0.171675 | 0.171429 |
| Yellow | 0.214286 | 0.205882 | 0.2 | 0.195052 | 0.192308 | 0.189035 | 0    | 0.185714 |
| White  | 0.25 | 0.235294 | 0.225 | 0.217391 | 0.211538 | 0.206857 | 0.203125 | 0    |
| Sum   | 1    | 1     | 1     | 1     | 1     | 1     | 1     | 1    |
| Mean  | 0.125 | 0.125 | 0.125 | 0.125 | 0.125 | 0.125 | 0.125 | 0.125 |
| Variance | 0.000090 | 0.000001 | 0.000513 | 0.0004096 | 0.004101 | 0.003057 | 0.003296 | 0.002946 |
| S. Dev | 0.081832 | 0.077466 | 0.072887 | 0.06853 | 0.064502 | 0.060805 | 0.05741 | 0.054281 |

Fig. 6. The Stochastic Matrix column wise.

| Z-Values | Black | Blue | Green | Cyan | Red | Magenta | Yellow | White |
|----------|-------|------|-------|------|-----|---------|--------|-------|
| Black    | -1.52753 | -1.23391 | -1.02899 | -0.87236 | -0.74530 | -0.03799 | -0.54433 | -0.46057 |
| Blue     | -1.09109 | -1.61357 | -0.88599 | -0.55514 | -0.44721 | -0.35444 | -0.27217 | -0.19739 |
| Green    | -0.65465 | -0.47458 | -1.71499 | -0.23792 | -0.14907 | -0.07089 | 0 | 0.05675 |
| Cyan     | -2.18222 | -0.04492 | 1.9616 | -1.82402 | 0.14071 | 0.21564 | 0.272165 | 0.328976 |
| Red      | 0.218218 | 0.284747 | 0.342997 | 0.39952 | -1.93793 | 0.496217 | 0.544331 | 0.592157 |
| Magenta  | 0.054024 | 0.064411 | 0.083994 | 0.712749 | 0.742336 | -2.035576 | 0.041997 | 0.025337 |
| Yellow   | 1.091089 | 1.0404074 | 1.028992 | 1.030667 | 1.042496 | 1.063232 | -2.1732 | 1.118518 |
| White    | 1.527525 | 1.423737 | 1.371989 | 1.341888 | 1.341614 | 1.306874 | 1.360828 | -2.30283 |
| Sum      | 0    | 0    | 0    | 0    | 0    | 2E-15 | 0    | 0    |
| Mean     | 0    | 0    | 0    | 0    | 0    | 2.5E-16 | 0    | 0    |
| Variance | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    |
| S. Dev   | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    |

Fig. 7. The ‘Z’ values.
Software imports video clip, identifies flickers by calculating probability, generates error frame and develops reconstructed image. The proposed algorithm develops a new frame called reconstructed frame that will be inserted between two frames and produces flicker-free video clip. The following result demonstrates flickers from a scene and related percentage (See Fig.).
1.4 Electronically Evaluation of phosphorous pixels:

The forthcoming result indicates the possibility of flicker prediction in any display terminal. Proposed algorithm proves the possibility flicker-free video. Phosphorous pixel P31, DP104 provides 90% luminance persistence. The flicker-free predictions will vary as a function of display phosphor persistence; refresh frequency, luminance, and display size www.wikipedia.com/stochastic

Graph 1 shows the relationship between monitor resolution and degree of visual angle. The visual angle of an image is increased by enhancing the monitor resolution. Graph2 and 3 shows the relationship between amplitude coefficient of D65, P31 and DP104 with its refresh rate. You can see that DP104 or graph 3 provides the better result among all. It means it is clear that it got higher refresh rate and less phosphorous persistence.

5. CONCLUSION

Monitor is composed of fast displaying phosphorus. It has relatively large amplitude coefficients which require less energy or relatively less screen luminance to be seen. This problem can be addressed by two methods one is stochastic theorem and second is Eriksson and Backstrom method of predicting flickers. The fast displays phosphors DP104 appears to be a less flicker at low luminance due to its short decay constant. The screen luminance for perceived flicker is proportional to the decay constant of the phosphor. Flicker prediction lies in stimulus energy which is the fundamental frequency of the temporally modulated screen luminance. Second method is related with stochastic theorem for flicker prediction in a movie. If a movie clip contains undesired noise which reflects the pixel transition between two frames are not smooth. The size of the image A is 998.10 K Byte and the size of the image B is 998.10 K Byte which is shown in fig.5. Here two different frames A and frame B are used (fig 2) then algorithm identifies flickers using stochastic model and shows almost 6% of flickers are there in a scene. The reconstructed frame D is an outcome of initial image A and the flicker frame (the image D) see Figure 4. Hence it reduces the flicker and provides smooth frame transition for high definition videos. Despite of flicker reduction this algorithm compresses frames and provides a new dimension for low bandwidth HD video transmission.

REFERENCES:
Donovan P. O’, (2011) Aaron Hertzmann, “AniPaint: Interactive Painterly Animation from Video”, IEEE Transactions on Visualization and Computer Graphics, March 2011 IEEE.

Farrell, J. E., B.L. Benson, and C. R. Haynie, (1987) "PREDICTING FLICKER THRESHOLDS FOR VIDEO DISPLAY TERMINALS", Hewlett Packard, Pa10 Alto, CA, Proceedings of the SID, Vol. (28/4): 1987.

Hsueh, Y. H., Wei-Da Chen, (2009) “An Implementation Frame Rate Control for Reducing Flicker in CSTN Panel”, 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009 IEEE.

Najim, K., E. Ikonen and A.K. Daoud (2006) "Stochastic Processes Estimation, Optimization & Analysis" printed by KOGAN PAGE SCIENCE London and Sterling, VA,2006.

www.wikipedia.com/stochastic

Yoon, J. (2011) In-Kwon Lee , Henry Kang, “Video Painting Based on a Stabilized Time-Varying Flow Field”, IEEE Transactions on Visualization and Computer Graphics, March 2011 IEEE.
