Flicker Reduction in LED-LCDs with Local Backlight

Nadernejad, Ehsan; Mantel, Claire; Burini, Nino; Forchhammer, Søren

Published in:
2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)

Link to article, DOI:
10.1109/MMSP.2013.6659307

Publication date:
2013

Citation (APA):
Flicker Reduction in LED-LCDs with Local Backlight

Ehsan Nadernejad, Claire Mantel, Nino Burini and Søren Forchhammer
Department of Photonics Engineering, Technical University of Denmark
2800 Kgs. Lyngby, Denmark
{ehsa,clma,nibu,sofo}@fotonik.dtu.dk

Abstract—Local backlight dimming of LCD with LED backlight can reduce power consumption and improve quality of displayed images and videos. However, important variations of LED over time produce a visually annoying artifact called flickering. In this work, we propose a new algorithm to reduce flickering while maintaining video quality. The proposed algorithm uses an adaptive second order Infinite Impulse Response (IIR) in which coefficients are calculated from the local image features. Experimental results show that the proposed method can reduce flickering while simultaneously keeping similar video quality in terms of PSNR and MSE.

I. INTRODUCTION

Nowadays, Liquid Crystal Displays (LCD) are used in different systems, like TV sets, computer monitors and etc. They have high dynamic range, high contrast ratio and low power consumption. Images are rendered by Liquid Crystal (LC) pixels filtering the light that comes from the backlight, located behind them [1], [2]. When the light source is placed behind the LC layer, the backlight is called direct-lit, otherwise it is referred to as an edge-lit backlight [3]. The backlight dimming algorithm controls the backlight luminance according to target image, dynamically. Conventional LCDs utilize only one backlight segment covering the whole display, defining global backlights. Global backlight dimming methods consume much power and cause light leakage in the black areas [3]. To overcome this, local backlights were introduced. Local backlight can be dimmed adaptively to match the image content, reducing power consumption and increasing contrast. Figure 1 shows an example of a local backlight dimming algorithm on an edge-lit display.

Several algorithms for backlight dimming have been proposed. Some algorithms use characteristic data of the target image [1], [4], [5], while more complex use the information of the Point Spread Function (PSF) to model light diffusion [2], [3], [6], [7]. By using these algorithms, LCDs achieves high contrast ratio and low power consumption.

A rendered video frame is formed by multiplying the light emitted by the backlight with the transmittance of the LC pixels [1]. In video sequences, when the image content changes or moves very quickly and frequently, the calculated backlight luminance could also be changed very rapidly and frequently as shown in Fig. 2. This phenomenon creates an unnatural trembling in the backlight segment, which defines the flickering artifact and is caused by temporal local or global dimming.

Some algorithms for flicker removal were proposed in the literature [8], [9]. The first algorithm uses Moving Average (MA) filter taking the average pixel values through several consecutive frames. Since, this algorithm is simply implemented, it has been generally applied to the local dimming LCDs. However, the MA filtering can not be generalized to the other dimming algorithms like PSF-based or histogram-based algorithms because it uses average luminance value of the target image for removing flickering artifact while most of the backlight dimming algorithm use other features (PSF or histogram) [1], [4], [5]. In addition, it causes a delay in the backlight when a scene change occurs. In [8] an adaptive MA filtering algorithm was proposed to remove flicker. This algorithm uses maximum 16 previous frames adaptively according to each backlight segment [8]. In Chen et. al [9], an Infinite Impulse Response (IIR) filter were proposed to smooth temporal backlight variation and eliminate flickering. This algorithm uses temporal and spatial backlight luminance to avoid flickering artifact [9]. While most flicker removal methods are designed for a specific backlight algorithm, they can not be generalized to be used in other algorithms. The
average values of the target image are considered as a feature in most flicker removal algorithms.

Chen et al [9] proposed a first order IIR filter for flicker removal in which the filter coefficients are calculated using average values of the target image. In this paper, we propose an adaptive flicker-reduction algorithm for LED-LCDs with dynamic backlight. The proposed algorithm is a second order IIR filter which uses information of the two previous spatial and temporal frames to calculate the backlight luminance. The filter coefficients are calculated adaptively using local features of the target image. The difference between maximum and average luminance values of the target image in each backlight segment is used to calculate the filter coefficients. The experimental results show that the proposed algorithm has a good performance in flicker removal. For some video sequences with very high flicker level, the algorithm can be repeated two or three times.

The rest of this paper is organized as follows. The proposed algorithm is described in Section II. Section III shows the experimental results on different video sequences and different local dimming algorithms. Finally, we conclude in Section IV.

II. FLICKER REDUCTION USING IIR FILTER

An annoying artifact which may be caused by local dimming is screen flicker. When the frame content of a video sequence change or move very quickly and frequently, the calculated LED luminous intensities may also change likewise. Rapid backlight can cause flickering. When backlight luminance intensity changes more than pixel values do, the flicker artifact will be more obvious. Therefore, a temporal filter is necessary to reduce the flicker artifact by temporally smoothing the change in the backlight. Figure 2 shows an example of the flickering artifact on an edge-lit display when the Cho algorithm [5] is applied to a video sequence.

An IIR filter can reduce the flickering artifact. One problem with using a smooth IIR filter is that the change in the backlight lags behind the change of the image signal. This artifact is more obvious at the moment of scene change, especially from a bright scene to a black scene. To reduce and remove screen flicker and backlighting lag artifacts an adaptive second order IIR is proposed. Figure 3 illustrates the block diagram of the proposed IIR filter.

The general form of the second order IIR filter in differential form is as follows:

\[ y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] \]  

where \( y[n] \) and \( x[n] \) are output and input of the filter. In order to calculate the temporal output of the backlight luminance, a linear combination of the spatial and the temporal variables is used as shown in Eq.1. The temporal backlight luminance is as follows:

\[ L_T^k = (a_1 L_T^{k-1} + a_2 L_T^{k-2} + b_0 L_S^k + b_1 L_S^{k-1} + b_2 L_S^{k-2})/\lambda \]  

where \( L_T^k \) and \( L_S^k \) are the temporal and the spatial LED values that correspond to \( y[n] \) and \( x[n] \) of the IIR filter, \( k \), \( k-1 \) and \( k-2 \) indicate the current and the last two frames, and \( a_1, a_2, b_0, b_1 \) and \( b_2 \) are the parameters controlling the smoothness of the IIR low-pass filter. The coefficients are selected adaptively according to the video content and are calculated as

\[ b_2 = 0; \]
\[ b_0 = \min(1, TH_1 + |diff f_m^k - diff f_m^{k-1}|); \]
\[ b_1 = \min(1, TH_2 + |diff f_m^{k-1} - diff f_m^{k-2}|); \]
\[ a_1 = 1 - b_0; \]
\[ a_2 = 1 - b_1; \]  

and

\[ \lambda = a_1 + a_2 + b_0 + b_1 + b_2; \]

where \( TH_1 \) and \( TH_2 \) are predefined parameters used to control the shape of IIR filter, \( k \) is the frame number, \( m \) is the backlight segment number, and \( diff f_m^k \) is the difference between the average and the maximum of the normalized pixel values in the entire segment. It is calculated as follows:

\[ diff f_m = max_m - avg_m \]  

where \( max_m \) and \( avg_m \) are the maximum and the average values of the normalized pixel values in physical domain. A larger difference between \( diff f_m^k \), \( diff f_m^{k-1} \) and \( diff f_m^{k-2} \) increases the possibility of scene change, so larger coefficient values are needed. This means that the temporal output is not smooth enough. Using the adaptive IIR filter, the changes in
the backlight output increases when there are changes in the scene, otherwise the backlight changes smoothly. In this way, both flicker and backlight lag are reduced.

III. EXPERIMENTS

The performance of the proposed algorithm was tested on some state-of-the-art backlight dimming algorithms. The algorithms were Albrecht’s algorithm [6], Cho’s algorithm [5], Zhang’s algorithm [4], Nadernejad’s algorithm [1] and the optimization algorithm [2], [11]. For Albrecht’s algorithm [6], we have implemented the first two steps (out of three, the third one being optional), which produces a clipper-free result. Two versions of the optimization algorithm were implemented [11]. These are based on Gradient Descent (GD) optimization algorithm with two leakage factors (GD with $\epsilon = 0.001$, and GD with $\epsilon = 0.0002$ [11]). All algorithms were simulated on an edge-lit display with 16 backlight segments placed in eight rows and two columns.

To assess the performance of our algorithm on video sequences, we tested them on 8 Full-HD (1920x1080) sequences. Four sequences (Volcano, Anemone, BBBunny and Diver) come from [12], two (Stars and Titles) come from [10] and two come from DVD content. These sequences present varying characteristics in terms of high and low luminance, color and details, as well as temporal variation (motion and scene change). Figure 4 shows a frame extracted from each of them. In the experiments $TH_1 = TH_2 = 0.125$.

Figure 5 shows the result of the proposed algorithm on different video sequences for segment number 13 of edge-lit display using Cho’s algorithm [5]. As can be seen, when the level of flicker is low like the Volcano and Concert sequences, the algorithm preserves the LED values. However, in the case of high flicker like Stars and Theatre, the fluctuations in LED values are controlled in an acceptable level. The control level follows the normal trend of the previous LED values. The effect of the proposed algorithm is the same for other segments. In Fig. 6, the average LED values (The average for all 16 segments of the edge-lit display) for one video sequence ‘Stars’ are shown. Another advantage of this method is that the average power consumption does not change significantly after flicker removal. The power level before and after filtering are shown at the bottom of Fig. 6.

In order to evaluate the performance of the proposed algorithm based on the objective quality measures the PSNR and MSE vectors for Stars video sequence after and before filtering are illustrated in Figs. 7 and 8 (The LC values are quantized to 10 bit). As shown in these figures, the proposed algorithm can produce a flicker-free result while keeping the quality of the video signal. Although, the PSNR and MSE after filtering were not always better compared to those of the original video sequence, their averages were almost the same as shown in the figures. In addition, their overall fluctuations were smoothed. The average results for different local dimming algorithms on different sequences are summarized in Table 1. These results are for the Title (251 frames), Theatre (207 frames), Stars (255 frames), Diver (300 frames), Concert (217 frames), Anemone (300 frames), Volcano (300 frames), and BBBunny (250 frames) video sequences. It is important to keep the quality of the videos at their initial level before filtering. Therefore, in this table, the minimum as well as the average PSNR are reported and due to its reverse relationship with the MSE, the corresponding maximum and average MSEs are shown. For power consumption, all the minimum, maximum and average values are reported. As observed in the table, the proposed filter does not have a large impact on PSNR, MSE and the power consumption, in terms of minimum, maximum and average values. In addition, Figure 9 shows the average quality versus power consumption of the different algorithms applied on all the sequences before and after filtering. In general, a flicker removal algorithm should not affect the quality and power consumption of LEDs. These results verify that the proposed algorithm preserves these two characteristics since the position of the points in the plot are close to each other before and after filtering. There is no objective metric for the flickering artifact in LCD displays with local backlight. However, we evaluated the performance of the filter both visually and objectively. The temporal variation is computed by the SSD as recommended in [13] to measure flickering caused by video compression. SSD is defined as:

$$SSD = \frac{1}{N_{frames}} \sum_n \frac{1}{N_{pixels}} \sum_i (Fr_{n-1}(i) - Fr_n(i))^2$$

(5)

where $Fr_n(i)$ is the pixel $i$ of the frame $n$. The SSD results for all sequences are shown in Table II. The average level of temporal variations is either smaller after temporal filtering (TF) than before applying the filter (Orgn) or similar. In some cases that the level of the flicker was quite high, the algorithm was applied two or three times for a successful removal. This is shown by the numbers inside the parentheses in Table II. In some cases, the removal was not completely successful and this is shown by (⋆) in the table that means not removed completely.

IV. CONCLUSION

In this paper an adaptive second order IIR filter for removing flickering artifact in LED-LCD with local backlight was proposed. The filter uses spatial and temporal information of LED values. The coefficients of the IIR were selected adaptively using local features of the target image. To show
Fig. 5. Performance of the proposed algorithm on different sequences, visualized using an edge-lit display in which the Cho’s algorithm [5] is used. The red and green line show the LED values before and after filtering, respectively. (LED No.13, Column: 1,...4 are: BBanny, Volcano, Stars, Title, and column. 2: 1,...4 are: Anemone, Diver, Theatre, Concert.)*

**TABLE I**

The average de-flickering results for different algorithms on all video sequences. The algorithms are: 1: GD with $\epsilon = 0.0002$ [2], 2: Albrecht [6], 3: Cho [5], 4: Nadernejad [1], 5: Zhang [4], and 6: GD with $\epsilon = 0.001$ [2], respectively.

<table>
<thead>
<tr>
<th>Metric</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.824</td>
<td>2.737</td>
<td>2.385</td>
<td>3.769</td>
<td>1.765</td>
<td>1.108</td>
<td>0.957</td>
<td>2.639</td>
<td>2.410</td>
<td>3.927</td>
<td>1.769</td>
<td>1.522</td>
</tr>
<tr>
<td>PSNR</td>
<td>47.537</td>
<td>42.090</td>
<td>39.878</td>
<td>38.304</td>
<td>41.419</td>
<td>45.092</td>
<td>47.005</td>
<td>42.595</td>
<td>39.080</td>
<td>38.024</td>
<td>41.349</td>
<td>44.285</td>
</tr>
<tr>
<td>Mean</td>
<td>47.537</td>
<td>42.090</td>
<td>39.878</td>
<td>38.304</td>
<td>41.419</td>
<td>45.092</td>
<td>47.005</td>
<td>42.595</td>
<td>39.080</td>
<td>38.024</td>
<td>41.349</td>
<td>44.285</td>
</tr>
<tr>
<td>Power</td>
<td>0.782</td>
<td>0.828</td>
<td>0.673</td>
<td>0.548</td>
<td>0.710</td>
<td>0.646</td>
<td>0.724</td>
<td>0.653</td>
<td>0.596</td>
<td>0.495</td>
<td>0.636</td>
<td>0.624</td>
</tr>
</tbody>
</table>

Fig. 6. The average LED values for 16 backlight segments before and after filtering applied on Stars using Cho’s algorithm [5].

Fig. 7. PSNR of the Stars sequences before and after de-flickering.

the performance of the proposed algorithm, some state-of-the-art local backlight dimming algorithms were implemented and tested on 8 Full-HD video sequences with different features. Experimental results demonstrated that the proposed algorithm can avoid abrupt variations of the LED signals and consequently reduce the flicker artifact.
## Table II

SSD measures of sequences before and after temporal filtering for different algorithms on all video sequences. (The numbers inside the parentheses show the number of times the filtering was repeated and (+) means that the flicker was not removed completely.)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Orgn TF</td>
<td>Orgn TF</td>
<td>Orgn TF</td>
<td>Orgn TF</td>
<td>Orgn TF</td>
<td>Orgn TF</td>
</tr>
<tr>
<td>Stars</td>
<td>Avg 64.83 59.79 67.34 66.43</td>
<td>65.37 64.04 55.79 52.87</td>
<td>65.24 63.72</td>
<td>59.66 52.37</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max 137.87 122.78 140.73 138.74</td>
<td>133.88 134.24 115.24 112.03</td>
<td>135.55 135.17</td>
<td>131.10 110.34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concert</td>
<td>Avg 0.71 0.70 (2) 0.63 0.56 (2+)</td>
<td>0.68 0.65 0.66 0.63 (2+)</td>
<td>0.69 0.66</td>
<td>0.69 0.66 (2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max 5.82 6.09 7.00 3.28</td>
<td>6.63 4.04 5.35 3.59</td>
<td>6.54 4.43</td>
<td>6.43 4.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iltis</td>
<td>Avg 117.94 115.71 72.13 71.60</td>
<td>92.15 92.36 70.48 69.62</td>
<td>92.54 91.85</td>
<td>108.09 103.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max 231.10 229.13 145.09 139.69</td>
<td>181.30 179.56 138.22 136.54</td>
<td>185.50 179.03</td>
<td>213.42 207.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Volcano</td>
<td>Avg 1.16 1.16 1.18 1.17 (2)</td>
<td>1.10 1.10 1.02 1.02</td>
<td>1.14 1.14</td>
<td>1.12 1.12</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max 1.88 1.88 1.90 1.88</td>
<td>1.76 1.76 1.65 1.64</td>
<td>1.85 1.85</td>
<td>1.82 1.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anemone</td>
<td>Avg 27.79 27.98 27.84 27.88 (2)</td>
<td>27.93 27.92 27.79 27.79</td>
<td>27.95 27.94</td>
<td>27.91 27.91</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max 78.29 78.28 78.24 78.28</td>
<td>78.16 78.15 78.01 78.04</td>
<td>78.22 78.20</td>
<td>78.23 78.28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BBBunny</td>
<td>Avg 72.56 72.39 (2) 72.39 72.34 (2)</td>
<td>71.25 70.90 (2) 70.93 70.81</td>
<td>72.01 71.86</td>
<td>72.37 72.02 (2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max 288.17 287.28 287.61 287.04</td>
<td>283.84 282.65 282.65 282.06</td>
<td>286.77 285.28</td>
<td>289.72 285.76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Theatre</td>
<td>Avg 32.37 32.71 (2+) 32.92 32.91 (3+)</td>
<td>30.84 30.89 30.84 30.89</td>
<td>28.04 28.09</td>
<td>31.90 31.91</td>
<td>31.35 31.36</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max 47.34 47.38 48.05 47.76</td>
<td>45.79 45.37 45.14 45.14</td>
<td>46.31 46.32</td>
<td>45.34 45.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diver</td>
<td>Avg 4.45 4.44 (2) 4.46 4.45 (2+)</td>
<td>4.27 4.26 4.32 4.31</td>
<td>4.33 4.32</td>
<td>4.34 4.31 (2)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Fig. 8

MSE of the Stars2 sequences before and after de-flickering.

### Acknowledgment

We would like to thank Jesper Meldgaard Pedersen, Bang & Olufsen for discussions and contributions.

## References


