DBS4video: Dynamic Luminance Backlight Scaling based on Multi-Histogram Frame Characterization for Video Streaming Application

Martino Ruggiero, Andrea Bartolini, Luca Benini
University of Bologna, DEIS
via Risorgimento 2,
40133 Bologna, Italy
martino.ruggiero@unibo.it, a.bartolini@unibo.it, luca.benini@unibo.it

ABSTRACT

Almost every modern portable handheld device is equipped with a coloured LCD display. The backlight of the LCD accounts for a significant percentage of the total energy consumption. Substantial energy savings can be achieved by dynamically adapting backlight intensity levels on such low-power portable devices. In this paper, we present the DBS4video framework which allows dynamic scaling of the backlight with a negligible impact on QoS for video streaming applications. DBS4video exploits in a smart and efficient way the hardware image processing unit integrated in almost every new multimedia application processor to implement a hardware assisted image compensation. The proposed approach overcomes CPU-intensive techniques by saving system power without requiring either a dedicated display technology or hardware modification. We introduce also a new image processing kernel based on multiple histograms collection for a single frame. We provide a real implementation of the proposed framework on a Freescale application development board based on the i.MX31 processor. We carried out a full characterization of the overall system power consumption versus QoS.

Categories and Subject Descriptors

K.6.4 [**System Management**]: Quality assurance

General Terms

Algorithms, Management, Performance

Keywords

LCD, Backlight scaling, Display, Power management, Video streaming

EMSOFT'08, October 19–24, 2008, Atlanta, Georgia, USA.

1. INTRODUCTION AND RELATED WORK

Despite the ever increasing advances in Liquid Crystal Display's (LCD) technology, their power consumption is still one of the major limitations to the battery life of mobile appliances such as smart phones, portable media players, gaming and navigation devices. There is a clear trend towards the increase of LCD size to exploit the multimedia capabilities of portable devices that can receive and render high definition video and pictures. Multimedia applications running on these devices require LCD screen sizes of 2.2 to 3.5 inches and more to display video sequences and pictures with the required quality. LCD power consumption is dependent on the backlight and pixel matrix driving circuits and is typically proportional to the panel area. As a result, the contribution is also likely to be considerable in future mobile devices.

Several power saving schemes and algorithms have been proposed in literature. Some of them exploit software-only techniques to change the image content to reduce the power associated with the crystal polarization [4][16], some others are aimed at decreasing the backlight level while compensating the luminance reduction using pixel-by-pixel image processing algorithms [5][7][8]. The major limitation of these techniques is that they rely on the CPU to perform pixel-based manipulations and their impact on CPU utilization and power consumption has not been assessed [6][9][12]. Moreover, very little work is focused and optimized for video streaming applications [13][10]. In such case, we have to face not only QoS degradation constraints, but also real-time performance guarantees[11].

We present an alternative approach, called DBS4video, which allows dynamic scaling of the backlight with a negligible impact on QoS for video streaming applications. It exploits in a smart and efficient way the hardware image processing unit (IPU) integrated in Freescale's multimedia application processor to implement a hardware assisted image compensation. The proposed approach overcomes CPU-intensive techniques by saving system power without requiring either a dedicated display technology or hardware modification. CPU processing, based on frame-by-frame histogram analysis on YUV image format, is minimized by means of hardware assisted image processing tasks. The hardware facilities exploited by the DBS4video implementa-

tion are present in almost every new embedded multimedia processor [2][3][1], so the DBS4video framework can be easily ported on other different platforms.

The optimal scaling factor of image compensation depends on the frame content and it varies frame-by-frame, consequently finding the optimal scaling factor implies an accurate video analysis. Selecting the right frame processing kernel is a key point in backlight luminance scaling technique, otherwise the power wasted on the CPU could be greater than that saved on the LCD. Histogram analysis is a good way to face this issue since histogram collection and processing is very efficient and intuitive [16]. Unfortunately, there is a significant drawback in using this kind of image classifier: histogram does not provide any information about pixels locality and image structure. Histogram can show only quantitatively how many bright pixels we have in a frame, but it can not say anything about their location or about their relation each other. When we apply luminance compensation, we know exactly how many pixels we are distorting, but we do not know where these pixels are located.

One main contributions of this work is the introduction of a frame processing step based on the collection of multiple histograms for a single frame. Basically, we divide the entire input frame in several sub-blocks and compute one histogram for each of them. In this way we lose less information about pixels locality, but at the same time we are maintaining histogram's benefits in terms of energy versus performance efficiency.

We provide a real implementation of the dynamic backlight scaling technique, embedded within a custom Video4Linux software subsystem running on a Freescale prototype development board based on the i.MX31 multimedia application processor and a 3.3-inch QVGA display [17]. By instrumenting this platform, we carried out a full characterization of both LCD and CPU power consumption versus QoS.

Another main contribution of this work is the introduction of a new way to measure QoS for backlight scaling techniques. Means of characterization and evaluation for the main assumptions and goals are very important for the issue we are facing. Modifying video sequences, and more in general images, requires an objective way to asses the final constraint satisfaction of no quality loss. More in detail, we need a QoS metric which compares the output of the display for two sequences of the same video: the original one and the backlight scaled one. Several QoS metrics have been proposed in literature [18][14], but none of them takes into account both the backlight contribution and modification. We built an accurate model of the LCD which produces as output the way how the target image will be reproduced onto the display exploiting also the backlight level.

To properly assess the effectiveness of the proposed technique, a video player application and a variety of video sequences were used. Results show power savings up to 24% considering both LCD and CPU contribution with bounded QoS degradation and real-time performance guarantees.

2. I.MX31 MULTIMEDIA PROCESSOR

The Freescale i.MX31 is an ARM-11 based Application Processor targeted for portable multimedia devices. The i.MX31 processor is aimed with an Image Processing Unit (IPU), where some computationally-intensive parts of video processing chain are offloaded from the ARM CPU and accelerated in hardware.

The Image Processing Unit is designed to support video and graphics processing functions and to interface to video/still image sensors and displays. The IPU is equipped with a DMA engine in order to perform its tasks with minimal involvement of the ARM CPU control and synchronization. As a result, in most cases, the CPU involvement is focused on processing tasks such as video decoding. Moreover, the system-on-chip integration combined with internal synchronization, avoids unnecessary access to system memory, reducing the load on the memory bus and reducing further the power consumption.

The IPU performs also some processing-intensive image manipulations, such as:

- Downsizing with independent integer horizontal and vertical ratios;
- Resizing with independent fractional horizontal and vertical ratios;
- Color space conversion (YUV to RGB, RGB to YUV, YUV to different YUV);
- Combining a video plane with a graphics plane (blending of graphics on top of video plane);
- 90 degree rotation, up/down and left/right flipping of the image.

Implementing our framework, we paid great attention in optimizing all memory accesses, CPU and IPU utilizations. The communication between the CPU, main memory and IPU's sub-modules has been made through DMA transfers, while synchronization has been handled by interrupts.

3. DYNAMIC LUMINANCE BACKLIGHT SCALING

The power consumption of the backlight is proportional to its luminance. The principle of dynamic backlight scaling is to save power by backlight dimming while restoring the brightness of the image by appropriate image compensation. Thus backlight luminance scaling adaptively dims the backlight with appropriate image compensation so that the user perceives similar levels of brightness and contrast with minor image distortion.

We can consider the luminance emitted from a x pixel of the LCD panel equal to equation:

$$L(x) = f(BL) * g(x) \quad (1)$$

Intuitively, this equation says that the luminance emitted from a pixel depends on functions *f()* of the backlight (BL) and *g()* of the pixel value itself, which sets its transmittance. We carried out a characterization of the LCD panel in order to specify and quantify these two functions.

3.1 LCD display characterization

This section describes a set of tests performed in order to better understand the display LCD optical properties and evaluate the non linearity effects existing between real displayed quantities and digital values description.

The main goal of these tests was to characterize the optical properties of the LCD display. More in detail we analyzed:

1. The relationship between pixel digital values in RGB space and emitted light intensities.

2. The relationship between pixel digital values in RGB space and perceived colours.

3. The impact of backlight dimming on colour perception evaluated in the RGB space.

4. The relationship between backlight values and emitted light intensities from LCD display.

A second goal was the creation of a display model thanks to which we can evaluate the rendering of an image on the LCD.

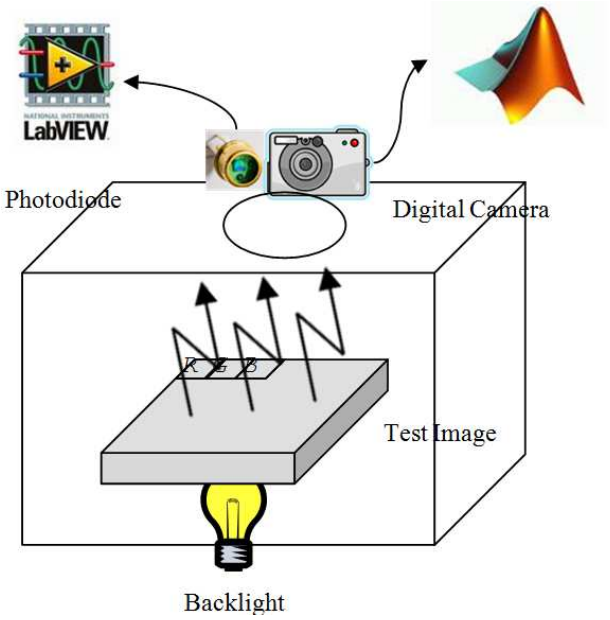

Figure 1: LCD characterization setup.

Figure.1 shows how we performed the tests. We displayed a set of image on the LCD and probed the emitted light with both a digital camera and a light sensor. The displayed images were built using a Matlab script which converts both jpg and bmp images into the iMX31 RGB compliant format and writes it directly on the iMX31 frame buffer device. In order to get a set of consistent measurements, the ambient light contribution was eliminated, performing the tests within a dark room.

For the first test, we used a photodiode IPL 10530DAW as light sensor. The light sensor produces as output a voltage linearly proportional to the intensity of the incident light emitted by the LCD. We tested the emitted light from the LCD displaying a monochromatic image which has only one RGB component which varies from 0 to 255, and the remaining components set to 0 ((RGB=X,0,0), (RGB=0,X,0), (RGB=0,0,X)).

Figure.2 shows the normalized light intensities on the y-axes and the normalized value of the three RGB components (X/255) on the x axes. The plot shows that the light emitted by pixels is not linear with the digital RGB value, but it exhibits a polynomial law. We fitted the measured value (the dots in the plot) with the function:

$$L(x) \propto offset + K * (R, G, B)^{\gamma} \quad (2)$$

Tab.1 reports the output fitted parameters of Eq.2.

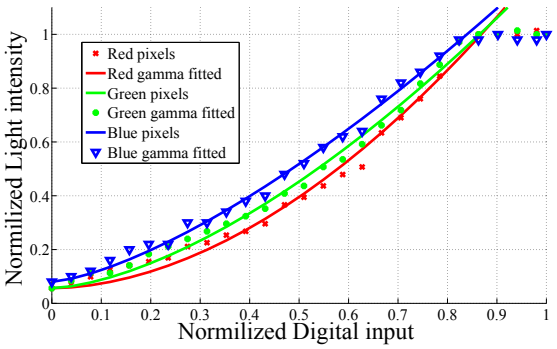

Figure 2: Light Intensities for R,G,B pixels and relative gamma fit.

	offset	K	γ	R^2
R	0,0563	1,226	1,855	0,984
G	0,0563	1,187	1,583	0,991
B	0,08	1,181	1,432	0,994

Table 1:

Subsequently we tested the emitted light from the LCD displaying a monochromatic image which has the three RGB components set to the same value (RGB=X,X,X). This set of images reproduces a gray scale characterized by a luminance component Y equal to the three RGB pixel digital value (Y = X). Figure.3 shows the normalized light intensity versus the luminance value Y. It shows the measured data (represented with dots) and the best obtained fitted function.

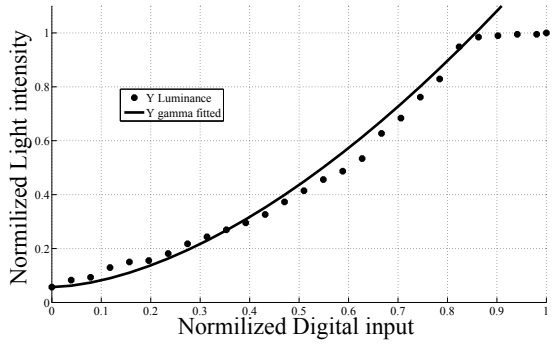

Figure 3: Light Intensity vs. Luminance (Y).

The plot reveals that some light is still emitted for black images (RGB=0,0,0). Moreover, a saturation effect can be noticed for pixel values bigger than 0.8. This means that from 0.8 to 1 the amount of light emitted from the LCD display does not change. The fitted function in this case is:

$$L(x) \propto offset + K * (Y)^{\gamma} \quad (3)$$

Tab.2 reports the output fitted parameters of Eq.3.

From this initial data, it can be noticed that a α multiplicative factor of the Y component of a pixel produces an increment in emitted light intensity equal to α^{γ}.

The following test was performed reproducing on the dis-

offset	K	γ	R^2
0,057	1,224	1,691	0,987

Table 2:

play a still image while, changing the value of the backlight and capturing its effect with the camera. This test was done for four different monochromatic images (White, Red, Green, Blue). The results of this test are shown in Figure.4.

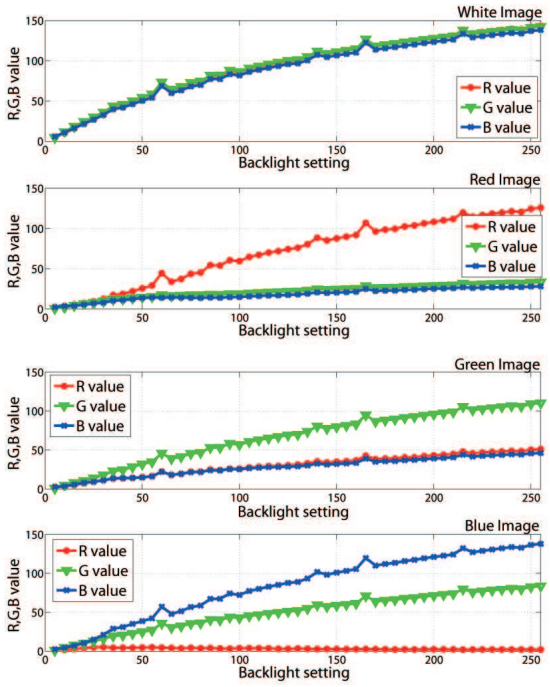

Figure 4: RGB Colour Space Camera sampled vs. Backlight digital value, for four different image show in the LCD: White, Blue, Green, Red.

It can be soon observed that the RGB space of the display and the RGB space of the camera are not aligned: the camera recognizes some intensity of other primary colours for pure monochromatic images. The plots show that a reduction on the backlight level has the same effect on the three components in RGB space. This effect can be modeled as a non linear reduction of the three components:

$$(R', G', B') \propto K * (R_{\text{MAX}}, G_{\text{MAX}}, B_{\text{MAX}}) * Backlight^{\theta} \quad (4)$$

This last equation states that the perceived color components (R',G',B') of a pixel depend on both the (R,G,B) values of the pixel itself and the backlight value.

Now that we have proved that the backlight dimming affects also the chromatic components of an image, we tried to quantify this effect in terms of emitted light using the photodiode as light sensor. The results of this characterization are shown in Figure.5: the dots show the measured data and the line shows the fitted equation.

The curve was obtained with the power fit:

$$L(x) \propto K * (Backlight)^{\theta} \quad (5)$$

Tab.3 reports the output fitted parameters of Eq.5.

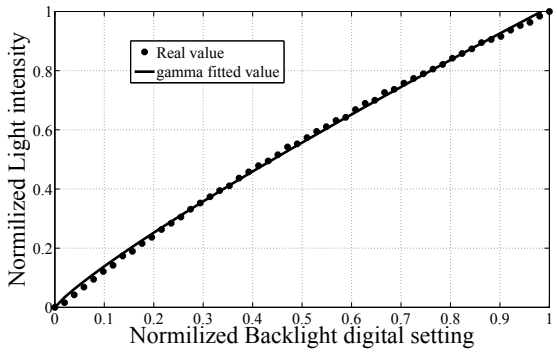

Figure 5: Normalized Light Intensity vs. Normalized digital Backlight value.

K	θ	R^2
1,014	0.8648	0,998

Table 3:

3.2 Problem model

As already mentioned in Eq.1, the emitted light from a pixel is function of both backlight and pixel transmittance value. If we want to save power, we need to dim the backlight and at the same time scaling the pixel transmittance in order to re-equilibrate the target pixel luminance.

From Eq.1, 3 and 5, we can formulate:

$$L(x) \propto (Y)^{\gamma} * (Backlight)^{\theta} \quad (6)$$

if we dim the Backlight value of a scaling factor β, in order to obtain the same target luminance we need also to scale the Y component of the pixel by an α factor:

$$L'(x) \propto (\alpha Y)^{\gamma} * (Backlight/\beta)^{\theta} \quad (7)$$

From 6 and 7 we obtain:

$$\alpha^{\gamma} = \beta^{\theta} \quad (8)$$

The last equation says that for a given α compensation value applied to an image, we need to scale the backlight of a $\beta = \alpha^{\gamma/\theta}$ factor in order to maintain the same target luminance.

4. THE DBS4VIDEO FRAMEWORK

We provide a real implementation of the DBS4video dynamic backlight scaling framework. It has been embedded within a custom Video4Linux software subsystem running on a Freescale prototype development board based on the i.MX31 multimedia application processor and a 3.3-inch QVGA Sharp display. The block diagram in Figure.6 describes how the entire framework operates.

We can divide the overall flow into three main sections:
1: pre-processing;
2: frame-processing;
3: post-processing.
In the following sub-sections each phase will be described in depth.

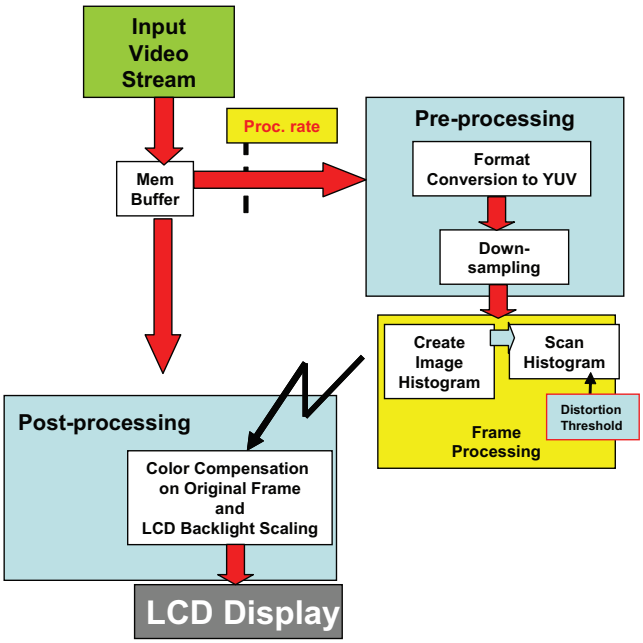

Figure 6: Dynamic Backlight Autoregulation block diagram.

4.1 Pre-processing

The task of the *pre-processing* is to off-load the CPU manipulating the input frame. Although histogram creation and analysis are a good compromise for an energy efficient image processing, handling images in RGB format is not convenient. The input frame is thus converted into YUV format, exploiting the hardware IPU unit, and subsequently down-sampled: the pre-processing phase allows to process a smaller image and compute over a smaller set of data, since the YUV format already contains the needed luminance information for each pixel into the image.

4.2 Frame-processing

Figure.7 shows a pictural overview of the frame-processing phase.

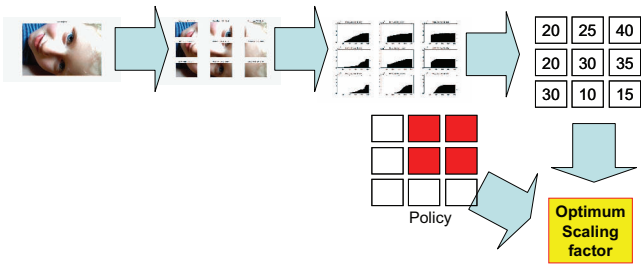

Figure 7: Overall frame processing phase.

After the pre-processing phase, the down-sampled and YUV converted image is fed to the *frame-processing* step. In this section of the framework, each pixel is used to compute the luminance histograms of the pre-processed image. The histograms computed take into account the amount of saturated pixels for a given α scaling factor. This approach is slightly different from creating a histogram which accounts the number of pixels for a given luminance value. In the latter histogram formulation there is not any direct information about distortion or any direct correlation between new backlight value and amount of distorted pixels. In the given implementation we can know the exact amount of distortion for a given scaling factor directly indexing the histogram, consequently critical regions in the image are easily recognizable.

Once histograms are created, they are scanned starting from bright region and, when the distortion threshold is reached, the optimal luminance α scaling factors for each histogram are found. More in detail during this phase, luminance histograms are individually analyzed to ascertain how much the image luminance may be increased for each sub-block, whilst satisfying a given distortion. The distortion is defined as the total number of saturated pixels after image compensation.

At this point, the framework generates a grid composed by the optimal scaling factors for each sub-block. The overall result of the frame-processing step is to determine the amount of brightness reduction achievable on the backlight, which compensates the brightness increase that can be applied to the entire image. Analyzing the grid, we can establish which compensation value apply following a given QoS policy. Exploiting the grid, several QoS policies can be easily defined, each accounting different levels or metrics of QoS vs power saving. In our experimental setup, the selected policy enables the most aggressive compensation value of the grid which does not saturate more than two in a group of four contiguous sub-blocks for the entire image.

4.2.1 *Single vs Multi histograms*

While accurate single-histogram analysis is very challenging and critic, we can obtain more control on quality degradation by means of multi-histograms. For a given input frame, we calculate more than one histogram, dividing the overall image into sub-blocks and calculating one histogram for every sub-block. Having more than one single histogram means loosing less information about image luminance locality.

Figure.8 shows an example in which multi-histogram performs better than single-histogram implementation. In the upper right corner the single histogram of the target picture is shown, while the remaining histograms are obtained from the multi-histogram implementation (the image is divided into 36 sub-blocks).

The circled region of the single histogram shows that there are quite a large number of bright pixels in the target picture. However the multi-histogram implementation highlights that these bright pixels are scattered on the overall area of the picture. This means that this picture can allow a more aggressive luminance scaling, since distorted pixels will not be easily recognizable from the eye of the viewer.

Figure.9 shows another example in which multi-histogram performs better than single-histogram implementation. In this case the multi-histogram one is able to recognize spotted bright pixels. In the single-histogram we can find high peaks at high luminance values, but we are not able to understand in which regions of the target picture the high peaks are located. With the multi-histogram implementation we are able to recognize where these regions are, and consequently we can apply the right scaling decision.

Figure 8: Example of sparse bright pixels.

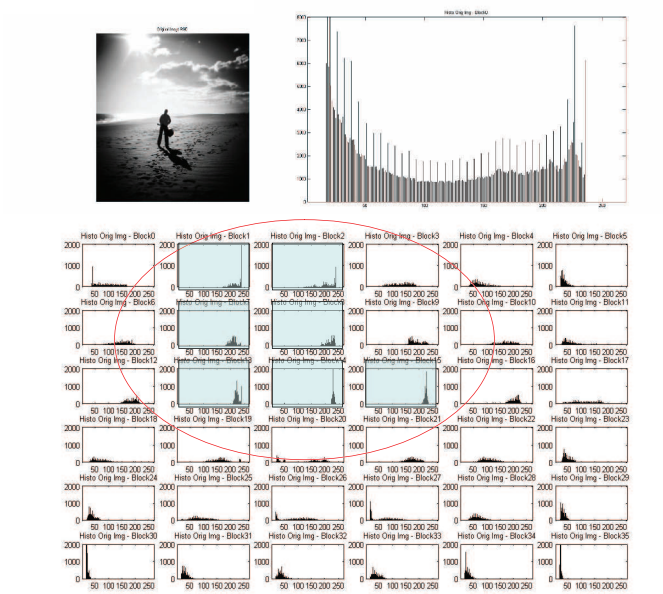

Figure 9: Example of spotted bright pixels.

4.3 Post-processing

The final step of our framework is the *post-processing* phase. The main task of this section is to apply the luminance scaling factor found in the frame-processing phase to both backlight and input video frame. The input frame is the starting image of the pre-processing phase with its original resolution and format. By applying the compensation to the original frame instead of the pre-processed one, we do not loose image quality, in terms of both resolution and color distortion.

The post-processing step is a function applied pixel-by-pixel and implementing it on CPU should result in a very power expensive step. In order to avoid this situation, we implemented it in hardware exploiting in a smart way the capabilities of the available image processing unit. One of the main functions performed by the hardware image processing unit is colour space conversion which is done through conversion matrices. The conversion matrix coefficients are programmable and they are stored in the IPU main memory region.

The conversion equations are:

$$Z0 = 2^{scale-1} * (X0 * C00 + X1 * C01 + X2 * C02 + A0) \quad (9)$$

$$Z1 = 2^{scale-1} * (X0 * C10 + X1 * C11 + X2 * C12 + A1) \quad (10)$$

$$Z2 = 2^{scale-1} * (X0 * C20 + X1 * C21 + X2 * C22 + A2) \quad (11)$$

Where X0, X1 and X2 are the component of the input format; Z0, Z1, Z2 the component of the output format; C00, .., C22 and A0, .., A2 the conversion matrix coefficients.

We decided to execute the luminance compensation in the post-processing step scaling the values of conversion matrix coefficients.

For example, if input image format is RGB, the default conversion matrix is the identity matrix:

$$CSC_{default} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

In order to compensate an image after backlight scaling down, we need to scale up the value of $CSC_{default}$ by an α factor:

$$CSC_{DLS} = \begin{vmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha \end{vmatrix}$$

As shown in the previous display characterization (see Section.3), the light intensity emitted by the display does not show a linear relationship with both pixel transmittance and backlight values. Hence the luminance compensation algorithm needs to take into account for it. An α multiplicative factor will generate a light intensity increment of a α^{γ}, while scaling the backlight value of a β factor will produce a light dimming of a β^{θ}. Consequently, having an input frame which allows a Y component increasing of a α factor, the reduction factor (β) for the backlight value should be obtained by the formula 3.2. This formula has been integrated inside the luminance scaling algorithm through a look up table, which is indexed by α and provide the corresponding β value.

5. QUALITY OF SERVICE METRICS

Several QoS metrics have been proposed in literature, but none of them takes into account the backlight contribution. To overcame this limitation we built a model of the LCD display which takes as input an image and a backlight level, and produces as output the way how the target image will be reproduced onto the LCD. To this output image we applied the selected QoS metrics. The most suitable metrics for our purposes were the SSIM and the CIELabΔE_{AB}: the former

performs well in finding structural distortion, the latter in evaluating the perceptual color distortion.

5.1 Structural Similarity (SSIM)

The first metrics implemented is the Structural Similarity (SSIM). As shown in [18], the SSIM is based on the assumption that the human visual system is efficient in extracting structural information from the viewing field. Whereas structural information of an image means an attribute which describes the structure of scene objects independently of average luminance and contrast. In Figure.10 we can see the schematic implementation of the SSIM metrics. The average luminance, contrast and structural information are computed for both target and sample images and then combined together to produce the quality index.

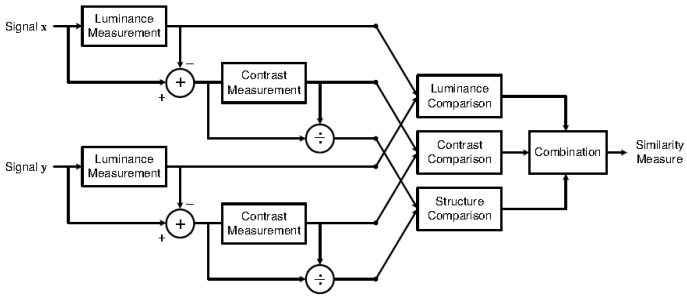

Figure 10: Diagram of (SSIM) Structural Similarity measurement system.

Let's consider for simplicity a gray scale image, the coloured issues will be discussed later. The average luminance is computed as overall pixel image average value (μ_x). The contrast information is, instead, obtained for both images (X and Y) as the standard deviation (σ_x σ_y). Since these values can vary a lot within the same image, both the luminance and contrast indexes are computed not globally, but locally, dividing the image in sub-block and computing them for each block. The structural information is represented by the vector $(X - \mu_x)/\sigma_x$. The SSIM index is then built comparing these three indexes for the two input images. The SSIM index is expressed by the following formula:

$$SSIM(X,Y) = l(X,Y) * c(X,Y) * s(X,Y) \qquad (12)$$

where l, c, s: $N^2 \rightarrow R[0,1]$ describe respectively the luminosity, the contrast and the structural degrees of similarity between the two images.

On colour images, the SSIM index can be obtained [18] after having transformed the image into the IPT colour space and applying the above formula at each channel. Doing so, we obtain one similarity index for each I,P,T channel ($SSIM_I$, $SSIM_P$, $SSIM_T$), then we can put all the three together:

$$SSIM = SSIM_I + SSIM_P + SSIM_T \qquad (13)$$

5.2 CIE Lab colour space ΔE_{AB}

CIE Lab colour space ΔE_{AB} is a pixel-wise measure corresponding to the Euclidean distance measured between two colour points in CIE Lab space [14]. It was developed to compare patches of constant colours, so it should be of limited accuracy for more complex images.

$$\Delta E_{AB} = mean_i(\sqrt{\Delta L(x_i,y_i)^2 + \Delta a(x_i,y_i)^2 + \Delta b(x_i,y_i)^2}) \qquad (14)$$

5.3 LCD Model and Experimental Setup

In order to evaluate how a certain image will be rendered on the LCD display, we modeled it using all the above characterization results (see Section.3). The model takes as input an image in RGB and a backlight value, and it produces in output how the image will look like on the LCD display.

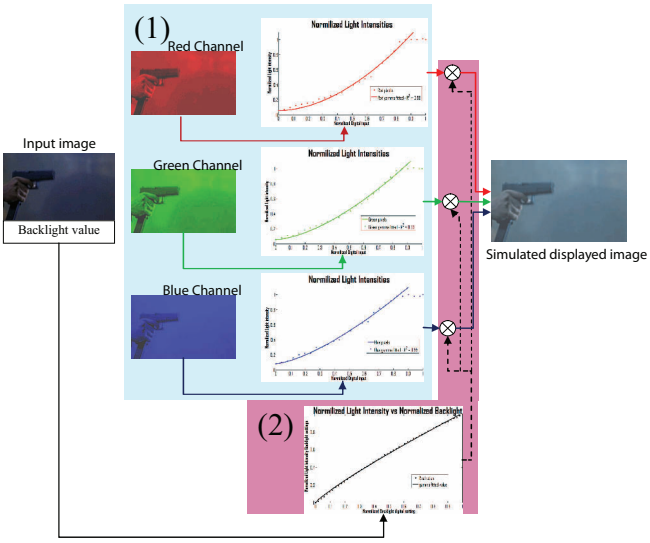

Figure 11: LCD model schematic.

The model can be split in three main steps (see Figure.11):

1. The incoming image is transformed by set of Eq.2, which account the non-linearity of LCD light trasmittance.

2. The input backlight is used to obtain a new value which considers the non-linearity showed in Eq.5.

3. The model combines the transformed image and the new backlight value to compute the simulated displayed image.

The proposed QoS metrics have been implemented in Matlab in order to evaluate the quality of service of our backlight scaling algorithm. Both sets of input images (from original and scaled video sequences) required by the QoS metrics have been taken at run time from the IPU post-process output. These images are the same ones that are reproduced on the real LCD screen. To capture the IPU output video flow, we used a modified Video4Linux driver which stores every outgoing frame into a file. This file contains both the frames in RGB format and the corresponding backlight values.

We captured both the original and the compensated video streams and imported them in Matlab. Using the described LCD model, we applied the backlight compensation to the video sequences. Finally, the QoS metrics have been computed frame-by-frame.

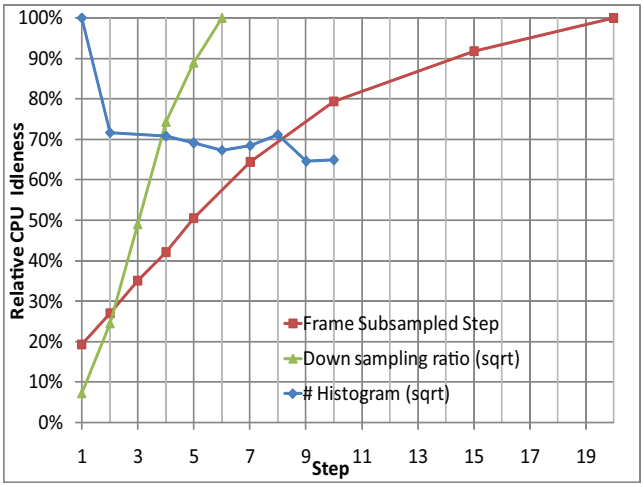

Figure 12: Framework parameters Vs performance

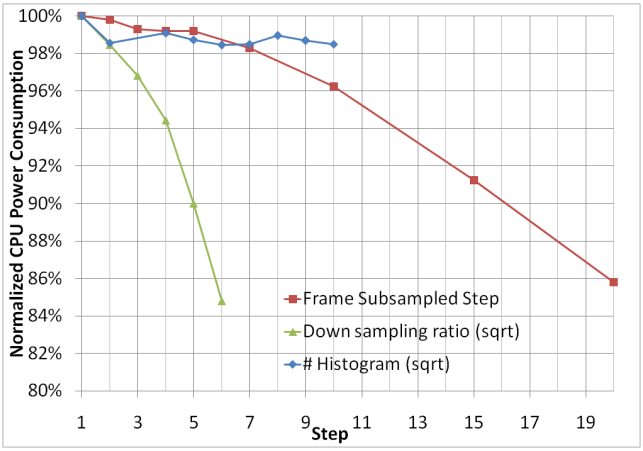

Figure 13: Framework parameters Vs power

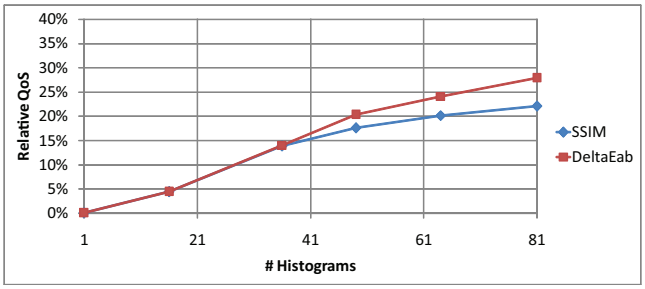

Figure 14: Number of histogram Vs QoS

6. EXPERIMENTAL RESULTS

In our framework we can set and modify four parameters, namely:
1. DBS4video Frame processing rate: it establishes how many frames will be fed to the main algorithm for calculating the new backlight and compensation level.
2. Downsampling ratio: it establishes the downsampling step of the input frame during the pre-processing phase.
3. Number of histograms: by means of this setting, we can set how many histograms will be created for each frame.
4. Distortion threshold: it establishes how many saturated pixels are admitted for the entire frame.

Tuning them we can change the trade-off between QoS and power savings. We carried out a parameters exploration study with regards to both performance, power and QoS metrics in order to better control this trade-off.

The plot in Figure.12 expresses how computational intensive the entire framework is varying the configuration parameters. The plot reveals that an increasing number of histograms does not affect the CPU idleness: there is only an initial overhead going from 1 histogram to 4 histograms, after that point adding more histograms does not correspond in an increasing of CPU load. The CPU idleness is more sensible on the downsampling and frame step tuning. These considerations are reflected also in the power consumption results presented in Figure.13.

Intuitively, with a low DBS4video frame processing rate we will waste less power on CPU side, but at the cost of a less responsiveness of the framework to average luminance changes of the input video stream. A low downsize ratio will result in a low CPU utilization at the cost of a less precise luminance value calculation for image compensation.

Figure.14 shows the results of the QoS exploration with an increasing number of histograms. Both SSIM and $DeltaE_{AB}$ metrics highlight the advantage of using a increasing number of histograms resulting in better QoS score improvements.

We measured the QoS and power consumption of the LCD and CPU during the playing of a video sequence from *Terminator 3* movie. Table4 shows the features of this video stream in terms of frame rate and resolution.

To evaluate if the total amount of emitted light from the LCD is preserved using DBS4video framework, we measured this quantity by means of the photodiode sensor. Figure.15 shows the results of this experiment for two different distortion thresholds compared w.r.t no-DBS4video use case. The selected distortion threshold were 4% and 40%, more conservative the former and aggressive the latter. It can be noticed that the three curves are almost overlapped for the overall video sequence duration, thus resulting in a good QoS for both distortion threshold settings. The average distance errors between DBS4video curves and the no-DBS4video one are of 7% for the conservative value and 9,4% for the aggressive one.

Table.16 shows QoS and power saving results for the overall target video using the DBS4video framework. Two different sections can be highlighted in the target video sequence(see Figure.15):
1. indoor scenes, characterized by the predominance of dark pixels (the translucent blue box);
2. outdoor scenes, with frame showing mainly bright images (the translucent orange box).
The power breakdown for the indoor and outdoor sub-sections is also reported. The power savings take into account both LCD, CPU and IPU module.

A distortion threshold of only 4% brings to a SSIM equal to 0,8278 and a $DeltaE_{ab}$ equal to 0,9381. Considering that the two QoS metrics provide a score equal to zero comparing a white againts a black image, the obtained scores are high QoS values.

It can be noticed that with only 4% of admitted distortion, the DBS4video framework is able to achieve almost 10% of

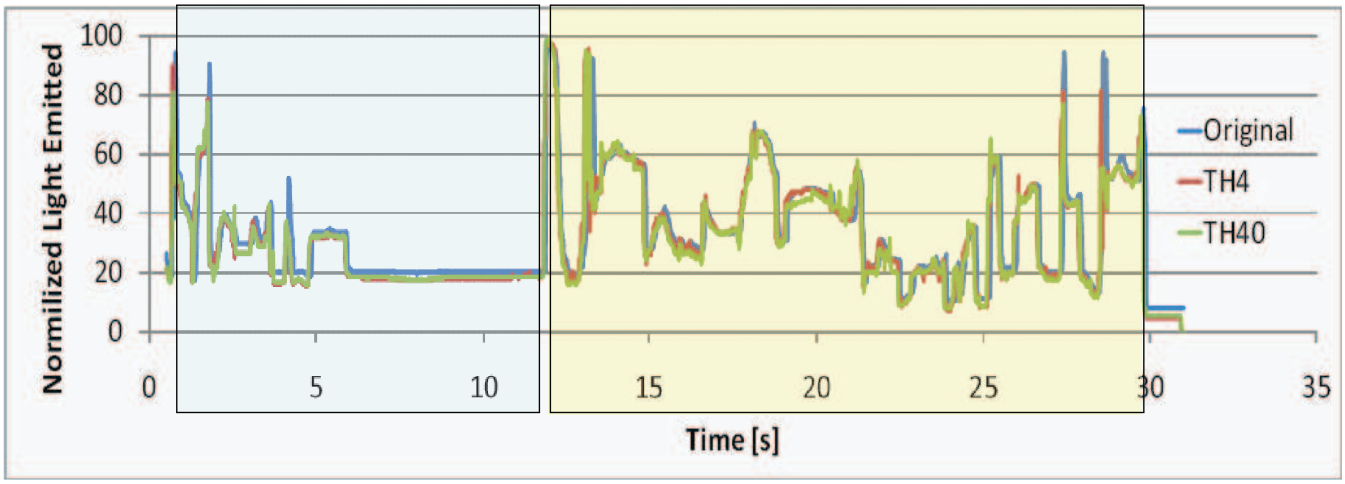

Figure 15: Terminator 3: emitted light traces

Video	Frame Rate	Frame Resolution
Terminator 3	24 fps	352x192

Table 4: Video stream features

Terminator 3	QoS		Power Savings		
Dth	*SSIM*	*DeltaEab*	*Entire Video*	*Indoor Scene*	*Outdoor Scene*
4%	0,8278	0,9381	9,8%	22,5%	4,4%
40%	0,7765	0,9134	16,4%	23,9%	12,6%

Figure 16: Terminator 3: Power savings and QoS results

power savings. We have to point out that the DBS4video power savings are frame dependent. Looking at the power breakdown for the indoor and outdoor sub-sections, the dark video sequence can allow more aggressive backlight scaling (which corresponds to a power saving equal to 22,5%) w.r.t the bright one, which forbids the DBS4video to achieve deeper savings (4,4%).

Figure.16 shows also that with a more aggressive distortion threshold setting, we can achieve a better power saving on the entire video (16,4%). If we compare the saving improvements between indoor and outdoor sequences, the latter is greater. Analysing the frame structure of both subsequences, the indoor scenes are characterized with higher number of bright spotted regions which prevents our DBS4video policy (see Section.4.2) to apply an even more aggressive backlight scaling.

7. CONCLUSIONS

Battery-operated hand-held systems are widely equipped with color LCDs which require backlight. This is one of the dominant power consumers of the total system power. We presented a techniques for reducing the power requirement of display, called DBS4video. The proposed algorithm is fast and effective for reducing the energy consumption while maintaining the designated video quality. Our experiment shows that by applying our technique, up to 23,9% power can be saved. The DBS4video framework is applicable to most battery operated mobile multimedia terminal devices.

8. REFERENCES

[1] NXP Nexperia mobile multimedia application processor PNX4009 http://www.nxp.com/

[2] STMicroelectronicsŠ Nomadik Application Processor. http://www.st.com/

[3] NEC Electronics MP211, an application processor for mobile phones. http://www.necel.com/

[4] Kerofsky L. abd Daly S. Brightness Preservation for LCD Backlight Dimming. In Sharp Technical Journal, ISSN:0285-0362, NO.95, PAGE.50–57 (2007)

[5] Shim H., Chang N. and Pedram M. A Backlight Power Management Framework for Battery-Operated Multimedia Systems. In IEEE Des. Test, ISSN:0740-7475, VOL.21, NO.5, PAGE.388–396 (2004)

[6] Chang N., Choi I. and Shim H. DLS: dynamic backlight luminance scaling of liquid crystal display. In Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, ISSN:1063-8210, VOL.12, NO.8, PAGE.837–846 (2004)

[7] Cheng W. and Chao C. Perception-guided power minimization for color sequential displays. In Proceedings of the 16th ACM Great Lakes Symposium on VLSI, 290-295, 2006.

[8] Cheng W., Hsu C. and Chao C. Temporal vision-guided energy minimization for portable displays. In Proceedings of the 2006 international Symposium on Low Power Electronics and Design, ISLPED '06. 89-94, 2006.

[9] Cheng W., Hou Y. and Pedram M. Power Minimization in a Backlit TFT-LCD Display by Concurrent Brightness and Contrast Scaling. In Proceedings of the Conference on Design, Automation and Test in Europe - Volume 1. 2004.

[10] Cheng L., Bossi S., Mohapatra S., El Zarki M., Venkatasubramanian N. and Dutt N. Quality Adapted Backlight Scaling (QABS) for Video Streaming to Mobile Handheld Devices. citeseer.ist.psu.edu/722345.html

[11] Cheng L., Mohapatra S., El Zarki M., Dutt N. and Venkatasubramanian N. Quality-based backlight optimization for video playback on handheld devices. Adv. MultiMedia 2007, 1 (Jan. 2007), 4-4.

[12] Iranli A. and Pedram M. DTM: dynamic tone mapping for backlight scaling. In Proceedings of the 42nd Annual Conference on Design Automation. DAC '05, 612-617. 2005.

[13] Iranli A., Lee W. and Pedram M. Backlight dimming in power-aware mobile displays. In Proceedings of the 43rd Annual Conference on Design Automation. DAC '06. 2006.

[14] Fairchild M. D. Color appearance models Wiley ed.

[15] Iranli A., Lee W. and Pedram M. HVS-Aware Dynamic Backlight Scaling in TFT-LCDs. IEEE Trans. on Very Large Scale Integration (VLSI) Systems, Vol. 14, N.10, 2006.

[16] Iranli A., Fatemi H. and Pedram M.. HEBS: Histogram Equalization for Backlight Scaling. Design, Automation and Test in Europe, 2005. Proceedings, pp. 346-351 Vol. 1, 2005.

[17] i.MX31: Multimedia Applications Processor http://www.freescale.com

[18] The SSIM Index for Image Quality Assessment http://www.ece.uwaterloo.ca/ z70wang/research/ssim/