Improving the SNR During Color Image Processing while Preserving the Appearance of Clipped Pixels

Sergio Goma, Milivoje Aleksic
AMD, 75 Tiverton Court, Markham, ON, Canada, L3R-9S3

ABSTRACT

An image processing path typically involves color correction or white balance resulting in higher than unity color gains. A gain higher than unity increases the noise in that respective channel, and therefore degrades the SNR performance of the input signal. If the input signal does not have enough SNR to accommodate the extra gain, the resultant color image has increased color noise. This is the usual case for color processing in cell phone cameras, which have sensors with limited SNR and high color crosstalk. This phenomenon degrades images more as illuminants differ from D65. In addition, the incomplete information for clipped pixels often results in unsightly artifacts during color processing. To correct this dual problem, we investigate the use of under unity color gains, which, by increasing the exposure of the sensor, would improve the resultant SNR of the color corrected image. The proposed method preserves the appearance of clipped pixels and the overall luminance of the image, while applying the appropriate color gains.

Keywords: color processing, improving SNR, reducing noise, clipped pixels, cell-phone camera;

1. INTRODUCTION

An increasingly important topic is the SNR response of a color sensor. Color filter design has a trade-off between color separation and overall luminance attenuation: higher color separation implies lower intensity signals at the pixel level, whereas higher pixel signal level implies lower color separation. Also, image quality requirements demands a good color separation which in context of linear color processing and RGB color filters implies high color gains.

This paper is structured as follows:

Section 2, “Problem Definition” provides a generic block diagram of a color processing pipe that is used to derive SNR degradation using error propagation calculation considering above unity overall color gains. This is used to propose a solution to improve SNR by exposing the sensor longer and use subunit color gains in color processing. The issues raised by this type of processing are examined and some solutions are suggested. This section concludes with a proposed solution to the problem of color processing using under unity overall gains. Section 3, “Implementation”, details a hardware implementation for a 5mpix sensor. Section 4 presents the results.

2. PROBLEM DEFINITION

Today’s CMOS sensors used in cell-phone cameras are in range of 5mpix with pixel sizes of 1.75um. They typically deliver a noise performance around 36dB SNR. Also, a significant use case for cell-phone camera is indoor low-light or office light, which both tend to have tungsten as the dominant illuminant (color temperature). For the purpose of this analysis, a generic image processing pipe as shown in Figure 1 is used. It has the following features:

Real-Time Image Processing 2008, edited by Nasser Kehtarnavaz, Matthias F. Carlsohn,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6811, 681102, © 2008 SPIE-IS&T · 0277-786X/08/$18

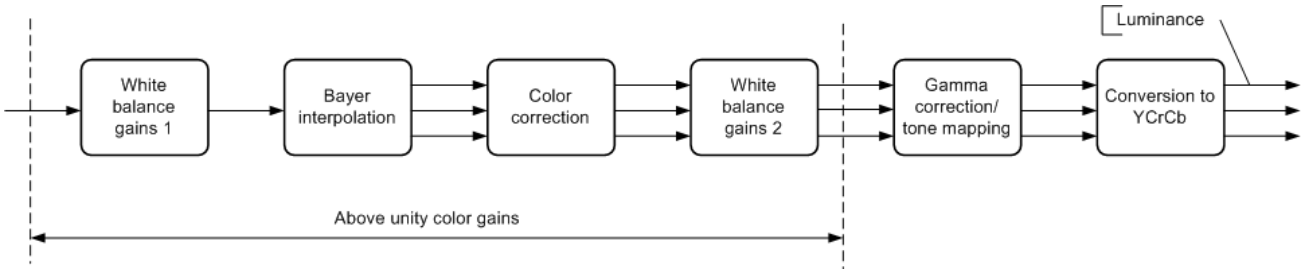


Figure 1 Generic image processing pipe

- Exposure is set using a closed loop algorithm[1]
- Color processing is done in an RGB space[2]
- linear color processing is done in form of multiplication with at least one color correction matrix and also 'white balance' is done in form of a multiplication with at least one set of gains[3];
- the color correction matrix is white point preserving[4];
- it is common practice to have 'white balance' gains above unity (or 'to clip on white');

Therefore a generic linear color processing in RGB space would have the form:

$$\begin{vmatrix} g_{RR} & g_{RG} & g_{RB} \\ g_{GR} & g_{GG} & g_{GB} \\ g_{BR} & g_{BG} & g_{BB} \end{vmatrix} = \begin{vmatrix} ag_R & 0 & 0 \\ 0 & ag_G & 0 \\ 0 & 0 & ag_B \end{vmatrix} \cdot \begin{vmatrix} cc_{RR} & cc_{RG} & cc_{RB} \\ cc_{GR} & cc_{GG} & cc_{GB} \\ cc_{BR} & cc_{BG} & cc_{BB} \end{vmatrix} \cdot \begin{vmatrix} pg_R & 0 & 0 \\ 0 & pg_G & 0 \\ 0 & 0 & pg_B \end{vmatrix}$$

Figure 2 Overall linear color processing

Where cc_i is the color correction matrix and ag_i and pg_i are sets of white balance gains and g_i are overall color processing gains;

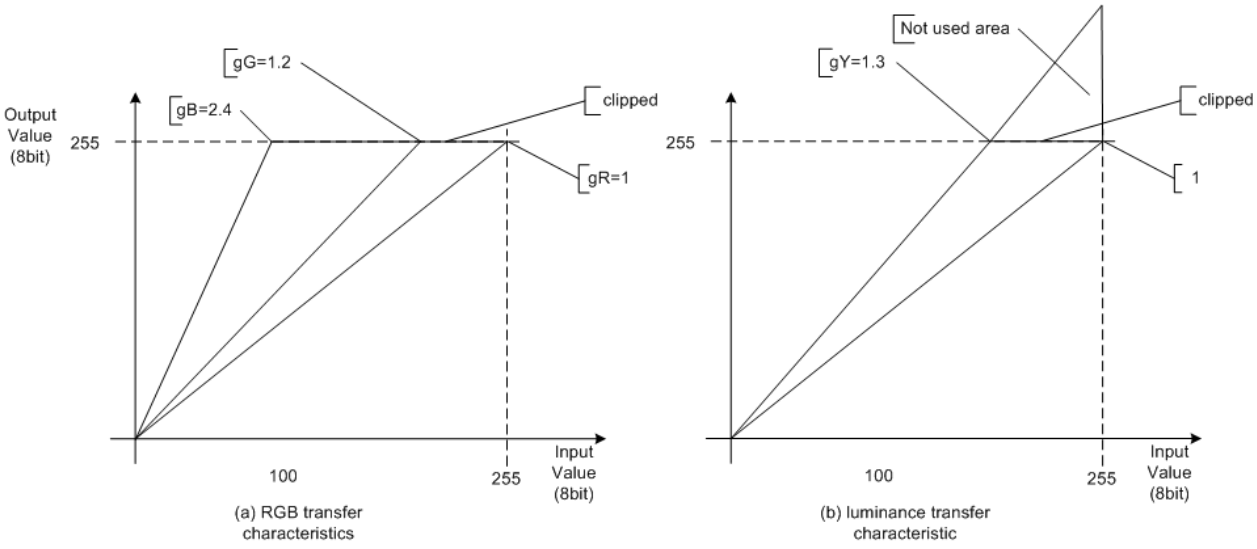


Figure 3 transfer characteristics (a) RGB after color processing (b) equivalent luminance after color processing

In order to have a picture correctly exposed, the exposure that would otherwise be used by the sensor must be scaled down by an amount equivalent to what the transfer characteristic of the luminance has increased due to over unity color processing. This is equivalent to considering the sensor to be under exposed to compensate for the gain introduced by the color processing. Considering this, there is an associated SNR degradation of input image because of color processing. If simple error propagation is used for a linear color processing as shown in Figure2, and the noise in the sensor, for simplicity, is considered the same for all color channels before color processing, then the degradation introduced by color processing can be expressed as:

$$Sdev_R = \sqrt{g_{RR}^2 + g_{RG}^2 + g_{RB}^2} \cdot \sqrt{3} \cdot Sdev_{sensor} \quad (1)$$

$$Sdev_G = \sqrt{g_{GR}^2 + g_{GG}^2 + g_{GB}^2} \cdot \sqrt{3} \cdot Sdev_{sensor} \quad (2)$$

$$Sdev_B = \sqrt{g_{BR}^2 + g_{BG}^2 + g_{BB}^2} \cdot \sqrt{3} \cdot Sdev_{sensor} \quad (3)$$

Where $Sdev_{sensor}$ is the read-out noise of a pixel. A more rigorous estimation for CMOS pixel noise can be used from[5], and similarly, a more complex color processing error propagation estimation can be found in[6,7]. In our case, the above is sufficient to illustrate that signal degrades fast in color processing and a rather high penalty needs to be paid to achieve good color separation and 'white balance', especially under illuminants that are not spectrally uniform (ie. tungsten or fluorescent).

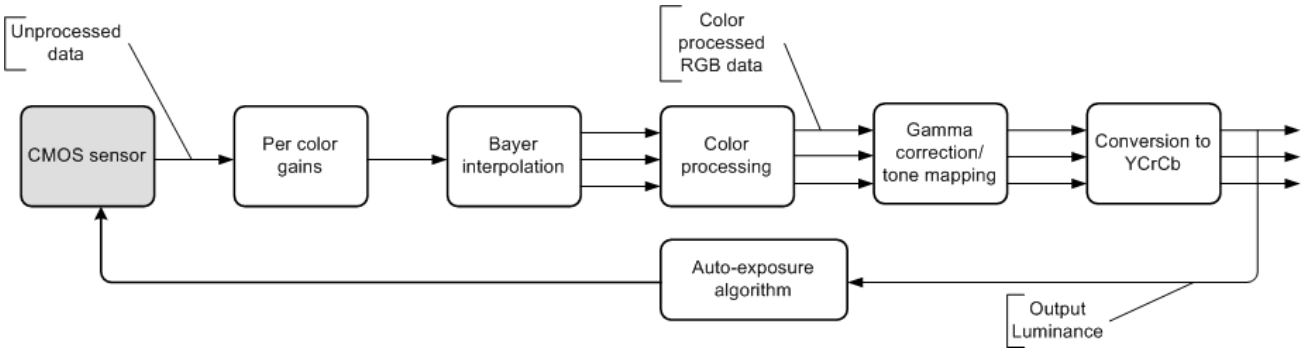


Figure 4 Real time linear color processing path with a closed loop auto-exposure

This is exemplified in Figure7, where a captured image was analyzed for color distribution as well as luminance.
The color distribution is depicted using a 2D projection in HSV space as described in [3], whereas for luminance the histogram was used. In Figure7 (a) the color map and the luminance histogram are calculated before any color processing and in (b), after color processing. It can be seen from these histograms that color processing with gains greater than unity will increase the luminance. In the real processing pipeline, luminance of the processed color is used to adjust exposure to obtain a histogram as depicted in (b). Therefore, the image delivered by sensor will be less exposed than when unity gains are used..

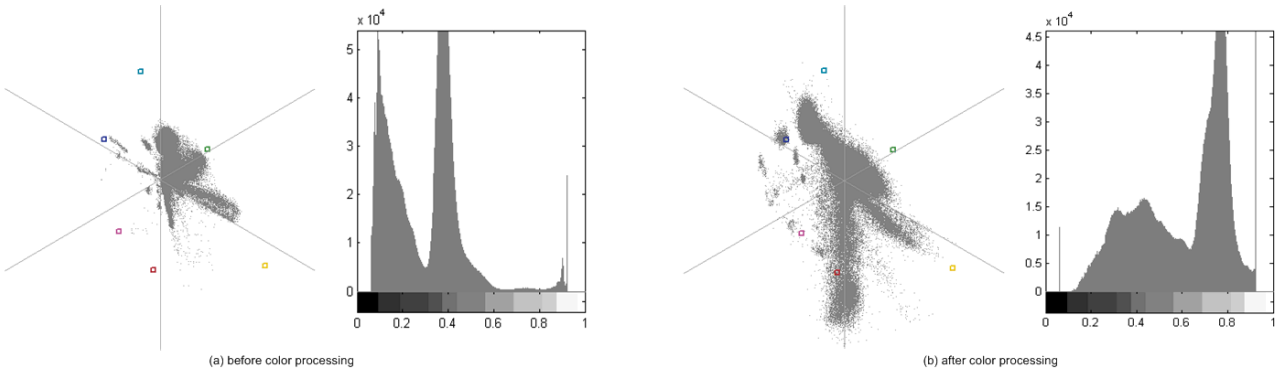


Figure 5 Color distribution and luminance histogram: (a) before color processing (b) after color processing

Also, the example image used to generate Figure7 has some clipped areas as is apparent in the histogram. This could make using gains less than unity problematic.

2.1 Proposed Algorithm

Generally, if no pixels are clipped in the captured image, under unity color processing cold be possible, as shown in Figure5. The steps involved are:

- on the captured image, before color processing, estimate input overall luminance;
- apply normal color processing and estimate overall output luminance;
- estimate the overall luminance increase;
- increase sensor exposure with the amount that luminance has increased after color processing;
- scale down the captured image before color processing by the same amount on all channels;

Of course, to increase the SNR, this implies a live closed loop, where at the system level, the auto-exposure algorithm would overexpose the sensor with the amount needed by color processing, and in this case, by scaling down the captured image before color processing, an overall under unity gain color processing can be achieved.

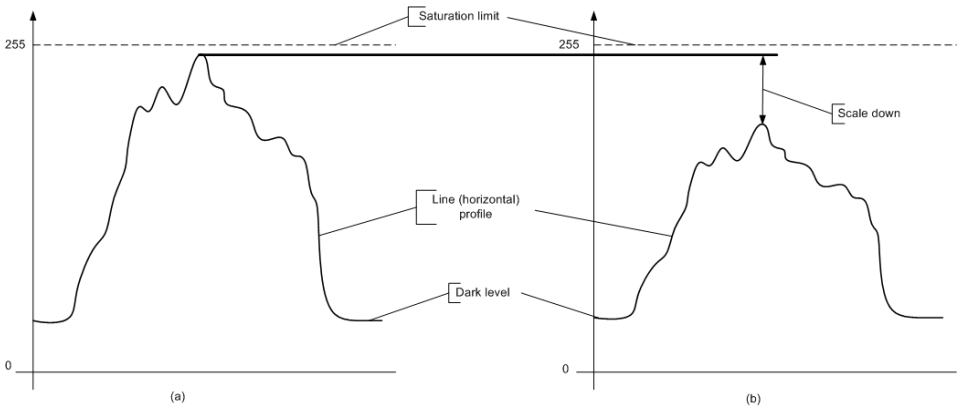


Figure 6 Under unity gains when profile is not clipped: (a) input profile (b) scaled down profile

Conceptually, this seems a straightforward idea. However, if the captured image happens to have clipped pixel areas before color processing, as is the case with scenes that are not particularly uniform, using this approach becomes problematic, as illustrated in Figure 6.

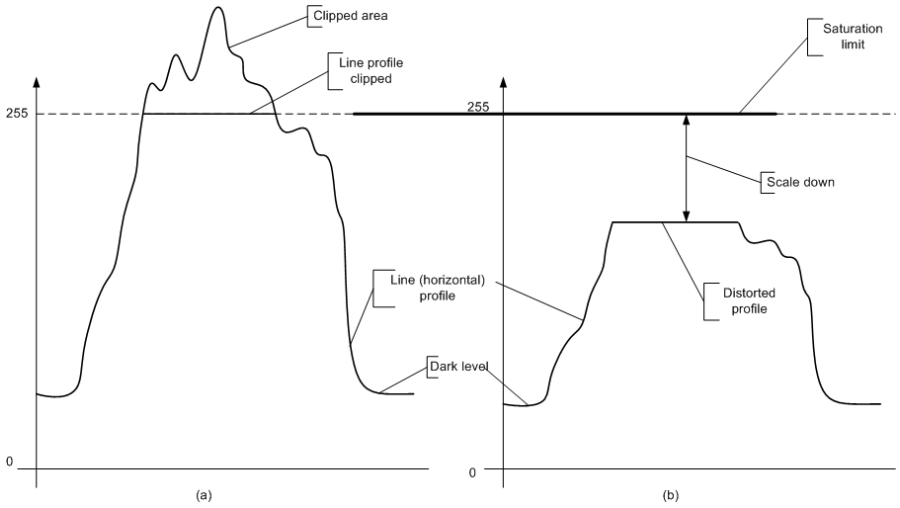


Figure 7 Under unity gains when profile is clipped: (a) input line profile – clipped area for reference (b) scaled down profile

Figure 5 and Figure 6 show a horizontal profile through one of the RGB color channels in the captured image. In Figure5 the profile has no clipped pixels, and it can be scaled down without problems, but Figure 6 has a clipped area (a), that when it's scaled down (b), the profile information will be distorted. As the other color channels might not be clipped in (a), the scaled down area in (b), when scaled down by the same amount on all channels – will likely have a different hue. A solution to this is to recover the clipped areas before color processing, as illustrated in Figure7 (a) or, to not scale the clipped areas. Using the latter approach, while hues are preserved a new problem arises: perceptually the boundary between clipped areas and adjacent regions becomes noticeable. To alleviate this issue, we propose blurring the boundary of clipped areas. Blurring the boundary will not restore the edge newly created by downscaling, but it will locally decrease the contrast. As the clipped area contains only hue information, this creates a less noticeable effect as suggested in Figure7(b). We found that a 3x3 blur filter gives reasonable results. A larger blur will smooth more the edge, but as the PSF of a good cell-phone camera module is between 3 to 5 pixels wide, will reduce the contrast more in the clipped areas making it again noticeable. However, a new problem arises when the clipped area is around 3x3 pixels or less. In this case, blurring the area becomes noticeable, as the filter is bigger than the original clipped area (this is also the case of any remaining bad pixels). To correct for this case, we also filtered the clipped areas and considered only areas of 3x3 or more. To summarize, the proposed algorithm comprises of following steps:

1. before color processing, extract from the captured image a clipped pixels area map (build a 1-bit map for each color plane with location of clipped pixels in that color channel);
2. apply a blur kernel to each color clip map to obtain a final clip mask for each color channel;
3. downscale pixels values in the captured image;
4. restore clipped areas in each plane by forcing the pixels in the downscaled image to clip according to the clip masks;
5. using the clip masks and result from step 4, blur each color plane of the image from step 4 according to the clip masks.
6. apply regular color processing afterwards.

For both blur operations, clips masks and image we used a regular 3x3 low-pass kernel: $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

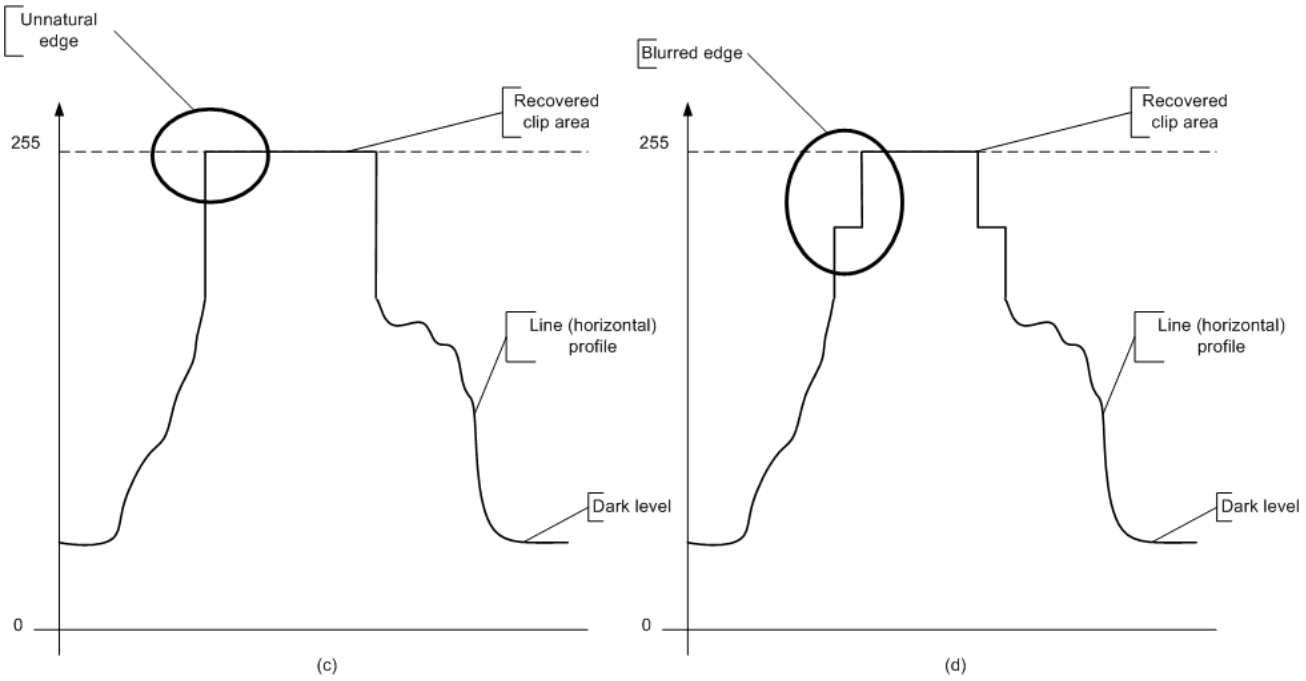


Figure 8 Restoring clipped areas in under unity gains when profile is clipped: (a) recovered clipped area with unnatural edges (b) blurred edge on clipped areas

3. IMPLEMENTATION

A real-time implementation of the proposed algorithm is suggested in Figure8.

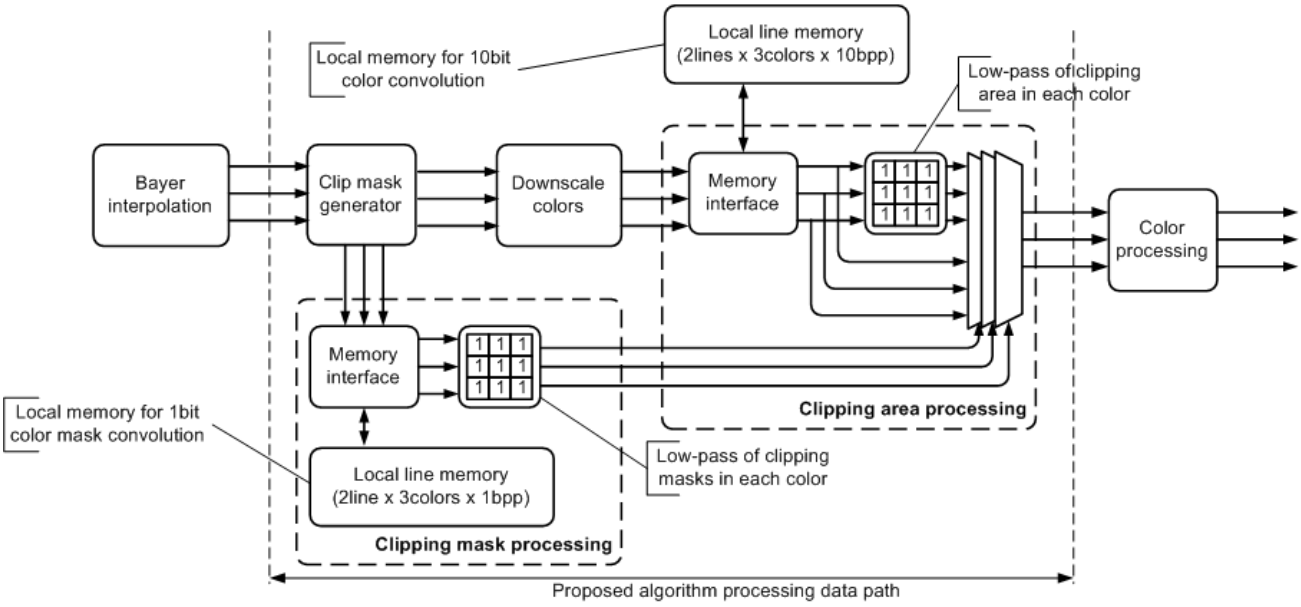


Figure 9 Hardware implementation of proposed algorithm

As this implementation was intended for a 10bit image processing hardware color pipe, it is a hardware implementation of the proposed algorithm. The implementation is designed to be inserted in the image processing data path before any color processing block as a stand alone block. The block accepts under unity gains in 'Downscale colors' and 10bit RGB color data inputs. The output of the block is RGB 10bit data. The under unity gains are controlled by a software auto-exposure algorithm. Also, for convolutions, two local memory blocks were used: one to store the clip masks and one to store image data. As memory blocks were local, and size of kernel was 3x3, the memory requirements were as follows (used to accommodate a 5mpix CMOS sensor):

- Image kernel: 2592x10x3x2 bits (2592 pixels/line, 10bit/pixel, 3 color planes, 2lines);
- Mask kernel: 2592x1x3x2 bits (2592 pixels/line, 1bit/entry, 3 color planes, 2lines);

Performance of the suggested implementation was 1pixel/clock in 90nm technology with at a max. 150MHz pixel clock.

4. RESULTS

The method works well to improve the SNR, which was expected, given the fact that the SNR improvement is achieved by exposing the sensor longer, and the improvement can be easily quantified objectively using simple error propagation metrics (as outlined in (1)(2)(3)). However, clipping area restoration it's done on a subjective basis, and the improvements are perceptual, therefore we'll present results highlighting this part of the proposed method.

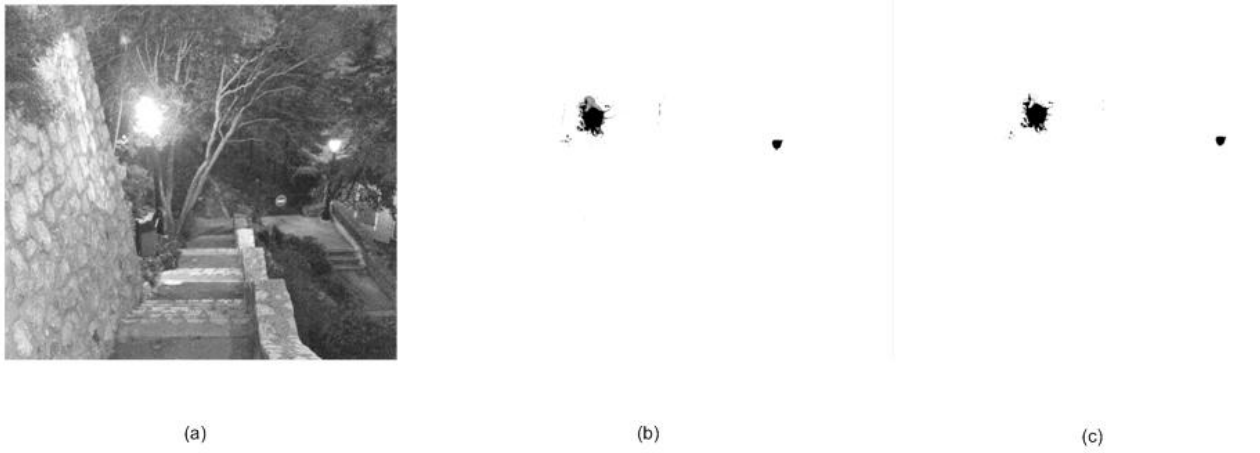


Figure 10 Image and clip masks (a) Processed picture (b) clip masks (c) clip masks blured

In Figure10 an example image was processed using the proposed approach. Image is taken under low-light conditions and the color of illuminant is tungsten. The overall color gains applied were:

gR=0.48 gG=0.62 gB=0.44

The luminance scaling down factor was 2.25x and the number of clipping pixels present in the picture was 20842 or around 0.41% of total size (image size is 2592x1948 or 5mpix). The noise improvement in this case was on average around 3dB.

The darkest clipping pixel had

R=1 G=0.6 B=0.47 (normalized value)

Average color point of clipped area was

R=0.99 G=0.94 B=0.91 which means that using under unity without restoration of clipping areas would've produced a reddish hue in clipped areas.

In Figure11 a zoom in area of the clip mask is shown. Here, clip masks from RGB planes are overlapped with different shades of grey. As it can be seen, there's a contour of green pixels that are clipping before red pixels, as they need a lower gain.

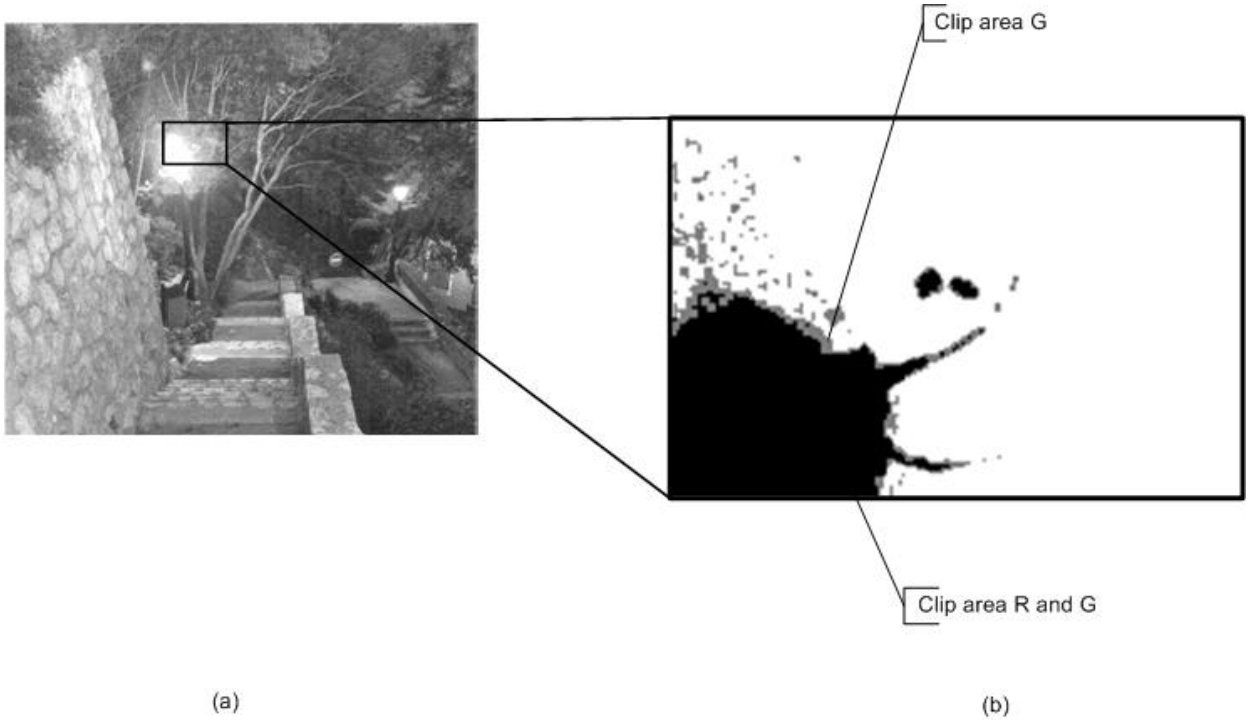


Figure 11 Clip masks (a) processed image (b) zoomed clipped area

Further in Figure12, the same area is presented to show the difference between the use of a blur of 3x3 and 5x5 on the clipped edge. A blur of 5x5 degrades the edge of clipped area, making the edge noticeable.

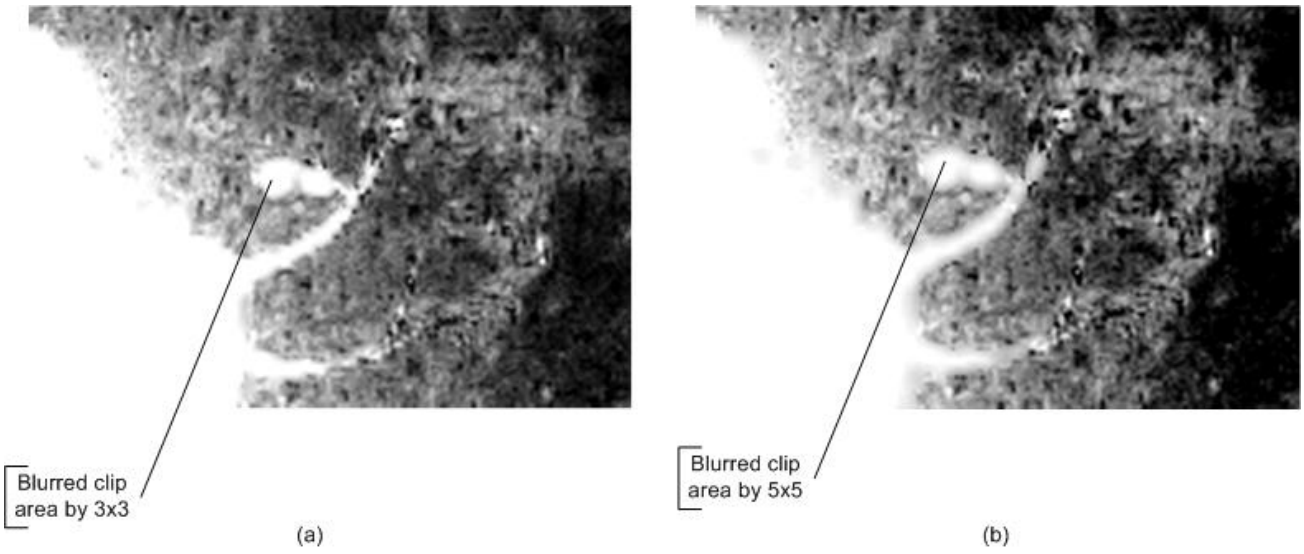


Figure 12 Blurred clipped edge detail (a) 3x3 blur (b) 5x5 blur

A similar situation arises when a clipping area less than 3x3 is blurred, the appearance is unnatural, creating similar artifacts as in Figure12(b), which is the reason the clip masks need to be blurred as well.

5. CONCLUSIONS

This paper presents a method for improving cell-phone camera SNR by extending sensor exposure time and down scaling the input signal with the amount increased by color processing. Down scaling the input signal alters clipped pixel areas in the image and requires special processing. To minimize the effect of clipped pixels a real time algorithm for tracking and processing those pixels is presented. Proposed method uses per color channel clipping pixels masks that require additional processing. Experimental results verify a reasonable perceptual preservation of the clipped pixel appearance as well as an increase of the SNR after color processing.

6. REFERENCES

1. Robert Kremens, et al. ”System Implications of Implementing Auto-Exposure on Consumer Digital Cameras “,Sensors, Cameras, and Applications for Digital Photography, Proceedings of the SPIE, Vol. 5678, pp. 100-107, 1999.
2. F. Xiao, X. Zhang, B. Fowler, “Color processing in camera phones: How good does it need to be?”, Digital Photography, Proceedings of the SPIE, Vol. 5678, pp. 96-104, 2005.
3. R. Ramanath, W. E. Snyder, Y. Yoo, M. S. Drew, “Color image processing pipeline”, Signal Processing Magazine, IEEE vol. 22, Issue 1, pp34-43, 2005
4. G. D. Finlayson, M. S. Drew, “White-point preserving color correction”, Proc. IS&T/SID 5th Color Imaging Conference, pp. 258-261, 1997
5. T. Hui, B. Fowler, A.E. Gamal, “Analysis of temporal noise in CMOS photodiode active pixel sensor”, IEEE Journal of Solid-State Circuits, Vol. 36, No. 1, pp92-101, 2001.
6. Peter D.Burns, Roy S.Berns, ”Error propagation analysis in color measurement and imaging”, Color Research & Application, vol.22, no.4,pp280-289, 1997.
7. Panagiotis E. Haralabidis, Christodoulos Pilinis, “Linear color camera model for a skylight colorimeter with emphasis on the imaging pipeline noise performance”, Journal of Electronic Imaging, Vol. 14, Issue 4, 043005, 2005.