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

Sergio Goma, Milivoje Aleksic – AMD

6811-1/EI 2008

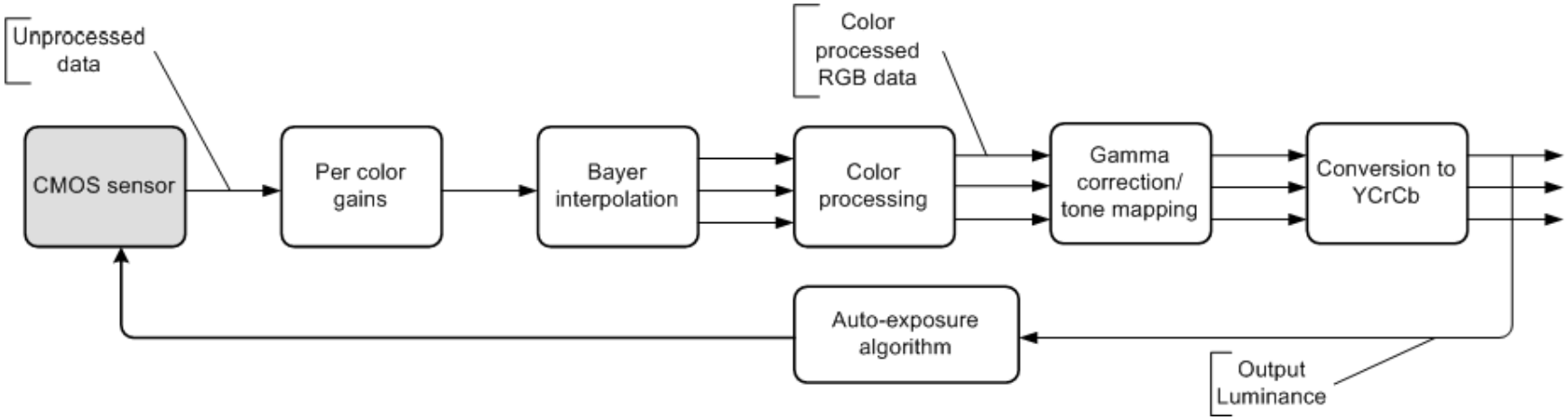
AMD

Outline

- Introduction
- Color processing with above unity gains;
- SNR degradation due to color processing -> the problem;
- Under unity color processing -> proposed solution;
- Implementation details for a hw pipe;
- Results
- Conclusions and remarks

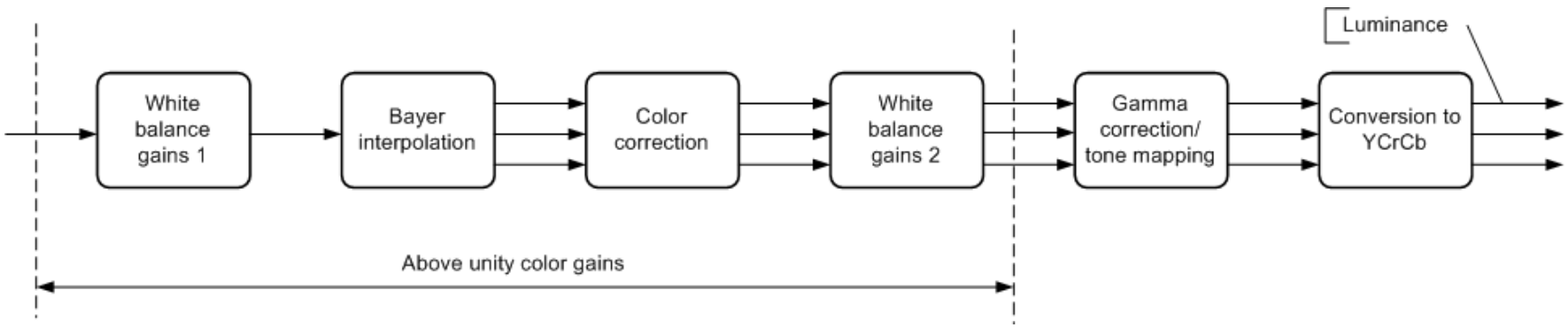
Real time linear color processing path with a closed loop auto-exposure

- System level perspective: exposure adjustment is done based on resultant luminance

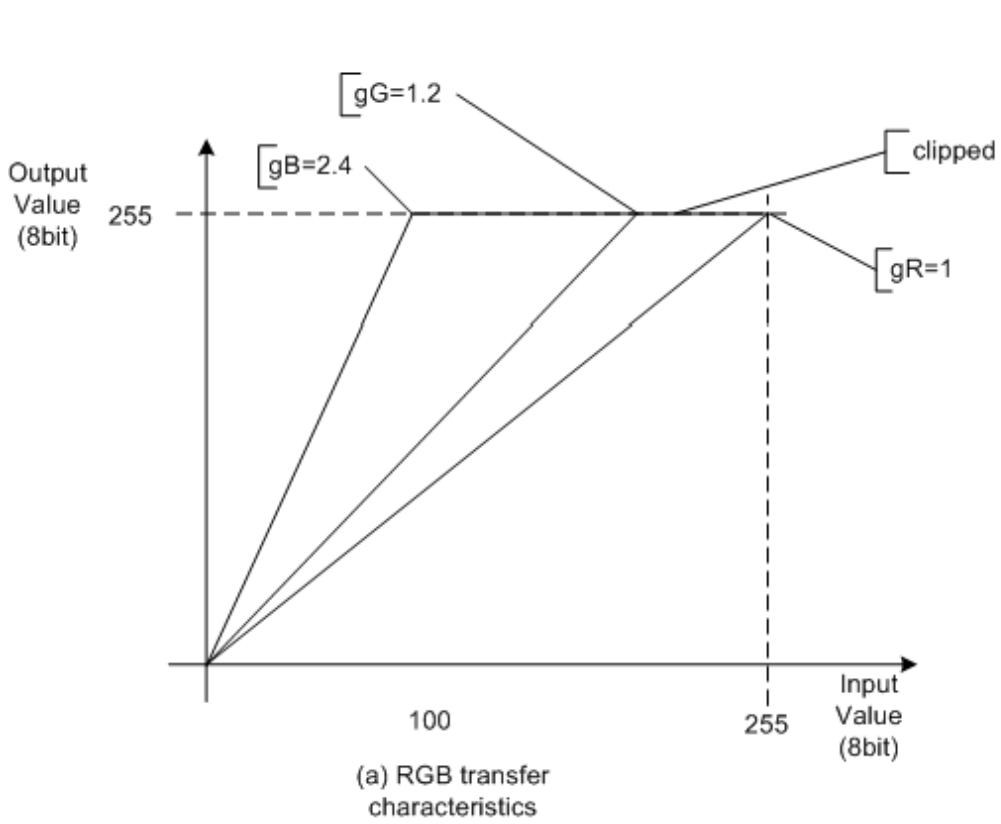


Generic color processing

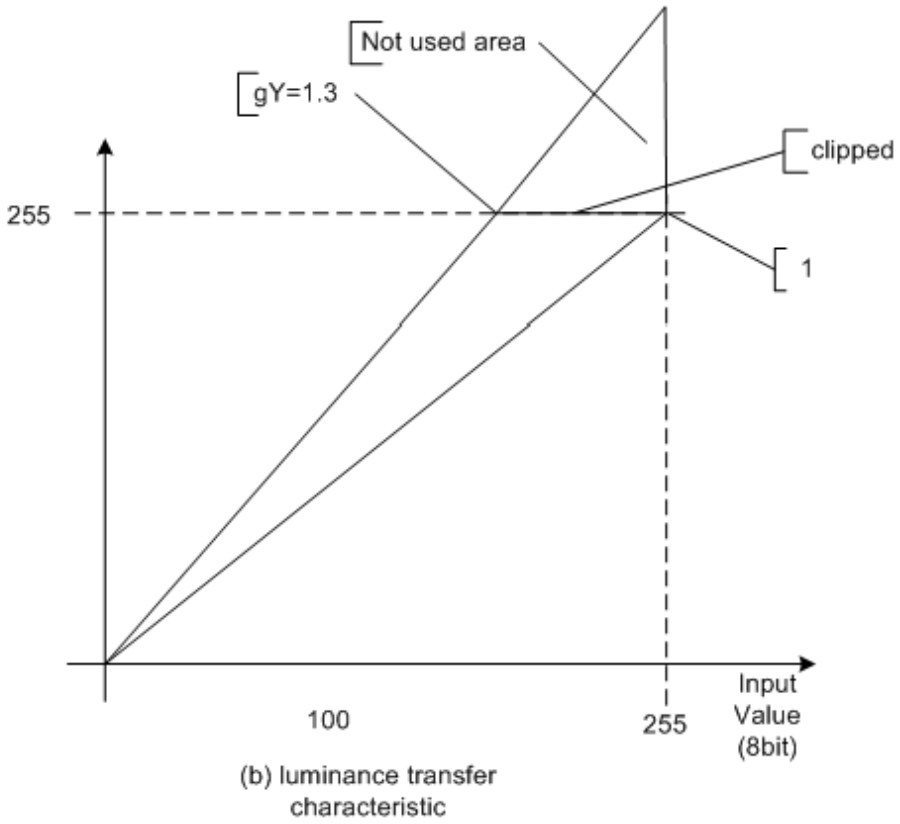
- Hardware implementation of the image processing data path is usually a good choice for the mobile market.
- The constrains are:
  - Power
  - Cost
  - Performance (speed, color quality)



Pixel Transfer Characteristics



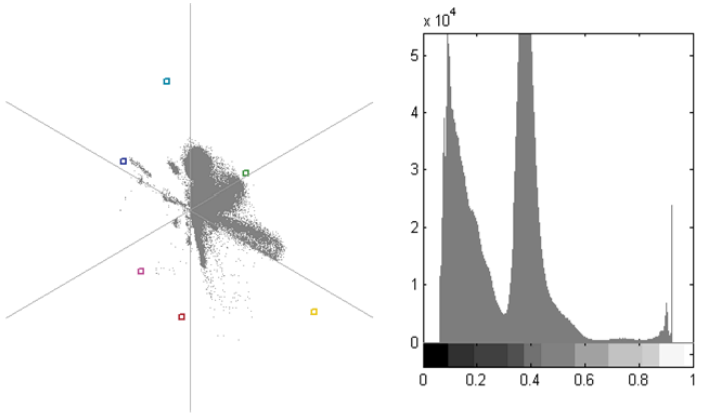
(a) RGB transfer characteristics



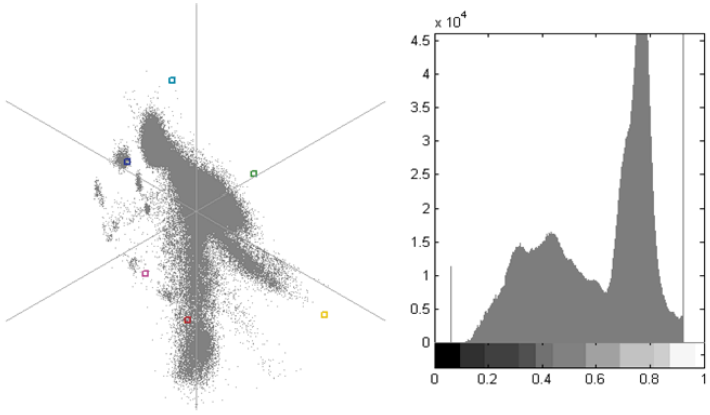
(b) luminance transfer characteristic

SNR degradation

- Linear color processing with above unity gains degrades sensor signal.



(a) before color processing



(b) after color processing

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

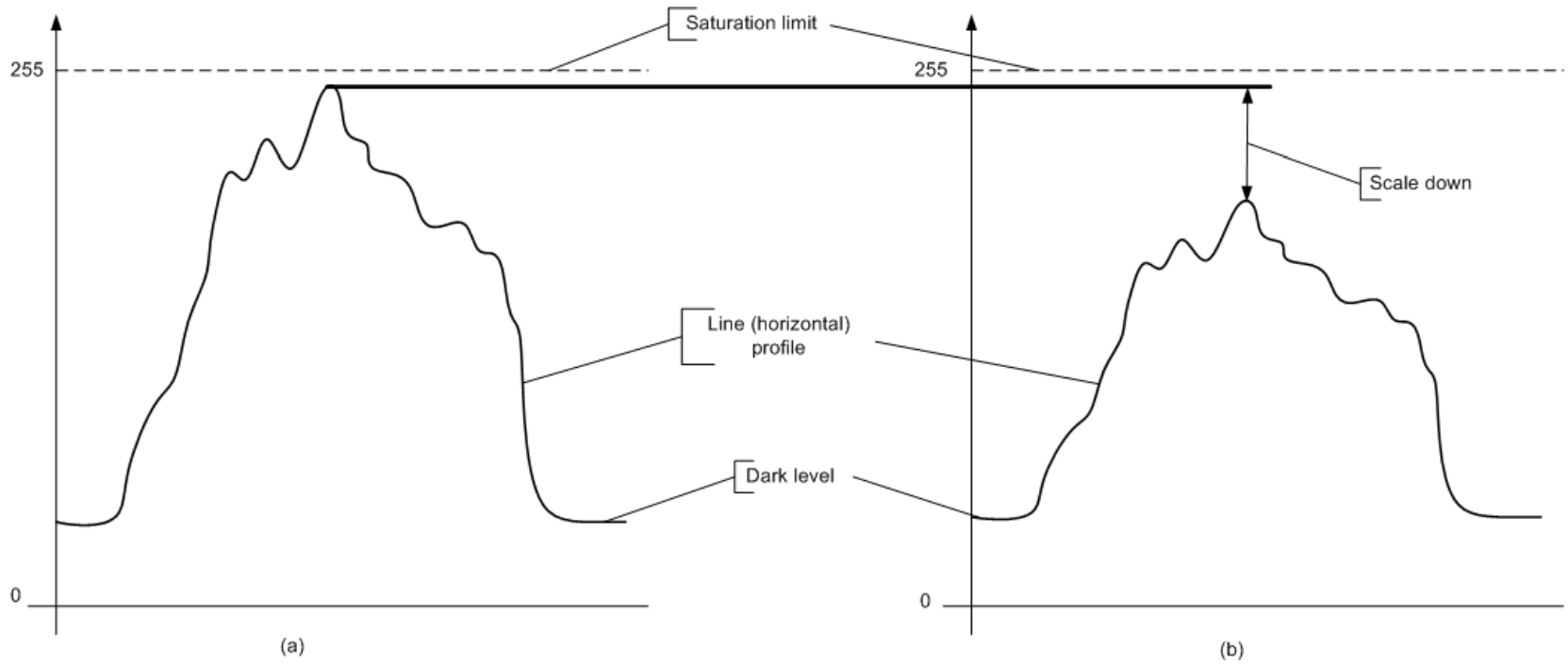
$$Sdev_R = \sqrt{g_{RR}^2 + g_{RG}^2 + g_{RB}^2} \cdot \sqrt{3} \cdot Sdev_{sensor}$$
$$Sdev_G = \sqrt{g_{GR}^2 + g_{GG}^2 + g_{GB}^2} \cdot \sqrt{3} \cdot Sdev_{sensor}$$
$$Sdev_B = \sqrt{g_{BR}^2 + g_{BG}^2 + g_{BB}^2} \cdot \sqrt{3} \cdot Sdev_{sensor}$$

Under unity color processing (first try)

- Processing steps:
  - on the captured image, before color processing, estimate input overall luminance;
  - apply ‘normal’ color processing and estimate overall output luminance (above unity gains);
  - estimate the overall luminance increase;
  - increase sensor exposure with the luminance increase due to color processing;
  - scale down the captured image before color processing by the same amount on all channels;

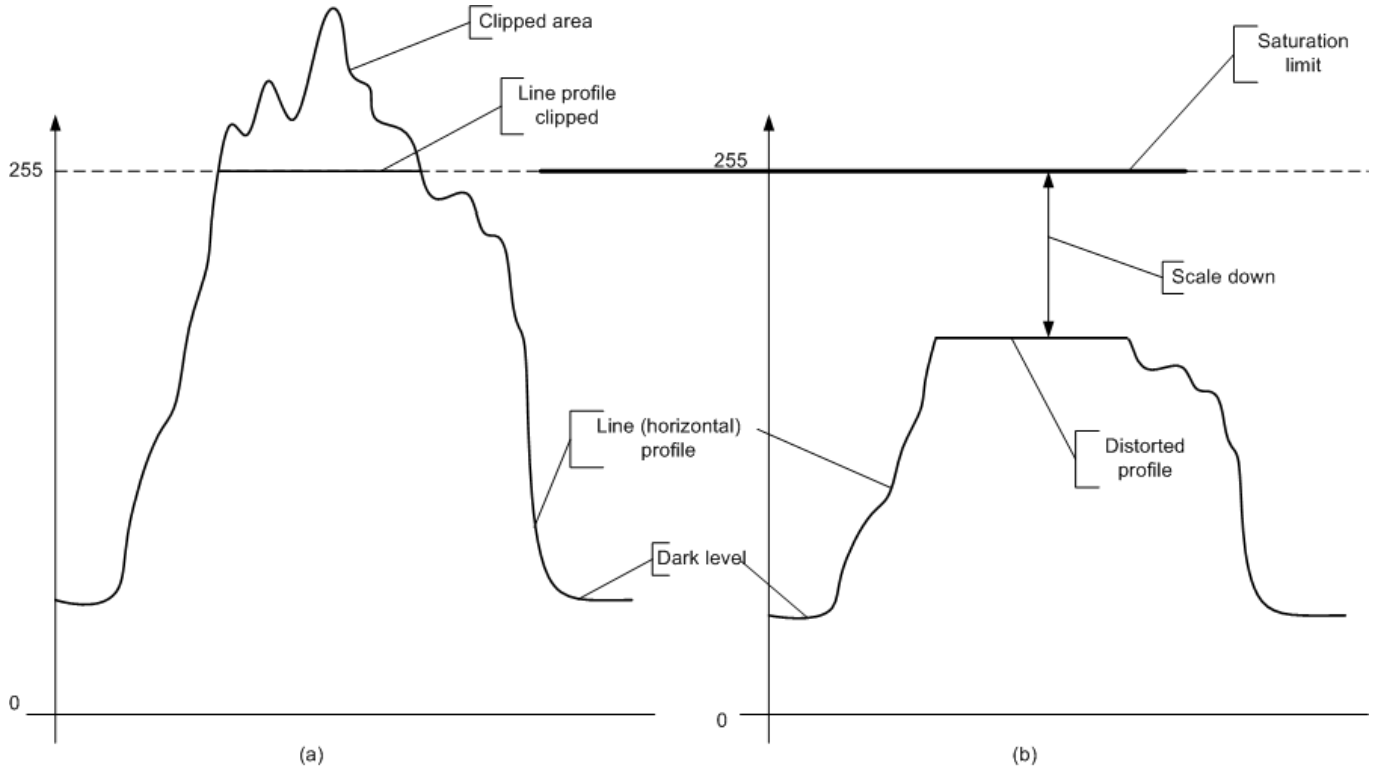
Under unity gain (no clipping)

If no clipping pixels are involved -> no problem!



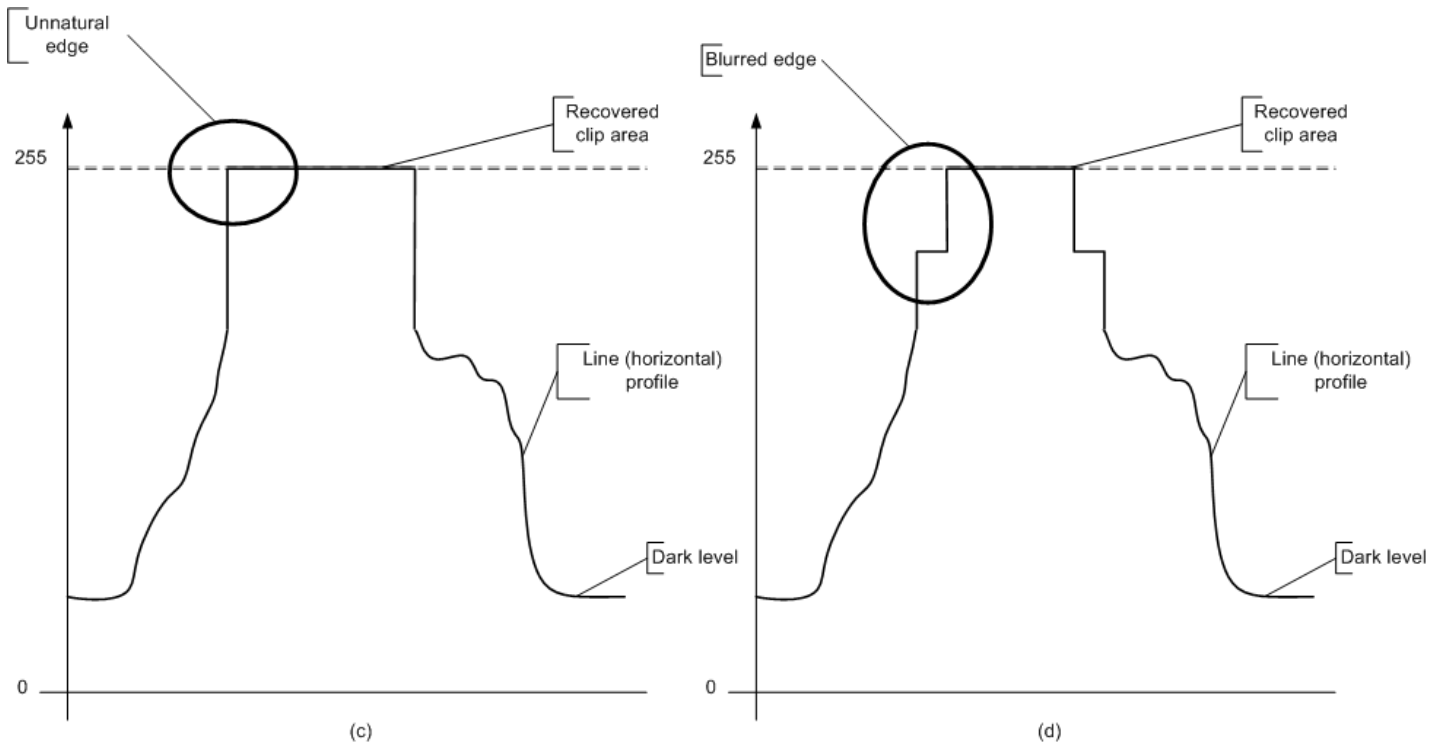
Under unity gain (clipping)

- Case2: clipped pixels are present -> downscaling the signal destroys color information in clipped area;



‘Recovered’ edge processing

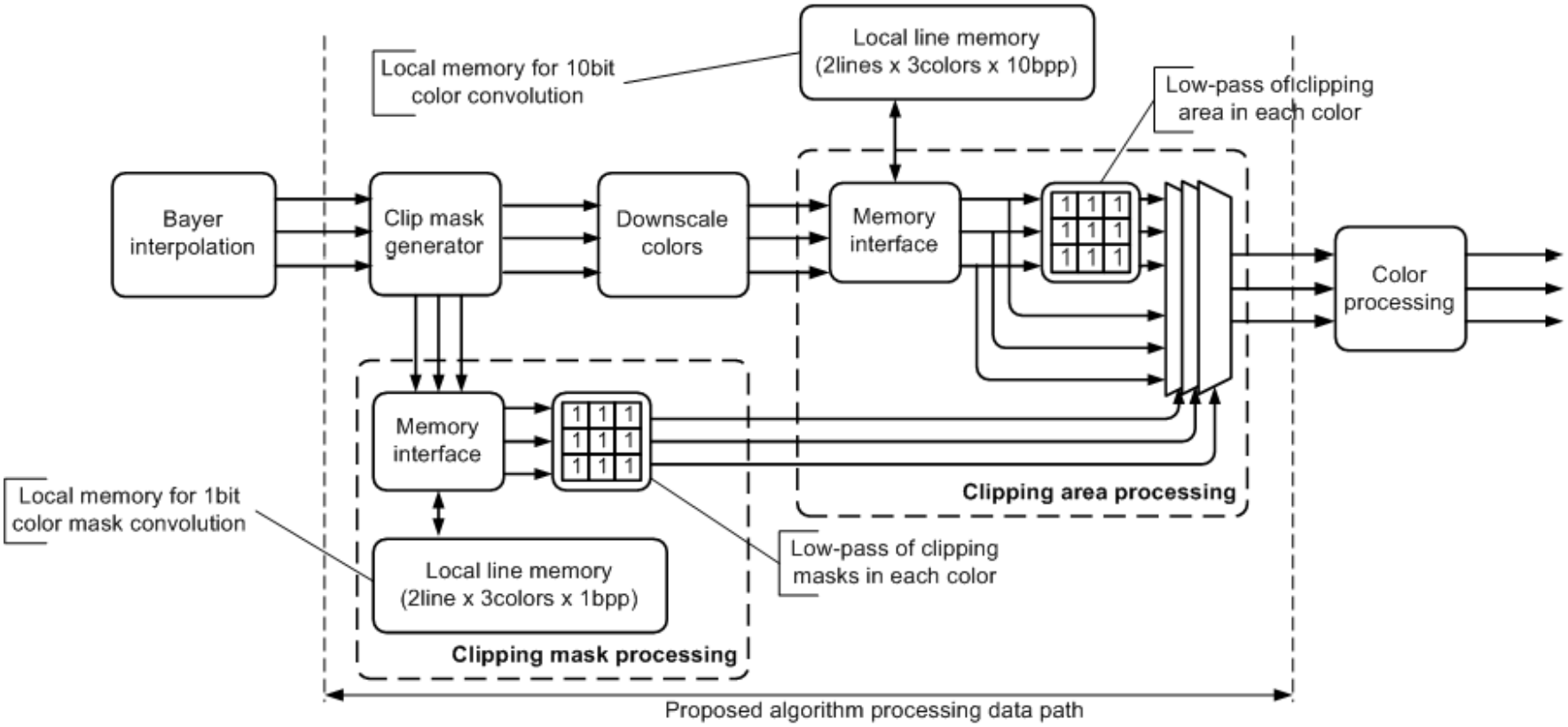
- Operation: filter clipped area contour with a simple 3x3 blur kernel;
- Underlying assumption: PSF is 3-5 pixels wide;



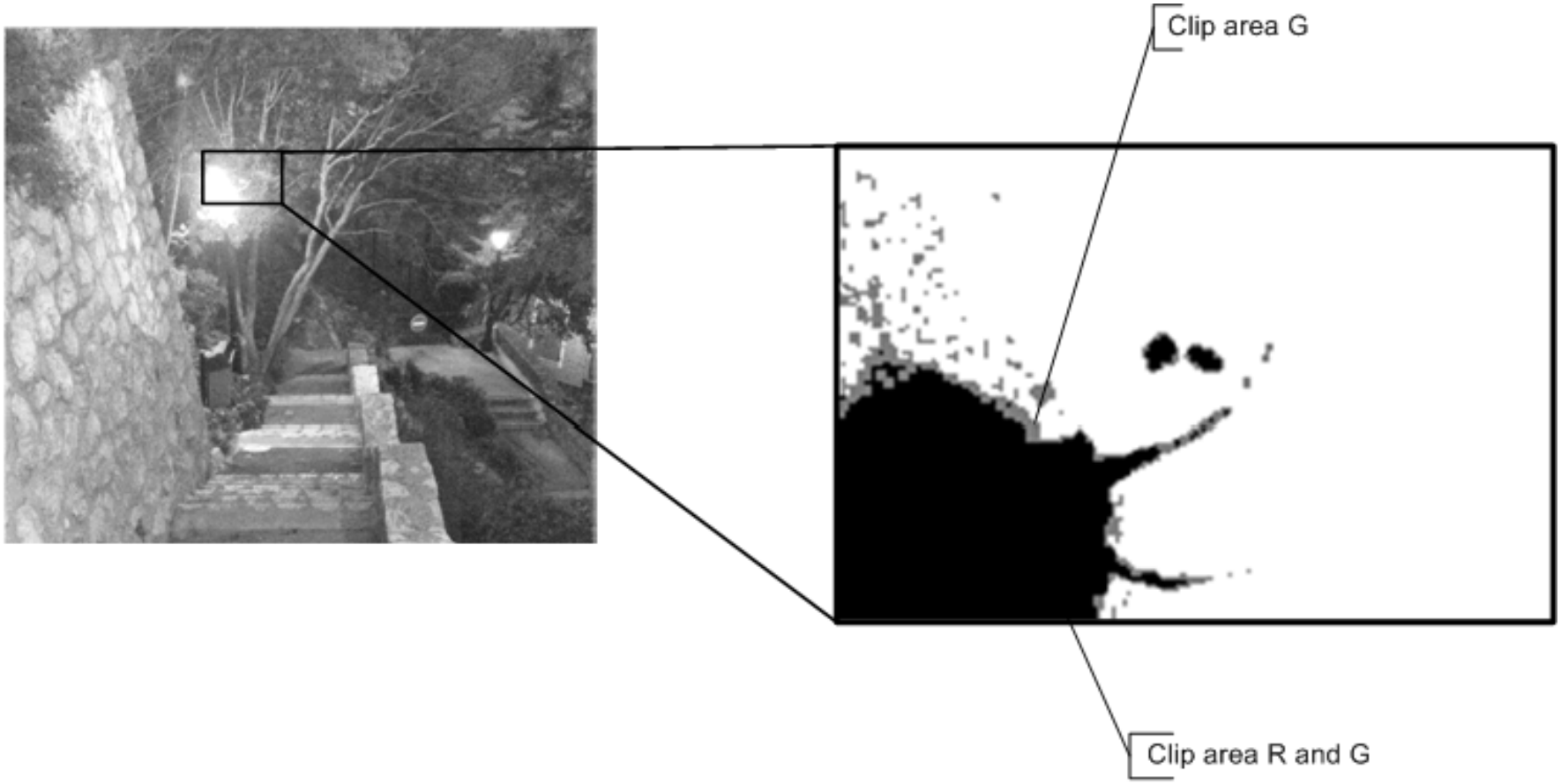
Under unity color processing – proposed method

- Processing steps:
  - on the captured image, before color processing, estimate input overall luminance;
    - **Capture clipped pixel map (blur to remove islands < PSF);**
  - apply ‘normal’ color processing and estimate overall output luminance (above unity gains);
  - estimate the overall luminance increase;
  - increase sensor exposure with the luminance increase due to color processing;
  - scale down the captured image before color processing by the same amount on all channels;
    - **Restore clipped pixel appearance (blur clipped area boundary to make it perceptually unnoticeable);**

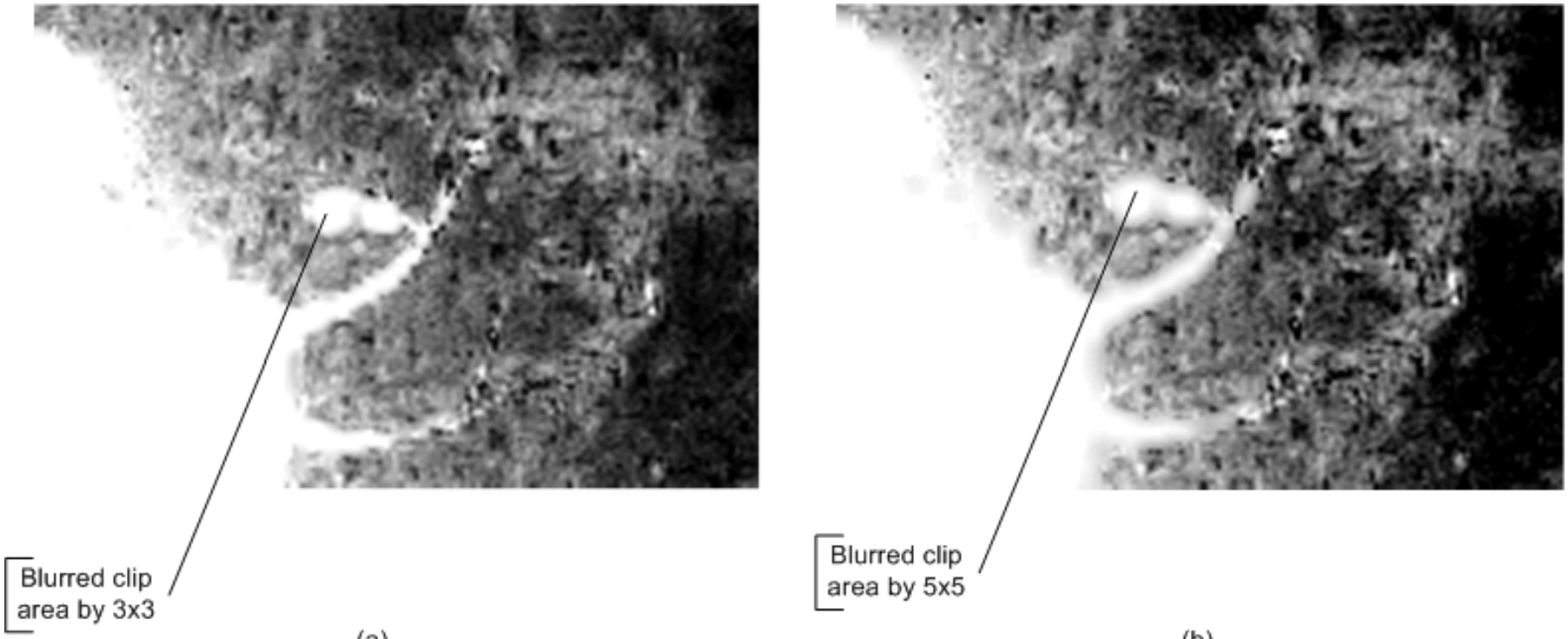
Implementation Detail



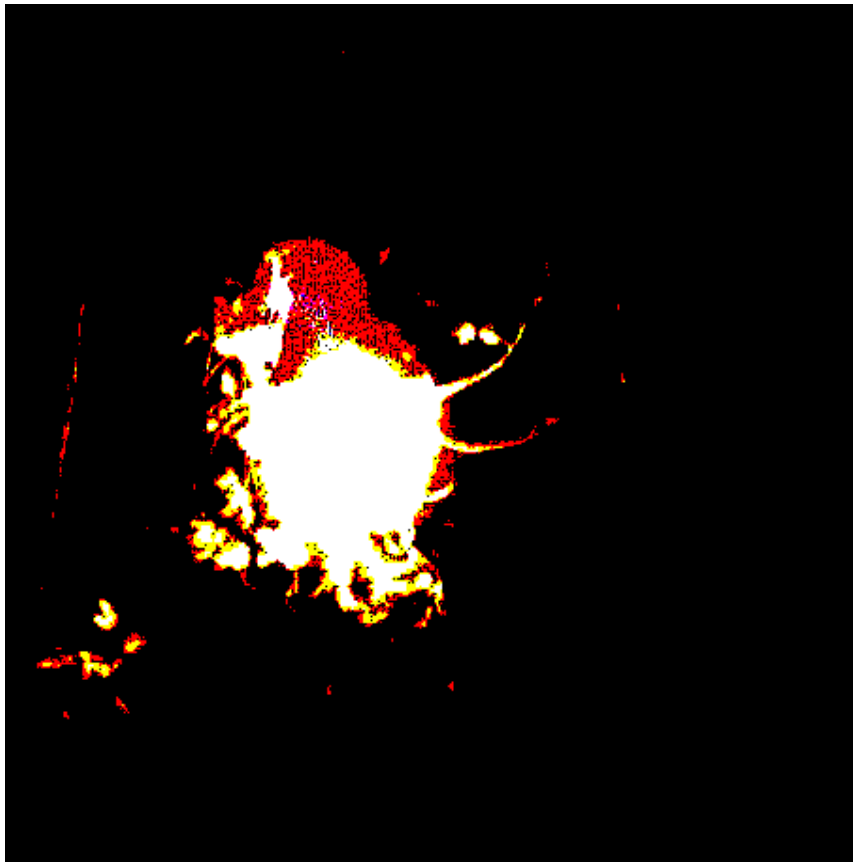
Results

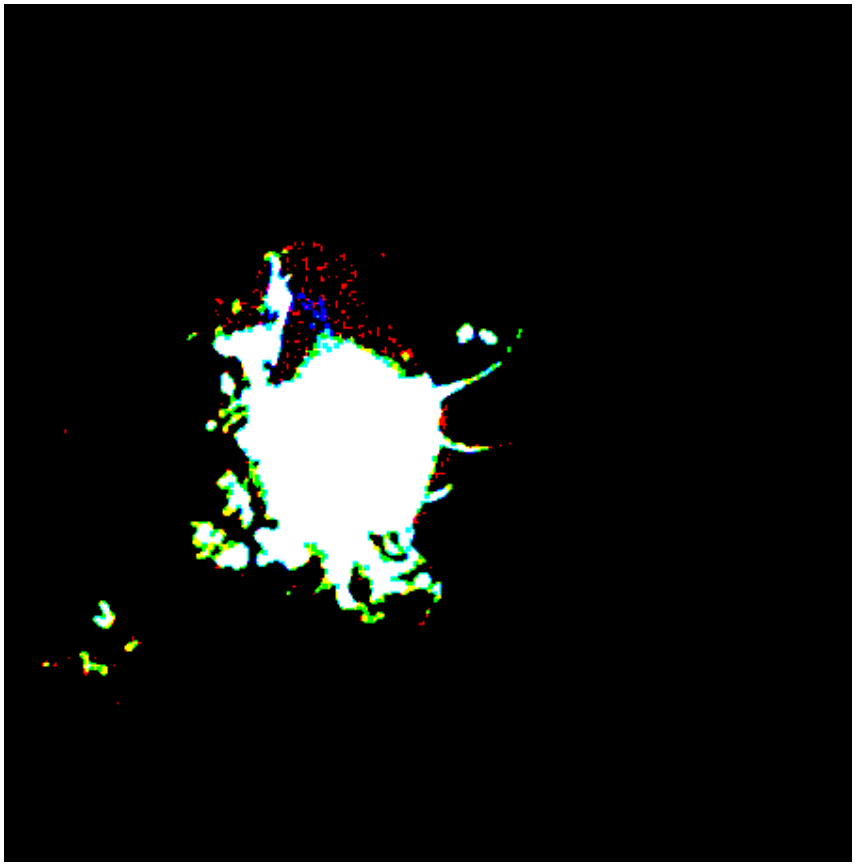


Results

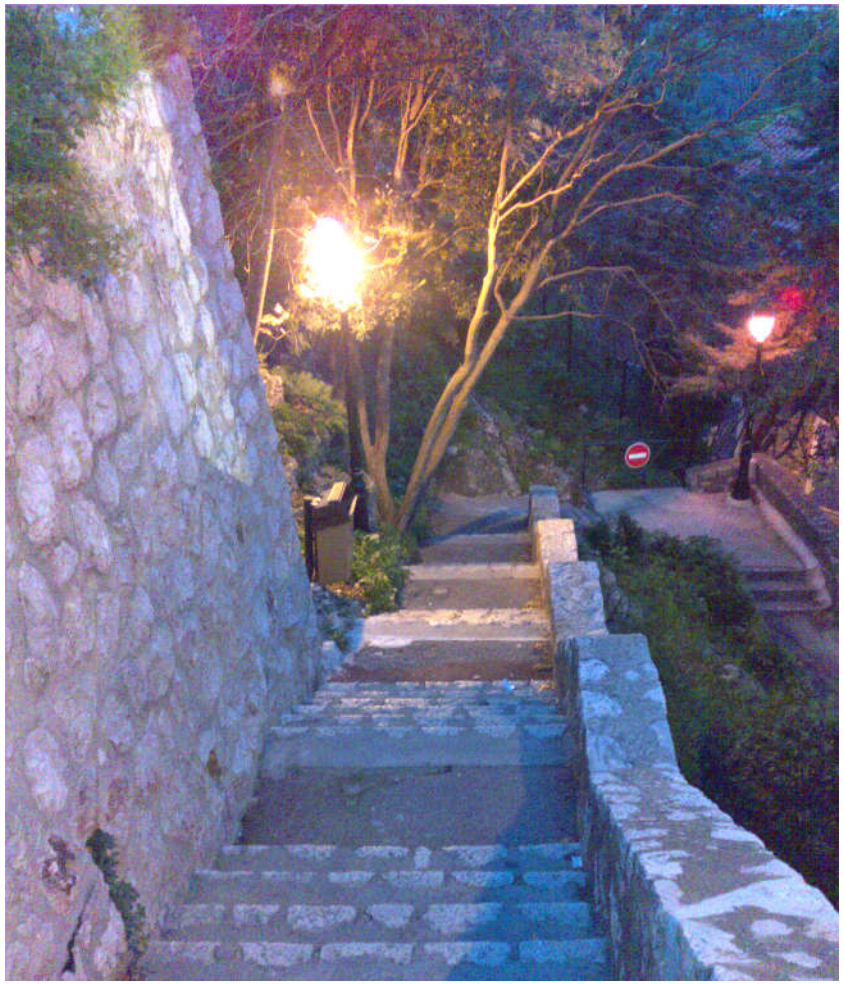


Results





Results





Conclusions

- A method that makes possible usage of under unity color processing in consumer image processing;
- The method preserves the appearance of clipped pixels;
- IF situation allows (motion blur or handshake is not a limiting factor), SNR is improved – usually ~3dB ;

End Remarks

- A high level system perspective often offers additional opportunities ;
- A perceptual perspective can help reduce design constraints ;
- In consumer imaging the boundary between analytical image processing and perceptual image processing is blurred ;