Loss-less Compression of Camera Image Data

Dr. Friedrich Dierks, Basler AG

Abstract *This paper shows how 10-12 bit camera image data can be loss-less compressed to 8 bit data by means of a simple look-up table. This enables 1394 cameras to deliver their full dynamic range on Mono8 data format instead of Mono16 format thus reducing bandwidth requirements and speeding up frame rate.*

1394 cameras can transfer monochrome image data either in Mono8 or Mono16 format. The Mono16 format requires two bytes per pixel and thus consumes twice the bandwidth and needs twice the transfer time compared to the Mono8 format.

A signal with a dynamic range of *d* bits can be transferred using digital numbers with *d*-1 bit resolution provided the noise on the signal can be modeled as Gaussian noise (for details see [**3**]). Using fewer than *d*-1 bits will lead so strong systematic non-linear errors due to the quantisation and thus reduce the dynamic range of the signal.

As a consequence cameras with more than 9 bit dynamic range require the Mono16 format in order to deliver the full dynamic range. Typical dynamic ranges are 9..10 bits for machine vision CMOS cameras and 10..11 bits for machine vision CCD cameras.

In [**1**][1] an idea from [**4**] is described how to compress image data with more than 9 bit dynamic range without loss of information so that is can be transferred in Mono8 format. This paper describes the method in detail and presents measurement results for the Basler CMOS camera A600f and the CCD camera A102f.

Basic Idea

From the basic linear camera model (Fig. 1) as

described in [**1**] Bernd Jähne, Digitale Bildverarbeitung, 5th edition, Springer 2002, ISBN 3-540-41260-3

[**2**]

$$\mu_y = K(\mu_e + \mu_d) \tag{1}$$

$$\mu_e = \eta\mu_p \tag{2}$$

$$\sigma_y^2 = K^2\left(\sigma_e^2 + \sigma_d^2\right) \tag{3}$$

$$\sigma_e^2 = \mu_e \tag{4}$$

$$DYN = \frac{n_{e.sat}}{\sigma_d} \tag{5}$$

follows

$$\sigma_y = K\sqrt{\frac{\mu_y - \mu_d}{K} + \sigma_d^2}\ . \tag{6}$$

For a better understanding of this formula it is simplified setting $K = 1$ and $\mu_d = 0$:

$$\sigma_y = \sqrt{\mu_y + \sigma_d^2}\ . \tag{7}$$

This equation states that the temporal noise in the image rises for bright images proportional to the square of the brightness level and is constant for dark images.

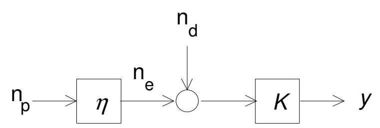


Fig. 1 : Model of a single pixel

As already stated the digital resolution of the gray value needs to be the same size as the standard deviation of the noise in order to make sure the quantisation noise can be neglected. On the other hand it does not make sense to use more bits for quantisation since theses additional bits would show only noise.

The idea is to apply a non-linear transformation

$$y' = f(y) \tag{8}$$

[1] Section 10.2 “Pixelverarbeitung”, sub-section “Äquivalisierung der Varianz des Rauschens”

to the gray values *y* which results in the same noise level over the whole value range for the transformed gray values *y'*.

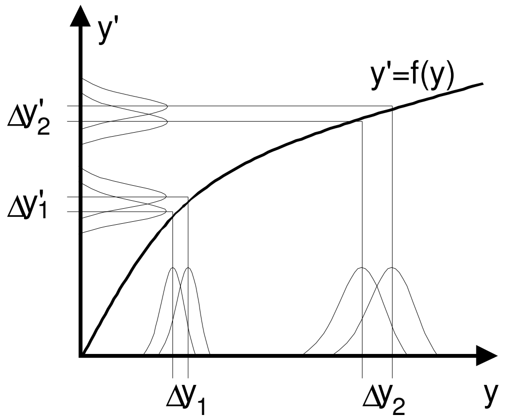


Fig. 2 : Gray value transformation

Fig. 2 shows an example : for small values of the original gray value data *y* there is small noise only and thus a small resolution Δy_1 is required; for larger values *y* the noise is stronger and thus a larger resolution Δy_2 is sufficient. The transformation is chosen in a way that the noise level is the same for all transformed gray values *y'*. As a result the required resolution $\Delta y'_1 = \Delta y'_2 = \Delta y'$ is the same for all values *y'*.

As can be seen in Fig. 2 the same data is described with a smaller range using the transformed values *y'* than using the original values y without loosing information because an adequate resolution of the quantisation is always given.

Deriving the Transformation

The step size of the transformed gray values can be computed by multiplying the step size of the original data and with the derivative of the transformation function:

$$\Delta y' = \frac{df(y)}{dy}\Delta y = \frac{dy'}{dy}\Delta y \qquad (9)$$

The quantisation step size of the transformed values is going to be the digital resolution of the output format of the camera so it is one.

$$\Delta y' = 1 \qquad (10)$$

On the other hand the quantisation step size of the original values is proportional to the standard deviation of the noise.

$$1 = \Delta y' = \frac{dy'}{dy}\Delta y = \frac{dy'}{dy} C\sigma_y \qquad (11)$$

Substituting (6) yields the differential equation

$$dy' = \frac{1}{CK\sqrt{\frac{\mu_y - \mu_d}{K} + \sigma_d^2}} dy\,. \qquad (12)$$

which can be integrated taking into account that $y = \mu_d$ should yield y'=0:

$$\int_0^{y'} d\hat{y}' = \frac{1}{C\sqrt{K}} \int_{\mu_d}^{y} \frac{1}{\sqrt{\hat{y} + K\sigma_d^2 - \mu_d}} d\hat{y} \qquad (13)$$

$$\begin{aligned} y' &= \frac{2}{C\sqrt{K}} \sqrt{\hat{y} + K\sigma_d^2 - \mu_d}\,\Bigg|_{\mu_d}^{y} \\ &= \frac{2}{C\sqrt{K}} \left(\sqrt{y + K\sigma_d^2 - \mu_d} - \sqrt{K\sigma_d^2} \right) \end{aligned} \qquad (14)$$

For $y < \mu_d$ the output value is $y' = 0$.

For a better understanding of this formula it is simplified setting $K = 1$ and $\mu_d = 0$:

$$y' = \frac{2}{C}\left(\sqrt{y + \sigma_d^2} - \sigma_d \right) \qquad (15)$$

This equation states that the transformation is in the large a square root function.

The coefficient C is found from the condition that an input value of

$$y_{\max} = 2^p - 1 \qquad (16)$$

needs to create an output value of

$$y'_{\max} = 2^{p'} - 1 \qquad (17)$$

Thus the coefficient C can be computed as

$$C = \frac{2}{y'_{\max}\sqrt{K}} \left(\sqrt{y_{\max} + K\sigma_d^2 - \mu_d} - \sqrt{K\sigma_d^2} \right) \qquad (18)$$

The limits to the compression are given by the postulation that the output step size needs to be at least the size of the standard deviation of the output noise in order to make sure the quantisation noise can be neglected. Taking into account

$$\sigma_y = \frac{dy}{dy'}\sigma_y' \qquad (19)$$

and equation (11) this yields to the inequality

$$\frac{\Delta y'}{\sigma_y'} = C \le 2 \qquad (20)$$

which needs to be fulfilled if the compression has to be de-facto loss-less (see [**3**]).

Most image processing task can be performed on the compressed data. The data con however be decompressed by applying the inverse function of equation (15) which yields

$$y = \frac{C^2 K}{4} y'^2 + CK\sigma_d y' + \mu_d \qquad (21)$$

Typically for de-compression $\mu_d = 0$ will be used. Note that applying this transformation will result in missing codes.

Example I : A600f

The camera A600f has a CMOS sensor with an integrated 10 bit analog-to-digital converter. It has an on-board 10 bit to 8 bit lookup-table which can be used to compress the image data.

The camera has the following data:

$$p = 10 \text{ bit} \qquad (22)$$

$$p' = 8 \text{ bit} \qquad (23)$$

$$\sigma_d = 106 \text{ e}^- \qquad (24)$$

$$n_{e.sat} = 53 \text{ ke}^- \qquad (25)$$

$$\mu_d = 50 \text{ DN} \qquad (26)$$

This results in

$$DYN = 9 \text{ bit} \qquad (27)$$

$$\frac{1}{K} = 48 \text{ e}^- / \text{DN} \qquad (28)$$

$$C = 1.06 \qquad (29)$$

Note that C is well below 2 so the compression is de-facto loss-less. Of course the 9 bit dynamic range of the A600f could be transferred without compression so the A600f is an example only that the compression really works (see measurement data below).

Fig 3 shows the characteristic curve of the lookup table. Note that due to $\mu_d = 50$ smallest 10 bit input value is 50. The curve is nearly straight. As a result no difference between the uncompressed and the compressed image can be seen when visually inspecting the live image.

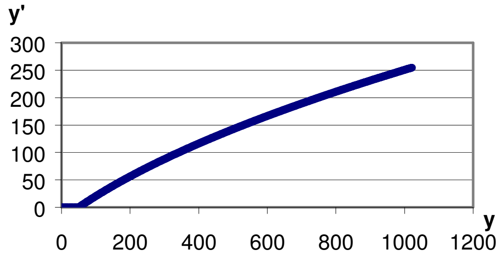


Fig 3 : Lookup table for A600f

To see the effect of the compression three measurements with different camera settings were made:

- Mono16 : The full 10 bit output data was transferred in Mono16 format
- Compression : The lookup table was loaded and the output data was transferred in Mono8 format
- Mono8 @ MinGain : The Gain was set to minimum which corresponds to the MSB-aligned 8 bit of the 10 bit sensor data.

Fig 4 shows the mean value - offset versus shutter time. Note that the Mono16 curve is drawn with respect to the left scale while all other curves are drawn with respect to the right scale. The Mono8 curve saturates below 255 because the input range of the look-up table has (1024 – 50) gray values only instead of 1024.

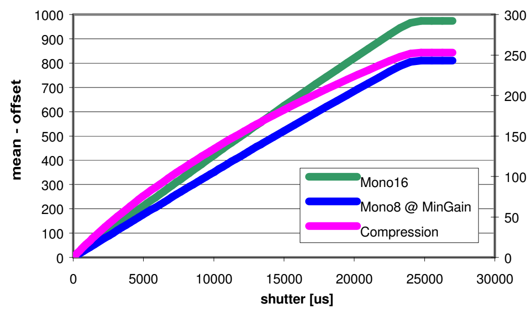


Fig 4 Mean value versus shutter time.

Fig 5 shows the standard deviation of the temporal noise. Note that when using Mono8 with minimum gain the noise value drops below 1 at half the input range. This corresponds with the fact that the sensor delivers 9 bit dynamic range but Mono8 can only deliver 8 bit so only half of the measurement values are transferred unaffected by quantisation noise. The noise of the compressed data constantly equals 1 and thus the full dynamic range of 9 bit is transferred.

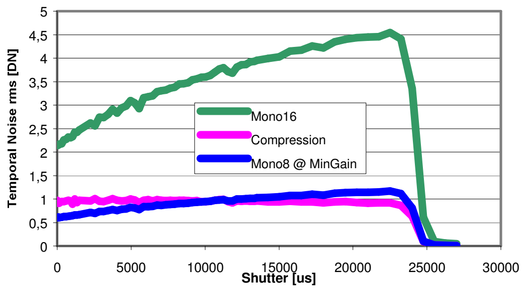
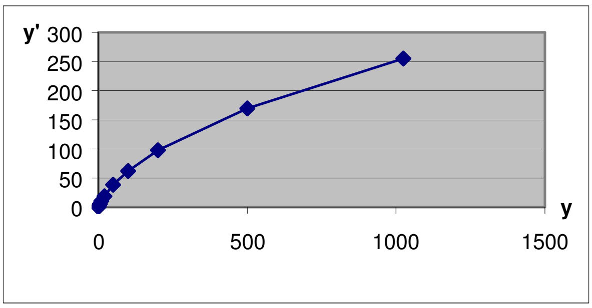


Fig 5 Temporal noise (rms) versus shutter time

Conclusion : with the compression LUT the A600f can deliver it's full dynamic range of 9 bit with Mono8 data format without any information loss and without any visual effects in the image.

Example II : A102f

p	10	bit
p'	8	bit
1/K	16	e-/DN
sigma.d	15	e-
K	0,0625	DN/e-
C	0,89266	1



Note that C is smaller than 2 so the compression is loss-less.

Fig 6 shows the temporal noise versus shutter time which is flat with compression as predicted.

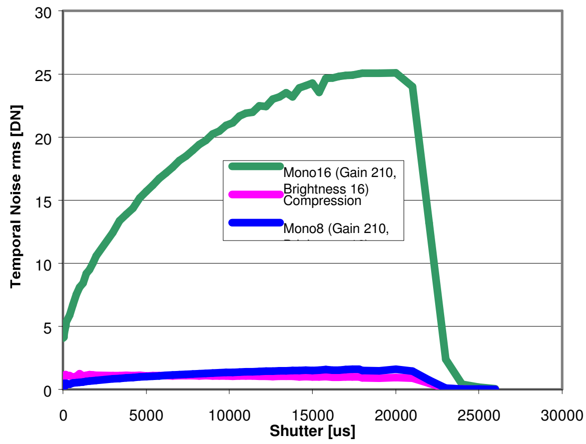


Fig 6 Temporal noise (rms) versus shutter time

Applications

If a camera has more analog dynamic range than 9 bit there are several ways do deal with that:

- Restrict the digital resolution to 8 bit and loose dynamic range. Analog gain can be used to adapt the camera to the lighting condition
- Use 16 bit for transfer. This is bandwidth consuming and with 1394a typically slows down the camera
- Compress the analog data by nonlinear transformation as proposed and decompress the data on the PC. This takes computing time which thus can be traded against bus bandwidth
- Compress the analog data by nonlinear transformation as proposed and use it as is. Most image processing algorithm assume a constant noise level which was ok in the days cameras were read noise limited. Today cameras for machine vision are photon noise limited so this assumption does not hold true any more. The transformed image however has constant noise level and most algorithms should be able to deal with the compressed data.

Problems with the proposed algorithm might arise because the digital look-up table changes the frequency of occurrence in the histogram yielding the so-called hedgehog-effect.

Literature

[**1**] Bernd Jähne, Digitale Bildverarbeitung, 5th edition, Springer 2002, ISBN 3-540-41260-3
[**2**] Dierks, F.; "Comparing Digital Cameras with Respect to Sensitivity and Noise"; Basler internal paper

[**3**] Dierks, F.; “Quantisation in Digital Camera Images”; Basler internal paper

[**4**] Wolfgang Förstner, Image Preprocessing for Feature Extraction in Digital Intensity, Color and Range Images, In: A. Dermanis, A. Grün, F. Sanso, Ed., Geomatic Methods for the Analysis of Data in the Earth Science, Lecture Notes on Earth Science no. 95, Springer, Berlin 2000