

Focal-Plane Compression Imager with Increased Quantization Bit Rate and DPCM Error Modeling

Fernanda Duarte Vilela Reis de Oliveira, Tiago Monnerat de Faria Lopes, José Gabriel Rodríguez Carneiro Gomes, Fernando Antônio Pinto Barúqui and Antonio Petraglia

Universidade Federal do Rio de Janeiro – COPPE/Electrical Engineering Program,
fernanda@pads.ufjf.br

Abstract— Focal-plane processing is the target of many studies due to its potential for enhancing the speed of the vision system flow. With focal-plane processing it is possible to perform parallel processing throughout the entire matrix. In order to alleviate A/D conversion and transmission constraints, analog image compression is implemented at the focal plane, thereby reducing the amount of data to be transmitted and the bandwidth requirements. The ADC is performed at the focal plane as well, after the compression operation whose realization is based on differential pulse-code modulation (DPCM), linear transform and vector quantization (VQ) applied on every 4×4 pixel block using current-mode circuits. This paper presents experimental results obtained from a second-generation version of the image sensor. Among these results we can point out the presentation of different captured images and the modeling of errors that were identified during the experimental tests. Since the source of these errors is the DPCM stage, the modeling concerns the bits that refer to the mean block luminance results. The error modeling procedure was developed considering the relationship between the pixel integration period and the DPCM quantizer threshold values. The main contributions of the second-generation chip in comparison to the previous realization are: increase of the vector quantizer complexity, number of bits per pixel, pixel matrix size, and the use of cascode current mirrors in the linear transform matrix. The image sensor advanced in this paper was fabricated in a standard 180 nm CMOS process.

Index Terms—CMOS image sensor, focal-plane compression, DPCM, vector quantization.

I. INTRODUCTION

The CMOS image sensors framework applies to countless number of applications [1]. Nowadays it is possible to find these sensors in systems ranging from simple cameras, such as the ones used for surveillance, to high quality cameras used by professional photographers. Industrial and academic fields invest resources on studying these sensors, because of the achievable image quality, and because of their flexibility [2]. An interesting feature of the CMOS image sensors is the possibility of introducing processing hardware in the same chip of the pixel matrix [3]. This allows for the design of an entire system on chip and has the potential for enhancing speed and reducing power of vision systems, which are very useful characteristics for embedded circuit applications. Furthermore, if we consider adding processing hardware inside the pixel matrix, every pixel would be part of a processing unit, and we can perform parallel processing throughout

the entire matrix. This technique is called focal-plane processing and has lately been the topic of many publications [4]-[10].

Usually, in a vision system chain, all the pixels values from the pixel matrix must be converted to digital, sent out of the capture chip, stored in a intermediate memory and sent to a digital processor that will perform a desired task such as image compression, object recognition, face detection, among others. The bottleneck of this chain is the data transmission from the pixel matrix to the intermediate memory. Analog-to-digital conversion, transmission out of the chip and memory storage require significant amount of time in the vision system flow. To accelerate the vision system flow, it is interesting to perform compression inside the pixel matrix chip, thereby also reducing the amount of data that need to be transmitted. Another advantage in doing compression inside the chip is to alleviate the bandwidth requirements of the system.

Our goal is to perform data compression at the focal plane. Current-mode analog circuits are used to implement differential pulse-code modulation, linear transform and vector quantization in every 4×4 pixel block [11]. Two generations of a compression imager have been designed, fabricated and tested. This paper presents experimental results for the second-generation chip and compares them with results from the first-generation chip [11].

Based on the experimental results from the first chip, improvements were included in the design of the second prototype. Among the modifications in the new design we highlight five main changes:

- VQ (vector quantizer) complexity: a new input component was added with the goal of being able to capture more details and improving the modulation transfer function. Five input dimensions are used now, instead of four, thus increasing the VQ complexity.
- Number of bits: because of the changes in the VQ, one additional bit is required for representing the additional component sign. To improve the representation of the five-component absolute value vector, we added two VQ bits, thus increasing the VQ bit rate from seven to nine bits per vector.
- Cascode current mirrors: the linear transform matrix was implemented using cascode current mirrors, instead of simple current mirrors, which were used in the first-generation chip. A study was performed in

order to verify that the cascode brings advantages to the implementation [12].

- Technology: the new chip was designed for fabrication in a 0.18 μm technology, while the previous one was designed for a 0.35 μm technology.
- Pixel matrix size: a 64×64 pixel matrix was designed for the second generation chip, in comparison with 32×32 for the first one.

After conducting the experimental tests, errors were identified in the DPCM (differential pulse-code modulation encoder). We will also present a modeling and proposed correction for these errors. To model the DPCM encoder errors, we performed experimental tests using a white uniformly illuminated image. The modeling is based in the relationship between the integration period and a reference voltage that defines the quantizers thresholds.

Section II gives a brief explanation of the algorithm, highlighting the differences between the designs. The circuits that are required for implementing the proposed data compression are shown in Section III. Section IV presents a qualitative comparison between the results obtained from both chips using similar targets and the DPCM errors modeling. Section V closes the paper with a final discussion and ideas for future work.

II. BLOCK-BASED IMAGE COMPRESSION

The implemented image compression technique is block-based, performed in every 4×4 pixel block. This technique is explained in detail in [11]. It is divided into two parts: compression of the mean value of the block, also called DC component, and compression of the details of the block, also called AC component.

For the DC component representation, we use a DPCM

encoder. The idea is to transmit, instead of the input signal itself, the difference between the input signal and a prediction of this signal [13]. With a good prediction, most of the values transmitted will be close to zero, which permits prioritizing the most likely values to be transmitted and even using fewer bits, if we can afford to lose the most unlikely values. For a natural image, there is a high probability of two neighboring pixels having close values, so a good prediction for a pixel is to use the value of a neighbor that was previously sent. This idea can be extended to the mean value of pixel blocks. That is, the mean value of a spatially adjacent block is a good prediction for the mean value of the block we want to transmit. In the following paragraphs, all algebraic symbols refer to Fig. 1, which summarizes the operation of DPCM, linear transform and VQ performed inside a single 4×4 pixel block. For convenience, because of circuit simplifications, in the proposed DPCM technique, instead of the mean value of a block, we compute the summation of the pixels inside the block, $s(n)$. Four bits are used for transmission, in which one, D_1 , is for the sign and three, D_2 , D_3 and D_4 , are for the absolute value of the difference between the summation of the pixel values inside a block and the summation of the pixel values inside the previous block at the same row, $s(n-1)$. The prediction value for the next block is given by the equation $\hat{s}(n+1) = s(n-1) + \hat{e}(n)$, where $\hat{s}(n)$ is the decoded value that represents $s(n)$ and $\hat{e}(n)$ is the predicted pixel summation for the current block. The maximum difference that can be encoded is approximately equal to the pixel output signal range center value. There are transitions from a white block (one) to a black block (zero), or the opposite, which are large enough for the DPCM to fail in properly representing them. The choice of not representing steps higher than the middle of the signal range comes from the DPCM design, which considers the PSNR (peak-signal-to-noise ratio [14]) of the reconstructed images and the maximum number of desired bits. For focal-plane image compression, we use a low bit rate, which is enough for keeping the image quality at an acceptable minimum.

DPCM is performed for every row of blocks. A reference value is considered for the first block of every line. The main difference between the first-generation design DPCM and the second-generation DPCM is this reference value. On the first-generation chip, this reference was equivalent to the lowest possible value from the signal range (zero), while in the second-generation chip this reference was designed to be equivalent to the middle value of the signal range. This change in DPCM initialization was based on system-level simulations that indicated a PSNR increase associated with the new DPCM initialization. The consequence of considering the lowest value of the signal range is that the blocks at the beginning of every row became dark. This is not a problem when the block average values are close to zero, but if the first block is brighter, then we may not be able to reach the correct average luminance value. On the other hand, when the first block reference value is closer to the middle of the range reaching darker or brighter average values is equally easy.

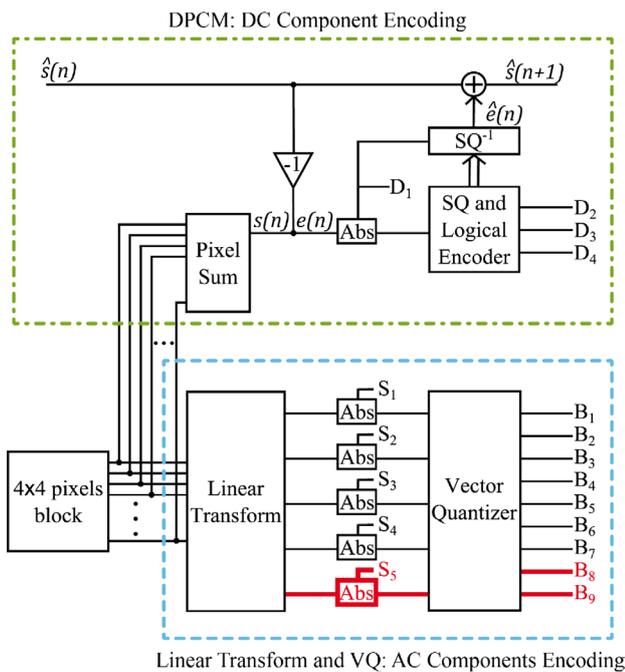


Fig. 1: Compression method block diagram.

In the case of the AC component, we use linear transform and vector quantization for signal compression. As mentioned before, neighboring pixels usually have very close values, which means that, inside a block, the pixel values are correlated. In other words, there is a significant amount of redundant data that does not need to be transmitted. The linear transform is responsible for changing the signal domain into a more efficient one, thus reducing redundancy and concentrating signal energy. The computation consists in multiplying a 16-dimension vector, which represents the block pixel values, by a carefully chosen matrix that aims at maximum decorrelation. The outputs of this operation are transform coefficients. If these coefficients are multiplied by the 4×4 pixel blocks that compose the transform basis, and the resulting pixel blocks are added up, then the pixel-block texture details are reconstructed. The linear transform basis is composed by 1 DC component and 15 AC components of increasing horizontal and vertical spatial frequency. The DC component was already previously encoded by the DPCM. Because of silicon area constraints, we encode a smaller set of AC components containing the highest-energy ones.

For the previous chip design, only four linear transform components were used. For the second chip design, we decided to use five linear transform components aiming at increasing the image quality [15]. After the linear transform, we compute the component absolute value, encode the component signs using one bit per sign (bits S_1 to S_4 in the previous chip, and bits S_1 to S_5 in the new chip), and finally convey the absolute values to a VQ for block coding.

The vector quantizer is a generalization of the scalar quantizer. It performs analog-to-digital conversions but, instead of considering each input component separately, it jointly encodes the vector components aiming at decreasing entropy and distortion [16]. Adding a new linear transform component leads to one-bit increase in the output bit rate. This additional bit represents the sign of the new component. The VQ complexity and bit rate increase as well. After including the fifth VQ input dimension, we have obtained better results when using 9 bits for the VQ (from B_1 to B_9), instead of 7 (from B_1 to B_7), as in the previous design. A theoretical comparison of both VQs is presented in [15] and justifies the addition of a linear transform component at the VQ input.

The DC and AC components data encoding flow can be seen separately in Fig. 1, highlighted by the boxes in dash-dot and dashed lines, respectively. The differences between the chip generations are indicated by the thicker lines. As it can be seen in the figure, in the previous design 15 bits were transmitted for each 4×4 pixel block. In the new design we have 18 bits per block. The bit rate increase is justified by the image quality improvement [15].

III. SCHEMATIC DIAGRAM

Current-mode circuits were used to implement the compression algorithm. Using current-mode circuits allows for

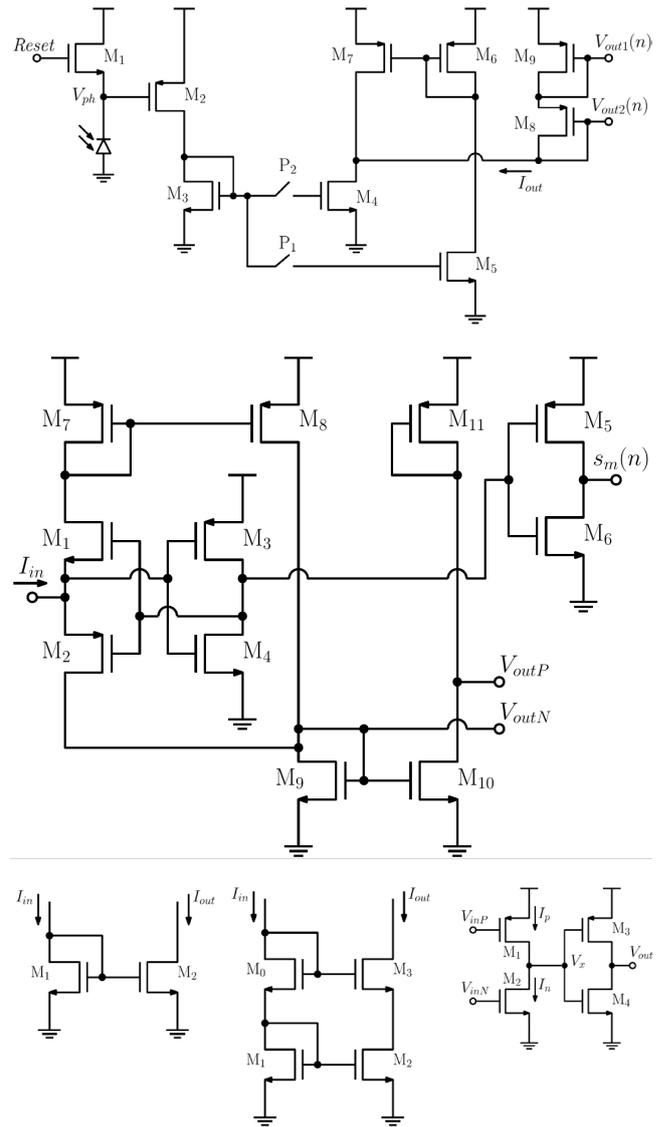


Fig. 2: Circuits used to implement the compression algorithm at the focal plane: (a) photodiode readout circuit, (b) absolute value circuit, (c) simple current mirror, (d) cascade current mirror and (e) current comparator.

maintaining the signal range as technology scaling leads to reduced power supply. In our case, the current mode simplicity yields an important advantage over voltage-mode implementations with respect to signal summation, multiplication and copying. The sum of two currents is performed by simply connecting both currents sources to the same node. To multiply a current by a scalar constant, current mirrors with properly adjusted width and length are used. Those basic operations are required for the image compression algorithm implementation.

Fig. 2(a) shows the photodiode readout circuit, which maps the photocurrent into an amplified current that will be processed by the circuits that implement data compression. The readout circuit has three control signals, Reset, P_1 and P_2 . When the reset is activated, the reset transistor, M_1 , which works as a switch, turns on and the photodiode node, V_{ph} ,

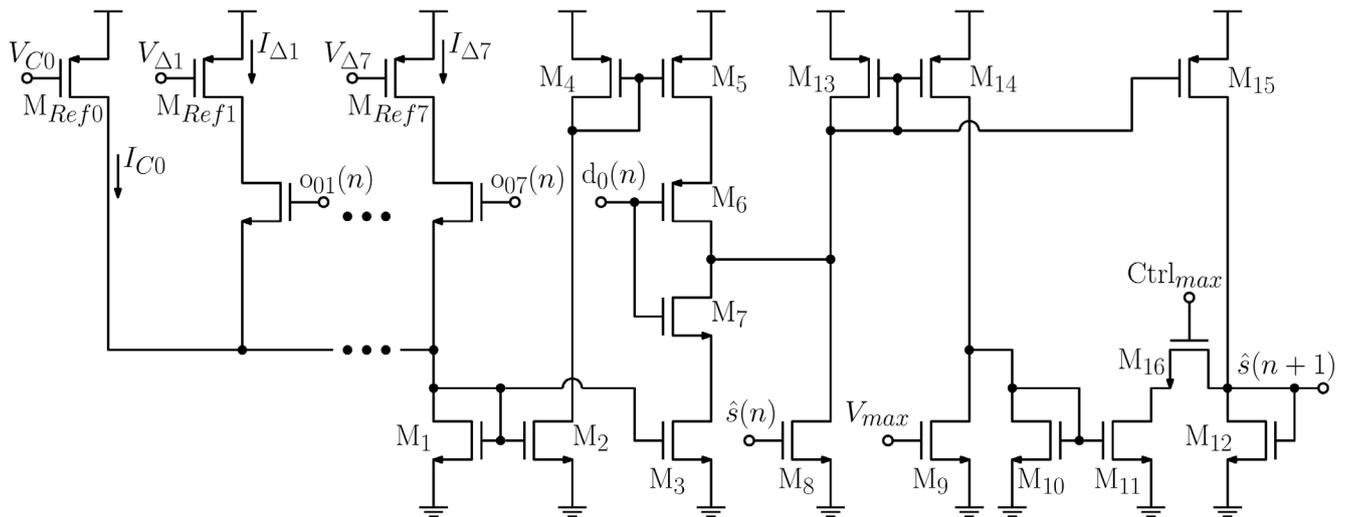


Fig. 3: DPCM reconstruction circuit with maximum current control.

is charged. As soon as the reset signal goes down and M_1 turns off, the initial V_{ph} voltage is sampled by turning off the switch controlled by P_1 . At the same time, the photodiode starts discharging the photodiode node, V_{ph} , proportionally to the incident light. The integration period is the time interval during which the photodiode is kept working, generating a current proportional to the incident light. By the end of this period we turn off the switch controlled by P_2 to obtain a second sample. The samples are stored in the parasitic capacitances of transistors M_4 and M_5 , thus maintaining the currents that flow through these transistors equal to a copy of the first and second sample, respectively. The first sample is copied and inverted using M_6 and M_7 with the goal of subtracting it from the second sample. The output current, I_{out} , is thus the difference between the second and the first sample. This technique, which is useful for reducing fixed-pattern noise, is called correlated double sampling [17]. Transistors M_8 and M_9 transform the output current into two output voltages, V_{out1} and V_{out2} , which are used by DPCM and linear transform circuitry to obtain copies of I_{out} .

For both the DC and AC encoding algorithms, it is necessary to execute absolute value operations, as it can be seen in Fig. 1. The circuit shown at Fig. 2(b) implements this operation. This circuit is required for DPCM and for the generation of VQ inputs. Its input signal, I_{in} , is a current of either direction and its outputs are a bit that represents the direction of the input current, s_m , and two voltage references, V_{outP} and V_{outN} , from where the absolute-valued current can be copied with the correct direction, depending on its multiplication by a positive or a negative scalar.

Simple and cascode current mirrors are used in the new design. These circuits are presented in Fig. 2(c) and (d), respectively. In both cases, the output current I_{out} will be equal to a constant multiplied by I_{in} , where the constant that multiplies the input current is equal to the ratio between the channel width/length ratio of the output transistor M_2 and the channel width/length ratio of the input transistor M_1 . Among

other uses of these circuits, they are employed to implement the linear transform operations. Since these operations can be written as a sum of weighted pixel values, the current mirror is used to perform the multiplication, while its input current is proportional to the pixel value. The first chip design used only simple current mirrors, but these circuits are not as accurate as the cascode current mirrors. The study presented in [12] showed that the use of cascode current mirrors for the linear transform operations mapping 16 pixel values into five components is very beneficial. To allow for VQ implementation using simple analog circuits, we use a sub-optimal approximation consisting of another linear transform followed by scalar quantizers. For this linear transform, which maps 5-D input vectors into 5-D feature vectors that have its components separately encoded by scalar quantizers, simple current mirrors are used in both designs, since the advantages of using cascode at this part of the circuit were not significant [12].

The current comparator is used for the quantization circuits in both VQ and DPCM. It is shown in Fig. 2(e). The output of this circuit is a bit that indicates which current is higher. If I_p , which is given by the input voltage V_{inP} is larger than I_n , given by V_{inN} , then the node V_x is charged and the output voltage V_{out} is low, to represent a logic zero bit. On the other hand, if I_n is larger than I_p , then the node V_x is discharged and the output bit is one. The current comparator is used for converting the result of the compression to digital. The reference currents required for performing the comparison are generated outside the pixel matrix, using several cascode current mirrors. Only one reference current, defined outside the chip, is required for generating the threshold values for VQ and DPCM, and for generating the prediction value for the first block of each row. This current is measured through the voltage across an external resistance whose value is well-known. This voltage across the external resistance will be mentioned very often in Section IV, so we allocate a particular symbol, γ , for it.

In Fig. 3, we show the circuit for generating the prediction value for each DPCM block. This circuit implements the equation . The first step is a digital-to analog conversion: the inputs are the DPCM bits, o_{01} until o_{07} , which will define which currents are added up, from $I_{\Delta 1}$ to $I_{\Delta 7}$, to represent the absolute quantized value of the error, e . The result of the sum between the delta currents and I_{C_0} yields an analog reconstruction value. The bit d_0 is the reconstructed prediction error sign, which defines the direction of the error current. The reconstructed prediction error is thus copied through M_3 or M_5 . The pixel sum prediction for the pixel block at the present spatial position, s , is copied through M_8 and the sum that defines s is performed at the drain node of this transistor. Consequently, the current that flows through M_{13} is the pixel sum prediction for the pixel block at the next spatial position, $s+1$. The output of this circuit is also different from the circuit in the first-generation chip. In order to guarantee that the circuit will not work with currents higher than designed, a circuit that defines the maximum pixel sum prediction current was included. It is composed by transistors M_9 to M_{16} . In this circuit, M_9 copies the maximum current the DPCM circuit is allowed to work with. If the current copied by M_{14} , that represents $s+1$, is higher than the maximum current, then the difference between these two currents will be copied using M_{10} and M_{11} . This difference is then used to subtract from the current by which it exceeds the maximum current, thus resulting in a prediction current that is equal to the maximum possible current. This circuit can be activated or deactivated using the transistor M_{16} . This circuit was included with the goal of controlling the luminance representation saturation at the DPCM decoder. The maximum current value is defined outside the chip, a reference current that is different from the one obtained with s . Therefore, two currents are generated outside the chip: the one proportional to s , which defines the thresholds s_{min} and s_{max} , and a second current which defines the maximum prediction current value, from columns 2, 3, ..., 16.

The present chip was designed using a 180 nm CMOS technology. The layout of the 4×4 pixel block that is used throughout the pixel matrix is presented in Fig. 4(a). The implementation of the data compression algorithm inside each pixel block requires 833 transistors per block. In the previous design, that used only simple current mirrors and four (instead of five) linear transform components, there were 607 transistors per block. Even with the increase in the

number of transistors, using a 180 nm technology, instead of the 350 nm technology from the first design, allowed for an increase in the fill factor, from 7.1% to 13.5%, and a reduction of the pixel pitch, from $37.5 \mu\text{m}$ to $27.2 \mu\text{m}$. Fig. 4(b) shows a photograph of the fabricated chip welded to the test board. The metallic structure shown in Fig. 4(b) is used to connect the lens for the chip tests.

IV. EXPERIMENTAL RESULTS

Experimental tests are being performed with the purpose of characterizing the new chip and identifying whether the modifications were advantageous. In order to do that, an experimental setup has been designed so that a micro-controller could be used as the interface between a computer and the chip. The chips from both generations use the same lenses and structure shown in [11], but a new circuit board was designed to adapt for the requirements of the new chip. For example, the transistors from the first generation operate with a power supply of 3.3 V, while the second generation ones work with 1.8 V.

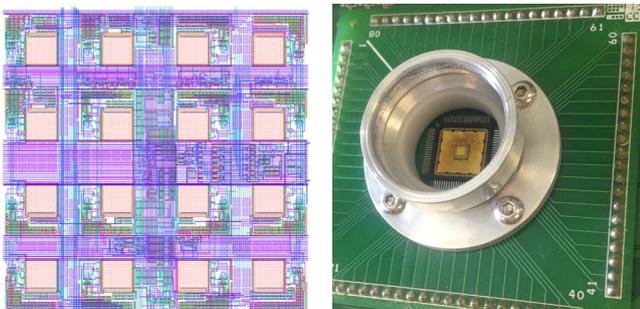


Fig. 4: (a) Pixel block layout and (b) photograph of the fabricated integrated circuit and the lens support

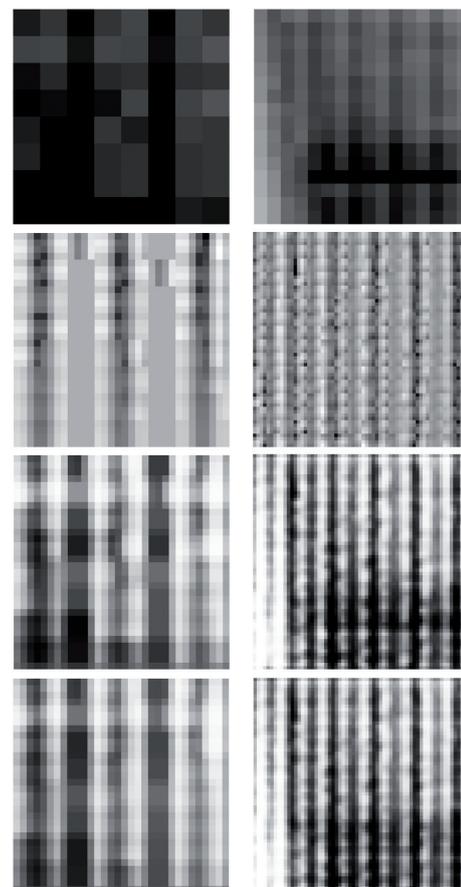


Fig. 5: Results for a black and white striped pattern. On the left column, results from the first generation chip, with a target of 1.67 cycle/cm spatial resolution; on the right column, results from the second generation chip, with 3.33 cycle/cm spatial resolution. From top to bottom: DPCM partial result, VQ partial result, decoded image after filtering and enhancement, and the average image, using 100 decoded images.

Aiming at comparing the results from both chips, the same targets used for the first generation [18] were employed to test the new chip. Those are black and white images with geometrical shapes that allow for the visualization and first evaluation of the DPCM and VQ stages, as well as the final decoded image.

Fig. 5 shows the results for both generations when a striped pattern is used as target. It is important to note that although the images are presented with the same size, the resolutions are different. The images generated by the first chip have 32×32 pixels while the images from the second chip have 64×64 pixels. In this figure, the reconstructed partial image using only the DPCM bits can be observed at the first row, the reconstructed partial image employing only the VQ bits is at the second row, the final image is presented at the third row, and the mean image after 100 captures is shown at the fourth row. The reconstructed final images have been filtered to remove noise and enhanced for a better visualization. For all the presented figures, the DPCM row starts on the left side of the image.

This figure is interesting because it shows an image in which the transition between white and black stripes are narrower than a DPCM block, which means that the VQ is the predominant signal in the image. In this figure, the new generation image has twice the number of black stripes compared to the first generation image. Since the resolution changes from the first to the second design, we depict these images to guarantee that a fair comparison is made. That is, in a 32×32 section of the new generation image there is approximately the same number of stripes as that of the first generation chip. Comparing the images in the second row of Fig. 5, we see that the VQ is more efficient for the new design, since the stripes have a better definition. The transition between black and white is now more precise and better defined.

On the other hand, observing the final reconstructed images, we can see that there is an error in the DPCM. It seems that the reference value for the first block, that was supposed to be in the middle of the dynamic range, is higher than expected. As a consequence, it takes more steps to reach darker values than what was expected. In this figure the employed first column prediction values were arbitrarily chosen to guarantee that the final image had no negative values, which resulted in the white blur that can be observed in the bottom left of the final image. In order to alleviate this effect, the DPCM is being carefully studied. An analysis of these errors has been developed and will be presented in the next section. Aside from this DPCM unexpected result that degrades the image quality, the overall result shows that the image quality has improved.

Additional consequences can be seen in Fig. 6, where the VQ partial results and a final image for various targets are displayed. In the final images from the second row of Fig. 6, we can see two errors generated from the DPCM: the first one is the white blur that appears in the image from the second column, and the second one is the black blur that appears

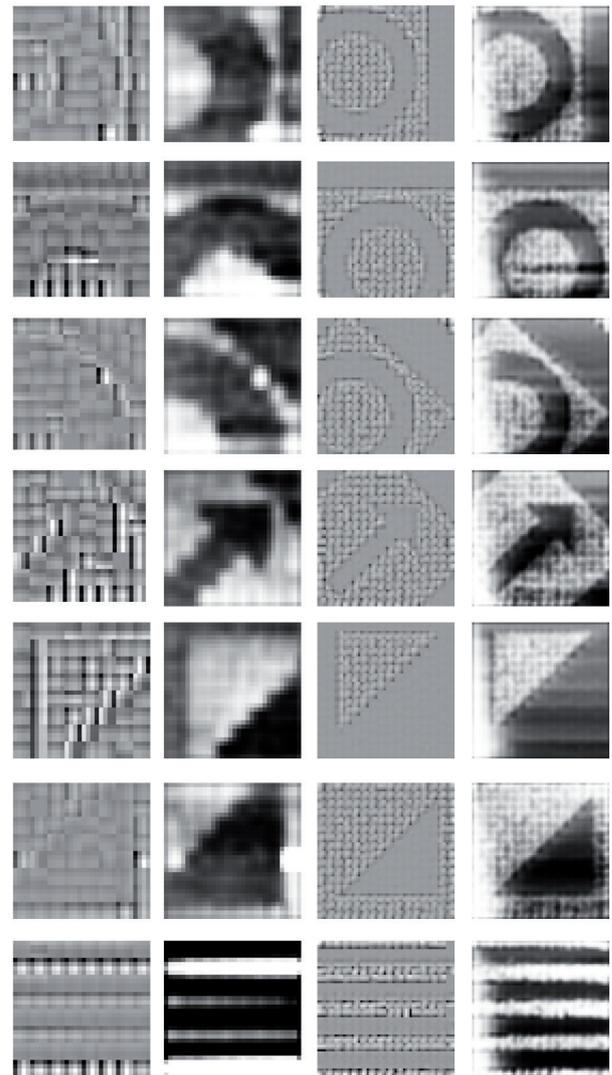


Fig. 6: (a) VQ partial results from the first generation chip and (b) corresponding final, (c) VQ partial results from the second generation chip and (d) corresponding final image.

in the image from the fourth column. Both errors seem to have the same nature, since the DPCM noise assumes values that are higher or lower than what was supposed to be and the total noise causes the blur. The VQ results confirm the conclusion drawn from Fig. 5, that there was a significant quality improvement in this section of the compression algorithm. On the images, which show the VQ partial results, the objects can be clearly seen. It should also be noted that on the darker regions there is much less noise in the results from the new design than from the former one. In the arrow image, for example, shown in the fourth row and third column of Fig. 6, the arrow can be hardly seen if we look just at the VQ result of the first-generation result, while the VQ from the new design clearly represents the arrow.

A. DPCM Experimental Tests and Modeling

With the goal of investigating the sources of the DPCM

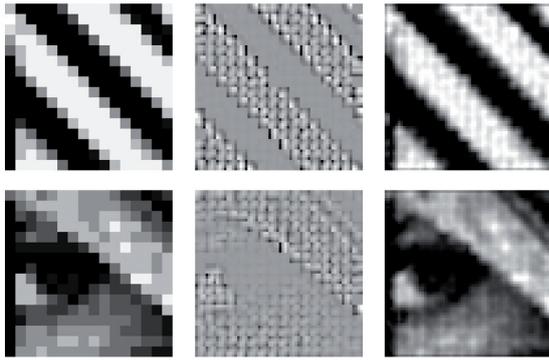


Fig. 7: Images generated when the imager uses PCM for the encoding of the DC component of the blocks. In the top row, the target was a black and white striped pattern and in the bottom row, the target was a piece of the Lena image. From left to right, PCM result, VQ result and corresponding final image.

errors and improving the image quality, experimental tests have been performed making use of the maximum current limiting circuit. This circuit is present at the reconstruction current circuit and can be seen in Fig. 3. If we set the maximum current as low as possible we are able to turn off the DPCM from column two onwards. This happens because the prediction current, i_p , is limited to a value close to zero. Consequently, i_p , the value quantized is proportional to the value of the pixel block sum and, instead of a DPCM, the DC component encoder is now working as a pulse-code modulation (PCM) of only three bits. This does not apply to the first column of the image, though, since the prediction value for the blocks of this column depends on an input reference current.

Regardless of the low bit-count, the achievable image quality is good, as can be seen in the two examples shown in Fig. 7. In this figure we can see the PCM, VQ and final image results, from left to right. The two targets used can be clearly identified from the two rightmost images in Fig. 7, which are a diagonal striped pattern for the top row and a piece of the Lena image, for the bottom row. The PCM result is able to give a good representation for the luminance of the blocks, while the VQ result is responsible for the finer details. The only post-processing applied to the final image was a low pass filter, with the goal of reducing noise. Some errors can be perceived in the first column, mainly in the Lena image, in which this column is entirely black. This happens because the current limiting circuit does not work for this column.

This result shows that it is possible to isolate the errors that are caused by the DPCM. Taking advantage of this fact, experimental tests were performed to analyze the response of each block of pixels, separated by columns. With these tests, we were able to establish a response model for the first column, as explained below.

The tests are based in discovering a relationship between the reference voltage V_{ref} , which defines the input current that sets the threshold currents and the first block prediction current, the integration period T_{int} and the quantization thresholds

from the scalar quantizer present in the DC component encoder. The imager target is a white image uniformly illuminated and the lens aperture and focus are maintained fixed during the entire test.

The test consists in finding, for a given V_{ref} , the integration period necessary from which V_{ref} crosses a determined threshold. This can be found by monitoring the occurrences of the DPCM index. Considering the last threshold, for instance, if V_{ref} increases, thus increasing the threshold value, a higher integration period is necessary. This happens because V_{ref} is proportional to the integration period. This test is repeated for every block of the pixel matrix, for different values of V_{ref} . The result is a set of points that relate the reference voltage with the integration period. This set of points were used for a curve fit using least squares method.

Fig. 8(a) shows the relationship between V_{ref} and T_{int} considering the largest threshold, V_{th} , for each row of block of pixels for columns one, two, three and four, from top to bottom. In this figure, the solid lines correspond to the derived models, which are consequences of the linear regression approximation. Every different line represents a different block row in the matrix. The results for the next columns are similar to the ones for columns three and four. The model equations found are simple linear equations in which the linear coefficients are ideally equal to zero and represent a time offset, the minimum amount of time necessary for V_{ref} to reach a threshold, and the angular coefficient is directly related to the threshold values and to the first blocks prediction current, since it represents the relationship between V_{ref} and T_{int} or, in other words, the relationship between V_{ref} and T_{int} . The angular coefficient can be thus used to understand and model the scalar quantizer response.

Comparing the rows in Fig. 8 we observe that there is a significant difference in the behaviors of the equation models angular coefficients for column one with respect to the behaviors of these coefficients for columns two, three and four. In order to understand this difference, we will start by analyzing the angular coefficient for the equations plotted in the first row of Fig. 8(a). Computing this figure average angular coefficient and comparing it with each angular coefficient, we find that it can reach a difference of over 30% with respect to the average for some block rows.

Performing the same analysis for column two, the maximum difference with respect to this column average angular coefficient is of 21%, which occurs only for the last row. The angular coefficient of row 15 is 15% higher than the average value and, for the other rows, the difference is smaller than 10%. Columns three and four have a similar behavior as that of column two, with a difference of around 20% for the angular coefficient of the last row with respect to the average value, and, for the other rows, with a difference smaller than 10%. Furthermore, the standard deviations of the mentioned angular coefficients for columns two, three and four are one order of magnitude smaller than the standard deviation considering column one.

It is important to point out that the angular coefficients

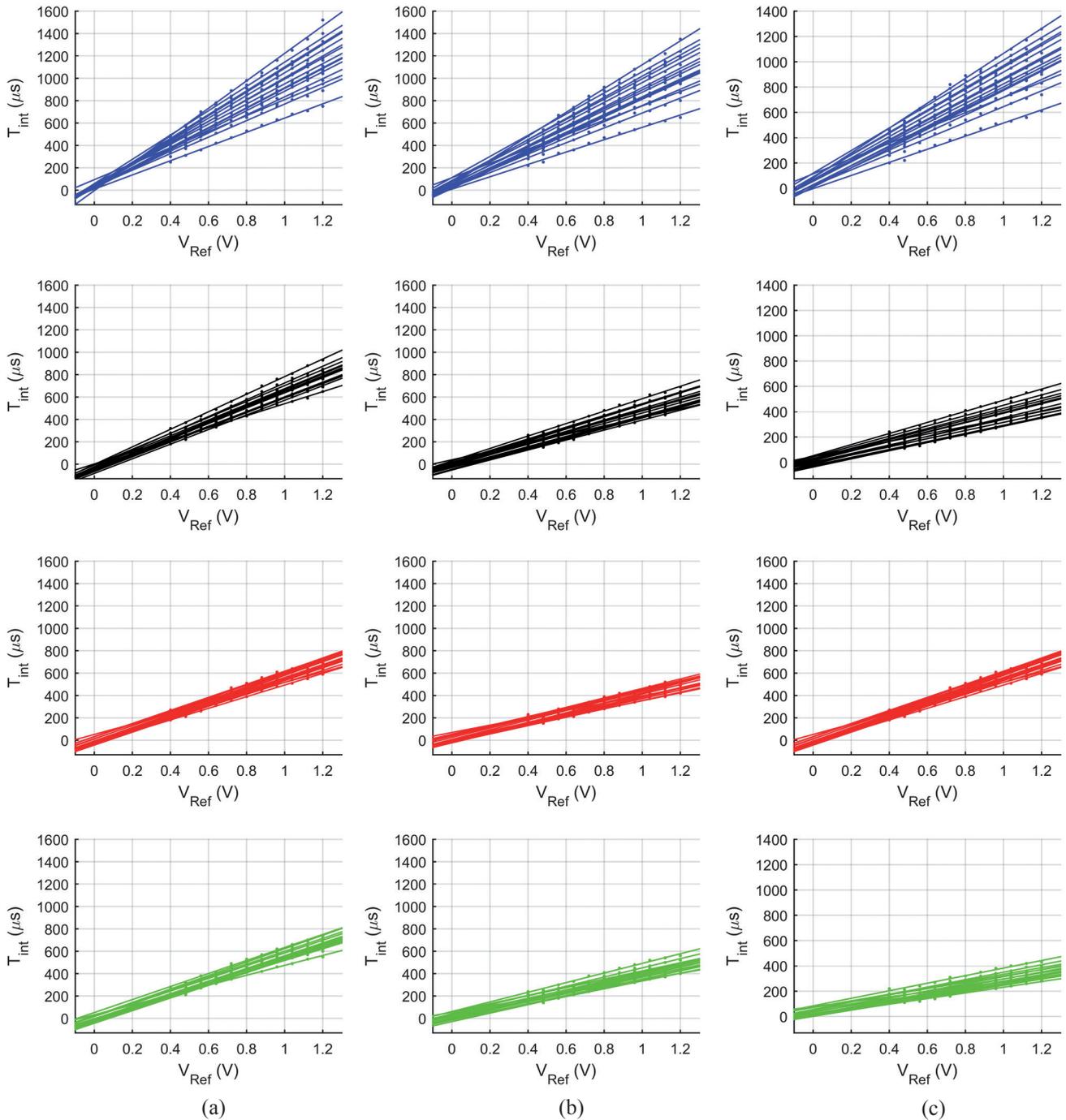


Fig. 8: From bottom to top, relationship between the reference voltage and the integration period for every row of columns one, two, three and four. Results considering (a) the seventh DPCM threshold, (b) the sixth DPCM threshold and (c) the fifth DPCM threshold.

of Fig. 8(a) for matrix columns two, three and four, as well as for the next columns, are proportional to the multiplying factor that maps into the highest threshold current, and thus proportional to the threshold. The exception is column one, because this is the only column that has a prediction value different from zero. Since depends on this prediction and the prediction is also proportional to it will represent a change in the angular coefficient. This change should be the same for every row, but, due to fabrication uncertainties, it is not constant in practice.

Similar results can be found by performing the same test for the previous threshold. These results can be seen in Fig. 8(b). Consequently, the angular coefficient is now proportional to. It is thus expected that the change in the angular coefficient is proportional to ratio between and, which is equal to. In fact, for columns two and beyond, the average ratio between the angular coefficient of the test considering and the angular coefficient considering is equal to 0.69, with a standard deviation of 0.03. For column one, on the other hand, the average ratio is 0.89, with a standard deviation of 0.01.

Considering yet another threshold, θ , the results are shown in Fig. 8(c). The expected ratio between the angular coefficients is equal to 0.68. In this case, the average ratio from column two on is equal to 0.67, with a 0.05 standard deviation. For column one, the average ratio is equal to 0.95, with a standard deviation of 0.02.

Since the main difference between column one and the rest of the columns is the prediction current, it can be assumed that this current is the reason for the difference among the observed angular coefficients. It is reasonable to consider that the model for this column will be equal to $\frac{C}{P}$, where C is the portion of the equation relative to the threshold and P is relative to the prediction value. For the other columns, C is equal to zero, due to the use of the current limiting circuit. It is thus possible to use the models from these columns to estimate the influence of the prediction current in the first column.

Instead of using column two models to perform this estimation, we have decided to use column three. This decision was based on the results shown in Fig. 8, in which the angular coefficients from column two have a higher standard deviation, indicating that there is still a small influence from the previous column. Column three, on the other hand, presents better behavior, showing that the circuit errors were isolated, which can be emphasized by its resemblance with column four behavior.

To estimate the prediction value influence in column one slopes, we have subtracted, row by row, the angular coefficients from column one and column three models. The resulting values should be proportional to the prediction current. By computing the ratio between each result (i.e. difference between column-three and column-one angular coefficients) and the corresponding difference at the previous row from the previous row, we are able to quantify the differences between each θ with respect to the previous row.

This procedure was repeated for thresholds seven, six and five, and for each one we found a vector with 15 values, starting with the ratio between the estimated θ of row two with respect to row one, and ending with the same ratio between rows 15 and 16. The mean vector was used as the proportionality vector. These relations are important to guarantee the compatibility between the decoder and the circuit encoder, but it is still necessary to define θ for any row, in order to find the prediction value for the other 15 rows. This issue was solved by using a white image as target, varying θ of the first row and performing a linear search aiming at minimum standard deviation between the reconstructed DPCM values for the first column of blocks.

The decoded values for the first column of a white image are shown in Fig. 10. The dashed line of this figure shows the result considering the designed θ , which is equal to 0.47 for every row. The standard deviation of this set of decoded values is equal to 0.30. As it can be seen from this figure, the decoded values decrease, row by row, which produces an image column that starts as white, but turns black by the end of the column. The results after applying the proposed

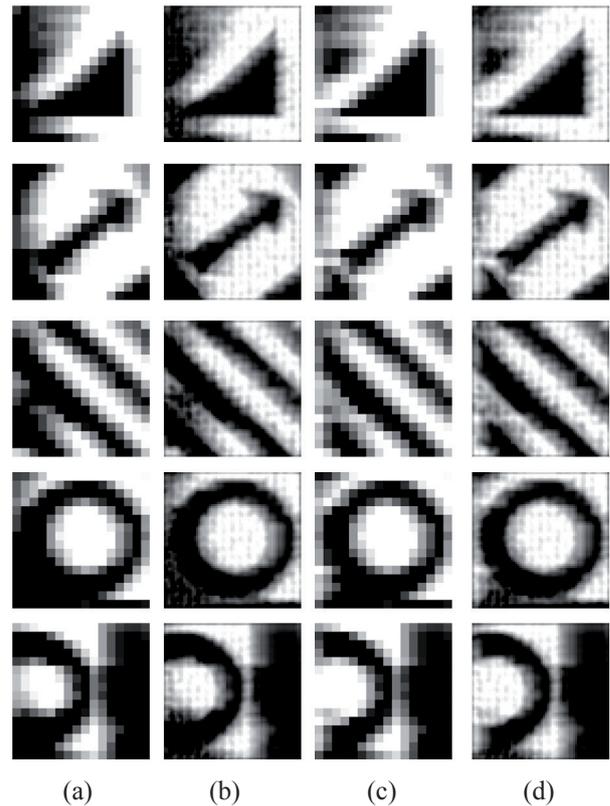


Fig. 9: Various images used to test the proposed correction method, (a) DPCM and (b) final image results without correction and (c) DPCM and (d) final image results applying the correction.

method to adjust the prediction values are shown in dashed-dotted lines and in solid line. The difference between each of these lines is the value chosen for θ for the first row, which varies from 0.16 until 0.44. Using these two values as examples, when θ for the first row is equal to 0.16 and the proposed correction is applied, an array that contains the first column decoded values will have standard deviation of 0.21.

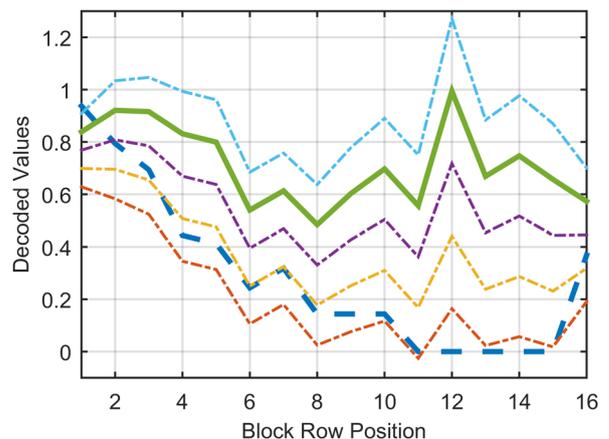


Fig. 10: First column decoded values considering the designed prediction value, in dashed line, and considering the proposed method for five different initial prediction values, in dashed-dotted line and in solid line. The solid line represents the best result, which is the one where the decoded values have lower standard deviation.

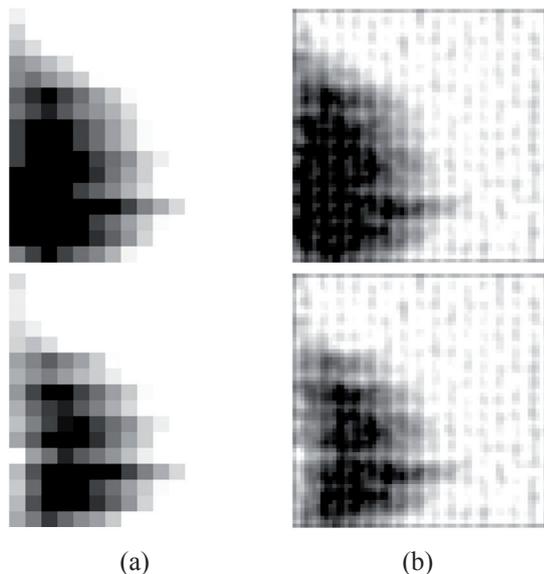


Fig. 11: In the top row, decoded white image without correction, and, in the bottom row, applying the proposed correction method, (a) DPCM decoded image, (b) and final image.

Performing the same computation when for the first row is equal to 0.44, the set of decoded first column values will have standard deviation of 0.16. The lowest standard deviation is 0.15, and it occurs when for the first row is 0.37, which is represented in Fig. 10 in solid line. Although the ideal result would be a decoded column with all the values equal to one and standard deviation equal to zero, the proposed approach has successfully decreased the standard deviation and made the decoded value more uniform.

Fig. 11 shows the effects of the correction applied in a white image. The top row of this figure displays the result without the correction and the bottom row is the result with the proposed method. As it can be seen, in the DPCM result, Fig. 11(a), the first column is more uniform when the correction is applied. A shadow can still be observed in the image, but the proposed correction was able to reduce it. Fig. 11(b) shows the final image, in which the effect of the correction method is also noticeable. In these figures, the maximum and minimum pixel values were defined and controlled by the decoder, not allowing block values to be negative or above one. The other figures from this section will also consider this adjustment. Another example can be seen in Fig. 12. In this case, since the image starts with a dark column, the results achieved with proposed method are worse than those obtained without correction for the first column. Since the correction method was found using a white image, better results are expected in such cases. On the other hand, we can see from the VQ results that the second column of the image should be white, so the correction method helps in turning this column brighter, while it is entirely black in the original result.

Fig. 9 presents additional examples used to test the proposed approach, where the DPCM and final image are dis-

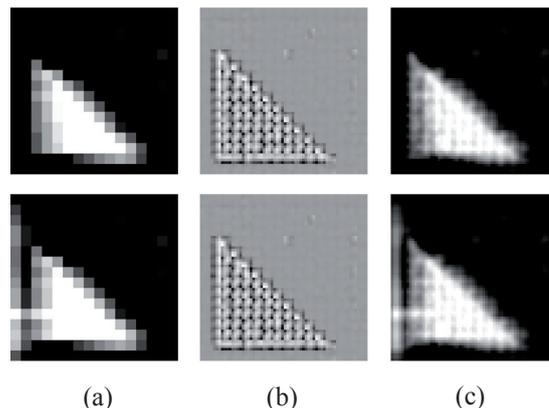


Fig. 12: In the top row, decoded white image without correction, and, in the bottom row, applying the proposed correction method, (a) DPCM decoded image, (b) VQ, (c) and final image.

played, with and without the proposed correction method. As expected, the proposed approach helps in increasing the image quality when the first column is bright, but it also whitens the first column even when the image is dark.

V. CONCLUSION

This paper compared the performances in terms of algorithm and experimental results produced by two IC design realizations, both capable of capturing and compressing images at the focal plane. Table I summarizes the main differences between the chips. The results showed that there was a significant improvement from the first design to the second one in the linear transform and vector quantization steps, since the borders of the images are better defined. On the other hand, the DPCM stage can be further improved. The DPCM errors were analyzed and modeled with the purpose of generating a method for compensating such errors. In order to do that, the circuit to limit the maximum current was used to uncouple column one from the rest of the columns. The proposed method was successful in making the first column of the image more uniform for white images. When the image starts in black the method whitens the result, but the effect in the following columns is minimum, causing little blur.

The paper also showed that it is possible to use the current limiting circuit to operate with the chip in a low-bit PCM mode. This option produced good results, even with a low bit count. Future work includes modulation transfer function [19] and fixed-pattern noise measurements, with the goal of fully characterizing the IC design.

Table I. Comparison Between First and Second Generation Chips

	1 st generation	2 nd generation
Bit rate	0.94 bpp	1.33 bpp
Transform coeffs.	4	5
Sign bits	4	5
VQ bits	7	9
Fab. Process	AMS 0.35 μm Opto	IBM 0.18 μm
Transistor count	607 per block	833 per block
Pixel area	37.5 $\mu\text{m} \times 37.5 \mu\text{m}$	27.2 $\mu\text{m} \times 27.2 \mu\text{m}$
Photodiode area	10 $\mu\text{m} \times 10 \mu\text{m}$	10 $\mu\text{m} \times 10 \mu\text{m}$
Fill factor	7.1%	13.5%
Chip area	2.4 mm \times 2.1 mm	2.8 mm \times 2.8 mm
Resolution	32 \times 32	64 \times 64
DPCM	0.0	> 0.0*
Power supply	3.3V	1.8V

*For the 2nd generation, is different for each row of the matrix, the method proposed in this paper aims at correcting this non-uniformity.

ACKNOWLEDGEMENTS

This work was supported by Brazilian higher education and research funding agencies: CAPES, CNPq, and FAPERJ. Special thanks to CMSatisloh for providing us with the optical setup necessary for the experimental tests.

REFERENCES

[1] A. N. Belbachir, *Smart Cameras*, Springer Science+Business Media, 2010.

[2] M. Kriss, *Handbook of Digital Imaging*, 1st ed., vol. 1, Jon Wiley & Sons, 2015

[3] J. Nakamura, *Image Sensors and Signal Processing for Digital Still Cameras*, 1st ed., Taylor & Francis Group, 2006.

[4] S. Vargas-Sierra, G. Liñan-Cembrano, and A. Rodríguez-Vázquez, "A 151 dB high dynamic range CMOS image sensor chip architecture with tone mapping compression embedded in-pixel," *IEEE Sensors Journal*, vol. 15, no. 1, 2015, pp. 180–195.

[5] J. Fernández-Berni, R. Carmon-Galán, and A. Rodríguez-Vázquez, "Single-exposure HDR technique based on tunable balance between local and global adaptation," *Circuits and Systems II, IEEE Transactions on*, 2015.

[6] S. Chen, A. Bermak, and Y. Wang, "A CMOS image sensor with on-chip image compression based on predictive boundary adaptation and memoryless QTD algorithm," *Very Large Scale Integration Systems (VLSI), IEEE Transactions on*, vol. 19, no. 4, 2011, pp. 538–547.

[7] W. D. Leon-Salas, S. Balkir, K. Sayood, N. Schemm, and M. W. Hoffman, "A CMOS imager with focal plane compression using predictive coding," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 11, 2007, pp. 2555–2572.

[8] J. A. Leñero-Bardallo, R. Carmona-Galán and Á. Rodríguez-Vázquez, "A Wide Linear Dynamic Range Image Sensor Based on Asynchronous Self-Reset and Tagging of Saturation Events," in *IEEE Journal of Solid-State Circuits*, vol. 52, no. 6, pp. 1605-1617, June 2017.

[9] J. Illade-Quinteiro, V. M. Brea, P. López and D. Cabello, "Time-of-flight chip in standard CMOS technology with in-pixel adaptive number of accumulations," *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, Montreal, QC, 2016, pp. 1910-1913.

[10] C. A. de M. Cruz, D. W. de L. Monteiro, E. A. Cotta, V. F. de Lucena and A. K. P. Souza, "FPN Attenuation by Reset-Drain Actuation in the Linear-Logarithmic Active Pixel Sensor," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 10, pp. 2825-2833, Oct. 2014.

[11] F. D. V. R. Oliveira, H. Haas, J. G. R. C. Gomes, and A. Petraglia, "CMOS imager with focal-plane analog image compression combining DPCM and VQ," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 60, no. 5, 2013, pp. 1331–1344.

[12] F. D. V. R. Oliveira, J. G. R. C. Gomes, and A. Petraglia, "Influence of cascode and simple current mirrors in inner product implementations for CMOS imagers," in *Circuits Systems (LASCAS), 2015 IEEE 6th Latin American Symposium on*, Feb 2015, pp. 1–4.

[13] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Second Edition. Prentice-Hall, 2002.

[14] D. Salomon, *Data Compression – The Complete Reference*. Third Edition. Springer, 2004.

[15] F. D. V. R. Oliveira, J. G. R. C. Gomes, and A. Petraglia, "Comparison of low-complexity image compression algorithms for analog circuit implementation," in *2014 14th International Workshop on Cellular/Nanoscale Networks and their Applications (CENNA)*, July 2014, pp. 1–2.

[16] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[17] J. Otha, *Smart CMOS Image Sensors and Applications*, CRC Press, 2008.

[18] F. D. V. R. Oliveira, H. L. Haas, J. G. R. C. Gomes, and A. Petraglia, "Current-mode analog integrated circuit for focal-plane image compression," in *Integrated Circuits and Systems Design (SBCCI), 2012 25th Symposium on*, Aug 2012, pp. 1–6.

[19] G. D. Boreman, *Modulation Transfer Function in Optical and Eletro-Optical Systems*. SPIE, 2001.