

Feature Detection in Analog VLSI

Alberto Pesavento
Department of Electrical Engineering
California Institute of Technology
Pasadena, CA, 91125
alberto@klab.caltech.edu

Christof Koch
Computation and Neural Systems Program
California Institute of Technology
Pasadena, CA, 91125
koch@klab.caltech.edu

Abstract

Feature detection and tracking is a fundamental problem in computer vision research. By detecting and tracking features in an image sequence it is possible to recover information on both the motion of the viewer and the structure of the environment. We designed and tested a CMOS imager with analog VLSI focal plane computation for feature detection. The chip implements a feature detection algorithm that is suitable for integration in a compact analog VLSI chip. We will review the algorithm, its analog VLSI implementation and results from the chip.

1 Introduction

In principle, from the stream of image frames produced by a moving camera, it is possible to recover the shape of objects in the field of view, and the motion of the camera. Information on both the structure and the motion of the viewer can be recovered from the displacement of key features in the image sequence. The selection and tracking of feature in an image stream is therefore an important problem in computer vision. The selection of the features is a computationally intensive task and usually is accomplished off-line on a sequence of images recorded from a camera. These solutions are bulky, expensive, require cameras and frame grabbers and have a very high power consumption. Existing real-time systems, of which very few exist, are the ASSET-2 [10] systems that uses four VME boards, equipped with custom hardware circuits, and the implementation by Benedetti and Perona [1] that uses a custom board with six FPGAs hosted in a Pentium PC. We designed and tested the first CMOS imager with focal plane computation for feature detection. The chip implements a modified version of a feature detection algorithm that is suitable for integration in a compact analog VLSI chip.

2 Feature detection algorithm

Not all the regions in an image contain motion information. Some researcher proposed to track corners, or windows with high spatial frequency content, or region where some mix of second-order derivatives was sufficient high. Tomasi and Kanade [11], instead of defining *a priori* the characteristics of the features to track, use a different approach. They are interested in a region only if that region can be tracked well and they omit a region if it is not good enough for the purpose. In this way the selection criterion is optimal by construction. We here review the selection method proposed by Tomasi and Kanade [11].

If $I(x, y, t)$ is the image brightness we can write, under the assumption of constant brightness,

$$\frac{dI(x, y, t)}{dt} = I_x v_x + I_y v_y + I_t = [I_x, I_y] \mathbf{v} + I_t = 0 \quad (1)$$

where $\mathbf{v} = [v_x, v_y]^T$ and the partial derivatives of $I(x, y, t)$ with respect of x , y and t are denoted by I_x , I_y and I_t respectively. If we make the assumption that all the N points in the region of interest are moving at the same speed, which is reasonable for small inter-frame displacements and a small neighborhood, we can formulate the linear problem

$$\begin{bmatrix} I_x^1 & I_y^1 \\ \vdots & \vdots \\ I_x^N & I_y^N \end{bmatrix} \mathbf{v} = - \begin{bmatrix} I_t^1 \\ \vdots \\ I_t^N \end{bmatrix} \quad \text{or} \quad \mathbf{A} \mathbf{v} = \mathbf{b}. \quad (2)$$

The velocity vector \mathbf{v} can be computed as the least squares solution to $\mathbf{A} \mathbf{v} = \mathbf{b}$, i.e. $\mathbf{v} = \mathbf{G}^{-1} \mathbf{A}^T \mathbf{b}$, where

$$\mathbf{G} = \mathbf{A}^T \mathbf{A} = \begin{bmatrix} \sum_{k=1}^N (I_x^k)^2 & \sum_{k=1}^N I_x^k I_y^k \\ \sum_{k=1}^N I_x^k I_y^k & \sum_{k=1}^N (I_y^k)^2 \end{bmatrix} = \begin{bmatrix} a & b \\ b & c \end{bmatrix}. \quad (3)$$

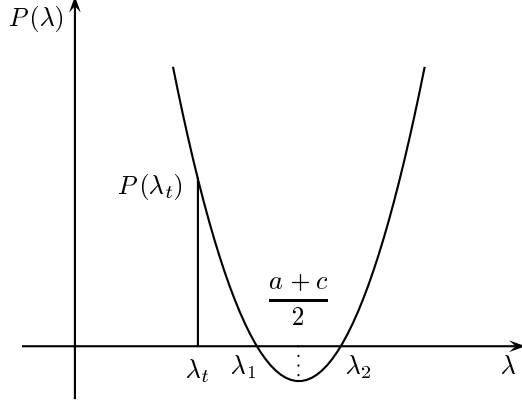


Figure 1. Requiring $\lambda_1 > \lambda_t$ is equivalent to the conditions $P(\lambda_t) > 0$ and $\frac{a+c}{2} > \lambda_t$, as shown in this picture.

The feature can be tracked from frame to frame if the systems (2) can be solved reliably. This means that \mathbf{G} must be both above noise and well-conditioned. The noise requirement implies that both eigenvalues λ_1 and λ_2 (with $\lambda_1 \leq \lambda_2$) of \mathbf{G} must be large, while the conditioning requirements means that they cannot differ by several order of magnitude, that is $\text{cond}(\mathbf{G}) = \frac{\lambda_2}{\lambda_1} < c_{th}$ is bounded.

Two small eigenvalues mean a roughly constant intensity profile within a window. A large and a small eigenvalue correspond to unidirectional pattern. Note that when one eigenvalue is zero, (for instance \mathbf{G} is rank deficient), there is an unreachable subspace in the solution for \mathbf{v} that corresponds to the velocity component parallel to the direction of the edge. This situation is often referred to as the Aperture Problem. Two large eigenvalues are found for windows with corners, salt-and-pepper textures, or any other pattern that can be tracked reliably.

For all practical purposes, when the smaller eigenvalue is sufficiently large to meet the noise criterion, the matrix \mathbf{G} is usually well conditioned. This is due the fact that the intensity variations in a window are bounded by the maximum allowable pixel value, so the greater eigenvalue, λ_2 , cannot be arbitrarily large.

This observation simplify the selection of the trackable windows as to the ones for which

$$\lambda_1 > \lambda_t \quad (4)$$

where λ_1 is the minimum eigenvalue and λ_t is a predefined threshold value.

The eigenvalues of the matrix \mathbf{G} are the roots of the characteristic polynomial

$$P(\lambda) = (a - \lambda)(c - \lambda) - b^2 \quad (5)$$

where a , b and c are the coefficients of \mathbf{G} , see (3). The minimum eigenvalue λ_1 is defined by

$$\lambda_1 = \frac{a+b}{2} - \sqrt{\left(\frac{a+b}{2}\right)^2 - (ac - b^2)}. \quad (6)$$

The expression of λ_1 in (6) it is too complex to be implemented in analog VLSI and therefore a complexity reduction is necessary.

From Fig. 1 we can see that the condition expressed in (4) is equivalent to

$$P(\lambda_t) > 0 \quad \text{and} \quad \frac{a+c}{2} > \lambda_t \quad (7)$$

and remembering that the coefficients a and c are positive by construction, a simpler sufficient condition is

$$P(\lambda_t) > 0 \quad \text{and} \quad a > \lambda_t \quad \text{and} \quad c > \lambda_t. \quad (8)$$

We summarize the algorithm to select features in an image as:

- 1) Compute I_x and I_y at every pixel location;
- 2) For the window centered at (x, y) compute a , b and c ;
- 3) Calculate $P(\lambda_t) = (a - \lambda_t)(c - \lambda_t) - b^2$;
- 4) The window contains a trackable feature if

$$P(\lambda_t) > p_n \quad \text{and} \quad a > \lambda_t \quad \text{and} \quad c > \lambda_t \quad (9)$$

where p_n is a second threshold introduced to eliminate any false positive given by image noise.

3 Analog VLSI implementation

The analog VLSI implementation of the algorithm used windows of size 3×3 pixels ($N = 9$). Every pixel at the edge of the window computes the spatial-derivative both along the x axis and along the y axis from the adjacent pixels and therefore the actual number of photoreceptors involved in the computation is 21 and the patch has the shape depicted in Fig. 2a.

At every pixel i the chip computes

$$\begin{aligned} (I_x^i)^2 &= I_x^i I_x^i = (V_{left}^i - V_{right}^i)(V_{left}^i - V_{right}^i) \\ (I_y^i)^2 &= I_y^i I_y^i = (V_{up}^i - V_{down}^i)(V_{up}^i - V_{down}^i) \\ I_x^i I_y^i &= (V_{left}^i - V_{right}^i)(V_{up}^i - V_{down}^i), \end{aligned} \quad (10)$$

where the quantities V_{left}^i , V_{right}^i , V_{up}^i and V_{down}^i are photoreceptors voltages of the four neighboring pixels, see Fig. 2b. We then need to add these three values to the general

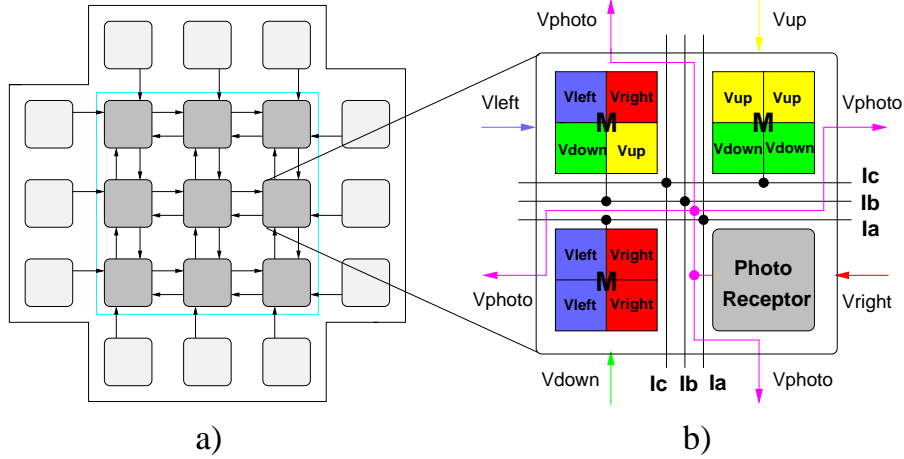


Figure 2. **a)** We used a 3×3 pixels window. The number of photoreceptors involved in the computation is 21. **b)** At every pixel the quantities $(I_x^i)^2$, $(I_y^i)^2$ and $I_x^i I_y^i$ are computed by three four quadrant multipliers and added to the global I_a , I_b and I_c wires.

summation that is carried out in the window. Both the multiplication and the summation is elegantly accomplished using three four quadrant multipliers with current output connected to the corresponding wires of the currents I_a , I_b and I_c that represent the quantities a , b and c .

The photoreceptor used in this imager is the adaptive photoreceptor proposed by Delbrück and Mead [2] that in a normal design has a peak-to-peak output voltage of about 500mV. If images with high contrast were focused onto the chip, a standard Gilbert four quadrant multiplier [5] in sub-threshold CMOS with a linear range of about 100mV would saturate, degrading the precision of the computation. We designed a multiplier with a larger linear range [9]. The multiplier (Fig. 3), has a linear range of about $\pm 1V$, a factor of 20 increase with respect to the normal 100mV, with less than double the number of transistors. The power consumption of the multiplier under normal bias condition is below $1\mu W$.

Finally, steps **3)** and **4)** of the algorithm are implemented using the simple network of Multi Input Translinear Elements (MITEs) [6, 7] reported in Fig. 4. The two nFETs in box 1 subtract the current I_t , representative of λ_t , to the two input currents I_a and I_c . The three MITEs output the current $(I_a - I_t)(I_c - I_t)/I_{ref}$ (i.e. $(a - \lambda_t)(c - \lambda_t)$) that is then mirrored to the node A. It is interesting to notice that this circuit, at the same time, test for the two conditions $I_a > I_t$ and $I_c > I_t$ (corresponding to the conditions $a > \lambda_t$ and $c > \lambda_t$) because if one of the two input currents I_a and I_c is less than the threshold current I_t the current output of the network of three MITEs is zero. In box 2, first the absolute value of the current I_b is computed, then, using the two MITEs, the current is squared to obtain I_b^2/I_{ref} . Finally the inverter connected to node A performs the final comparison:

its output is low if $(I_a - I_t)(I_c - I_t)/I_{ref} - I_b^2/I_{ref} > I_n$ (condition $P(\lambda_t) = (a - \lambda_t)(c - \lambda_t) - b^2 > p_n$) and high otherwise.

4 Experimental results

We designed, fabricated and tested a chip with a 8×8 array of pixels that allows us to detect features in four windows of 3×3 pixels. It is worth noticing that, even if the four windows are non-overlapping, the actual 21-pixels patches used in the algorithm are overlapping and only the photoreceptor of the central pixel of the window contribute to only one patch. The chip was fabricated in a Tiny-Chip dye in a $1.2\mu m$ double-poly double-metal process available through the MOSIS fabrication service.

We tested the chip with different stimuli and it performed reliably in all the tests we performed. In Fig. 5a the image of a pen was projected onto the chip; as expected none of the four patches reported a feature. In Fig. 5b the tip of the pen was just on the top-right patch and the feature was correctly detected by the corresponding thresholding circuit. Additional examples, with different stimuli, can be found at the web site http://www.klab.caltech.edu/~alberto/-feature_chip.html where we posted movies generated with the experimental data directly recorded from the chip.

5 Conclusion

We presented a CMOS imager with focal plane computation for feature detection. We reviewed the algorithm, its analog VLSI implementation and presented results from the

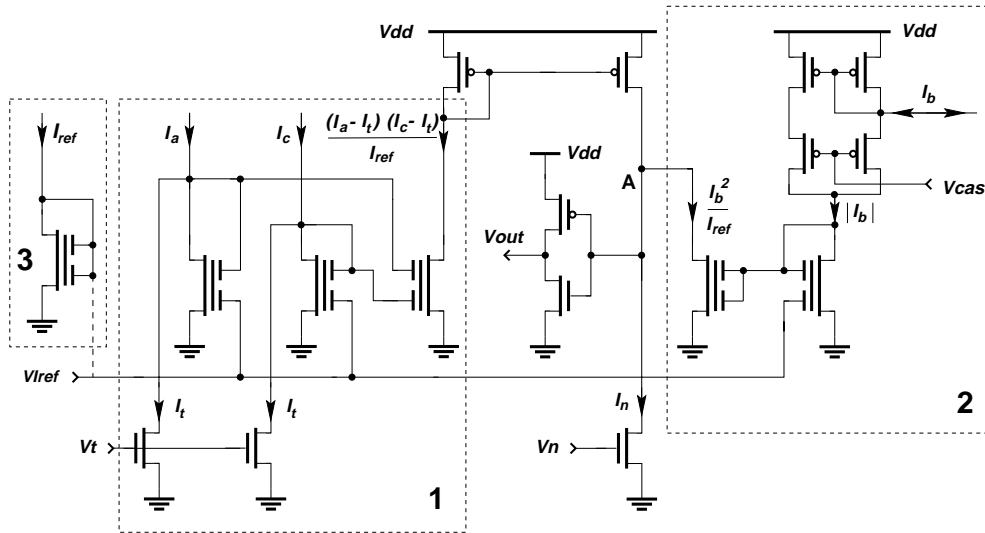


Figure 4. Network of MITEs that implements steps 3) and 4) of the algorithm. If $(I_a - I_t)(I_c - I_t)/I_{ref} - I_b^2/I_{ref} > I_n$ the output of the inverter goes low to indicate the presence of a feature and stays high otherwise. The voltage $V_{I_{ref}}$ could be either supplied as an external bias or generated injecting a current in a diode-connected MITE as illustrated in box 3.

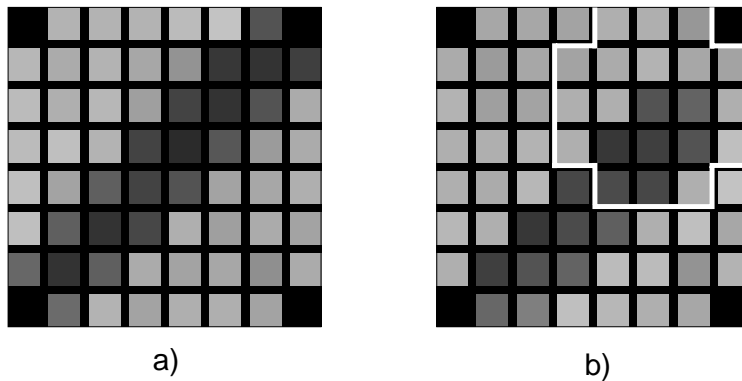


Figure 5. a) The image of a pen is projected onto the chip. The tip is not in the field of view and the chip correctly reports that there are no features present. b) When the tip of the pen was in the field of view of the top-right patch the chip correctly signals the presence of the feature.