

Applying neuromorphic vision sensors to planetary landing tasks

Garrick Orchard[†], Chiara Bartolozzi[∇], and Giacomo Indiveri^{*}

[†]Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore MD, USA

^{*}Institute of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland

[∇]Robotics, Brain and Cognitive Sciences Dept., Italian Institute of Technology, Italy

Email: gorchard@jhu.edu, chiara.bartolozzi@iit.it, giacomo@ini.phys.ethz.ch

Abstract—Recently there has been an increasing interest in application of bio-mimetic controllers and neuromorphic vision sensors to planetary landing tasks. Within this context, we present combined low-level (SPICE) and high-level (behavioral) simulations of a novel neuromorphic VLSI vision sensor in a realistic planetary landing scenario. We use results from low level simulations to build an abstract description of the chip which can be used in higher level simulations which include closed-loop control of the craft.

I. INTRODUCTION

Landing is a critical stage of any planetary exploration mission. As smooth a landing as possible is desired to protect on-board equipment. In comparison to landing on Earth, landing on other planets is made more challenging by the constraints under which it must be achieved. No existing infrastructure is available on the surface for guidance, and the atmosphere (or lack thereof) limits the sensing modalities available. Furthermore, as with any space mission, cost rises rapidly with weight. Therefore custom lightweight vision sensors with low power consumption to limit battery weight are an attractive alternative to standard machine-vision systems. Reliability is also crucial because repair of a remote craft is not usually feasible. We propose to use vision sensors and bio-mimetic control principles for planetary landing tasks. Specifically, we propose to implement the motion sensing and computation operations required to control planetary landing tasks by using a set of compact neuromorphic vision sensors distributed on the craft external surface. The low-power vision processing capabilities of the neuromorphic sensors, combined with the distributed approach allow both a low weight budget and a redundant fault-tolerant architecture.

Neuromorphic vision sensors are custom VLSI devices that process images directly at the focal plane level. These sensors typically use bio-inspired circuits which implement hardware models of the first stages of visual processing in biological systems [1], [2]. In biological retinas, early visual processing is performed by receptors and neurons arranged in a manner that preserves the retinal topography with local interconnections. Neuromorphic circuits have a similar physical organization: photo-receptors, memory elements, and computational nodes share the same physical space on the silicon surface and are combined into local circuits that process, in real-time, differ-

ent types of spatio-temporal computations on the continuous analog brightness signal.

The highly distributed nature of physical computation in neuromorphic systems leads to efficient processing that would be computationally expensive on general-purpose digital machines. For example, like their biological counterparts, neuromorphic sensors can operate over an input range covering many orders of magnitude, despite limited bandwidth [1], [3]. This extraordinary performance is achieved by a simple but densely parallel process that involves continually adapting local reference signals to the average signal statistics prevailing there.

In this work we present the combined circuit and behavioral software simulation results of a novel tracker motion sensor device, in a realistic planetary landing scenario which includes the closed-loop control of the landing craft.

Detailed circuit simulations are typically carried out using SPICE simulators. However the closed-loop control requirements imply that current sensor output signals determine how future sensor inputs change. For this reason SPICE simulations are not well suited. In addition, the small time-steps typically used in SPICE simulations are ill-suited for use in the behavioral landing simulations, of which the sensor is only one component.

We approach the problem by initially simulating individual circuit blocks in SPICE and then using these results to build a higher level model of each block. We and then combine these blocks into a full chip simulation from which we derive a simple behavioral model of the sensor for use in the landing simulations.

II. THE NEUROMORPHIC VISION SENSOR

The Neuromorphic Vision Sensor used in this work is an asynchronous event-driven motion chip recently developed and fabricated, named as Tracker Motion Sensor (TMS). The TMS extracts the spatial derivative and the temporal derivative of the temporal contrast change detected by each photoreceptor and extracts edges of moving objects and their velocity. The chip readout can be either “traditional” frame-based scanning of the pixels output, or asynchronous. In the latter case the output is read in order of decreasing pixel saliency, implementing a form of stimulus-driven attention.

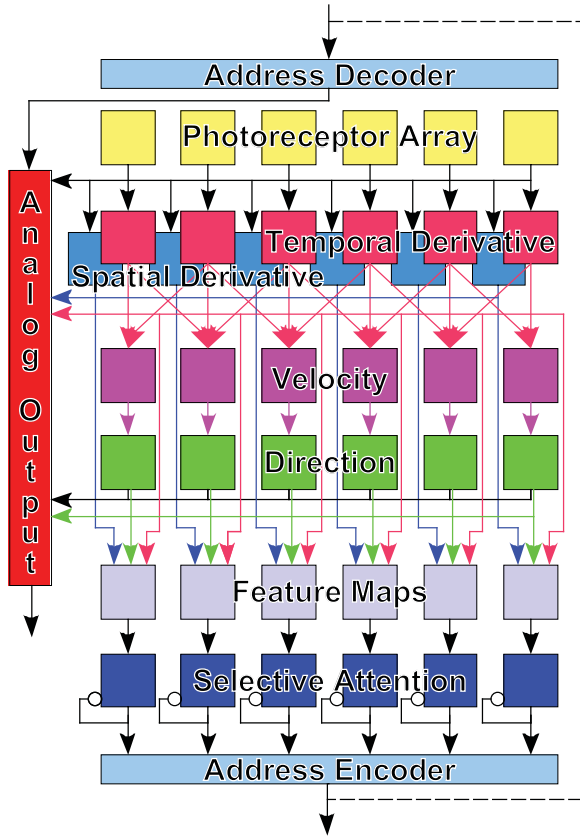


Fig. 1. Tracker Motion Sensor Block Diagram: the photo-receptor’s output is spatially and temporally derived, fast transitions of the photo-current triggers an event and the computation of velocity and direction of the event movement over the array. The temporal and space derivative and the speed are summed to compute the saliency map. A WTA network with self inhibition scans the saliency map in order of decreasing saliency. The identity of the winning pixel is a digital output of the chip and can be used to steer the readout of the analog values corresponding to the winning pixel.

Figure 1 shows the block diagram of the TMS. The input is an adaptive logarithmic photoreceptor [4], that responds to temporal variations of the logarithm of light intensity impinging on the photo sensitive area; this makes the photoreceptor sensitive to the relative contrast change, rather than to absolute illumination, furthermore the photoreceptor is capable of adapting its response to the global scene illumination level, accomplishing a wide dynamic range.

The temporal derivative circuit detects fast variations of the photoreceptor output, typically corresponding to edges of objects moving in front of the sensor. Such an event triggers the computation of velocity as time to travel of the event across neighboring pixels, by “Facilitate-and-sample” circuits based on [5].

The attentional read-out is implemented by means of a Winner-take-All circuit (WTA) that selects the pixel with highest activity in the saliency map [6], computed by summing with tunable weights the outputs of the spatial and temporal derivative and of the velocity circuits. The attentional system comprises also a self-inhibition mechanism that implements shifts of attention and allows to scan the pixels output in order

of decreasing saliency [7]. The output of the chip indicates the location of the object moving with highest saliency and can be used to select the analog outputs of the corresponding pixel. In addition it is possible to read-out the analog outputs (spatial and temporal derivative, and velocity) of each individual pixel.

In the current work we intend to use the velocity measurements of the TMS as the main sensor output for controlling the craft landing. The sensor has not been fully characterized yet and was therefore simulated to obtain velocity measurement estimates.

III. SIMULATION

Following our distributed sensor approach we plan to place sensors in pairs perpendicular to each other allowing for measurement of optical flow along two orthogonal axes. This arrangement of sensors was simulated on three different levels. The lowest-level simulations involve simulating individual circuit blocks at the transistor level, with specified control inputs, to verify their operation and determine their transfer function. These transfer functions are used to implement full-chip mid-level simulations, in which input signals are generated from realistic landing scenarios, to investigate the effect of transistor mismatch and the direction of motion on the output. The high level simulations determine sensor outputs as a function of the rotational and translational velocities of the chip and the accuracy determined from the mid-level simulations.

A. Low-Level Simulation

Circuit blocks described in Fig. 1 were simulated at a transistor level to gain a better understanding of their operation and to predict their response to known signals. These simulations are typically run with a microsecond time-step and span only a few milliseconds.

B. Mid-Level Simulation

When possible, we derived analytically the equations describing the response properties of each circuit block (from circuit analysis). The low-level simulations were instrumental for determining the unknown parameters in the equations. To account for worst case transistor mismatch we randomly varied transistor widths in the block level equations from -20% to +20%. For more complex circuit blocks the output waveforms computed by the low-level SPICE simulations were stored and used as look-up tables by the mid-level simulation.

At this level, we connected all circuit blocks together (as shown in Fig. 1) and carried out full chip simulations, from simulated optical input to final voltage output.

The optical inputs were derived from artificially generated images of the moon surface obtained from PANGU (Planet and Asteroid Natural-Scene Generation Utility) (see Fig. 2). The images were up-sampled and blurred to simulate the effect of the sensor’s finite pixel size and field of view (FOV). Typically a 5° FOV was used, but this can be changed in simulation or on the actual sensor by mounting a different lens. After scaling and blurring the image, we subtracted its zero to account for the fact that we assume the photo-receptors are fully adapted

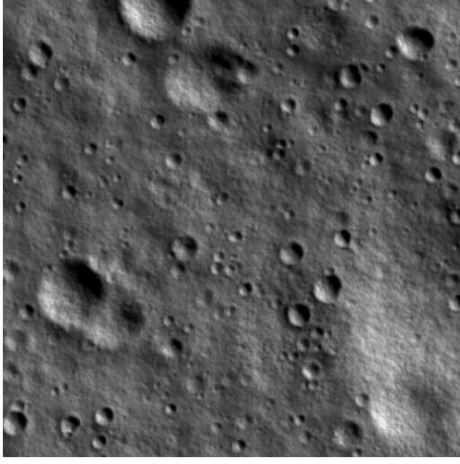


Fig. 2. A typical image of the moon surface generated by the PANGU software.

to the current lighting conditions. For each pixel we traced a path through the image, based on the array orientation, speed and direction of movement. Pixel values along this trace are scaled and used as an input to the photoreceptor circuits.

In simulation the user can set the chip parameters which are controlled by bias voltages (the user specifies the desired parameters and the simulation calculates the appropriate circuit bias voltages). These simulations are typically run with a microsecond time-step.

C. High-Level Simulation

High level simulation of the sensor was implemented as a Matlab function which is called by the landing simulation. The function has no explicit time-step, but rather returns a result when requested. The function works by calculating the actual motion of the image on the optical plane (1) before taking into account error introduced by the aperture problem (2) [8] and the transistor mismatch. The sensor is assumed to be fixed in position and orientation relative to the craft. Forward kinematics are used to infer the sensor position, orientation, rotational velocity and translational velocity from the craft position, orientation and motion. With this knowledge the image motion can be calculated at each pixel as:

$$\begin{aligned}
 \frac{dx_{pix}}{dt} &= \frac{64}{2 \tan(\frac{FOV}{2})} \frac{dx_s}{dt} \\
 \frac{dy_{pix}}{dt} &= \frac{64}{2 \tan(\frac{FOV}{2})} \frac{dy_s}{dt} \\
 \frac{dx_s}{dt} &= \frac{v_{zs}x_s - v_{xs}}{z_s} - \omega_y + \omega_z y_s + \omega_x x_s y_s - \omega_y x_s^2 \\
 \frac{dy_s}{dt} &= \frac{v_{zs}y_s - v_{ys}}{z_s} - \omega_x - \omega_z x_s + \omega_y x_s y_s - \omega_x y_s^2
 \end{aligned} \quad (1)$$

Where v is velocity, subscript s denotes co-ordinates relative to the sensor, subscript pix indicates pixel co-ordinates and ω is rotational velocity about the sensor axes. The first two equations serve to convert from sensor to pixel co-ordinates (64 pixels in the array).

D. The aperture problem

When viewing only a small section of an edge, the component of motion in the direction of the edge is ambiguous. This is the aperture problem which all visual sensors are prone to, including the human visual system. The problem is even more serious when using a linear array to detect motion because the array measures only the component of the motion that lies along the direction of its axis. If the array axis orientation is different from the direction of motion of the edge this will introduce an error (see Fig. 4). The measured edge velocity can be described by:

$$v_{measured} = \frac{v_{edge} \sin(\phi)}{\sin(\phi - \theta)} \quad (2)$$

Where v_{edge} is the speed at which the edge moves across the image plane, ϕ is the angle between the edge and its direction of motion, θ is the angle between the array and the direction of motion of the edge.

We assume that image edges will mainly occur at crater boundaries and therefore be circular. The symmetry allows us to define the probability of encountering an edge with orientation ϕ to the sensing array as:

$$p(\phi) = \frac{\sin(\phi)}{2}$$

Thus we are most likely to encounter an edge with orientation $\phi = \frac{\pi}{2}$ which greatly simplifies our equations. In the following simulation results we assume that ϕ is always equal to $\frac{\pi}{2}$. This is equivalent to measuring the normal flow [8].

IV. SIMULATION RESULTS

A. Low-level simulations

Simulation of the photoreceptor circuit reveals that it can be accurately approximated as a constant gain on the input signal for low frequencies and short time-scales. This does not hold true for high frequencies, but the finite pixel size and the lens transfer function of our current application will cause blurring, which will remove high frequencies from the incoming signals. Over longer simulation times the assumption of constant gain also fails because the photoreceptor adapts.

Simulation results for four adjacent pixels derived using the image in Fig. 2 are shown in Fig. 3.

B. Mid-level simulations

When the array and velocity are aligned ($\theta = 0$) and no transistor mismatch is present the full chip simulation results are exactly as expected. Inclusion of mismatch causes a random error of -15 to 15% in each pixel output. One of the terms defining the strength of each pixel in the saliency map is the measured velocity at that pixel. This causes the sensor to be biased towards higher velocities, which results in a typical overestimate the optical flow by 15% (see Fig. 4).

The TMS is biased to measure the flow at the point of highest contrast in an image and indicates whether sufficient contrast is present to make a reliable measurement. Smooth transitions of the WTA output between adjacent pixels suggest

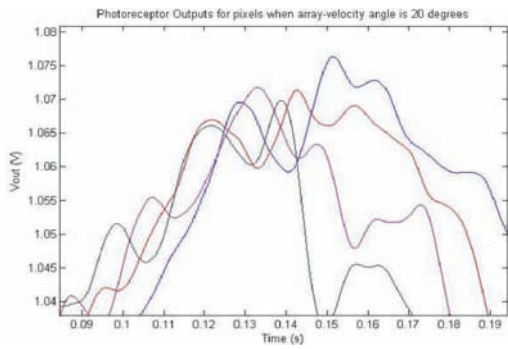


Fig. 3. An example of photo-receptor outputs for four adjacent pixels derived using the image in Fig. 2. In this case $\theta = 20^\circ$. The pixels are adjacent to each other and are therefore all focused on a small portion of the image. They have very similar outputs due to the low spatial frequency content of the image.

that a single edge is being tracked across the image plane. A constant velocity output from the TMS as the WTA transitions between pixels suggests that the edge is moving with constant velocity. We can therefore derive a confidence estimate in the output of a single sensor by monitoring the WTA and velocity outputs. By considering an orthogonal pair of sensors we can determine both the direction and magnitude of the normal flow and by making a comparison between outputs of nearby orthogonal pairs we can further refine the confidence estimate.

C. High-Level Simulation

The high level simulation allows the output of the sensor to be determined for an arbitrary position, orientation, rotation and linear velocity of the craft. Multiple sensors can be simulated by just specifying their positions and orientations relative to the craft center of mass (COM).

V. CONCLUSIONS

We have simulated the operation of a neuromorphic vision sensor in realistic planetary landing scenarios, at three different levels. At the lowest level we carried out SPICE simulations to characterize the response properties of the sensor's individual blocks; at the intermediate level we used transistor equations to derive analytically the individual circuit block behavior and carry out full-chip simulations, by providing realistic inputs from planetary landing scenarios; at the last level we derived velocity estimates for controlling the craft in the landing task, by combining outputs from perpendicularly oriented sensors, placed on the surface of the craft. Our results are useful for evaluating the overall performance of the chip in this specific task. The simulations achieve their aim of enabling us to verify bio-inspired control strategies for planetary landing using different configurations of sensor placements on the surface of the craft.

ACKNOWLEDGMENT

This work was supported by the European Space Agency (ESA) Ariadna Grant 08/6303 "Neuromorphic computation of optic flow data" and by the EU ICT FP7 Grant ICT-231467-eMorph: "Event-Driven Morphological Computation

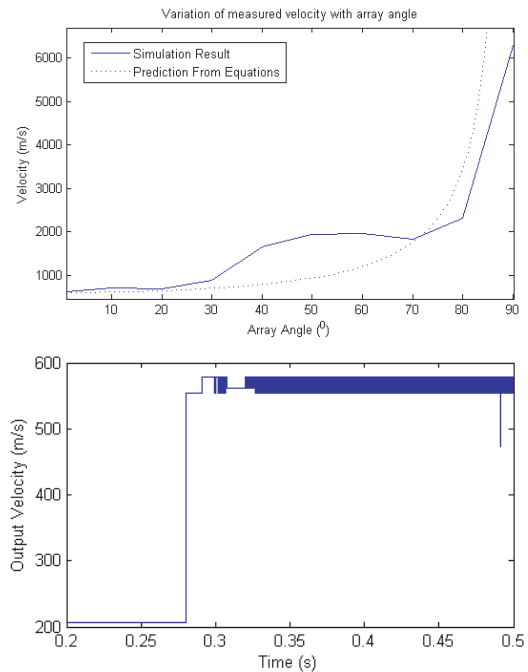


Fig. 4. **Top:** A comparison between the optical flow predicted by equations and simulation results as a function of θ . The actual velocity is 500m/s . For $\theta = 0$ the measurement is accurate as expected. Deviation between the measured and expected result can be explained by the effect of the edge orientation (ϕ) which is assumed to be $\frac{\pi}{2}$ for the prediction curve.

Bottom: An example of a simulation result at a velocity of 500m/s , 2km from the surface. Before 0.28s no pixels have measured optical flow and the minimum output is given. At 0.28s a pixel makes a valid flow measurement and becomes the current pixel of interest. At around 0.29s another pixel makes an even larger measurement of optical flow and becomes the new pixel of interest. The inhibition of return (IOR) circuit block causes the output to jump between these two pixels until a third pixel makes a valid measurement at 0.31s , after which the IOR block causes the output to jump between all three pixels. It is likely that other pixels made valid measurements during this time, but any measurements significantly smaller than these three readings will not be reflected at the sensor output. The distance to the planet surface in simulation is used to determine the actual velocity.

for Embodied Systems". We are grateful to Steven Fry, Vasco Medici, and Tobias Seidl, for the fruitful interactions and discussions on the project topics.

REFERENCES

- [1] C. Koch and B. Mathur, "Neuromorphic vision chips," *IEEE Spectrum*, vol. 33, no. 5, pp. 38–46, May 1996.
- [2] R. Etienne-Cummings and J. Van der Spiegel, "Neuromorphic vision sensors," *Sensors and Actuators*, vol. SNA056, no. 1, pp. 19–29, 1996.
- [3] G. Indiveri and R. Douglas, "ROBOTIC VISION: Neuromorphic vision sensor," *Science*, vol. 288, pp. 1189–1190, May 2000.
- [4] S.-C. Liu, "Silicon retina with adaptive filtering properties," *Analog Integrated Circuits and Signal Processing*, vol. 18, no. 2/3, pp. 243–254, February 1999.
- [5] J. Kramer, R. Sarpeshkar, and C. Koch, "Pulse-based analog VLSI velocity sensors," *IEEE Transactions on Circuits and Systems II*, vol. 44, no. 2, pp. 86–101, February 1997.
- [6] L. Itti and C. Koch, "Computational modeling of visual attention," *Nature Reviews Neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.
- [7] C. Bartolozzi and G. Indiveri, "Selective attention in multi-chip address-event systems," *Sensors*, vol. 9, no. 7, pp. 5076–5098, 2009. [Online]. Available: <http://www.mdpi.com/1424-8220/9/6/5076>
- [8] A. Stocker, *Analog VLSI Circuits for the Perception of Visual Motion*. Wiley & Sons, 2006.