Neuromorphic VLSI & Bat Echolocation

Timothy K. Horiuchi

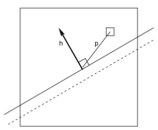
Smart Sensors: Horizon Detection Chip

Unmanned micro-aerial vehicles are rapidly being developed for use as a low-cost, portable, aerial surveillance platform for semi-autonomous operation.

While they are successfully achieving flight, the sensors needed for autonomous flight are lacking.

Obstacle avoidance and the control of basic flight parameters such as altitude, roll, and pitch remain a problem for such small vehicles with tiny weight and power budgets.

Inspired by a visual-horizon estimation algorithm implemented through a closed-loop radio-telemetry video link at the Univ. Florida, we designed a single-chip vision sensor that finds the visual horizon using less than 2.5mW (@ 5V).



Consider an image where each pixel is assigned a horizontal and vertical coordinate (x, y) with the origin in the center of the image. The sign of the dot-product of a horizon vector (***h***) and the pixel vector (***p***) plus a bias parameter determines the horizon line. The dotted line represents the horizon line with a positive bias.

Each pixel then is either in the 'sky' class or the 'ground' class.

$$class(p^{\mu}, h) = \mathrm{sign}((p^{\mu} \cdot h) + b - \theta)$$

In the chip, each pixel calculates its class using a circuit that computes the dot product with its coordinates and the globally distributed horizon vector information.

$$O^{\mu} = g\left(h \cdot p^{\mu} + b - \theta\right) = g\left(\sum_i h_i \cdot p_i^{\mu} + b - \theta\right)$$

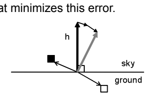
By comparing a pixel's intensity with the average intensity of the two existing classes of pixels, we define the "correct" output class as the one that best matches the pixel's intensity. We can thus define an error for a given choice of ***h*** by the sum of the squares of the number of mismatched pixels.

$$E(h) = \frac{1}{2}\sum_{\mu}\left(\zeta^{\mu} - O^{\mu}\right)^2$$

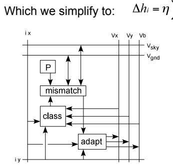
We can now define a gradient-descent learning rule on ***h*** that minimizes this error.

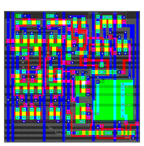
$$\Delta h_i = -\eta \frac{\partial E}{\partial h_i}$$

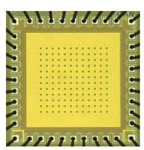
$$= -\eta\left(-\sum_{\mu}\left(\zeta^{\mu} - O^{\mu}\right) g'\left(\sum_i h_i \cdot p_i^{\mu} + b - \theta\right) p_i^{\mu}\right)$$



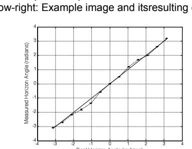
Which we simplify to: $\Delta h_i = \eta \sum_{\mu} \delta^{\mu} p_i^{\mu}$

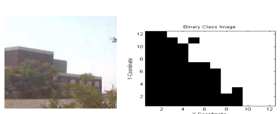




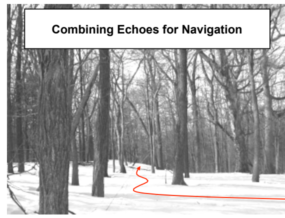


Left: Block diagram of the processing that occurs in a single pixel. *vx*, *vy*, and *vb* represent the x-y components and bias parameter of the horizon vector and each pixel's coordinate (x, y) is represented by currents encoded by the *voltages ix* and *iy*. Vsky and Vgnd represent the average intensity of pixels within the selected sky and ground regions. Middle: VLSI layout masks of a single pixel. Right: Die photograph showing the holes in top metal to expose the photodetectors. Below Left: Estimated roll angle output for a high-contrast test image. Below-right: Example image and itsresulting class image.



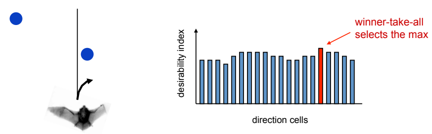


Neural Models: Computation in Neuromorphic VLSI



The problem of fast obstacle avoidance has long been the fascination of roboticists and scientists. As an example, bats and birds can fly quickly through dense forests or complex 3D terrain (e.g., caves and buildings) avoiding dynamic obstacles and hunting prey.

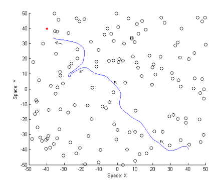
While obstacle field problems have long been tackled using potential fields (attractive or repulsive forces) or occupancy grid search algorithms, these approaches require an explicit representation of the space. In this work, we develop a simple approach where obstacles eliminate directions of travel, but the final choice of direction is based on a "openness" evaluation in combination with desired directions of travel.

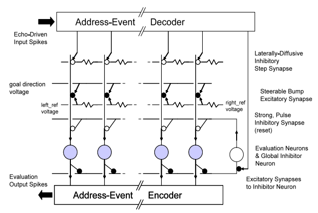


The model involves an array of neurons that represent different directions of travel (or different motor actions). The input to the array is a spatial pattern that represents the relative desirability of different directions given a task. For example, if the bat wishes to fly straight forward (possibly towards a goal), but doesn't mind deviating if there are obstacles, we might provide a pattern that is a DC constant + a shallow, wide Gaussian.

When obstacles are detected, they produce additive Gaussian-shaped suppressions that are wider and deeper for closer obstacles. A winner-take-all mechanism (i.e., max() function) selects the winning direction (or motor action). We can thus think of a field of neurons that are firing in a fashion to represent "open space".

This simple algorithm works well and under a wide range of parameters.

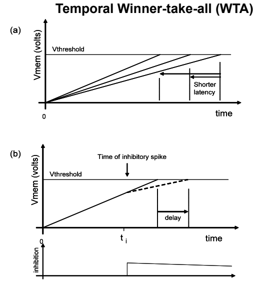




We have implemented the algorithm in a neuromorphic VLSI chip using spiking neurons and a winner-take-all inhibitory network. While using spiking neurons with the mean firing rate representing the activation of different directions works well, we can also operate the neurons in a mode where the winner-take-all is based on spike latency.

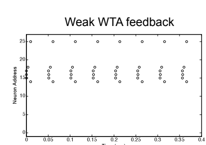
In this race to first spike, neurons with larger input currents reach their threshold voltage first. By using a strong feedback inhibition, only the winning (or competitive neurons) can be allowed to fire spikes.

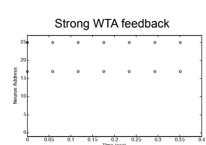
Temporal Winner-take-all (WTA)



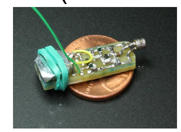
Left: the membrane voltage of a neuron charges faster with larger input currents. We can use this latency to implement WTA. If an echo (obstacle) is detected, we can inject a weak step function that starts at the time of the echo. This delays the spike, effectively inhibiting the neuron.

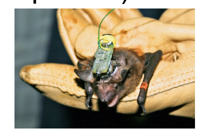
Right: Example of repeated cycles of competition for a Gaussian shaped activation current for weak and strong inhibitory feedback.





Instrumentation for Behavioral Studies: Ultrasonic Microphone Telemetry (Moss and Krishnaprasad)



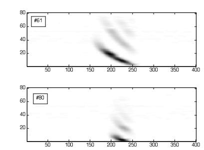


In collaboration with ISR members P. S. Krishnaprasad and Cynthia Moss, I have been developing a radio telemetry device for recording the ultrasonic vocalization of the flying bat. Mounted on the head, this microphone can provide insight into how the intensity, bandwidth, duration changes in response to the task and the environment, even as it turns its head in 3D.

In particular, we have been interested in determining if there is evidence for "switching between controllers" as the bat transitions from one part of a task to another.

In this work, we have compared vocalizations by their spectrograms. First, we extracted the power spectrogram P(f, t) (frequency vs. time) which results in a 2D image.

Power Spectrum $P(f,t)$



The spectrogram is L^2 normalized so that overall gain is discounted.

$$P'(f,t) = \frac{P(f,t)}{\sum_{f,t} P(f,t)}$$

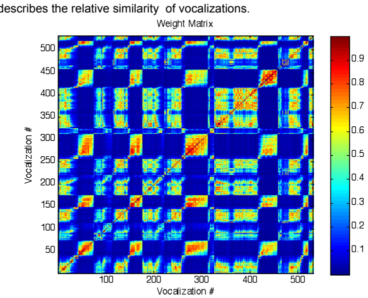
Then the vocalizations are pairwise compared (dot-product), resulting in a large correlation matrix.

$$C_{ij} = \sum_f \sum_t \sqrt{P'_i(f,t)} \cdot \sqrt{P'_j(f,t)}$$

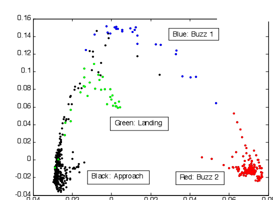
We further compute the geodesic $Dist_{ij} = \arccos\left(C_{ij}\right)$

And create a "weight matrix" that describes the relative similarity of vocalizations.

$$W_{ij} = e^{-\frac{Dist_{ij}^2}{L^2}}$$



From the weight matrix, we can see that sequential vocalizations are often very similar (square-like regions) and that there are sharp transitions between vocalization types.



Using the weight matrix, we performed the Laplacian Eigenmaps technique (Belkin and Niyogi, 2003) to find a low-dimensional projection that might expose distinct clusters of vocalizations as are done with human speech.

1. Create the weight matrix $\boldsymbol{W_{ij}}$
2. Create the Laplacian matrix $L = D - W$, where D is a diagonal matrix created from the column sums of W.
3. Solve the generalized eigenvalue problem
4. Use the 2 eigenvectors (f_1, f_2) with the lowest non-zero eigenvalues (2-D).
5. plot (f_1, f_2,'k.')

This work was supported by grants from the Air Force Office of Scientific Research (AFOSR: FA95500410130, FA95500710446), the National Institutes of Health (NIH: R01 MH56366; NIH/NIBIB (CRCNS 1R01EB004750), and the National Science Foundation (NSF: CAREER: CCF0347573)