2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS) | 978-1-6654-1913-0/20/$31.00 ©2021 IEEE | DOI: 10.1109/AICAS51828.2021.9458450

A Bio-Inspired Motion Detection Circuit Model for the Computation of Optical Flow: The Spatial-Temporal Filtering Reichardt Model

Hsin-Yu Wu[1#], Wei-Tse Kao[2#], Harrison Hao-Yu Ku[2], Cheng-Te Wang[3], Chih-Cheng Hsieh[2], Ren-Shuo Liu[2], Kea-Tiong Tang[2], Chung-Chuan Lo[3,4*]*

*#: contributed equally. *: corresponding authors*

[1] Department of Physics, National Tsing Hua University, Hsinchu 30013, Taiwan

[2] Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan

[3] Institute of Systems Neuroscience, National Tsing Hua University, Hsinchu 30013, Taiwan

[4] Brain Research Center, National Tsing Hua University, Hsinchu 30013, Taiwan

Contact information: cclo@mx.nthu.edu.tw, cheng-te@lolab-nthu.org

Abstract—***Optical flow is the pattern of apparent motion in a visual scene produced by the relative movement between objects and an observer. Optical flow is used in many engineering applications such as optical odometry. A variety of optical-flow algorithms has been proposed in the past few decades; however, most of these algorithms involve complex computation, making them difficult to be implemented in neuromorphic systems that operate based on neural networks. Interestingly, studies have shown that insect visual systems are able to perform complex optical flow algorithms. Inspired by the classic Reichardt motion detection model proposed for insects, we designed a spatial-temporal filtering Reichardt (STR) model. This model computes optical flow based on simple filters in the spatial and temporal domains. The STR model is hardware friendly: it does not require time-consuming iteration processes nor computationally intensive multi-layer convolutional networks, which are typical in other optical flow algorithms. We systematically investigate the performance of the STR model with different parameters including: object size, speed, luminance, and filter forms. We also compare the performance of the STR model to the classical Farneback algorithm, and we demonstrate that the STR model is comparable to the classical algorithms while requiring much less computational power.***

Keywords—Optical flow, Drosophila, motion detection, bio-inspiration

I. Introduction

Motion detection of image is an important topic in computer vision. In the traditional approach, a number of CPU-based algorithms such as Lucak-Kanade, Farnebäck, Horn-Schunck and DIS [1]–[4] were proposed and they can estimate motion vector (optical flow) of each pixel from multi frames. The fundamental assumptions of most optical flow algorithms are (1) the intensity of pixel is a function of spatial and time, (2) the displacement of patterns between images are small and smooth, and (3) the luminance of scenario is constant. Therefore, the function can be expanded by Taylor series and the gradience of the function of pixel intensity is a moving vector (optical flow). The Lucak-Kanade algorithm compute the optical flow based on the partial derivatives of patches and the algorithm needs to solve the inverse matrix during the process. The Farnebäck's method, on the other hand, fits the displacement of a local patch by quadratic polynomial and uses a symmetric matrix and a coefficient vector to describe the movement. More recent machine learning approach, such as Flownet, PWC-net, and RAFT [5]–[7], utilizes artificial neural networks to learn to calculate the optical flow.

The traditional approach requires complex computation such as curve fitting or matrix inversion that are difficult to be implemented in neural networks, while the machine learning approach requires a large number of neuron layers and is also computationally intensive. Taking the Flownet for example, it uses 6 convolution layers and 6 transpose convolution layers to estimate the optical flow. Therefore, both approaches are not suitable for low-powered computation on edge devices such as small drones. To address this issue, we design a hardware friendly optical flow algorithm that is inspired by the insect visual system.

In the past few decades, the motion vision of insects has been extensively studied and several models have been proposed to explain the underlying neural mechanisms. Among these models, the Hassenstein-Reichardt (HR) model [8] is the one mostly discussed. The HR model detects motion direction of objects by implementing asymmetric latencies between two photo receptors (Figure 1A). Although the HR model is simple, it requires a huge number of elementary circuits: four (up → down, down → up, left → right and right → left) for each pixel.

There are also a number of biologically realistic models proposed base on the HR model. For example, the Borst 2018 model [9] is developed based on extensive experimental data and is able to reproduce many empirical observations. However, this model is more complex than the classic HR model (Figure 1B) because it uses three processing units for each pixel and each direction, and a total of 12 units are required to cover all four directions for a given pixel.

Therefore, in order to design an optical flow circuit model that has a decent performance while is sufficiently simple for implementation in custom-designed chips, we decide to develop a circuit model based on the classic HR model [9]–[11].

Our model extends the HR model by implementing a layer of spatial filters and a layer of temporal filters, and is therefore termed spatial-temporal filtering Reichardt model (STR model). The STR model is purely feedforward, computationally simple (no curve fitting and no matrix

978-1-6654-1913-0/21/$31.00 ©2021 IEEE

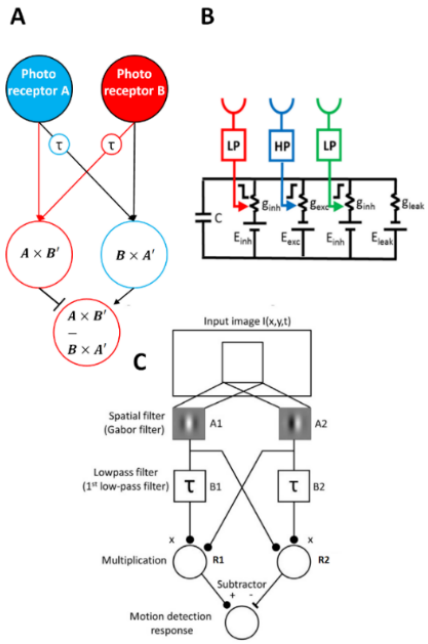

Figure1. Biophysics models of the insect motion detector. A: The classical Hassenstein-Reichardt (HR) model. B: The Borst 2018 [9] model. C: Our spatial-temporal filtered Reichardt (STR) model.

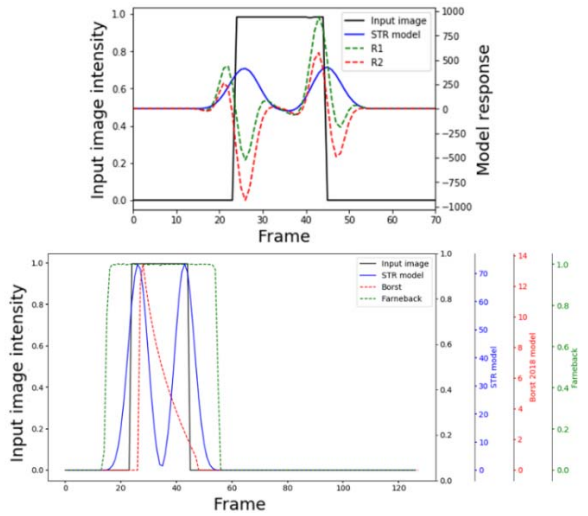

Figure 2. The sensitivity of the STR model to moving edges. Top: the responses of the R1 (A2B1, green) and R2 (A1B2, red) components and the output of the model (blue) to the stimulus produced by a moving square (black) in the preferred direction. The model generates a strong response to each event of edge crossing. The square has a size of 20x20 pixels and the moving speed is 1 pixel/ frame. Bottom: the responses of the STR model (blue), Borst 2018 model (red) and the Farneback algorithm (green) to the same stimulus (black) as in the top panel. The STR model is highly sensitive to moving edges.

inversion) and requires much fewer neurons and elementary circuits. Therefore, although the total number of operations per pair of frames in the STR model is comparable to the Lucak-Kanade method, the nature of feedforward computation makes the STR model easier to be implemented in hardware.

II. The STR model

Each elementary circuit of the STR model (Figure 1C) processes a small patch (24x24 typically) of an input image. The circuit computes motion in the window with four compute layers. (1) The patch is first convolved by two different spatial filters (A1 and A2), which generate two channels of signals. (2) Each channel passes through a temporal filter which produces a latency (B1 and B2). (3) The delayed signal is then combined with un-delayed (un-filtered) signal from the other channel through multiplication (R_1 and R_2). (4) The model output is generated by computing the difference between the signals from the two channels and normalizing the difference by a factor H, which is a function of the average input intensity. The purpose of the normalization is to produce an output that is independent of the intensity of the object. The determination of the factor H is discussed in the next section. The four-layer of process can be described by the following four sets of equations:

(1) $g_n(x,y;\lambda,\theta,\psi_n,\sigma,\gamma) = e^{-\frac{x^2+\gamma^2y^2}{2\sigma^2}} \cos\left(2\pi\frac{x}{\lambda}+\psi_n\right)$

$A_n(x,y,t) = I(x,y,t) * g_n(x,y)$

(2) $B_n(t) = (1-\alpha)B_n(t-1) + \alpha A_n(t)$

(3) $R_1 = A_2B_1,$

$R_2 = A_1B_2$

(4) $output = (R_1 - R_2)/H$

The temporal filter in the second layer passes signals below a certain frequency range with a latency. The latency can be adjusted by changing the parameter α and the system only needs to store the signal in the last timestep. There is no need for storing long arrays of previous signals for large latency. Therefore, the system is hardware friendly. By combining the delayed signal from one channel and un-delayed signal from another channel through multiplication, the two channels (R_1 and R_2) generate responses to the intensity changes caused by moving objects (Figure 2 top).

The difference between the two channels arises from the phase difference in the Gabor filters which detect edges or textures at the different locations. Finally, R_2 is subtracted from R_1 and the circuit produces a positive responses for both edge-passing events (on→off and off→on) when an object moves in the preferred direction (Figure 2 top).

To understand the characteristics of the STR model, we compare it with the widely used Farneback algorithm and the biologically realistic Borst 2018 model (Figure 2 bottom). The three models respond to the stimulus produced by a moving squire differently. The Borst 2018 model generates a sharp peak to the onset of the stimulus (the first edge) but not the offset of the stimulus (the second edge), while the Farneback algorithm responds to the entire square during its passing. The STR model, in contrast, responds mainly to the edge-passing events for both onset and offset. Therefore, the STR model is highly sensitive to edges in the images.

A true motion detector should detect the direction of movement rather than the direction of intensity change. We test this by showing the STR model with four stimulus conditions: (1) a white square moving rightward in a black background, (2) same as in (1) but moving leftward (Figure 3 top), (3) a black square moving rightward in a white background and (4) same as in (3) but moving leftward (Figure 3 bottom). The STR model always produces a positive response to the rightward movement regardless whether the object is white or black. Interestingly, the

STR model produces a negative response with the same magnitude if the object moves in the opposite direction. This is one of the important advantages of the STR model over other biophysical models because most of those models only respond to the preferred direction but not the opposite direction. Therefore, two sets of the elementary circuits are required if two opposite directions need to be detected, while only one circuit is required in our STR model.

III. ANALYSIS AND TEST

To compute the optical flow, a motion detector should be able to indicate not only the direction, but also the speed of the motion, and should be insensitive to the intensity of the objects. We analyze the peak response of the STR model to a moving square of various speeds. The raw response ($R = R_1 - R_2$) of the model is nearly linear to the velocity of the object but the absolute magnitude depends on the λ of the Gabor filter, time constant α of the low pass filter and intensity of the object (Figure 4). A smaller α produces a steeper R versus velocity curve but with a smaller linearly region (Figure 4 top-left). The raw response R is also modulated by λ in a similar fashion but with a larger λ producing a steeper curve. Therefore, we can obtain a desired R versus velocity relationship by carefully choosing the values of α and λ. To produce the true optical flow which is independent of the image intensity, we normalize the raw response R based on the average input intensity as indicated in Equation (4). To determine the actual form of the normalization factor H, we observe the relation between the raw response of the model, the average input intensity I read off from the Gabor filters (A_1 and A_2) and the true speed v of the object. We found that the relation can be described by:

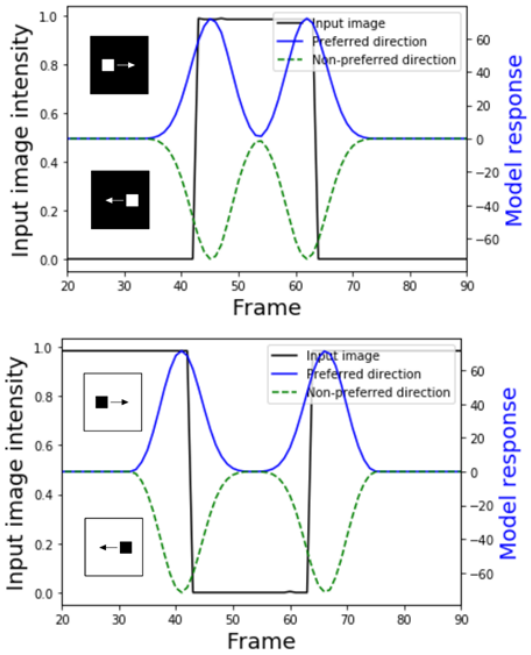

Figure 3. the response of the STR model with different stimulus conditions. Top: White square moving in the black background. Bottom: Black square moving in the white background. Black curves: image intensity at the center of the window. Blue curves: the response of the STR model when object moves in the preferred direction. Green curves: the response of the STR model when object moves in opposite to the preferred direction.

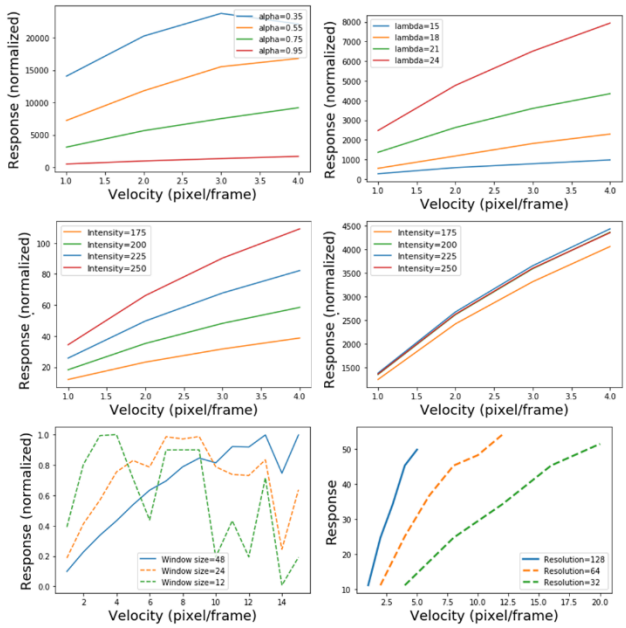

Figure 4. Response of the STR model as a function of the object speed with different stimulus or model parameters. Top-left: the raw response curve with different time constant α. Top-right: the raw response curve with different λ of the Gabor filter. Middle-left: the raw response cruve with different speeds and intensities. Middle-right: same as in the top-left panel but with normalized response, which is linearly correlated with the object speed and is insensitive to the intensity. Bottom-left: the normalized response with different window sizes. Bottom-right: the normalized response with different image resolutions. The x-axis indicates the velocity in the original images (128 x 128), not in the rescale resolutions.

$$R = f(v)I^k \tag{5}$$

We find that $f(v)$ is a simple linear function and k=6. Therefore, the normalization factor is determined to be I^6. By dividing the raw response by H, we recover the true speed of the object and the normalized response of the STR model is independent of the intensity of the object (Figure 4 middle-right). A drawback of the model is that its linearity only covers the speed range between 1.0 and 4.0 pixel/frame. A simple solution to increase the speed range is to use different window sizes. We discover that a small window gives rise to a sensitive linear response to small speed, as indicated by a steep curve in the small speed range (Figure 4 bottom-left, green curve). A larger window can produce a linear response covering a larger speed range, but the response curve is relatively shallower (Figure 4 bottom-left, orange and blue curves). Based on the observation, we can combine two circuits with two different window size to cover a larger speed range while still maintaining the sensitivity to slow objects.

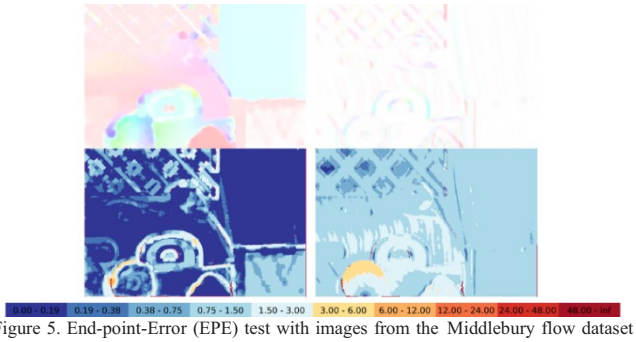

Figure 5. End-point-Error (EPE) test with images from the Middlebury flow dataset [12]. Top-Left: the optical flow from the Farneback's algorithm. The color encodes the direction of the flow. Top-Right: the optical flow from the STR model. Bottom-Left: EPE of the optical flow from the Farneback algorithm. The color encodes the magnitude of the error. The flow error, defined as the percentage of the pixels with EPE > 3.0 is 2.37%. Bottom-Right: EPE of the optical flow from the STR model. The flow error is 3.26%.

However, constructing two motion detector arrays with two different window sizes means doubling the computational load of the system because the total number of operations is proportional to the pixel size of the image, not the window size. To reduce the computational load, we can use the same window size but with reduced resolution, creating an effective window with a large size. We test this idea with three different image resolutions (128x128, 64x64 and 32x32) and confirm that for the same window size (24x24 pixels), a large image resolution leads to a sensitive response to slow movement in a narrow speed range (Figure 4 bottom-right, blue) while a small image resolution leads to a response to the speed in a broader range (Figure 4 bottom-right, green).

Finally, we test the STR model using the images from the Middlebury flow dataset [12] with the evaluation method proposed in KITTI [13]. We also perform the same test for the Farneback algorithm for comparison (Figure 5). The STR model produces a comparable flow error to the Farneback algorithm at the small-speed range, while producing a slightly larger errors at the large-speed range due to the narrower dynamical range of the STR model. The overall flow error is 2.37% for the Farneback algorithm and 3.26% for the STR model.

IV. Discussion

We demonstrate that the proposed STR model produce optical flow with decent performance. Most importantly, the STR model is computationally simple and only requires convolution, low-pass filtering, multiplication and subtraction in a feedforward fashion. These operations are relatively easy to be implemented in custom-designed neuromorphic chips.

As the first version of the STR model, it still has room to be improved. Three approaches will be taken in the follow-up study. First, we will systematically test the effect of each model parameter including the type of spatial filter on the performance. This will provide us a picture on how optimal parameters depend on the stimulus properties such as object size, speed and texture. Second, we will implement a pyramidal approach by using 2-3 levels of STR detectors and each level process the same image with different resolution. This will greatly improve the dynamical range of the model. Third, we will design a learning algorithm so that the STR model can dynamically optimize itself based on the momentary stimulus properties in a changing environment.

[1] G. Farnebäck, "Two-Frame Motion Estimation Based on Polynomial Expansion," in *Image Analysis*, Berlin, Heidelberg, 2003, pp. 363–370, doi: 10.1007/3-540-45103-X_50.

[2] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *undefined*, 1981. /paper/An-Iterative-Image-Registration-Technique-with-an-Lucas-Kanade/a06547951c97b2a32f23a6c2b5f79c8c75c9b9bd (accessed Dec. 04, 2020).

[3] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, no. 1, pp. 185–203, Aug. 1981, doi: 10.1016/0004-3702(81)90024-2.

[4] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool, "Fast Optical Flow Using Dense Inverse Search," in *Computer Vision – ECCV 2016*, Cham, 2016, pp. 471–488, doi: 10.1007/978-3-319-46493-0_29.

[5] Z. Teed and J. Deng, "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow," *ArXiv200312039 Cs*, Aug. 2020, Accessed: Jan. 07, 2021. [Online]. Available: http://arxiv.org/abs/2003.12039.

[6] P. Fischer *et al.*, "FlowNet: Learning Optical Flow with Convolutional Networks," *ArXiv150406852 Cs*, May 2015, Accessed: Jan. 22, 2021. [Online]. Available: http://arxiv.org/abs/1504.06852.

[7] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume," *ArXiv170902371 Cs*, Jun. 2018, Accessed: Jan. 22, 2021. [Online]. Available: http://arxiv.org/abs/1709.02371.

[8] B. Hassenstein and W. Reichardt, "Systemtheoretische Analyse der Zeit-, Reihenfolgen- und Vorzeichenauswertung bei der Bewegungsperzeption des Rüsselkäfers Chlorophanus," *Z. Für Naturforschung B*, vol. 11, no. 9–10, pp. 513–524, Oct. 1956, doi: 10.1515/znb-1956-9-1004.

[9] A. Borst, "A biophysical mechanism for preferred direction enhancement in fly motion vision," *PLOS Comput. Biol.*, vol. 14, no. 6, p. e1006240, Jun. 2018, doi: 10.1371/journal.pcbi.1006240.

[10] A. Borst, "Drosophila's View on Insect Vision," *Curr. Biol.*, vol. 19, no. 1, pp. R36–R47, Jan. 2009, doi: 10.1016/j.cub.2008.11.001.

[11] A. Borst, "Fly Vision: Moving Into the Motion Detection Circuit," *Curr. Biol.*, vol. 21, no. 24, pp. R990–R992, Dec. 2011, doi: 10.1016/j.cub.2011.10.045.

[12] S. Baker, S. Roth, D. Scharstein, M. J. Black, J. P. Lewis, and R. Szeliski, "A Database and Evaluation Methodology for Optical Flow," in *2007 IEEE 11th International Conference on Computer Vision*, Oct. 2007, pp. 1–8, doi: 10.1109/ICCV.2007.4408903.

[13] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 3061–3070, doi: 10.1109/CVPR.2015.7298925.