Interacting Maps for Fast Visual Interpretation

Matthew Cook, Luca Gugelmann, Florian Jug, Christoph Krautz, and Angelika Steger

Abstract— **Biological systems process visual input using a distributed representation, with different areas encoding different aspects of the visual interpretation. While current engineering habits tempt us to think of this processing in terms of a pipelined sequence of filters and other feed-forward processing stages, cortical anatomy suggests quite a different architecture, using strong recurrent connectivity between visual areas. Here we design a network to interpret input from a neuromorphic sensor by means of recurrently interconnected areas, each of which encodes a different aspect of the visual interpretation, such as light intensity or optic flow. As each area of the network tries to be consistent with the information in neighboring areas, the visual interpretation converges towards global mutual consistency. Rather than applying input in a traditional feed-forward manner, the sensory input is only used to weakly influence the information flowing both ways through the middle of the network. Even with this seemingly weak use of input, this network of interacting maps is able to maintain its interpretation of the visual scene in real time, proving the viability of this interacting map approach to computation.**

I. Introduction

One of the key differences between biological and engineered visual systems is that engineered solutions traditionally use a feed-forward sequence of processing stages and filters [1][2], whereas biology (e.g. in primates) uses strong recurrent connectivity between different brain areas which process different types of information in parallel [3][4]. Understanding this biological style of visual processing could help with the long term technological goal of matching or surpassing the visual capabilities of biological systems.

Some key architectural properties that currently are largely unique to biological vision systems include the strong recurrent connectivity between cortical areas [4], the ability of seemingly weak input to dominate the activity of a multi-area system [6], and the ability of a distributed representation to arrive at a coherent interpretation of weak or noisy input [7].

As a first step towards a better understanding of such an approach to computation, we designed a system having these properties, which analyzes visual input with a network of recurrently connected areas. Each area represents a different aspect of the visual interpretation, such as light intensity or optic flow, as shown in Fig. 1. The input is not fed into one end of the network for serial processing, rather, it merely modulates information flowing within the network. There is also no designated output. Each area converges to represent its aspect of the distributed interpretation and could be used as an output if desired. This is similar to the brain's approach to cognitive operation, where many different aspects of an interpreted scene are available for conscious reasoning [7].

The parts of our network are connected by mild constraints, none strong enough to single-handedly restrict the solution significantly. The flow of information in the network occurs through these constraints, each of which influences the areas it relates, gently pushing each area towards satisfaction of the constraint. Given this structure of the network, it is only through the recurrent feedback of information reverberating throughout the entire network that a mutually consistent state is reached. This state then provides a coherent multi-faceted visual interpretation.

The Structure of the Network

Since each area is internally retinotopically organized, we refer to the areas as *maps*. The maps in our network are not intended to correspond to specific brain areas of any particular species, rather, we designed them to form a small but coherent vision system composed of clearly understandable parts and relationships. The network contains an optic flow map $\mathbf{F}$, a light intensity map $\mathbf{I}$, a spatial intensity gradient map $\mathbf{G}$, a temporal intensity derivative map $\mathbf{V}$, a camera calibration map $\mathbf{C}$, and a single (non-mapped) estimate of the three-dimensional rotation $\mathbf{R}$ of the camera, which we assume to be fixed at a single point but free to rotate, like an eye in its socket. If we wanted to handle 3D translation in addition to 3D rotation, a larger network would need to be used, as discussed below.

The relationships between these maps, as shown mathematically in the rectangles in Fig. 1, are (i) that $\mathbf{G}$ should be the gradient of $\mathbf{I}$, (ii) that spatial variation $\mathbf{G}$ in brightness should match time variation $\mathbf{V}$ according to the local optic flow $\mathbf{F}$, and (iii) the optic flow $\mathbf{F}$ at each point should correspond to the camera motion $\mathbf{R}$ with respect to the direction $\mathbf{C}$ that the pixel is aimed in. Together, these relations simply express the idea that the input should be explainable in terms of some kind of camera rotation in front of some kind of image. As will be discussed in more detail in Section II, these constraints come from (i) the definition of $\mathbf{G}$ and $\mathbf{I}$, (ii) the optical flow constraint equation [8] defining the relationship of $\mathbf{V} = d\mathbf{I}/dt$, $\mathbf{F} = d\vec{x}/dt$, and $\mathbf{G} = d\mathbf{I}/d\vec{x}$, and (iii) the three-dimensional geometry. That is, the constraints are given simply by the mathematics of the meanings of the maps, not by any special requirements of this type of network.

These relations connect the maps in a point-wise way. In fact, the only place where neighboring pixels are connected at all is through the relation between $\mathbf{G}$ and $\mathbf{I}$, in which $\mathbf{G}$ depends on the relative values of neighboring pixels in $\mathbf{I}$, and

The authors are listed in alphabetical order. Matthew Cook is with the Institute of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland (cook@ini.phys.ethz.ch). Luca Gugelmann, Florian Jug, Christoph Krautz, and Angelika Steger are with the Institute of Theoretical Computer Science, ETH Zurich, Switzerland (email: {lgugelmann,fjug,ckrautz,asteger}@inf.ethz.ch).

This work was supported by ETH Research Grant ETH-23 08-1 and EU Project Grant FET-IP-216593.

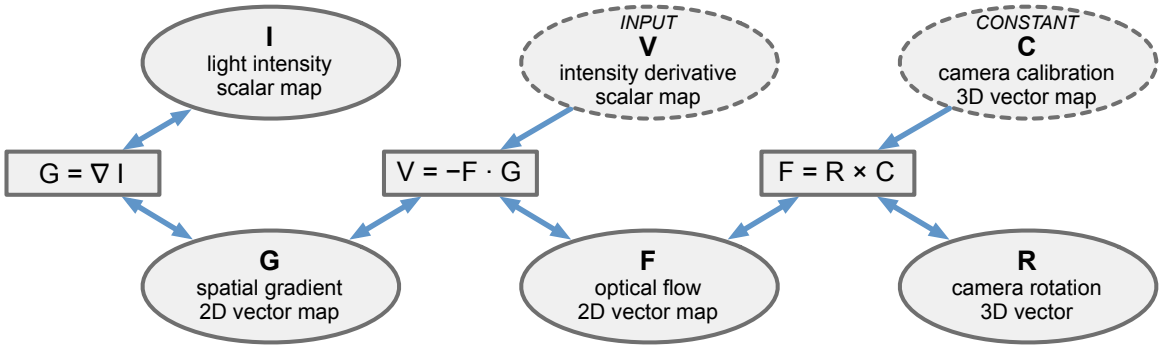


Fig. 1. The network architecture. The relationships, shown in rectangles, apply independently for each pixel in the maps. See section I for descriptions of the maps and relationships, and section II for a more detailed description of the network and its operation.

through the relation between **F** and **R**, in which all pixels (not just neighbors) must coordinate their local optical flow estimate with the one single estimate of the camera's rotation contained in **R**.

The input to the network is given by a neuromorphic vision sensor [9] which reports the temporal derivative of the light intensity at each pixel with 'on' and 'off' spike events. We bin these spike events during a short time window to generate the input for map **V**. This is the sole input to the system. Note that the map **I**, corresponding to a traditional grayscale camera image, is not an input to our system, but must be inferred from the input **V** by simultaneously inferring the rest of the network. To our knowledge there is no feed-forward method for finding **I**, **G**, **F**, and **R** given **V**. To get a feel for the problem, the reader can try to infer **I**, **G**, and **F**, given only **V** as shown in figure 5. Attempting this inference will make it clear that a solution necessitates the interpretation of **V** as the result of a particular camera motion in front of a particular image.

Given the input **V**, it is not possible to simply solve for **F** and **G**. The only constraint on the vectors **F** and **G** at any pixel is that their scalar product should be -**V**. This is a very weak constraint, eliminating only one out of the four degrees of freedom in **F** and **G** at each pixel. Even if either **F** or **G** is known, the other is still underconstrained, being limited only to a line of possibilities. When trying to solve for **F**, this is known as the *aperture problem* [10].

The other constraints, namely that the optic flow **F** at each pixel should be consistent with some overall camera rotation **R**, and that **G** should be the gradient of some map **I** (i.e., a conservative vector field), clearly do not constrain **R** or **I** at all. Thus all of the constraints in the system are weak constraints, and it is a priori not clear that the system will be able to find a correct interpretation of the input.

For our purposes, this weak effect of the input is an advantage. Although we could use regular video input, thus avoiding the need to infer **I**, our purpose here is to explore the viability of a system with a distributed internal representation where the parts of the representation are in general not directly observable.

This small visual system is simple enough that it is easy to implement and verify for correctness, yet complex enough to solve non-trivial problems such as inferring the grayscale image or the optical flow. Of course if the goal were to determine the grayscale image, optical flow, or ego-motion, there are specialized methods and sensors for each of these. Our goal here is to see whether an interconnected distributed representation can reach reasonable conclusions in each of its areas, even when the input is too weak to infer any area individually.

As described in section III, our network in practice converges to a consistent visual interpretation. It can actually do so using just a single input frame, even if all maps are randomly initialized. Note that a single frame from a traditional camera is insufficient for determining optical flow, as it contains no temporal information. The neuromorphic sensor we use, inspired by features of biological retinas [11], reports the time derivative of the light intensity at each pixel rather than the light intensity itself. A single frame of this information surprisingly turns out to be enough to infer both the standard grayscale image and the optical flow.

Extendability

Our approach is in principle not limited to the specific network shown in Fig. 1. The network could be extended, for example, to allow camera translation and infer three-dimensional depth information ('structure from motion'). To do this, one would express the total optical flow **F** as a sum of an overall image shift **S** due to camera rotation **R**, and a much more subtle contribution of perspective **P** due to camera translation among objects in three-dimensional space, in which nearby objects appear to move faster than far objects. This map **P** would be related to a distance map **D**, along with a global vector for the camera's translation **T**. In general, any type of information that could help with the analysis of the visual scene could be added to the network, connected by its relationships to the other types of information in the network.

Relationship to factor graphs

The overall architecture of our network is quite similar in spirit to the design of a factor graph [12][13]. The structure shown in Fig. 1 not only expresses relationships between different aspects of the interpretation, but also provides conditional independence information just like a factor graph diagram with the maps (ovals) corresponding to sets of variables, and the relations (rectangles) corresponding to sets of factors. For example, the light intensity **I** and the optical flow **F** are conditionally independent given the spatial gradient **G**.

This similarity can be made precise. The network diagram shown in Fig. 1 can be viewed as a simplified version of a very large loopy factor graph. Each map represents a grid of random variables where the variables take on either one-dimensional (**I**, **V**), two-dimensional (**G**, **F**), or three-dimensional (**C**, **R**) values.

Given our camera resolution of 128×128 pixels, the factor graph corresponding to our network would contain over 60,000 random variables, each having a one, two, or three-dimensional probability distribution. Since factor graph algorithms such as loopy belief propagation are quite slow on such large structures, we do not keep track of distribution estimates in our simulations, but rather simply maintain a current best guess for the value of each variable.

Optical flow

The traditional problem of reconstructing optic flow from visual input by using the brightness constancy assumption was originally proposed by Horn and Schunck [8], and has been studied quite heavily since then. Most of the work on this problem (see [14] for a survey) utilizes traditional feedforward processing and relaxation methods [15]. One approach to optical flow that does bear some resemblance to our network is that of Ringbauer et al. [5], who have shown that it is possible to use a multi-map network to compute optical flow based on what is known about the dorsal pathway of primate visual cortex. Their network uses a combination of feed-forward processing and recurrence, and all the maps in the system represent optical flow of some sort, providing a nice example of map-based processing. In contrast with their network, ours does not directly compute the flow using Reichard-like schemes. Instead, similar to traditional relaxation methods [15], our network relies on recurrent interactions between local constraints located throughout the network, which co-influence information as it is transformed between maps, and optical flow is just one of several different aspects we infer about the visual scene.

It is worth reiterating that our network was *not* designed for the purpose of extracting optical flow. Optical flow is just one of several simultaneously inferred quantities. The grayscale image, for example, cannot be inferred without also inferring the optical flow. The purpose of our system is to show that a seemingly weak input can be used in a distributed interconnected system to infer maps which cannot be calculated in a simple feed-forward way.

II. How the relations update the maps

The world we model consists of a still scene which is recorded by a camera provided with a neuromorphic vision sensor [9]. The pinhole of the camera is centered at the origin of the world's coordinate system and its view axis is aligned with the world's z-axis (see Fig. 2). For the purpose of the formulas, we treat camera rotation via the equivalent assumption that the camera is fixed while the world rotates around it, and we let **R** be the rotation vector for the world, which we treat as being painted on the interior of a rotatable unit sphere centered on the camera. The resolution of the camera we used is 128×128. We thus discretized all maps as arrays of size 128×128, except for **I** which has size 129×129 (so that there are 128 differences in the x and y directions to match **G**, cf. Equation 2 below) and **R** which is a single three-dimensional vector.

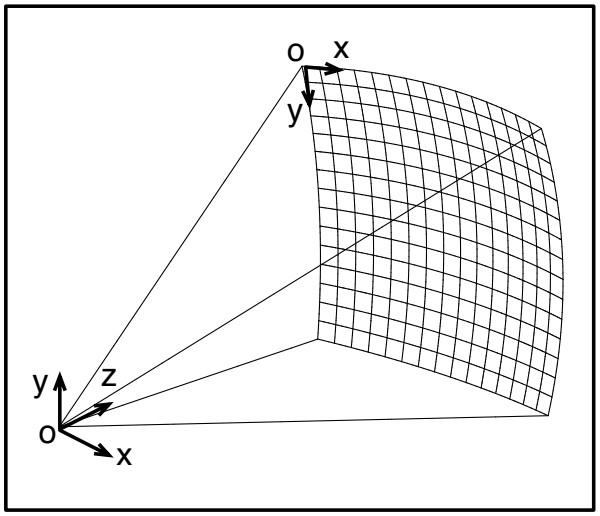


Fig. 2. The relationship between the 2-D pixel coordinates and 3-D space.

There are three relations in this network. The relation between **V**, **F**, and **G**, which is often called the *optical flow constraint equation* [8], is defined as

$$-\mathbf{V}_{x,y} = \vec{\mathbf{F}}_{x,y} \cdot \vec{\mathbf{G}}_{x,y} \tag{1}$$

and is derived from the observation that the change in brightness over time is given by the speed of the optic flow times the change in brightness in the direction of the optic flow. As shown in Fig. 3, this change in brightness depends on the angle between the flow vector and the spatial gradient.

Since our input is **V**, there is an unavoidable scalar ambiguity in the values reached by the rest of the network: For any non-zero real value β, it is possible to multiply all values in **G** and **I** by β, and to multiply all values in **F** and **R** by $1/\beta$, and they will be consistent with exactly the same **V** as before. In other words, given a frame of input, it is impossible to distinguish between a slow-moving high-contrast image and a fast-moving low-contrast image. This ambiguity extends to the sign of β: The image might have inverted contrast and be moving in the opposite direction.

In practice, our system resolves this ambiguity arbitrarily in the first frame (influenced by the scale of the random numbers used to initialize the maps), and then by initializing the maps with the converged state from the previous frame, the relative weight of **F** and **G** (i.e., β) stays roughly the same, leading to consistent interpretations over time.

The relation between maps **I** and **G** requires that

$$\vec{\mathbf{G}}_{x,y} = \nabla \mathbf{I}_{x,y} = \begin{pmatrix} \mathbf{I}_{x+1,y} - \mathbf{I}_{x,y} \\ \mathbf{I}_{x,y+1} - \mathbf{I}_{x,y} \end{pmatrix} \tag{2}$$

holds, where $\nabla \mathbf{I}_{x,y}$ is simply the forward difference approximation of the gradient.

The remaining relation is between the optic flow **F** and the global rotation **R** of the world. It is defined as

$$\vec{\mathbf{F}}_{x,y} = m_{32}(\vec{\mathbf{R}}_{x,y} \times \vec{\mathbf{C}}_{x,y}), \tag{3}$$

where $m_{32} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ maps a vector from world space to image space [16][17]. The relation follows from the observation that in world space the optic flow of a pixel at (x, y) is always perpendicular to the vector of rotation **R** and tangential to the surface of the unit sphere, i.e., perpendicular to the vector $\vec{\mathbf{C}}_{x,y}$ which points in the direction of pixel (x, y) on the surface of the unit sphere. Note that $\vec{\mathbf{C}}_{x,y}$ is not changed by the dynamics of the network but derived by calibration of the view angle of the camera before the experiments, although there is no reason in principle why the network could not slowly update it and thus be self-calibrating.

Based on these relations we derive our update rules following a simple pattern. To update a given map based on a given relation, we assume that the other maps in that relation contain correct information, and we compute a *candidate* map, which is the map closest to the current map that would satisfy the relation. We then compute the new values for the map by taking a small step towards the candidate map. This essentially amounts to a relaxation method [15], applied to specific local constraints rather than to a boundary-constrained differential equation.

The relation between **V**, **F**, *and* **G**.

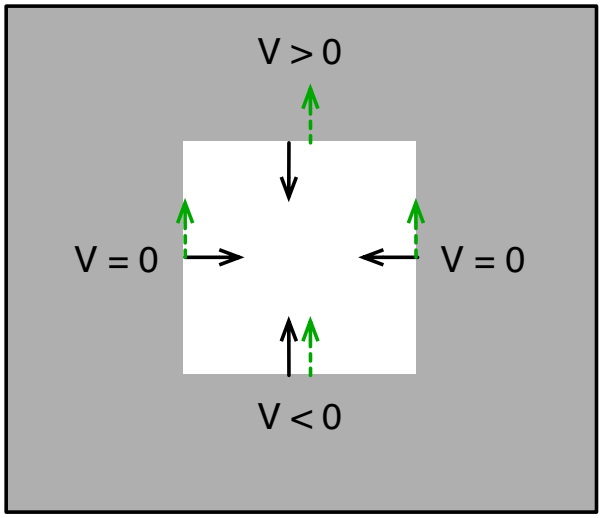
Equation 1 cannot be solved directly for $\vec{\mathbf{F}}_{x,y}$ or $\vec{\mathbf{G}}_{x,y}$ because for a given $\mathbf{V}_{x,y}$ and $\vec{\mathbf{G}}_{x,y}$ there exist many possible solutions for $\vec{\mathbf{F}}_{x,y}$ which all lie on a line perpendicular to $\vec{\mathbf{G}}_{x,y}$ passing through $\vec{\mathbf{G}}_{x,y}/|\vec{\mathbf{G}}_{x,y}| \cdot -\mathbf{V}/|\vec{\mathbf{G}}_{x,y}|$ (see Fig. 3(b)). However, from Equation 1 we can derive a quadratic error function

$$Q_{\mathbf{V},\mathbf{G}}(\vec{\mathbf{F}}_{x,y}) = (\mathbf{V}_{x,y} + \vec{\mathbf{F}}_{x,y} \cdot \vec{\mathbf{G}}_{x,y})^2 \tag{4}$$

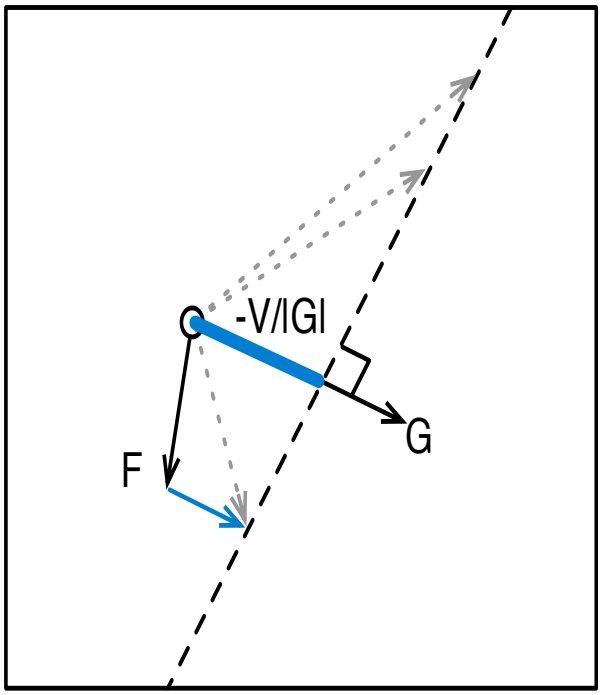
which is zero when Equation 1 is fulfilled and greater than zero otherwise. Calculating the gradient of $Q_{\mathbf{V},\mathbf{G}}(\vec{\mathbf{F}}_{x,y})$ with respect to $\vec{\mathbf{F}}_{x,y}$ yields

$$\frac{dQ_{\mathbf{V},\mathbf{G}}(\vec{\mathbf{F}}_{x,y})}{d\vec{\mathbf{F}}_{x,y}} = 2\vec{\mathbf{G}}_{x,y}(\mathbf{V}_{x,y} + \vec{\mathbf{F}}_{x,y} \cdot \vec{\mathbf{G}}_{x,y}). \tag{5}$$

We can optimize **F** by taking a step in the direction of this gradient, or we can directly solve for the nearest $\mathbf{F}'$ that



(a)



(b)

Fig. 3. The relation between the temporal gradient **V**, the optical flow **F** and the spatial gradient **G**. **(a)** shows a bright square moving upwards on a dark background. Black solid arrows indicate the spatial gradient and green dashed arrows the optical flow for a selection of pixels. On the bottom border both the optical flow and the spatial gradient point in the same direction, thus the pixel at that position becomes darker ($\mathbf{V} < 0$). In contrast, on the top the opposite occurs ($\mathbf{V} > 0$) and on the left and right there occurs no brightness change at all ($\mathbf{V} = 0$) as both vectors are perpendicular. **(b)** $-\mathbf{V} = \mathbf{F} \cdot \mathbf{G}$ can not be solved uniquely for **F** or **G**. Given a spatial gradient **G** and a temporal gradient **V**, the solutions for **F** (e.g. the dotted gray vectors) lie on the dashed black line (perpendicular to **G**). This is known as the *aperture problem* [10]. In our network, the relationship adjusts **F** towards this dashed line, in the direction of the blue arrow.

satisfies Equation 1 and then take a step towards that $\mathbf{F}'$. These approaches yield the same resulting direction, with a possible difference of magnitude if one takes a step size proportional to the gradient. Experimentally both methods seem to work well.

The update rule from $\mathbf{V}$ and $\mathbf{F}$ to $\mathbf{G}$ is defined analogously.

The relation between $\mathbf{I}$ *and* $\mathbf{G}$.

Equation 2 yields an obvious candidate. Thus the update rule from the light intensity map $\mathbf{I}$ to the spatial gradient map $\mathbf{G}$ can be simply given as

$$\vec{\mathbf{G}}_{x,y} = (1 - \delta_{\mathbf{IG}})\vec{\mathbf{G}}_{x,y} + \delta_{\mathbf{IG}} \nabla \mathbf{I}_{x,y}. \tag{6}$$

Note that a gradient descent approach by turning Equation 2 into a difference and squaring it to make an error function yields exactly the same rule (6).

In order to update the light intensity map from the gradient map we compare in a temporary map

$$\vec{\mathbf{\Psi}}_{x,y} = \vec{\mathbf{G}}_{x,y} - \nabla \mathbf{I}_{x,y} \tag{7}$$

current values of $\vec{\mathbf{G}}_{x,y}$ with the hypothetical gradient $\nabla \mathbf{I}_{x,y}$. As changing $\mathbf{I}_{x,y}$ affects also the gradients of the neighbors $(x-1, y)$ and $(x, y-1)$, the effect in the x-direction is computed as

$$\hat{\vec{\mathbf{\Psi}}}^{(x)}_{x,y} = \vec{\mathbf{\Psi}}^{(x)}_{x,y} - \vec{\mathbf{\Psi}}^{(x)}_{x-1,y}, \tag{8}$$

with out-of-bounds entries of $\vec{\mathbf{\Psi}}$ set to 0.

Together with the analogous definition for $\hat{\vec{\mathbf{\Psi}}}^{(y)}_{x,y}$ the update rule is defined as

$$\mathbf{I}_{x,y} = (1 - \delta_{\mathbf{GI}})\mathbf{I}_{x,y} + \delta_{\mathbf{GI}}(\mathbf{I}_{x,y} - \hat{\vec{\mathbf{\Psi}}}^{(x)}_{x,y} - \hat{\vec{\mathbf{\Psi}}}^{(y)}_{x,y}). \tag{9}$$

Again, this update rule can also be derived with a gradient descent approach.

The relation between $\mathbf{F}$, $\mathbf{C}$ *and* $\mathbf{R}$.

The update rule from the rotation to the optical flow map can be defined according to Equation 3 as

$$\vec{\mathbf{F}}_{x,y} = (1 - \delta_{\mathbf{RF}})\vec{\mathbf{F}}_{x,y} + \delta_{\mathbf{RF}} \cdot m_{32}(\vec{\mathbf{R}} \times \vec{\mathbf{C}}_{x,y}) \tag{10}$$

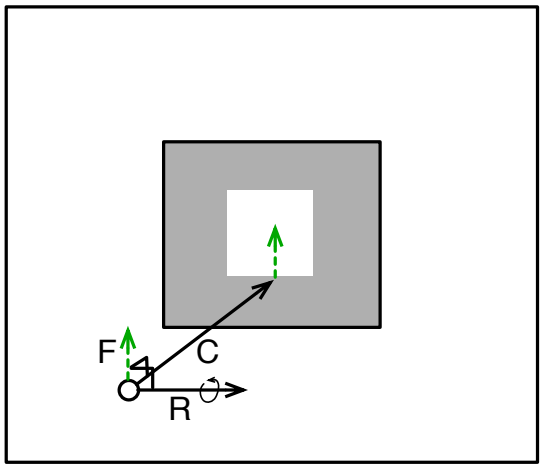
where m_{32} has to be applied to project the result of the cross-product back to image space [16][17].

In the opposite direction each flow vector $\vec{\mathbf{F}}_{x,y}$ restricts the set of possible solutions for $\vec{\mathbf{R}}$ to a straight line (see Fig. 4(b)) defined by

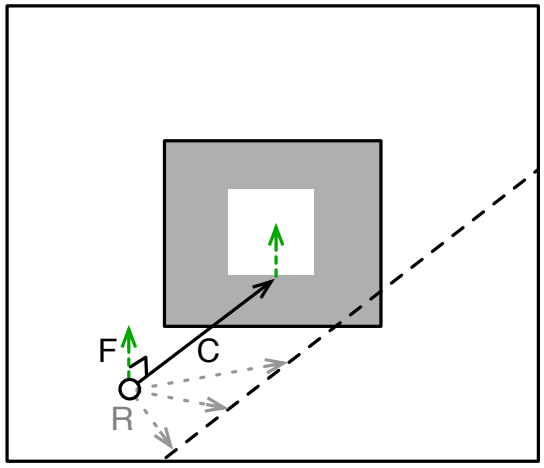
$$\vec{\mathbf{R}}_{x,y} = \vec{\mathbf{C}}_{x,y} \times m_{23}(\vec{\mathbf{F}}_{x,y}) + k \cdot \vec{\mathbf{C}}_{x,y}, \tag{11}$$

where $m_{23} : \mathbb{R}^2 \to \mathbb{R}^3$ maps a vector from image space to world space [16][17]. In a noise free setup the intersection of all $\vec{\mathbf{R}}_{x,y}$ would yield a unique solution for $\vec{\mathbf{R}}$. In our system each optical flow vector can only contribute one opinion about the solution for $\vec{\mathbf{R}}$. For every possible $\vec{\mathbf{R}}$ we can compute its distance to each $\vec{\mathbf{R}}_{x,y}$ as

$$d^2_{\vec{\mathbf{R}}_{x,y}} = |\vec{\mathbf{R}} - \vec{\mathbf{C}}_{x,y} \times m_{23}(\vec{\mathbf{F}}_{x,y})|^2 - (\vec{\mathbf{R}} \cdot \vec{\mathbf{C}}_{x,y})^2. \tag{12}$$



(a)



(b)

Fig. 4. The relation between $\mathbf{F}$ (dashed green vectors) and $\mathbf{R}$. **(a)** $\vec{\mathbf{F}}_{x,y} = \vec{\mathbf{R}} \times \vec{\mathbf{C}}_{x,y}$ follows from $\vec{\mathbf{F}}_{x,y}$ being perpendicular to $\vec{\mathbf{R}}$ and $\vec{\mathbf{C}}_{x,y}$. **(b)** Each entry of the flow map restricts the possible solutions for $\mathbf{R}$ to a straight line (dashed black line) passing through $\vec{\mathbf{C}}_{x,y} \times m_{23}(\vec{\mathbf{F}}_{x,y})$ and parallel to $\vec{\mathbf{C}}_{x,y}$.

By minimizing the error function $Q_{\mathbf{F},\mathbf{C}}(\vec{\mathbf{R}}) = \sum_{x,y} d^2_{\vec{\mathbf{R}}_{x,y}}$ we compute the new candidate of $\vec{\mathbf{R}}$. Therefore, the update rule from the optical flow map to the rotation is defined as

$$\vec{\mathbf{R}} = (1 - \delta_{\mathbf{FR}})\vec{\mathbf{R}} + \delta_{\mathbf{FR}} \cdot \arg\min_{\vec{\mathbf{R}}} Q_{\mathbf{F},\mathbf{C}}(\vec{\mathbf{R}}). \tag{13}$$

The minimization can be carried out by a linear least squares fit.

III. Experimental Results

We tested our network with data recorded using the vision sensor of [9], which has an array of 128×128 pixels, each of which sends an event whenever the relative change in light intensity since the last event sent exceeds a threshold. We bin these events into a sequence of frames, each covering

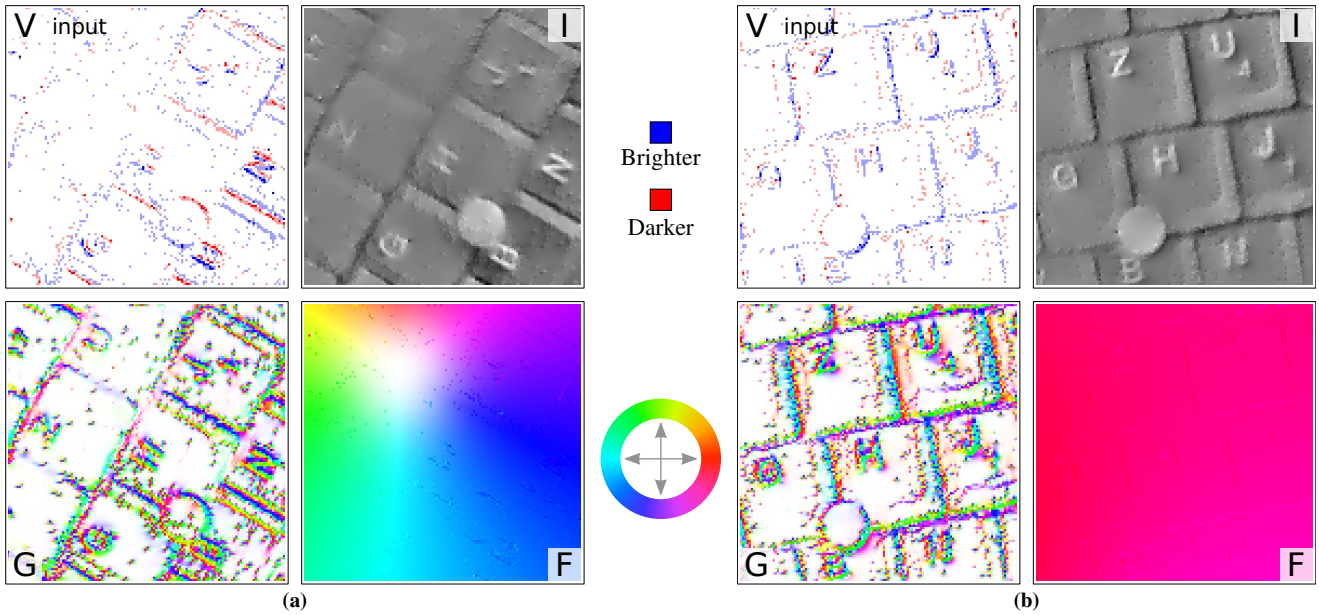


Fig. 5. Two sample results of the simulations. The input **V** shows input from the neuromorphic sensor as it moves above part of a keyboard. Red pixels in **V** represent *off* events, Blue areas *on* events. The other maps are all inferred from **V**. The hue in **G** and **F** represents the direction of the vector at the given pixel (cf. the legend at the center), and the saturation is proportional to the vector norm. **(a)** shows the result of a clockwise rotation of the input around the image center. **(b)** shows the result of a left to right motion of the input.

a short time window. We feed each frame to the system as input **V**, then let it converge to a stable state. All other maps are initialized with the results from the previous converged state or, at the beginning, random data. The system is able to process the data at around 10 frames per second using a C++ program on a 2.66Ghz mobile processor.

Two example states from our experiments are shown in Fig. 5. The system correctly identifies a rotation centered in the top left quadrant of the image in (a), and a translation from left to right in (b). Note that the light intensity map **I** correctly captures the shading of the keyboard despite the sparse input **V**.

The system always converges to a stable state, but when starting from random data sometimes an incorrect local optimum is reached. For example, if the true rotation is about the center of the image, the network can reach an incorrect local optimum which has the flow vectors pointing to the left throughout the image instead of correctly following a circular pattern. This error amounts to the β ambiguity (see p. 3) being resolved with β varying with the vertical position in the image. In contrast, when started from a nearly correct state, as when advancing from frame to frame, such spurious optima can be avoided. As an example of this information propagation note that in Fig. 5(a), **V** has only very sparse input in the upper left corner, but the corresponding part of the keyboard is still visible in **I**.

IV. Discussion

We have explored a new approach to network design, by creating a network consisting of recurrently interconnected maps, each of which represents a different aspect of the visual interpretation of the input. This system works surprisingly well on live data, without needing any arbitrary measures such as explicit smoothing or clipping of outliers to compensate for the noisy data.

The performance of our system demonstrates that it is feasible to process input by using loosely coupled maps that mutually influence each other towards a coherent interpretation. This approach also is not sensitive to whether the problem is mathematically underconstrained or overconstrained; it simply tries to find a reasonable solution. Such networks can be engineered in a relatively straightforward way, based on traditional mathematical relationships, even when there is no clear feed-forward way to compute the solution.

Our network shares several features with neural circuits. It is highly parallelizable, consisting of only local operations which could be performed asynchronously, and it uses a biologically inspired neuromorphic sensor feeding input that is some ways similar to what real, biological retinas send along the optic nerve [11]. It is our hope that studying this interacting map approach might help in understanding how the brain converges to its interpretations, especially given that the input arriving to cortical areas is known to provide an anatomically weak but computationally critical influence on their operation [6].

It would be interesting to convert this network into a spiking form, as opposed to being frame-based. This could allow the interacting map approach to be implementable on neuromorphic spiking hardware (such as [18] [19] [20] [21]), and might expose new issues that are relevant to understanding processing in the brain.

There are several possible extensions to the functionality

of our network, such as finding 'structure from motion' [10], (as described on page 2), tracking multiple objects (by having multiple vectors for **R**), or performing pixel-by-pixel self-calibration of the camera (by making the map **C** be influenceable).

Future work could bring us closer to the vision of a comprehensive neurally plausible visual scene interpretation system efficiently implementable on parallel neuronal hardware.

REFERENCES

[1] T Morris. Computer vision and image processing. *Palgrave Macmillan*, 2004.

[2] M.H. Hayes. *Statistical digital signal processing and modeling*. John Wiley & Sons, 1996.

[3] WR Hendee and PNT Wells. The perception of visual information. *Springer-Verlag*, 1997.

[4] D Felleman and DC Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1:1–47, Jan 1991.

[5] S Ringbauer, P Bayerl, and H Neumann. Neural mechanisms for mid-level optical flow pattern detection. *Lecture Notes in Computer Science*, 4669:281–290, 2007.

[6] NM Costa and KAC Martin. The proportion of synapses formed by the axons of the lateral geniculate nucleus in layer 4 of area 17 of the cat. *J Comp Neurol*, 516(4):264–76, Oct 2009.

[7] Robert Van Gulick. Consciousness. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2009 edition, 2009.

[8] B Horn and B Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[9] P Lichtsteiner, C Posch, and T Delbruck. A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, pages 2060–2069, 2006.

[10] S Ullman. The interpretation of visual motion. *MIT Press*, 1979.

[11] David H. Hubel. *Eye, Brain, and Vision (Scientific American Library, No 22)*. W. H. Freeman, 2nd edition, May 1995.

[12] F Kschischang, B Frey, and H Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, Jan 2001.

[13] J Yedidia, W Freeman, and Y Weiss. Understanding belief propagation and its generalizations. *Exploring Artificial Intelligence in the New Millennium*, pages 239–260, Jan 2003.

[14] S Beauchemin and J Barron. The computation of optical flow. *ACM Computing Surveys (CSUR)*, 27(3):433–467, Jan 1995.

[15] J Stoer and P Bulirsch. Introduction to numerical analysis. *Springer Verlag*, 1993.

[16] JD Foley, Av Dam, SK Feiner, and JF Hughes. Computer graphics: principles and practice. *Addison Wesley*, Jul 1997.

[17] HSM Coxeter and SL Greitzer. Geometry revisited. *The Mathematical Association of America Textbooks*, 1967.

[18] M Khan, D Lester, and L Plana. Spinnaker: Mapping neural networks onto a massively-parallel chip multiprocessor. *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 2849 – 2856, 2008.

[19] J Fieres, J Schemmel, and K Meier. Realizing biological spiking network models in a configurable wafer-scale hardware system. *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 969 – 976, 2008.

[20] G Indiveri, E Chicca, and R J Douglas. A VLSI reconfigurable network of integrate-and-fire neurons with spike-based learning synapses. *Proceedings of 12th European Symposium on Artificial Neural Networks (ESANN04)*, pages 405–410, Jan 2004.

[21] R Silver, K Boahen, S Grillner, N Kopell, and K L Olsen. Neurotech for neuroscience: unifying concepts, organizing principles, and emerging tools. *J Neurosci*, 27(44):11807–19, Oct 2007.