INCIDE the Brain of a Bee: Visualising Honeybee Brain Activity in Real Time by Semantic Segmentation

Martin Strauch[1,2*], Marc P. Broeg[1], Clemens Müthing[1], Paul Szyszka[2], Oliver Deussen[1], C. Giovanni Galizia[2], Dorit Merhof[1]

[1]Interdisciplinary Center for Interactive Data Analysis, Modelling and Visual Exploration (INCIDE), University of Konstanz, Germany
[2]Neurobiology, University of Konstanz, Germany

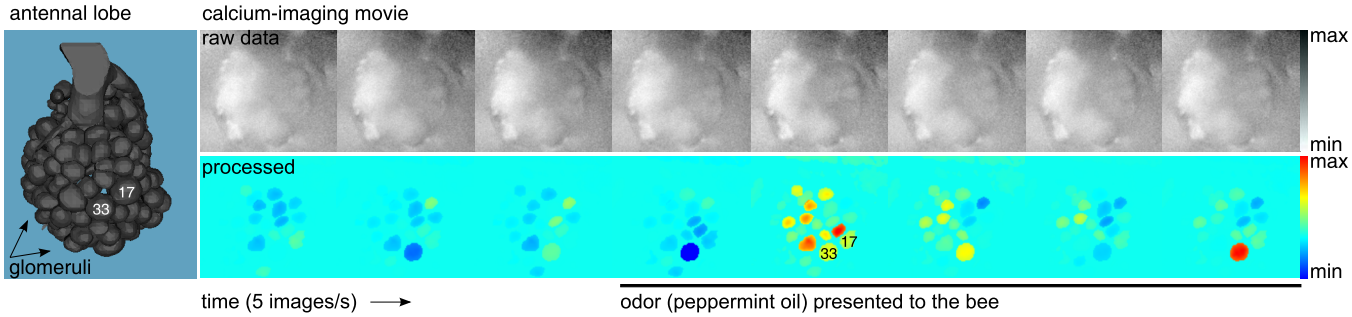

Figure 1: The honeybee brain encodes odors by activity patterns of units called glomeruli in the antennal lobe (AL). These patterns can be observed in calcium imaging movies. For orientation, glomeruli 17 and 33 are labelled. *Left:* Frontal view onto an AL model with a total number of 160 glomeruli. *Right:* Raw data *(upper row)* and visualisation result after processing with the presented method *(lower row)*. We show consecutive images before and during odor application.

Abstract

We present a software solution for processing recordings of honeybee brain activity in real time. In the honeybee brain, odors elicit spatio-temporal activity patterns that encode odor identity. These patterns of neural activity in units called glomeruli can be recorded by calcium imaging with fluorescent dyes, but so far glomerulus segmentation was only possible offline, making interactive experiments impossible. Our main contribution is an adaptive algorithm for image processing, along with a fast implementation for the graphics processing unit that enables semantic segmentation in real time. Semantics is based on the temporal dimension, relying on the fact that time series of pixels within a glomerulus are correlated. We evaluate our software on reference data, demonstrate applicability in a biological experiment, and provide free source code. This paves the way for interactive experiments where neural units can be selected online based on their past activity.

1 Introduction

Biological knowledge discovery is essentially based on manipulation of living organisms. In the study of neural circuits, it is desirable to modify activity of neural units conditional on the activity they exhibited before. Such interactions call for visualisation of brain activity in real time, which allows for identifying the active units already during the course of the experiment.

In our small-scale model circuit, the honeybee antennal lobe (AL), odor molecules smelled by the bee are encoded as combinatorial activity patterns in neural units called glomeruli [12]. Glomeruli in the AL are connected by a network of interneurons. Manipulating glomeruli in this network by means of neuropharmacology, such as in [13], can help to unravel network structure and function.

Activity patterns in the AL are accessible through calcium imaging with fluorescent dyes (see e.g. [27, 24]). The goal of image processing in this domain is to estimate the true signals from noisy movie data and to perform semantic glomerulus segmentation based on the fact that time series from the same glomerulus are correlated.

Extending our previous method [33], we present an adaptive algorithm that can process imaging movies at constant cost per time point. To the best of our knowledge, this is the first adaptive algorithm in this domain. Embedded in a matrix factorisation framework, the algorithm leads to a rank-reduced representation of the movie matrix such that glomerulus signals are highlighted, as shown in Figure 1.

We provide an efficient General Purpose Computation on Graphics Processing Unit (GPGPU) implementation written in C for CUDA (Compute Unified Device Architecture) [3] to enable real-time processing of calcium imaging data acquired from the honeybee AL. Existing software tools are only suitable for post-hoc analysis after the experiment [6, 7, 22, 33], which limits the range of possible biological experiments.

Our software makes it possible to perform interactive experiments in the honeybee AL, where interactions can range from changes in recording parameters to neuropharmacological interventions, such as artificially inducing activity in a particular glomerulus using time cues that originate from the recorded data itself. Real-time visualisation permits targeted interventions, as glomeruli can be chosen both based on their anatomical position and their involvement in activity patterns (see Figure 1).

In the following, we provide background on the biological application scenario and summarise related work (Sections 2 and 3).

*Corresponding author, email: martin.strauch@uni-konstanz.de

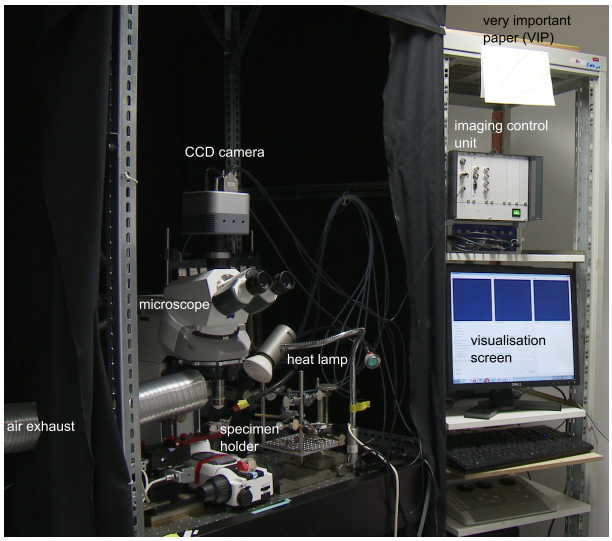

Figure 2: Experimental setup (curtains closed during experiments with fluorescent dyes). Movie streams from the camera on top of the microscope are processed in real time and appear on the visualisation screen. The imaging control unit triggers camera and excitation light source. Dissolved odorants are applied with syringes (not shown) and sucked out through the exhaust. The heat lamp keeps the brain at $\approx 28\,^{\circ}\mathrm{C}$.

We then develop an adaptive algorithm for processing imaging movies from the honeybee AL (Section 4). In Section 5, we evaluate both the algorithm and the software on reference data and demonstrate applicability in a biological experiment.

2 Biological background

2.1 Observing the olfactory code

The olfactory system is a nice example for how brains process information in activity patterns across neurons. In the bee, 60,000 receptor neurons converge onto the 160 glomeruli (30-50 μm each) of the AL, creating an information space with 160 dimensions: Each activity pattern at any time can be fully described by a vector with dimensionality 160. In higher-order brain centers, this space is then read out by 160,000 target neurons (the Kenyon Cells in the mushroom bodies), only to then collapse onto less than 100 "executive neurons" [11, 38].

Across species, olfactory coding is based on receptors with overlapping response ranges. Due to the vast number of chemical molecules, there is typically not a one-to-one correspondence between odor molecules and receptors, but it is the combined activity pattern of the entire receptor repertoire that encodes the odor [21]. This activity pattern is relayed to the AL, and a specific odor reliably elicits the same glomerular pattern in different bees [12].

In vivo imaging with calcium-sensitive fluorescent dyes makes it possible to record brain activity in the AL of the honeybee *Apis mellifera*. Figure 1 shows glomerular activity before and after presentation of the odorant peppermint oil. Note that we have a frontal view onto the AL that reveals between ca. 20 and 40 glomeruli depending on the focal plane.

2.2 Experimental setup

Figure 2 shows the experimental setup used in this work to record the calcium imaging movies. A CCD camera (Andor Clara, Andor Technology PLC, Belfast, Northern Ireland) is mounted on top of a fluorescence microscope (Axio Imager D.1, Zeiss, Göttingen, Germany) equipped with a water immersion objective (20×, NA 0.95, Olympus, Tokyo, Japan). Activity is recoreded from the opened honeybee brain, processed incrementally and visualised on a computer screen.

Experiments were performed with honeybee foragers (*Apis mellifera*). The day before the experiments, bees were caught from hives and the projection neurons of lateral and medial antennoprotocerebral tract (l- and m-APT) were stained with the calcium-sensitive dye Fura-2 dextran (Invitrogen, Molecular Probes, Eugene, OR, USA). Projection neurons are the neurons that project from the glomeruli to higher-order brain centers.

Calcium-imaging is a versatile technique for measuring neural activity [14]. In the honeybee, changes in intracellular calcium correlate with changes in projection neuron firing rate [10]. By exciting the dye with UV light, we can measure the intracellular calcium level and thus activity in all the glomeruli in the current focal plane.

A light source (Polychrome V, TILL Photonics, Gräfelfing, Germany) provided excitation light. For each recording, four double frames per second were recorded with 340 and 380 nm excitation light. The input signal for processing was the ratio between consecutive images taken at 340 and 380 nm. This is a standard procedure for this dye (see [23]). For further details on the experimental protocol, see [34].

3 Related work

Calcium imaging data from the honeybee AL is typically analysed with semi-automatic methods, such as described in [6, 7, 8]. These methods start with preprocessing to filter out noise and to visualise correlations between the signals of neighbouring pixels, and then require user interaction to select glomerulus positions based on visual inspection. This is not suitable for real-time processing where decisions need to be made in fractions of a second.

A more advanced computational approach [30] involves fitting of pre-defined model functions to the movie, e.g. components corresponding to noise, signal, artifacts, slow and fast calcium dynamics, etc. This synthetic method is designed for post-hoc data interpretation rather than for real-time visualisation.

Analytic methods attempt to decompose the data into components that can be seen as latent factors underlying the data. They do not require prior knowledge on e.g. the nature of slow and fast calcium dynamics. Common approaches are Principal Component Analysis (PCA), Non-negative Matrix Factorisation (NMF) and Independent Component Analysis (ICA). Especially ICA has been applied to various kinds of imaging data [2, 25, 31, 22, 32].

Methods based on ICA solve a blind source separation problem, where signal sources are estimated from a signal mixture without knowledge about the mixing process (denoted as "blind"). The set of solutions is, however, constrained by assuming a statistical model. ICA estimates statistically independent source signals based on the model assumption that the true sources, e.g. glomerulus signals, are in fact statistically independent, and, for all except one, non-Gaussian.

The most recent approach is the convex cone algorithm [33]. As opposed to the ICA-based methods which are variants of a general independence paradigm, the convex cone algorithm (see Section 4) is based on a data model explicitly designed for honeybee imaging movies. No assumptions on statistical properties, such as independence or non-Gaussianity, are made. Following a simple greedy strategy, it is also faster than common iterative ICA methods (such as [16]). This renders it suitable for repeated execution on a movie stream, which is the strategy we pursue in Section 4.

Software is available for some of the methods described above [6, 22, 7], yet none of these programs is able to read and process imaging data in real time.

4 ALGORITHM

4.1 Matrix factorisation framework

A movie can be represented by a movie matrix $A^{m\times n}$, where m refers to the number of time points, and n to the number of pixels. Individual images from the movie are flattened into length-n vectors and filled into the rows of matrix A. In the following, these m rows of A are denoted as $A_{(i)}$ and the n time series in the columns of A are denoted as $A^{(j)}$.

As matrix A contains noise and irrelevant signals, an approximation to A with rank $k \ll m,n$ can be obtained without losing biological signals. In fact, the approximation can serve to highlight relevant features in the movie.

Our goal is to compute such a rank-k approximation A_k to the movie matrix A. Formally, we write A_k as the product of a time series matrix $T^{m\times k}$ and an image matrix $S^{k\times n}$:

$$A^{m\times n}:\ A_k = T^{m\times k}\, S^{k\times n} = \sum_{r=1}^{k} T_{Ir}\, S_{rJ} \qquad (1)$$

This is illustrated in Figure 3, which shows a general matrix factorisation framework. Please note that different method-specific constraints will lead to different solutions. For example, Principal Component Analysis (PCA) [18] would lead to k mutually orthogonal principal component vectors (images) in S.

Ideally, the k image vectors in the rows of S should indicate the positions of the neural units and the k time series vectors in the columns of T should be the corresponding signals. In our previous paper [33] we have introduced a non-negative mixture model for imaging data from the honeybee AL that leads us to a sparse solution to Equation (1) where this is in fact the case.

4.2 Convex cone algorithm

In the following, we briefly review the mixture model and the convex cone algorithm from [33], before we then concentrate on adapting it to the data stream domain.

An imaging movie can be modelled by non-negative combinations of basis time series in T, plus the residual noise N.

$$A = TS^{0+} + N \qquad (2)$$

The model assumption is that vectors $A^{(j)}$ from the movie are either pure glomerular signals (plus noise) or linear combinations of these pure glomerular signals with non-negative coefficients. In the middle of a glomerulus, pure, unmixed signals exist, whereas in regions of contact between the glomeruli signal mixtures can occur due to light scatter from neighbour glomeruli. I.e. a weak signal can appear stronger due to additive light scatter from a neighbour.

Glomeruli each have their individual signal due to individual background activity and as they respond differentially to odors. Our approach is to select a pure, unmixed time series from the middle of each of the glomeruli into the columns of T. Geometrically, these are the generating extreme vectors of a convex cone that contains the data. Each time series vector from the movie can be reconstructed as a conic combination of the generating vectors, i.e. a linear combination with non-negative coefficients.

The 2D example in Figure 4 illustrates the special status of extreme vectors. Given the non-negative mixture model, the generating pure signals from the middle of the glomeruli are the extreme vectors of a convex cone that contains all time series vectors from the movie. For more background on convex geometry refer also to [26, 5].

The goal is thus to find the extreme, generating vectors as basis vectors in the columns of matrix T in Equation (1). The algorithm follows a greedy strategy to quickly enumerate c extreme time series vectors. It selects the column from $A_{\{1\}} := A$ which is least explained by combinations of the vectors selected so far, i.e. the column with the largest euclidean norm: $\text{argmax}_p \left\| A^{(p)}_{\{1\}} \right\|$.

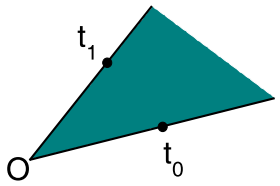

Figure 4: 2D example for the convex cone view. Two basis vectors, $\mathbf{t}_0$ and $\mathbf{t}_1$, that contain pure, unmixed signals have been selected. They are the extreme vectors of a convex cone. All mixed signals, i.e. data points in the open-ended, coloured area, can be reconstructed by conic combination of the generating extreme vectors.

Before finding the next column, the influence of the selected column $T^{(1)} := A^{(p)}_{\{1\}}$ is removed, forming the residual matrix $A_{\{2\}}$: $A_{\{2\}} := A_{\{1\}} - T^{(1)} S_{(1)}$, where $S_{(1)} := A^T_{\{1\}}\, T^{(1)}$.

4.3 Working on a movie stream

For visualisation in real time we need to process a movie stream. Now, A is a streaming matrix that grows by one row (image) per time point. In terms of computational complexity, the convex cone algorithm from [33] is in the order $\mathcal{O}(mnk)$, i.e. time consumption depends linearly on the movie dimensions m and n. However, repeatedly running the algorithm to obtain an updated solution at every time point would require a much higher cost in the order $\mathcal{O}(1nk + 2nk + ... + mnk)$.

Clearly, for fast processing in real time we should avoid costs that grow with the time dimension. The idea is to utilise an incremental PCA (IPCA) with constant cost per time point. IPCA serves to buffer the movie stream, at each time point updating a matrix V_k of the top-k principal components, such that the convex cone algorithm is performed on the small, constant-size V_k instead of the growing movie A.

Apart from dimensionality reduction to $k \ll m,n$ principal components, tasks such as selecting extreme vectors [33] and computing independent components [32] on imaging movies also benefit from noise reduction by PCA. Signals that contribute to the variance are typically concentrated in the top principal components, whereas noise components with lower eigenvalue can be cut off.

Thus, with appropriate processing by IPCA we can compute a matrix V_k that is both noise reduced and has constant size at each time point, allowing for robust and fast execution of the convex cone algorithm on a movie stream. In the literature, a multitude of IPCA algorithms is available [28, 29, 37, 41, 43, 15]. In particular, Weng et al. [37] proposed the CCIPCA algorithm that has constant computational cost of $\mathcal{O}(nk)$ per time point for updating the top-k principal components. Several publications [40, 4, 19] have already employed CCIPCA as a building block for incremental algorithms that update the principal component matrix V_k with CCIPCA and then perform the algorithm to be incrementalised, e.g. Independent Component Analysis (ICA) [4]. Here, we rely on this principle to incrementalise the convex cone algorithm.

A common way of computing PCA is by eigenvector decomposition of the covariance matrix. For large movie files, the dimensionality of the covariance matrix can, however, become inconveniently high. In contrast, the CCIPCA algorithm avoids costly operations on the covariance matrix. CCIPCA is an incremental approximation to PCA that estimates each principal component as the mean of the samples seen so far in the respective subspace, i.e. the first principal component corresponds to the mean of the data vectors, the second

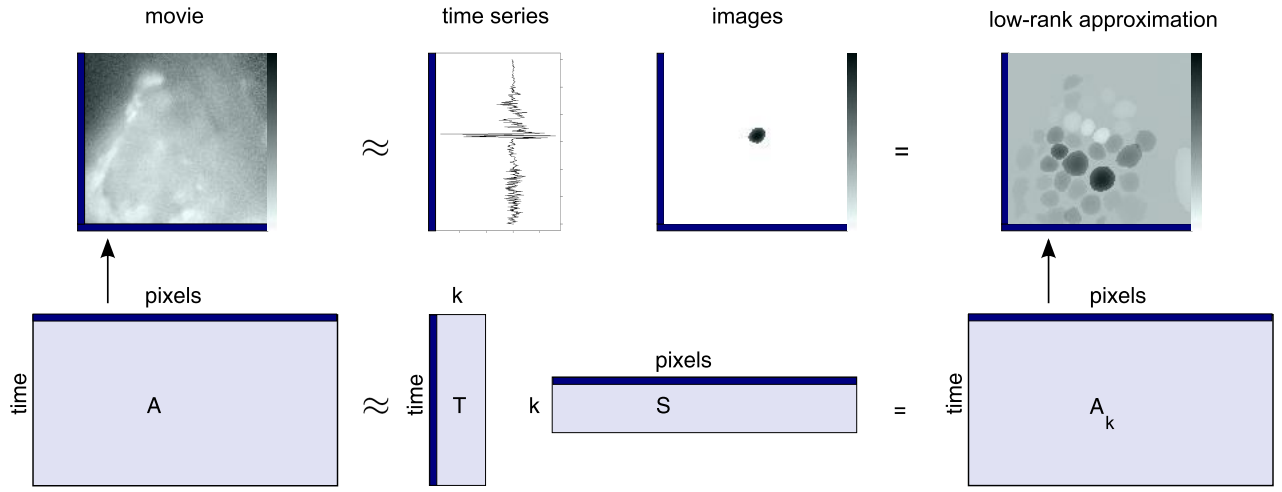

Figure 3: Illustration of the matrix factorisation framework. The movie matrix A is factorised into components with a temporal interpretation in T and components with spatial interpretation in S. When spatial components are sparse and restricted to a single neural unit, the low-rank (rank-k) reconstruction A_k provides a denoised version and visualisation of the original movie A.

principal component corresponds to the mean of the data vectors after subtracting the projection onto the first principal component, etc. For a convergence proof, see [42].

Briefly, CCIPCA is initialised with k random, orthogonal vectors in V_k. At each time point i, one image $A_{(i)}$ arrives from the stream and is used to update the $r = 1,\ldots,k$ rows of V_k. For ease of notation, let $V := V_k$.

$$V_{(r)}^{\{i\}} := \frac{i-1}{i} V_{(r)}^{\{i-1\}} + \frac{1}{i} A_{(i)} A_{(i)}^T \frac{V_{(r)}^{\{i-1\}}}{\left\| V_{(r)}^{\{i-1\}} \right\|} \tag{3}$$

Here, $V_{(r)}^{\{i\}}$ denotes $V_{(r)}$ at time point i. After updating the respective $V_{(r)}$, its influence is removed from the current image $A_{(i)}$:

$$A_{(i)} := A_{(i)} - A_{(i)}^T \frac{V_{(r)}^{\{i\}}}{\left\| V_{(r)}^{\{i\}} \right\|} \frac{V_{(r)}^{\{i\}}}{\left\| V_{(r)}^{\{i\}} \right\|} \tag{4}$$

This is repeated for all k principal components (Algorithm 1).

In practice, we preprocess the images $A_{(i)}$ by z-score normalisation, i.e. for each pixel we first subtract the mean μ and then divide by the standard deviation σ, where both parameters are updated incrementally.

4.4 Visualisation in real time

With the convex cone algorithm (Section 4.2) and the incremental PCA (IPCA, Section 4.3) we now have tools at hand to process the movie in real time and to provide a visualisation by low-rank approximation to the movie matrix. For a summary of the procedure, see `Cone_updating` (Algorithm 2): At each time point i, we update the principal component matrix V with the current image $A_{(i)}$ and then run the convex cone algorithm to find c extreme vectors of the constant size, dimensionality-reduced $k \times n$ matrix V.

The convex cone algorithm selects extreme time series vectors into matrix T, which are the pure glomerular signals, and it computes the corresponding non-negative images in matrix S, which indicate the spatial distribution of the signals.

The visualisation shown to the biologist when performing an experiment is the low-rank approximation $A_k = TS$, which can be seen as images in S being modulated by the time series in T. On a movie stream, we obtain a low-rank approximation to each image $A_{(i)}$ by projecting it onto the current version of the sparse matrix S at time point i: $A_{(i)} S^{\{i\}} S^{\{i\}}$. Thereby, we incrementally construct a low-rank approximation to the entire movie A (see also Figure 3).

Note that `Update_IPCA` is peformed at each time point i using only the current image $A_{(i)}$ and the prior version of the principal component matrix at time point $i-1$, $V^{\{i-1\}}$. Then, the `Convex_cone_algorithm` [33] is performed on $V^{\{i\}}$.

Algorithm 1 : $V^{\{i\}}$ = `Update_IPCA` $(V^{\{i-1\}}, A_{(i)}, k, i)$

for all $r \in [0, k-1]$ **do**

$$V_{(r)}^{\{i\}} := \tfrac{i-1}{i} V_{(r)}^{\{i-1\}} + \tfrac{1}{i} A_{(i)} A_{(i)}^T \frac{V_{(r)}^{\{i-1\}}}{\left\| V_{(r)}^{\{i-1\}} \right\|}$$

$$A_{(i)} := A_{(i)} - A_{(i)}^T \frac{V_{(r)}^{\{i\}}}{\left\| V_{(r)}^{\{i\}} \right\|} \frac{V_{(r)}^{\{i\}}}{\left\| V_{(r)}^{\{i\}} \right\|}$$

end for

Algorithm 2 : S = `Cone_updating` $(A^{(m \times n)}, c, k)$

initialise $V^{\{1\}}$
for all $i \in [0, m-1]$ **do**
  $A_{(i)}$:= `z_score_normalise` $(A_{(i)})$
  if $i > 1$ **then**
    $V^{\{i\}}$:= `Update_IPCA`$(V^{\{i-1\}}, A_{(i)}, k, i)$
    $S^{\{i\}}$:= `Convex_cone_algorithm` $(V^{\{i\}}, c)$
    $\widehat{A}_{(i)} := A_{(i)} S^{\{i\}} S^{\{i\}}$ // low-rank approximation to image $A_{(i)}$
  end if
end for

For display, we apply a high-pass filter (0.025 Hz) to the images $\widehat{A}_{(i)}$ in order to remove long term trends such as dye bleaching during the recording. This also compensates for level differences between glomeruli such that all glomerular signals can be displayed using the same colour scale.

Implementation	ms/frame	complete in min
Java_offline	134	68,15
Java_online	65	39,26
GPGPU_online	23	18,31

Table 1: Average computation time per frame and total computation time (for all 11 movies).

4.5 Implementations

Reference implementations for the offline convex cone algorithm from [33] and the incremental online variant proposed in this work were written in Java and performed within the data pipelining environment KNIME [1]. In the following, we refer to these implementations as Java_offline and Java_online, respectively.

Additionally, we consider an implementation of the incremental online algorithm which takes advantage of GPGPU (GPGPU_online). In particular, the convex cone algorithm was performed on the CPU, whereas z-score normalisation and PCA were performed with GPGPU using the NVIDIA CUDA [3] Basic Linear Algebra Subroutines (cuBLAS) [1] and the CUDA Linear Algebra library (CULA) [2].

We used the TILL Photonics Live Acquisition (LA) Software 2.0 [35] to configure experimental hardware, to determine excitation light intensity and the focal plane of the recording. A software interface provided by TILL Photonics allowed us to access the movie stream directly, bypassing LA.

5 Results and Discussion

In this section, the performance of the three implementations is assessed (Section 5.1) which comprises evaluation of computing times and a comparison between the results of the offline reference and the online algorithm. Experiences from applying the approach in a biological experiment are provided in Section 5.2, along with a discussion of advanced experimental setups that require interactive analysis of functional compartments of the AL.

5.1 Performance measures

5.1.1 Computing time

We evaluated computing time for each of the three implementations, Java_offline, Java_online and GPGPU_online (see Section 4.5) on reference data from [33]. The reference dataset consists of 11 honeybee imaging movies. Movies vary slightly in image size and recording time ($\approx 170 \times 130$ pixels, ≈ 3500 frames recorded with variable frequency, in total ≈ 15 minutes recording time).

All time measurements were performed on an Intel Core i7 950 (3.07 GHz) machine with 4 GB RAM and a NVIDIA GeForce GTX 285 (648 MHz, 1024 MB) graphics card on a 64 Bit Windows 7 system. For the sake of comparability, running times for Java_offline and Java_online do not include data transfer between nodes in the KNIME workflow.

Computing times are reported in Figure 5 and Table 1. Replacing the exact PCA from Java_offline with the incremental approximation (Java_online) already lead to an approximately 1.5-fold speedup. Running times were more variable for Java_offline due to different convergence times in the iterative PCA approach [39]. On top of the algorithmical speedup, GPGPU_online achieved another approximately 2-fold speedup compared to the CPU implementation of the same algorithm in Java (Java_online).

The speedup by GPGPU_online is currently limited by initial memory transfer overhead, but the parallelisation capabilities of GPGPU ensure that running times scale favorably with future increases in image size and resolution.

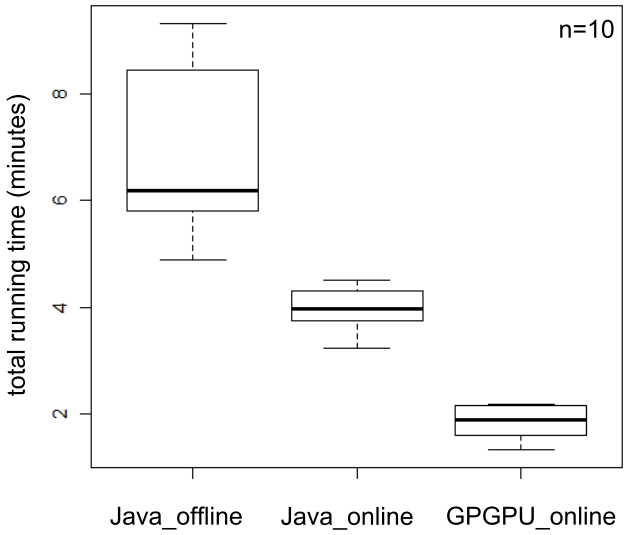

Figure 5: Running times on 11 honeybee imaging movies of approximately 15 minutes length each (≈ 3800 frames recorded with variable frequency). We tested the three implementations from Section 4.5: Java_offline, Java_online and GPGPU_online. Boxplots indicate median running time.

In summary, running times of GPGPU_online are sufficient for real-time application in biological experiments. Compared to the offline reference implementation, improved running time can be explained both by efficient computation with GPGPU and the incremental approximation to PCA. Moreover, incremental computation has another major advantage in practice: Once enough data has been recorded to detect all glomeruli, the incremental online algorithm is already able to present a solution which will be refined further as more data is acquired, whereas the offline algorithm would only start computing.

5.1.2 Approximation quality

In the next step, the quality of the results computed by the incremental algorithm (GPGPU_online implementation) was evaluated. Using the reference data from [33], we constructed glomerulus maps of the AL by collapsing all spatial components from matrix S into one image (Figure 6). Glomerulus maps provide a quick overview of AL anatomy and they allow us to compare whether the same anatomy could be recovered by the offline reference implementation Java_offline and the incremental algorithm (GPGPU_online). In both cases we used identical parameter settings (PCA $k = 50$, convex cone algorithm $c = 50$, Gaussian filter with width 7 as preprocessing).

While glomerulus maps are not identical between the implementations, the incremental online algorithm can produce useful glomerulus maps that provide visual orientation for the experimenter (Figure 6). We observed that quality of the results was limited by the incremental PCA that was very sensitive to imperfect z-score normalisation in the online setting. Problems with incremental z-score normalisation were caused by mean drifts during the course of the movie stream. In future work, we plan to evaluate alternative normalisation strategies, such as band-pass filters.

[1] http://developer.nvidia.com/cublas
[2] http://www.culatools.com/

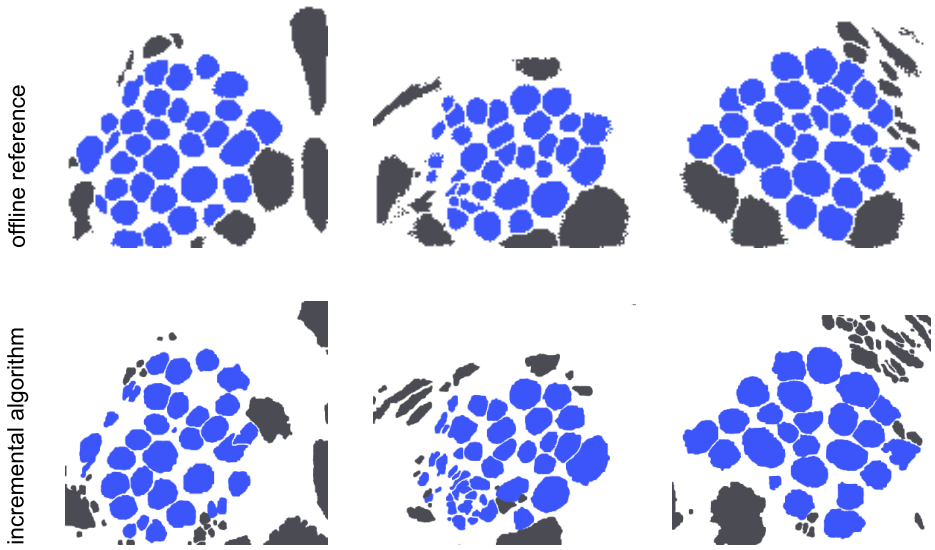

Figure 6: *Top:* Glomerulus maps for reference bees computed with the offline reference implementation Java_offline (reproduced from [33]). *Bottom:* Corresponding AL maps computed with the adaptive online algorithm (GPGPU_online).

5.2 Biological experiment

Having evaluated algorithms and implementations on reference data, we went into the lab to employ our software in an actual biological experiment. We used the experimental setup described in Section 2 and the fastest of the three implementations, GPGPU_online, to record and process imaging movies from two bees.

For a screenshot of our software, see Figure 7. During the experiments, images are constantly updated, showing raw data, the incrementally updated glomerulus map, and the low-rank approximation to the movie matrix. Figure 8 documents the development of a glomerulus map (see Section 5.1.2) during the course of the experiment. As we perform a semantic segmentation based on glomerulus activity, more information is available at later time points, which leads to a gradual buildup of the map. Experimental parameters and the level of glomerular activity in the particular bee determine convergence of the method. In this case, the glomerulus map converged quickly towards its final state. All glomeruli were already present in the map after about 1000 time points, while the additional iterations improved the visual quality of the map. We have already given an example for the low-rank approximated movie in Figure 1 that shows glomerular activity patterns in response to an odor stimulation. For another example, see Figure 9 where glomeruli exhibit spontaneous background activity without odor stimulation when the AL is apparently in an idle state. In humans, ongoing activity can cause sensory experiences in the absence of external stimuli. This has been analysed extensively in tinnitus research [36]. Results from [9] suggest that spontaneous activity could contain information about the last odor the bee has smelled, i.e. a sort of short term memory. Being able to respond to patterns that occur in spontaneous background activity offers new possibilities to the experimenter, such as performing manipulations conditional on the presence of certain patterns, addressing the perceptual role that ongoing activity may play in neural networks.

From a biological point of view, real-time analysis of calcium imaging movies allows for advanced experimental setups that require interactive analysis of functional compartments of the AL. This comprises experiments where neural units can be selected online based on their past activity, or in fact based on their instantaneous activity. This will allow us to address one of the basic questions in neuroscience: How does ongoing brain activity influence the way that sensory input is processed? If an odor is given at a point in time when the spontaneous activity resembles the combinatorial code of that odor, does that facilitate that particular pattern? Conversely, does a complementary spontaneous activity pattern act as a negative filter on incoming sensory experiences? From a human perspective: If spontaneous activity in the brain is already similar to "smelling a rose", does the real rose become more vivid?

5.3 Supplementary Material

For a video documentation of an imaging experiment refer to the movies in the online supplementary material that also comprises all source code used in this work. Note that our software requires TILL Photonics LA 2.0 [35] for configuring experimental hardware.

6 Conclusions and Outlook

We have introduced software for visualising activity patterns in calcium imaging movies of the honeybee AL. Due to an adaptive algorithm and efficient implementation, visualisation results can be obtained in real time already during the course of an experiment. This is a significant advantage over our previous offline implementation [33] and all other software tools in the field [6, 22, 7] that are only suitable for post-hoc data analysis after the experiment.

Visualisations include an incrementally updated glomerulus map (Figure 8) that provides orientation for the experimenter, allowing to target specific glomeruli based on their position in the AL. A low-rank approximation to the calcium imaging movie highlights activity patterns (Figures 1, 9).

This enables e.g. targeted pharmacological manipulations of glomeruli that responded to an odor. Furthermore, decisions can be based on the occurrence of patterns in spontaneous activity. With our adaptive algorithm and GPGPU-based implementation we pave the way towards novel, interactive experiments in honeybee neuroscience.

While the algorithm originates from a data-specific mixture model, this does not limit its wider applicability. Apart from honeybees, we have already applied our method to similar imaging data from the mosquito AL. The convex cone algorithm may also find application in other domains, e.g. data mining [17] or general scientific data analysis [20] with column-based matrix factorisations.

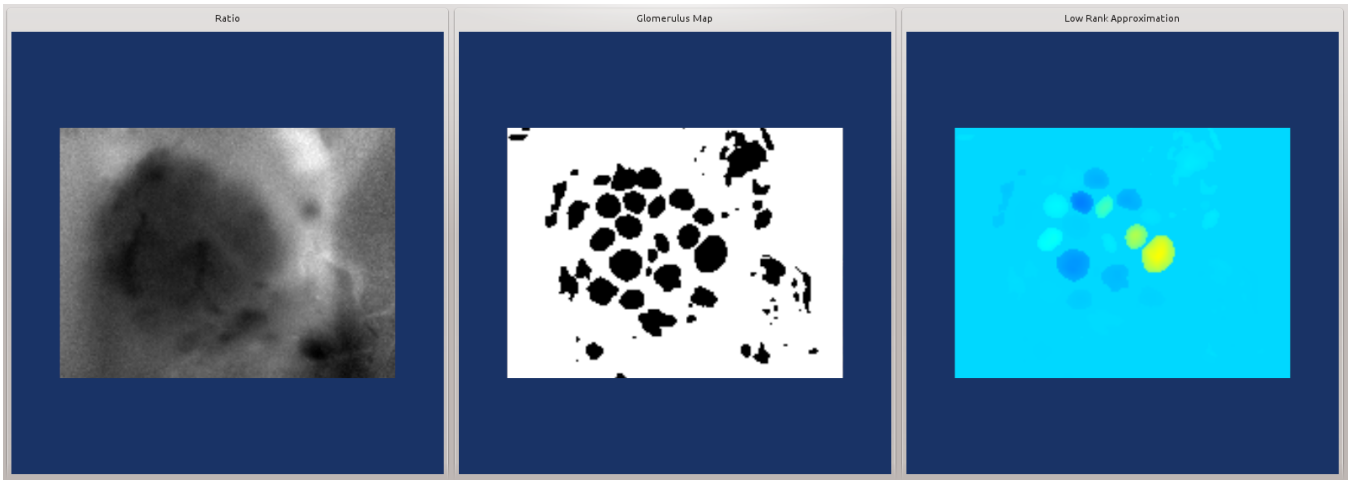

Figure 7: Screenshot. *Left*: Raw movie (fluorescence 340/380 nm), *Middle:* Incrementally updated glomerulus map. *Right:* Low-rank approximation to the raw movie. We employ a min-max (blue-red) colour scale, where min and max are updated over the course of the experiment.

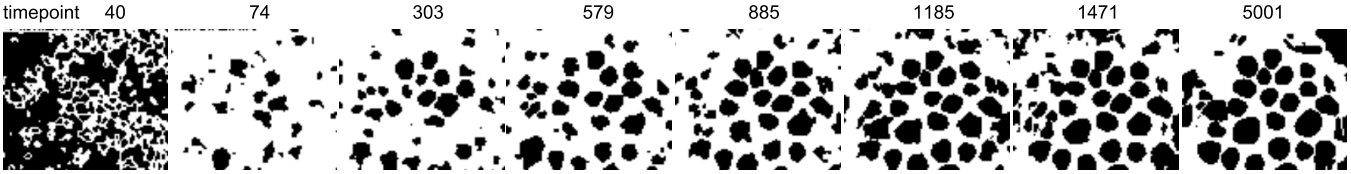

Figure 8: Incremental map construction for one of the bees processed online with our imaging system. Parameter settings: $k = 50$ principal components for CCIPCA, $c = 50$ for the convex cone algorithm. Principal components were spatially filtered with a Gaussian kernel (width=9).

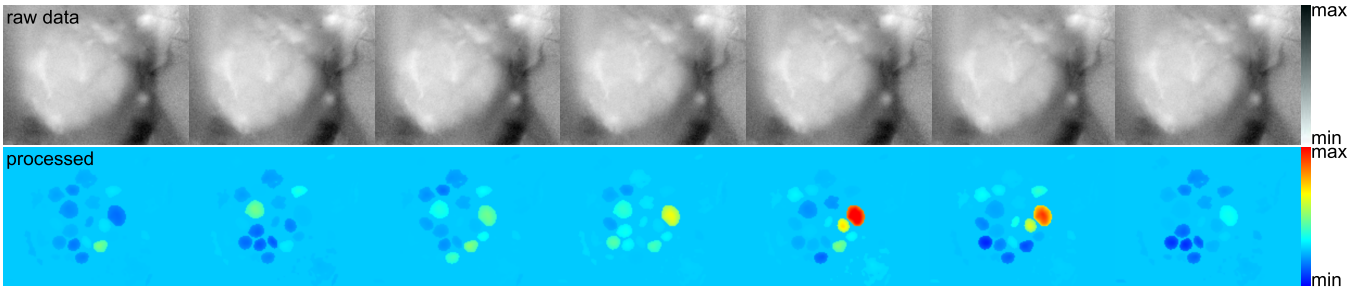

Figure 9: Consecutive images from one one of the movies processed online with our imaging system. *Top:* raw data (ratio 340/380). *Bottom:* processed. Glomeruli exhibit spontaneous background activity.

ACKNOWLEDGEMENTS

We are grateful to TILL Photonics GmbH (Munich, Germany) for providing a software interface for their imaging system. We would like to thank Julia Rein for the reference data from Section 5, and Jacob Stierle for assistance with handling and preparing bees.

REFERENCES

[1] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, and B. Wiswedel. KNIME: The Konstanz Information Miner. In *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*. Springer, 2007.

[2] G. D. Brown, S. Yamada, and T. J. Sejnowski. Independent component analysis at the neural cocktail party. *Trends in Neurosciences*, 24(1):54–63, 2001.

[3] N. Corporation. *NVIDIA CUDA Compute Unified Device Architecture Programming Guide*. NVIDIA Corporation, 2007.

[4] I. Dagher and R. Nachar. Face recognition using IPCA-ICA algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(6):996–1000, 2006.

[5] J. Dattoro. *Convex Optimisation & Euclidean Distance Geometry*. Meboo Publishing, Palo Alto, CA, 2011. ISBN: 0976401304.

[6] M. Ditzen. *Odor concentration and identity coding in the antennal lobe of the honeybee Apis mellifera*. PhD thesis, Freie Universität Berlin, http://www.diss.fu-berlin.de/2005/211/indexe.html, 2005.

[7] F. Dupuy, J. Casasa, A.-G. Bagnresa, and C. R. Lazzaria. Openfluo: A free open-source software for optophysiological data analyses. *Journal of Neuroscience Methods*, 183(2):195–201, 2009.

[8] P. C. Fernandez, F. F. Locatelli, N. Person-Rennell, G. Deleo, and B. H. Smith. Associative Conditioning Tunes Transient Dynamics of Early Olfactory Processing. *J. Neurosci.*, 29(33):10191–10202, 2009.

[9] R. F. Galán, M. Weidert, R. Menzel, A. V. M. Herz, and C. G. Galizia. Sensory memory for odors is encoded in spontaneous correlated activity between olfactory glomeruli. *Neural Comput*, 18(1):10–25, Jan 2006.

[10] C. G. Galizia and B. Kimmerle. Physiological and morphological characterization of honeybee olfactory neurons combining electrophysiology, calcium imaging and confocal microscopy. *J Comp Physiol A Neuroethol Sens Neural Behav Physiol*, 190(1):21–38, Jan 2004.

[11] C. G. Galizia and R. Menzel. The role of glomeruli in the neural representation of odors: results from optical recording studies. *J. Insect Phys.*, 47:115–129, 2001.

[12] C. G. Galizia, S. Sachse, A. Rappert, and R. Menzel. The glomerular code for odor representation is species specific in the honeybee Apis mellifera. *Nat Neurosci*, 2(5):473–478, May 1999.

[13] C. Girardin and C. Galizia. The "where" and "who" in brain science: Probing brain networks with local perturbations. *Cognitive Computation*, 4:63–70, 2012. 10.1007/s12559-011-9122-3.

[14] C. Grienberger and A. Konnerth. Imaging calcium in neurons. *Neuron*, 73(5):862–885, Mar. 2012.

[15] D. Huang, Z. Yi, and X. Pu. A new incremental PCA algorithm with application to visual learning and recognition. *Neural Processing Letters*, 30(3):171–185, 2009.

[16] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[17] S. Hyvönen, P. Miettinen, and E. Terzi. Interpretable nonnegative matrix decompositions. In *Proceedings of the 14th ACM SIGKDD, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 345–353, 2008.

[18] I. T. Jolliffe. *Principal Component Analysis*. Springer, New York/Berlin/Heidelberg, 2nd edition, 2002. ISBN: 0387954422.

[19] V. R. Kompella, M. D. Luciw, and J. Schmidhuber. Incremental slow feature analysis. In *Proceedings of IJCAI, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 1354–1359, 2011.

[20] M. W. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *Proc Nat Acad Sci USA*, 106(3):697–702, January 2009.

[21] B. Malnic, J. Hirono, T. Sato, and L. B. Buck. Combinatorial receptor codes for odors. *Cell*, 96(5):713 – 723, 1999.

[22] E. A. Mukamel, A. Nimmerjahn, and M. J. Schnitzer. Automated analysis of cellular signals from large-scale calcium imaging data. *Neuron*, 63(6):pp.747 – 760, 2009.

[23] N. O'Connor and R. B. Silver. Ratio imaging: Practical considerations for measuring intracellular Ca^{2+} and pH in living cells. In G. Sluder and D. E. Wolf, editors, *Digital Microscopy, 3rd Edition*, volume 81 of *Methods in Cell Biology*, pages 415 – 433. Academic Press, 2007.

[24] L. Rath, C. Giovanni Galizia, and P. Szyszka. Multiple memory traces after associative learning in the honeybee antennal lobe. *European Journal of Neuroscience*, 34(2):352–360, 2011.

[25] J. Reidl, J. Starke, D. Omer, A. Grinvald, and H. Spors. Independent component analysis of high-resolution imaging data identifies distinct functional domains. *NeuroImage*, 34(1):94–108, January 2007.

[26] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970. ISBN: 0691080690.

[27] S. Sachse and C. Galizia. Role of inhibition for temporal and spatial odor representation in olfactory output neurons: a calcium imaging study. *J Neurophysiol*, 87(2):1106–1117, Feb 2002.

[28] D. Skocaj and A. Leonardis. Weighted and robust incremental method for subspace learning. In *Proceedigns of ICCV, 14-17 October 2003, Nice, France*, pages 1494–1501, 2003.

[29] D. Skocaj, A. Leonardis, and H. Bischof. Weighted and robust learning of subspace representations. *Pattern Recognition*, 40(5):1556–1569, 2007.

[30] M. Stetter, H. Greve, C. G. Galizia, and K. Obermayer. Analysis of calcium imaging signals from the honeybee brain by nonlinear models. *Neuroimage*, 13(1):119–128, Jan 2001.

[31] M. Strauch and C. G. Galizia. Registration to a neuroanatomical reference atlas - identifying glomeruli in optical recordings of the honeybee brain. In *Proceedings of GCB, Sep 9-12, 2008, Dresden, Germany*, 2008.

[32] M. Strauch and C. G. Galizia. Fast PCA for processing calcium-imaging data from the brain of Drosophila melanogaster. In *Proceedings of DTMBIO, Oct 24th 2011, Glasgow, Scotland*. ACM, 2011.

[33] M. Strauch, J. Rein, and C. G. Galizia. Signal extraction from movies of honeybee brain activity by convex analysis. In *Proceedings of ICCABS, Feb 23-25 2012, Las Vegas, USA, IEEE*, 2012.

[34] P. Szyszka, C. Demmler, M. Oemisch, L. Sommer, S. Biergans, B. Birnbach, A. F. Silbering, and C. G. Galizia. Mind the gap: Olfactory trace conditioning in honeybees. *The Journal of Neuroscience*, 31(20):7229–7239, 2011.

[35] TILL Photonics. Live acquisition software. Website, 2011. Available online at http://www.till-photonics.com/Products/la.php.

[36] N. Weisz, K. Dohrmann, and T. Elbert. The relevance of spontaneous activity for the coding of the tinnitus sensation. In B. Langguth, G. Hajak, T. Kleinjung, A. Cacace, and A. Møller, editors, *Tinnitus: Pathophysiology and Treatment*, volume 166 of *Progress in Brain Research*, pages 61 – 70. Elsevier, 2007.

[37] J. Weng, Y. Zhang, and W. Hwang. Candid covariance-free incremental principal component analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25:1034–1040, 2003.

[38] R. I. Wilson and Z. F. Mainen. Early events in olfactory processing. *Annual Review of Neuroscience*, 29(1):163–201, 2006.

[39] H. Wold. *Multivariate Analysis*, chapter Estimation of principal components and related models by iterative least squares, pages 391–420. Academic Press, NY, 1966.

[40] J. Yan, B. Zhang, S. Yan, Q. Yang, H. Li, Z. Chen, W. Xi, W. Fan, W.-Y. Ma, and Q. Cheng. IMMC: incremental maximum margin criterion. In *Proceedings of KDD, Seattle, Washington, USA, August 22-25, 2004*, pages 725–730, 2004.

[41] S. Yan and X. Tang. Largest-eigenvalue-theory for incremental principal component analysis. In *International Conference on Image Processing (ICIP), Genoa, Italy, September 11-14, 2005*, pages 1181–1184, 2005.

[42] Y. Zhang and J. Weng. Convergence analysis of complementary candid incremental principal component analysis. Technical report, Comput. Sci. Eng., Michigan State Univ., East, 2001.

[43] H. Zhao, P. C. Yuen, and J. T. Kwok. A novel incremental principal component analysis and its application for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics B*, 36:873–886, 2006.