A strategy for presenting computational models

Maximilian Pohlmann, Lynn Schmittwilken, Marianne Maertens
marianne.maertens@tu-berlin.de, m.pohlmann@tu-berlin.de, l.schmittwilken@tu-berlin.de

Motivation

- computational vision models are invaluable tool to systematize + test our knowledge
- computer programs that simulate the operation of parts of the visual system [1]
- to be useful to scientific community -> open science principles

https://open-science-training-handbook.gitbook.io/book/open-science-basics/open-concepts-and-principles#undefined-4

status quo: patchy publication standards for code

desirable: "literate programming" (Donald Knuth)
- computer program is given an explanation of its logic in a natural language

JOV an ARVO journal ← eLife → RESCIENCE C https://rescience-c.github.io/

"*... publish work of the highest standards ... exploring creative new ways to improve how research is assessed and published ... eLife invests in open-source technology innovation to [...] improve online tools for sharing, using and interacting with new results.*" https://elifesciences.org/about

Goal

- reproduce computational routines in a paper that models some aspect of early vision
- Test case: model of M and P cell kernels with spatial + temporal response properties to predict contrast sensitivity function CSF
- paper from eLife 2019 [2]

> All data generated or analysed during this study are included in the manuscript and supporting files.

reproducibilty
- obtain experimenal result again when experiment is replicated with the same methodology
- document code such that anew computation yields identical results

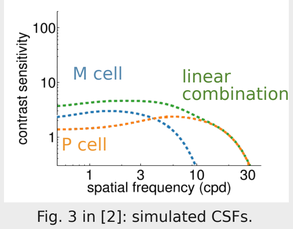
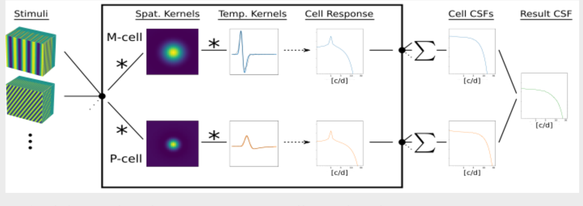


Fig. 3 in [2]: simulated CSFs.

Model overview



** vision scientists not necessarily trained programmers*

Spatial Kernels

$$K(\boldsymbol{f}) = C(K_c \pi r_c^2 e^{-\pi r_c |\gamma \boldsymbol{f}|^2} - K_s \pi r_s^2 e^{-\pi r_s |\gamma \boldsymbol{f}|^2})$$

$\boldsymbol{f}$ - spatial frequency K, r, γ - kernel parameters (obtained from physiological data)

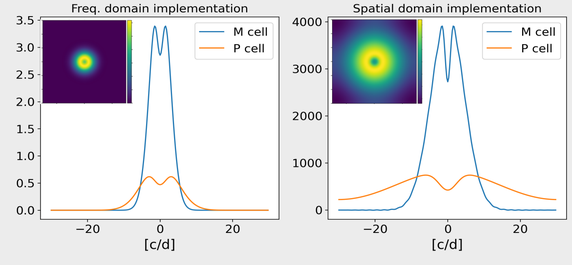
Ambiguities due to implementation decisions

- f: input frequency matrix depends on degree to pixel conversion
- implementation in spatial or frequency domain? Gaussians are Gaussians after Fourier transform

Our implementation

- in frequency domain -> kernels for both cell types (left panel)
- spatial domain implementation by the author in matlab

-> kernels for both cell types + Fourier-transformed them (right)
- results didn't match (we used freq. dom. impl. going forward)



Temporal Kernels

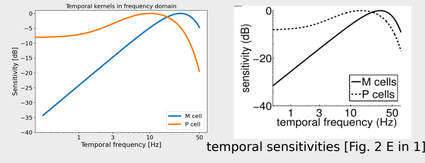
$$H(\omega) = Ae^{-i\rho 2\pi\omega D}\left(1 - \frac{H_s}{1 + i\rho 2\pi\omega\tau_S}\right)\left(\frac{1}{1 + i\rho 2\pi\omega\tau_L}\right)^N$$

ω - temporal frequency N, A, D, H, τ - kernel parameters (obtained from physiological data)

Ambiguities due to conventions

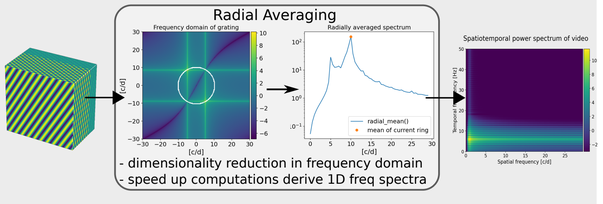
- units of kernel parameters i.e. microseconds or seconds
- amplitude of sensitivity expressed in decibel not obvious for us (convention in signal processing)

Our implementation



temporal sensitivities [Fig. 2 E in 1]

Conceptual steps vs. improving efficiency



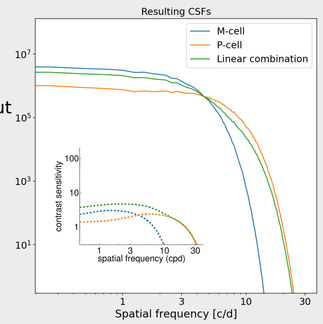
- dimensionality reduction in frequency domain
- speed up computations derive 1D freq spectra

Reproducibility

Summary

- ambiguities in descriptions could not always be resolved
- published code contained plotting routines not computations
- authors provided code upon request -> delays compared to directly downloadable code
- implicit assumptions about what is standard
- conventions (notation in vision science vs. engineering)

- we did not manage to reproduce the CSFs



- no major shortcoming, but the devil is in the details

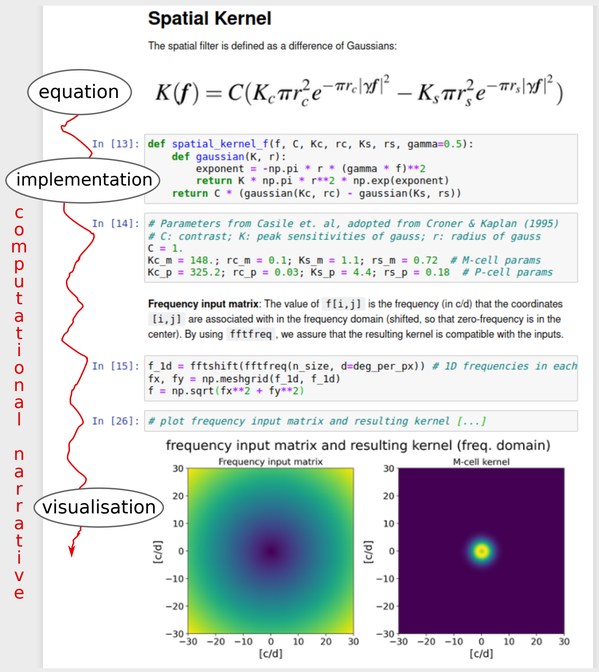
-> time: individual + scientific community

-> frustration

jupyter

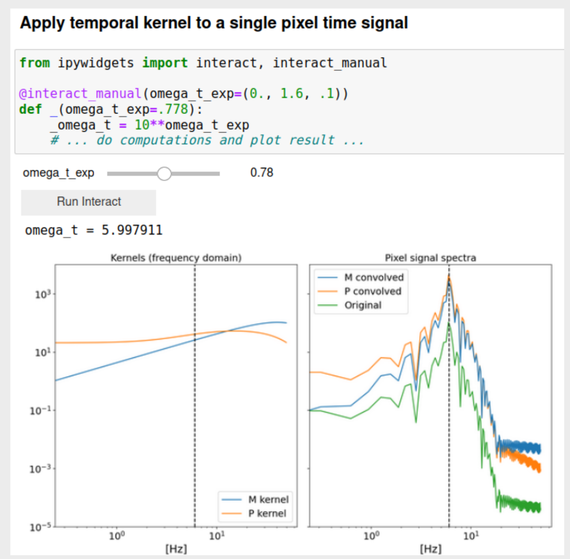
"*[I]n order for ... computations that process ... data to be useful for humans, they must be embedded into a narrative - a computational narrative — that tells a story for a particular audience and context. ... computational narratives ... combine live code, equations, narrative text, interactive user interfaces and other rich media.*" [3]

Notebook Interface



Interactive Visualisations

- exploration of model parameters: how do model parts behave for different parameter values
- visualising multidimensional structures: interactive parameters add another dimension to visualisations



Discussion

- showcase computational models in vision in an accessible and user-friendly way not 'just publish' code
- provide direct link between equations in ***paper*** and functions in ***code*** - **computational narrative**

Recommendations

1. publish code at time of article
2. link paper + code via interactive tools, such as Jupyter or Matlab LiveView
3. open access to all code (+ software i.e. python) is required to reproduce computations in the paper

-> accessible
-> reusable
-> available & free
-> transparent

https://github.com/computational-psychology/A-strategy-for-presenting-computational-models/

References
[1] Landy, M. A. & Movshon, J.A. (1991). Computational models of visual processing. Cambridge, MA: MIT Press.
[2] Casile, A., Victor, J. D., & Rucci, M. (2019). Contrast sensitivity reveals an oculomotor strategy for temporally encoding space. ELife, 8, e40924.
[3] Perez, F., & Granger, B. E. (2015). Project Jupyter: Computational narratives as the engine of collaborative data science. Retrieved September, 11(207), 108.

Acknowledgements
This work was supported by the German Research Foundation (grants MA 5127/3-1, 4-1, 5-1 and EXC 2002/1 "Science of Intelligence" – project number 390523135)