



Outils et bibliothèques de visualisation

Nicolas Rougier

► **To cite this version:**

Nicolas Rougier. Outils et bibliothèques de visualisation. High Performance Computing Magazine France, HPC Médias, 2013, 1 (8), pp.64-65. <hal-00907289>

HAL Id: hal-00907289

<https://hal.inria.fr/hal-00907289>

Submitted on 21 Nov 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Outils et bibliothèques de visualisation

[Nicolas Rougier]

Il existe aujourd'hui énormément d'outils et de bibliothèques susceptibles de nous aider dans le domaine de la visualisation scientifique. Qu'il s'agisse de réaliser une figure pour un article ou de surveiller à l'écran une expérience en temps réel, des outils Open Source adaptés peuvent nous faire gagner un temps précieux. Cependant, même en utilisant ces outils, le scientifique reste responsable de la qualité de sa visualisation, sachant que le chemin entre une bonne et une mauvaise visualisation peut être très court. Aussi, avant même de commencer, il est nécessaire de se poser au préalable un certain nombre de questions:

Pour quelle audience ?

- Vous
- Communauté scientifique
- Etudiants
- Médias / Grand Public

Pour quelle utilisation ?

- Visualisation temps réel
- Figure
- Illustration
- Démonstration

Avec quelles contraintes ?

- Qualité
- Vitesse
- Temps de développement

Sur des données de quelle nature ?

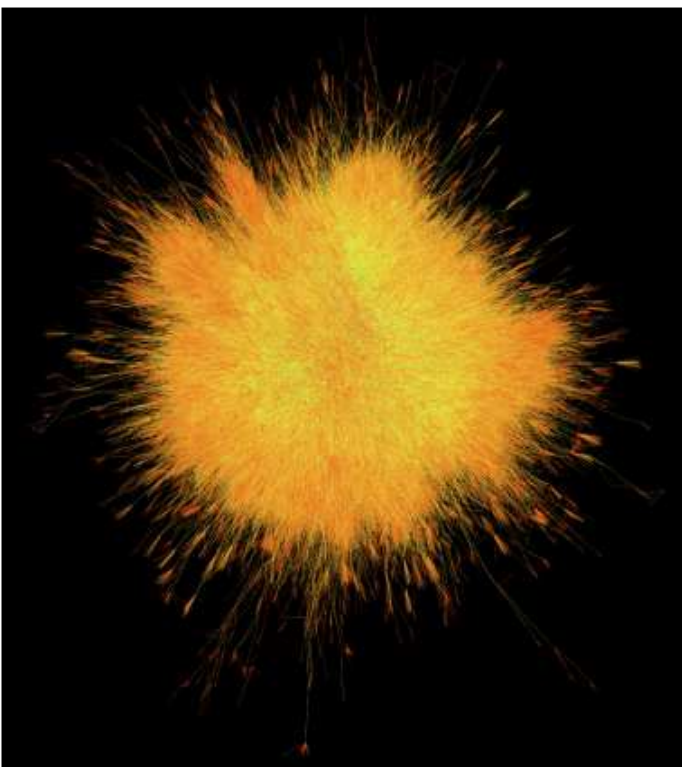
- 2D, 3D, 4D...
- Continues, discrètes...
- Numériques, symboliques...

Une fois le contexte défini, on peut alors choisir dans la très longue liste des outils de visualisation celui qui paraît le mieux adapté aux besoins du moment. Parmi les principaux, on citera :

- [Inkscape](#), outil de dessin vectoriel permettant de travailler des PDF ;
- [The Gimp](#), outil de dessin bitmap permettant de travailler des images ;
- [graphviz](#), qui permet de créer des graphes structurés assez facilement ;
- [tikz](#), qui permet de créer des schémas directement depuis Latex ;
- [d3js](#), pour visualiser des données en ligne de façon interactive ;
- [ImageMagick](#), dédié au traitement d'images à la chaîne ;
- [Matplotlib](#), outil de visualisation pour Python ;
- [ffmpeg](#) pour l'enregistrement, la conversion et le streaming vidéo ;
- [ParaView](#) ou [VisIt](#) pour la visualisation de gros volumes de données ;
- etc.

La **Fig. 6** page suivante montre une description graphique composite réalisée dans le cadre d'une modélisation du cortex somato-sensoriel (neurosciences computationnelles). La moitié haute décrit l'architecture du modèle et se décompose en deux parties. La partie gauche a été conçue à l'aide du logiciel Inkscape, qui permet de manipuler très facilement les dessins de type vectoriel, alors que la partie droite a été réalisée à l'aide du langage tikz, directement accessible

Figure 5 : Représentation d'un graphe par Yifan Hu utilisant l'algorithme *sfdp* de *graphviz*. Ce graphe contient près de 500 000 nœuds et 1 460 000 arêtes.



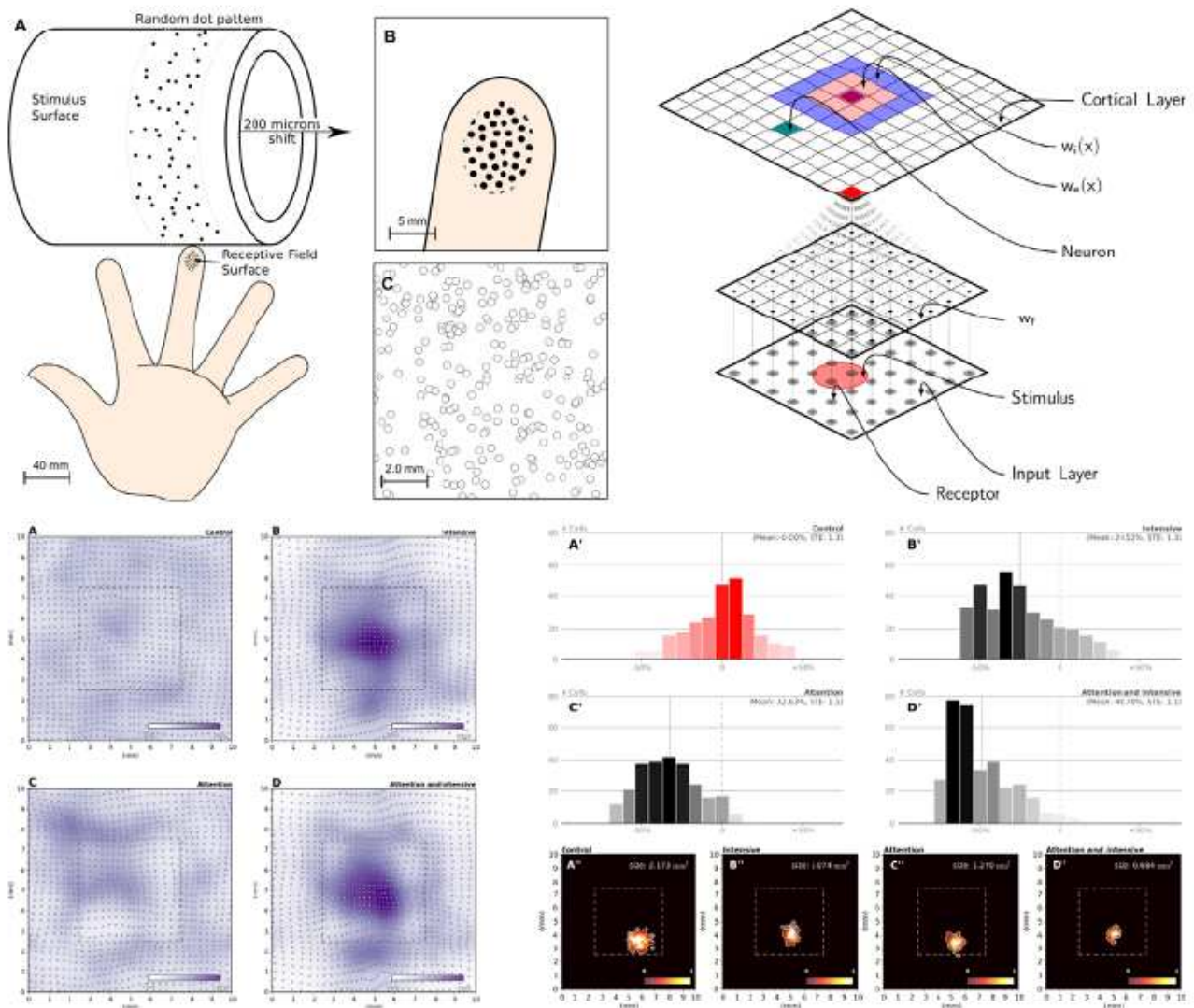


Fig. 6 - En haut, schéma décrivant l'architecture d'un modèle du cortex somato-sensoriel (neuroscience computationnelles). La partie droite de la figure a été réalisée avec Inkscape, la partie gauche avec Tikz. En bas, présentation des résultats du modèle réalisée avec la librairie matplotlib (python).

depuis LaTeX. La moitié basse présente quant à elle les résultats du modèle. Ces résultats sont rendus via matplotlib, qui est devenue la librairie standard de visualisation scientifique en Python. Le collage des sous-figures a été effectué avec la commande pdfnup disponible dans les distributions LaTeX standards. De même, les bords blanc ont été rognés avec la commande pdfcrop.

S'agissant des grandes masses de données, des outils libres tout à fait performants sont également disponibles. Si chacun connaît probable-

ment VisIt et ParaView, qui sait que les algorithmes à l'oeuvre dans graphviz permettent de construire des graphes de 500 000 nœuds et 1 500 000 arêtes (cf. Fig. 5) ? A chaque problème sa solution. Quels que soient nos besoins, il existe très certainement un outil permettant de les satisfaire. Et si tel n'était pas le cas, ce serait l'occasion idéale de contribuer à l'un ou l'autre de ces projets...

Pour aller plus loin : <http://devlog.cnrs.fr/jdev2013/t7>

