Comment retrouver une constellation dans la galaxie ?

Alexis Zubiolo, Kawssar Harb, Michèle Studer, Eric Debreuve, Xavier Descombes

Comment retrouver une constellation dans la galaxie ?

Alexis ZUBIOLO[1], Kawssar HARB[2], Michèle STUDER[2], Éric DEBREUVE[3], Xavier DESCOMBES[4]

[1]Université Nice-Sophia Antipolis, CNRS, Laboratoire I3S, UMR 7271
2000 route des Lucioles, Les Algorithmes, bât. Euclide B, 06900 Sophia Antipolis, France

[2]iBV - Institut de Biologie Valrose, CNRS, UMR 7277, Inserm U1091, Université Nice Sophia Antipolis
Parc Valrose, 06108 Nice, CEDEX 2, France

[3]CNRS, Laboratoire I3S, UMR 7271
2000 route des Lucioles, Les Algorithmes, bât. Euclide B, 06900 Sophia Antipolis, France

[4]Inria Sophia Antipolis Méditerranée
2004 route des Lucioles, BP 93, 06902 Sophia Antipolis CEDEX, France

alexis.zubiolo@i3s.unice.fr, kawssar.harb@unice.fr, michele.studer@unice.fr,
eric.debreuve@cnrs.fr, xavier.descombes@inria.fr

Résumé – Dans cet article, nous proposons une approche pour retrouver un sous-ensemble de points de $\mathbb{R}^n$ dans un ensemble de plus grande taille. L'idée est de mettre en correspondance des groupes d'objets détectés dans des images à différentes résolutions. Pour cela, nous utilisons un modèle de graphe biparti complet dont chacune des arêtes est pondérée par le coût d'un problème d'appariement. La méthode proposée est testée sur des données générées aléatoirement mettant en évidence plusieurs cas particuliers ainsi que sur des cas réel de mise en correspondance de neurones entre deux types d'images 3D de cortex de souris extraites par microscopie confocale ainsi que sur une image du ciel sur laquelle une galaxie est automatiquement détectée.

Abstract – In this paper, we propose an approach to find a subset of n-dimensional points within a larger set. The idea is to match objects detected in images at different resolutions. To do so, we use a complete bipartite graph model whose vertices are weighted by the cost of assignment problems. The proposed method is tested on randomly generated data emphasizing on some particular cases as well as on real data (matching neurons in two different types of 3D microscopic images of mice cortices and automatically detecting a constellation in an image of the night sky).

1 Introduction

La mise en correspondance d'informations issues de sources différentes est un problème classique en traitement du signal et de l'image. Par exemple, de nombreuses méthodes d'extraction de descripteurs (comme les SIFT [1]) ou encore de recalage d'images [2] répondent à ces défis dans divers contextes.

Dans ce papier, nous présentons une méthode permettant de mettre en correspondance des points de $\mathbb{R}^n$ issus d'un ensemble d'une part, et d'un sous-ensemble de cet ensemble d'autre part, dans le but d'apparier des neurones présents sur deux types d'images 3D et de détecter une constellation dans une image du ciel.

Pour ce faire, la méthode proposée se fonde sur des distances entre paires de points, ce qui permet d'obtenir une propriété d'invariance par translation et par rotation. Nous proposons donc de construire un graphe biparti complet entre les points des deux ensembles. Le poids d'une arête entre deux points de ce graphe sera lui-même défini comme le coût de l'appariement des distances de ces points aux autres points des ensembles auxquels ils appartiennent.

2 Méthode proposée

2.1 Formalisation

Soit $\mathcal{X} = \{x_i, 1 \leq i \leq n\}$ et $\mathcal{Y} = \{y_j, 1 \leq j \leq N\}$ (avec $n \leq N$) deux ensembles munis d'une même distance d. Le problème est de trouver une fonction de correspondance (de $\mathcal{X}$ vers $\mathcal{Y}$) $\phi : [\![1, n]\!] \to [\![1, N]\!]$. Nous proposons de minimiser la fonction objet

$$\sum_{1 \leq i_1 < i_2 \leq n} \left| d(x_{i_1}, x_{i_2}) - d(y_{\phi(i_1)}, y_{\phi(i_2)}) \right|$$

en ayant recours à des outils d'appariement de nœuds dans un graphe biparti. Soit donc un graphe biparti complet $\mathcal{G} = (\mathcal{X}, \mathcal{Y}, \mathcal{E})$ où $\mathcal{E} = (e_{ij})_{(i,j) \in [\![1,n]\!] \times [\![1,N]\!]}$ est une matrice d'arêtes pondérées (que l'on va définir par la suite) entre les points de $\mathcal{X}$ et ceux de $\mathcal{Y}$.

Chaque point $x_i \in \mathcal{X}$ (respectivement $y_j \in \mathcal{Y}$) peut être décrit par une liste de distances aux autres points de $\mathcal{X}$ (respectivement de $\mathcal{Y}$) $d_i^{\mathcal{X}} = (d_{ii'}^{\mathcal{X}})_{1 \leq i' \leq n} = (d(x_i, x_{i'}))_{1 \leq i' \leq n}$ (respectivement $d_j^{\mathcal{Y}} = (d_{jj'}^{\mathcal{Y}})_{1 \leq j' \leq N} = (d(y_j, y_{j'}))_{1 \leq j' \leq N}$).

Soit $(x_i, y_j) \in \mathcal{X} \times \mathcal{Y}$. Si x_i et y_j se correspondent, alors leurs listes de distances respectives $d_i^{\mathcal{X}}$ et $d_j^{\mathcal{Y}}$ ont des valeurs communes (ou relativement proches). Nous proposons donc d'envisager un problème d'appariements des distances. Soit $M^{ij} = (m^{ij}_{i'j'})_{(i',j') \in [\![1,n]\!] \times [\![1,N]\!]}$ la matrice solution de ce problème d'appariement (c'est-à-dire $m^{ij}_{i'j'} = 1$ si $d^{\mathcal{X}}_{ii'}$ et $d^{\mathcal{Y}}_{jj'}$ sont égales ou très proches et 0 sinon). On définit alors le poids e_{ij} entre x_i et y_j comme étant le coût optimal du problème d'optimisation linéaire suivant :

$$e_{ij} = \min_{M^{ij} \in \mathcal{M}_{n,N}(\{0,1\})} \sum_{i'=1}^{n} \sum_{j'=1}^{N} \left| d^{\mathcal{X}}_{ii'} - d^{\mathcal{Y}}_{jj'} \right| m^{ij}_{i'j'} \qquad (\mathrm{P}^{ij})$$

$$\text{avec} \qquad \forall i' \in [\![1,n]\!], \sum_{j'=1}^{n} m^{ij}_{i'j'} = 1 \qquad (1)$$

$$\text{et} \qquad \forall j' \in [\![1,N]\!], \sum_{i'=1}^{n} m^{ij}_{i'j'} \leq 1 \qquad (2)$$

où la contrainte (1) signifie que chaque distance de $d_i^{\mathcal{X}}$ correspond exactement à une distance de $d_j^{\mathcal{Y}}$, (2) signifie que chaque distance de $d_j^{\mathcal{Y}}$ correspond à au plus une distance de $d_i^{\mathcal{X}}$, et (P^{ij}) signifie que l'ont souhaite minimiser la somme des différences entre les distances correspondantes. Si e_{ij} est proche de 0, cela veut dire que les ensembles de distances $d_i^{\mathcal{X}}$ et $d_j^{\mathcal{Y}}$ se correspondent et que l'hypothèse $j = \phi(i)$ est raisonnable. Sinon, cela veut dire qu'il existe au moins une distance de $d_i^{\mathcal{X}}$ qui n'apparaît pas dans $d_j^{\mathcal{Y}}$, ce qui impliquerait $j \neq \phi(i)$.

Une fois que tous les e_{ij} sont calculés, l'ensemble des arêtes $\mathcal{E}$ est défini et la correspondance ϕ est le résultat de l'appariement du graphe biparti $\mathcal{G}$.

2.2 Résolution des problèmes d'appariements

L'algorithme hongrois [3] est une méthode classique d'appariement de graphes bipartis. Or, ce dernier est initialement conçu pour le cas où $\mathcal{X}$ et $\mathcal{Y}$ sont de même cardinal. Il est généralisable à des ensembles $\mathcal{X}$ et $\mathcal{Y}$ de tailles quelconques en ajoutant à la matrice de coût $C^{ij} = (|d^{\mathcal{X}}_{ii'} - d^{\mathcal{Y}}_{jj'}|)_{(i',j') \in [\![1,n]\!] \times [\![1,N]\!]}$ des valeurs fictives plus grandes que sa valeur maximale de sorte à la rendre carrée. En revanche, ce procédé augmente inexorablement la taille du problème. Dans notre cas où $n \leq N$, chaque C^{ij}, initialement de taille $n \times N$, deviendrait de taille $N \times N$ au moment de l'appel de l'algorithme hongrois. Ceci peut poser problème lorsque la différence entre n et N devient importante. Ce phénomène sera illustré dans la section 3.

Nous privilégions donc, dans ce cas, une résolution différente. Il est en effet possible de reformuler P^{ij} de sorte à obtenir un problème d'optimisation linéaire en variables entières équivalent et dont la matrice de contraintes est totalement unimodulaire [4]. Sa solution étant par conséquent un des sommets du polyèdre délimité par l'ensemble de ses contraintes, le choix du simplexe semble approprié puisqu'il apportera la solution optimale.

2.3 Cas d'ambiguïté

Dans la description faite en section 2.1, nous partons du principe que si $x_i \in \mathcal{X}$ et $y_j \in \mathcal{Y}$ se correspondent, alors nous retrouverons dans $d_j^{\mathcal{Y}}$ des valeurs proches de celles de $d_i^{\mathcal{X}}$. La réciproque est fausse, notamment dans le cas où le motif $\mathcal{X}$ est répété plusieurs fois dans $\mathcal{Y}$.

Cependant, c'est un phénomène rare (et d'autant plus rare que $|\mathcal{X}|$ est grand), et quand bien même il se produirait, la méthode proposée est capable de le détecter. En effet, dans un tel cas, il existerait $m \geq 2$ motifs $\mathcal{X}_1, \cdots, \mathcal{X}_m$ de même cardinal pour lesquels $d^{\mathcal{X}}_{i_1}, \cdots, d^{\mathcal{X}}_{i_m}$ ont des valeurs similaires à celles de $d_j^{\mathcal{Y}}$ pour un certain $(i_1, \cdots, i_m, j) \in [\![1,n]\!]^m \times [\![1,N]\!]$, si bien que les pénalités respectives $e_{i_1 j}, \cdots, e'_{i_m j}$ seraient faibles. On en déduirait alors les ensembles $\mathcal{X}_1, \cdots, \mathcal{X}_m$ des matrices $M^{i_1 j}, \cdots, M^{i_m j}$ des problèmes $\mathrm{P}^{i_1 j}, \cdots, \mathrm{P}^{i_m j}$, comme l'illustre la section 3.3.

3 Applications

Dans cette section, nous testons l'algorithme proposé sur des données générées (éventuellement bruitées ou présentant des ambiguïtés évoquées dans la section 2.3), et nous l'illustrons sur des données réelles de microscopie confocale tridimentionelles et une image du ciel dans laquelle on retrouve une constellation (Section 3.4). Par ailleurs, bien que l'utilisation d'une autre distance puisse être envisagée, nous nous limiterons au cas où d est la distance euclidienne sur $\mathbb{R}^k$ (où $k \in \mathbb{N}^\star$). L'algorithme a été implémenté sous MATLAB en utilisant `linprog` pour la résolution du problème du simplexe.

3.1 Données simulées non bruitées

Nous générons un ensemble $\mathcal{X}$ de n points en dimension 2 dont chacune des composantes suit une loi uniforme sur $[0,1]$, puis un ensemble $\mathcal{Y}$ composé des points de $\mathcal{X}$ et de $N-n$ autres points eux aussi générés selon une loi uniforme sur $[0,1]^2$. Nous comparons dans la figure 1 les temps de calculs mis par la méthode en ayant recours à l'algorithme hongrois d'un côté, et au simplexe de l'autre.

On constate effectivement que le simplexe devient plus intéressant lorsque la différence entre n et N devient trop importante ($n \leq N/5$ sur la plage de données testée). L'algorithme hongrois, lui, l'emporte largement lorsque n s'approche de N.

3.2 Données simulées bruitées

Nous générons d'abord les ensembles $\mathcal{X}$ et $\mathcal{Y}$ de la même façon qu'en 3.1. Nous fixons $N = 100$ et étudions plusieurs valeurs de $n : \{5, 10, 15, 20\}$. Nous appliquons ensuite un bruit additif de loi uniforme sur $[-\epsilon, +\epsilon]$ (avec $\epsilon \in [0, 10^{-1}]$) sur chaque composante des n premiers vecteurs de $\mathcal{Y}$ (celles correspondant aux points de $\mathcal{X}$). Nous souhaitons étudier la robustesse de la méthode proposée en suivant l'évolution de le taux de réussite de l'algorithme en fonction de ϵ. Le taux de réussite

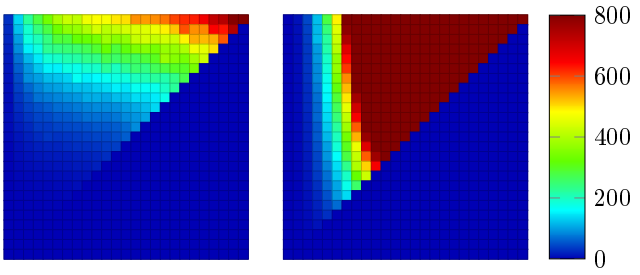

FIGURE 1 – Temps de calcul (en secondes) de la résolution du problème d'appariment pour $1 \leq n \leq N \leq 100$ (avec un pas de discrétisation de 4 pour n et N) pour l'algorithme hongrois (à gauche), et pour le simplexe (à droite). Pour chaque graphique, l'axe des abscisses correspond à n et l'axe des ordonnées à N. La partie triangulaire inférieure droite est forcée à 0 parce qu'elle correspond au cas où $n > N$. Nous avons forcé la valeur maximale de temps de calcul à 800 secondes pour que le graphe correspondant à l'algorithme hongrois soit lisible, ce qui explique la saturation du graphe correspondant au simplexe. En réalité, le simplexe met plusieurs heures pour le cas $n = N = 100$.

est simplement la moyenne des rapports entre le nombre de points de $\mathcal{X}$ correctement appariés avec leurs correspondants dans $\mathcal{Y}$ et n sur 50 réalisations de bruit. Les résultats sont illustrés dans la figure 2.

On constate que la robustesse de la méthode augmente avec n, ce qui est dû au fait que plus n est grand plus le motif $\mathcal{X}$ sera difficile à reproduire dans $\mathcal{Y}$.

3.3 Données simulées présentant des ambiguïtés

Nous nous intéressons au cas d'ambiguïté évoqué dans la section 2.3. Pour cela, nous générons un ensemble $\mathcal{X}_1$ de n points aléatoires selon une distribution uniforme sur $\left[0, \frac{1}{2}\right]^2$, un ensemble $\mathcal{X}_2$, image de $\mathcal{X}_1$ par une translation de vecteur $(\frac{1}{2}, \frac{1}{2})$ puis d'une rotation de centre $(\frac{3}{4}, \frac{3}{4})$ et d'angle $\frac{\pi}{2}$. $\mathcal{Y}$ est alors défini comme étant la réunion de $\mathcal{X}_1 \cup \mathcal{X}_2$ et de $N - 2n$ points générés aléatoirement selon une loi uniforme sur $[0, 1]^2$.

On constate alors que dans la matrice d'adjacence du graphe, pour chaque colonne (c'est-à-dire pour chaque $x \in \mathcal{X}$), deux valeurs sont nulles, alors que les autres sont strictement positives. L'algorithme peut alors renvoyer une alerte pour indiquer qu'un cas d'ambiguïté a été détecté.

3.4 Applications et illustrations

3.4.1 Mise en corresondance de données multi-résolution

Nous appliquons l'algorithme proposé à des images 3D de cortex de souris acquises par microscopie confocale (ZEISS LSM 710), et dont les neurones sont mis en évidence par fluorescence grâce à la protéine Thy1-eYFP-H [5]. Nous disposons de plusieurs ensembles d'images, chacun étant composé de deux types d'images : une image au zoom x10 montrant

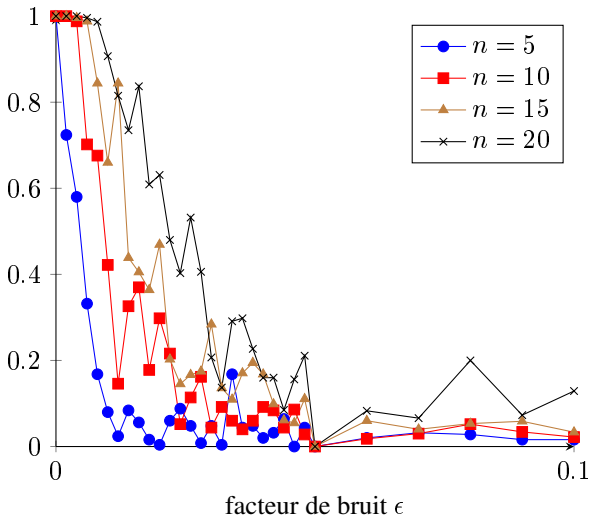

FIGURE 2 – Taux de réussite en fonction du facteur de bruit ϵ, pour $N = 100$ et différentes valeurs de n.

les neurones dans leur ensemble et quelques images au zoom x40 focalisées sur le corps cellulaire des neurones (comme le montre la figure 3). Nous souhaitons calculer des paramètres descriptifs de la morphologie des neurones sur les deux types d'images dans le but d'en faire une étude statistique. Nous avons donc besoin de retrouver les neurones de l'image 40x dans l'image 10x pour combiner les caractéristiques extraites. Le but étant de mettre en correspondance les neurones et non les images, il semble plus judicieux d'éviter le recalage des images beaucoup plus coûteux en temps de calcul.

Nous proposons de représenter chaque neurone par le barycentre de son corps cellulaire. Pour ce faire, les corps cellulaires des neurones sont dans un premier temps segmentés en appliquant à l'image un seuillage par hystérésis suivi d'outils de morphologie mathématique [6]. L'extraction du barycentre de chacune des composantes connexes résultantes donne une idée suffisamment précise de la position du neurone. Dès lors, l'algorithme proposé peut établir une correspondance entre les barycentres (et par conséquent entre les neurones) sur les deux types d'image. Le résultat est illustré par la figure 3. Il est important de noter que dans cet exemple, les distances sont exprimées en µm et non pas en pixels, puisque l'échelle est différente entre les deux types d'images. Les éventuelles erreurs commises lors du calcul du barycentre du corps cellulaire du neurone constituent un bruit suffisamment faible dans la mesure où tous les neurones présents sur nos images ont été correctement appariés.

3.4.2 Détection d'une constellation dans une image du ciel

Pour cette dernière illustration, nous proposons de détecter une constellation d'étoiles dans une photo du ciel. Nous partons donc d'une image du ciel sur laquelle apparaît la Grande Ourse. Nous détectons à la main les étoiles de la constellation pour en déduire les distances entre paires d'étoiles. Après seuillage de

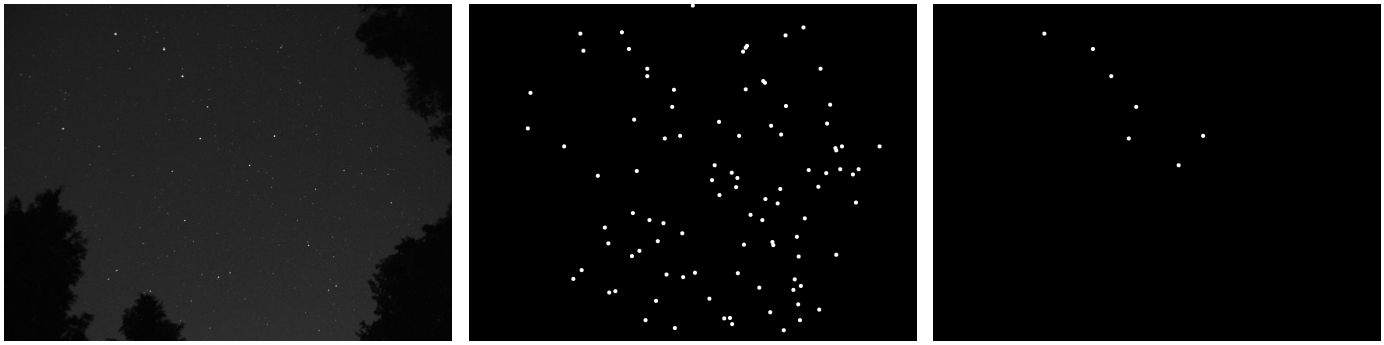

FIGURE 4 – L'image initiale (à gauche), les barycentres des 95 composantes connexes de l'image obtenues après seuillage (au milieu), et la constellation détectée en 25 secondes par l'algorithme (à droite).

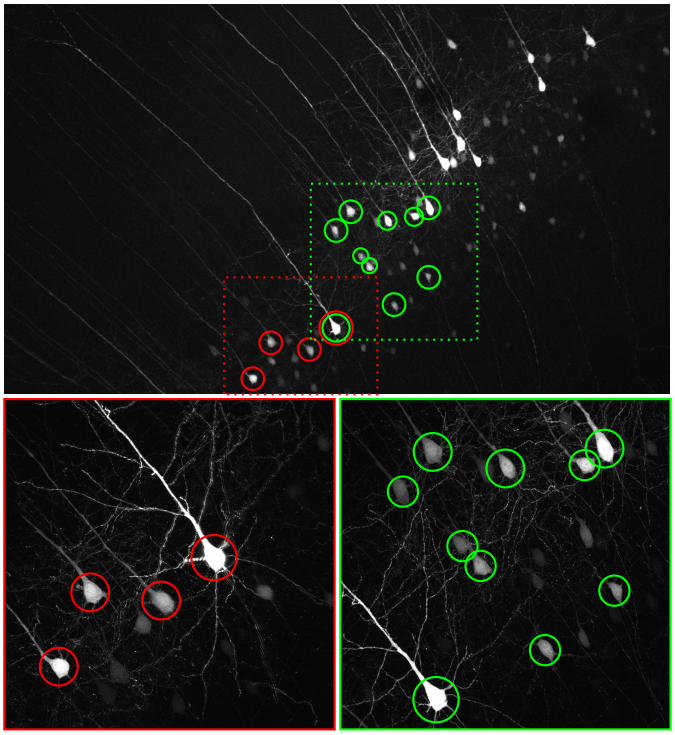

FIGURE 3 – Illustration de la carte de correspondance sur la projection du maximum d'intensité d'images 3D de microscopie confocale. Les têtes des neurones correspondants sont entourées de la même couleur sur l'image 10x (en haut) que sur les images 40x (en bas).

l'image, les 95 étoiles les plus lumineuses (ainsi que du bruit) sont segmentées. Nous appliquons donc l'algorithme pour qu'il cherche la Grande Ourse (définie par le tableau des distances calculé précédemment) parmi les 95 étoiles segmentées. Les résultats apparaissent dans la figure 4.

Il est important de noter que dans cette application, l'algorithme proposé n'a pas eu recours à l'information d'intensité des étoiles, mais s'est juste basé sur les coordonnées de leur centre de gravité.

4 Conclusion

Dans cet article, nous proposons une méthode permettant d'apparier des points correspondants tirés de deux ensembles différents en se basant sur les distances entre paires de points.

Il est possible d'améliorer le temps de calcul de l'algorithme en diminuant le nombre de problèmes de programmation linéaire résolus. Pour chaque $(i, j) \in [\![1, n]\!] \times [\![1, N]\!]$, on peut en effet ignorer les valeurs de $d_j^{\mathcal{Y}}$ supérieures à $\max d_i^{\mathcal{X}}$ (puisque de telles valeurs correspondent à des points de $\mathcal{Y}$ plus éloignés que le point le plus loin de x_i dans $\mathcal{X}$) et ainsi associer une valeur arbitrairement grande au e_{ij} correspondant.

Au delà des améliorations algorithmiques possibles, il est naturel de se demander si l'algorithme peut être étendu au cas où le facteur d'échelle est inconnu.

Références

[1] W. Pratt. *Correlation techniques of image registration*. Aerospace and Electronic Systems, 1974.

[2] D.G. Lowe. *Distinctive image features from scale-invariant keypoints*. International journal of computer vision, 2004.

[3] H.W. Kuhn. *The Hungarian method for the assignment problem*. Naval research logistics quarterly, 1955.

[4] A. Schrijver. *Combinatorial optimization : polyhedra and efficiency*. Springer Science & Business Media, 2003.

[5] G. Feng, R.H. Mellor, M. Bernstein, C. Keller-Peck, Q.T. Nguyen, M. Wallace, J.M. Nerbonne, J.W. Lichtman et J.R. Sanes. *Imaging Neuronal Subsets in Transgenic Mice Expressing Multiple Spectral Variants of GFP*. Neuron, 2000.

[6] A. Zubiolo, K. Harb, M. Studer, E. Debreuve, X. Descombes. *Morphological Analysis and Feature Extraction of Neurons from Mouse Cortices Multiscale 3D Microscopic Images*. International Conference of the IEEE Engineering in Medicine and Biology Society, 2015.