


Programmation différentiable pour données spatiales

JT Sexton



Yann LeCun 

5 janvier 2018 · 



OK, Deep Learning has outlived its usefulness as a buzz-phrase.
Deep Learning est mort. Vive Differentiable Programming!

Plan de match

0) Concepts préliminaires

1) Étude de cas: algorithme différentiable pour segmenter une pièce industrielle dans un nuage de points

2) Structures de données spatiales différentiables

3) Perspectives: différentier à travers l'optimisation

4) Conclusion

Définition «DeepMind» (Elements of Differentiable Programming)

Definition 1.1 (Differentiable programming). Differentiable programming is a programming paradigm in which complex computer programs (including those with control flows and data structures) can be differentiated end-to-end automatically, enabling gradient-based optimization of parameters in the program.

La clef: Vector Jacobian Product (LIVE!)

“Mathematics is a game played according to certain simple rules with meaningless marks on paper.”
- David Hilbert



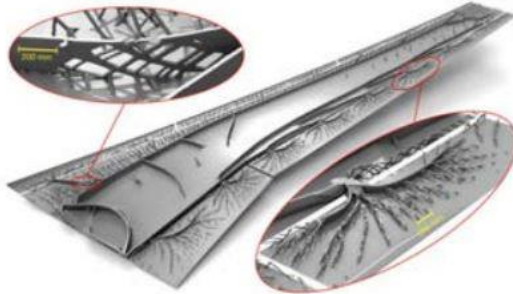
Quelques exemples

- Tirés du cours MIT 18.S096 Matrix Calculus for Machine Learning and Beyond (disponible sur youtube/OCW, chaudement recommandé), utilisation à fin éducatives (fair use)

Applications: Physical Problems

Topology-optimized aircraft wing

~ 10^9 parameters

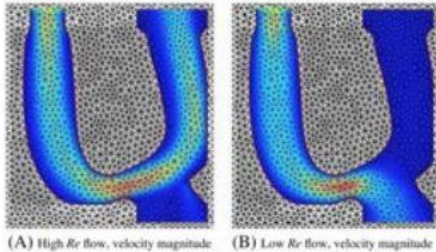


Aage, Niels, et al. "Giga-voxel computational morphogenesis for structural design." *Nature* 550.7674 (2017): 84-86.

Goal: maximize stiffness under external loads, utilizing limited amount of material
→ Light but strong
→ 100s tonnes of fuel saving

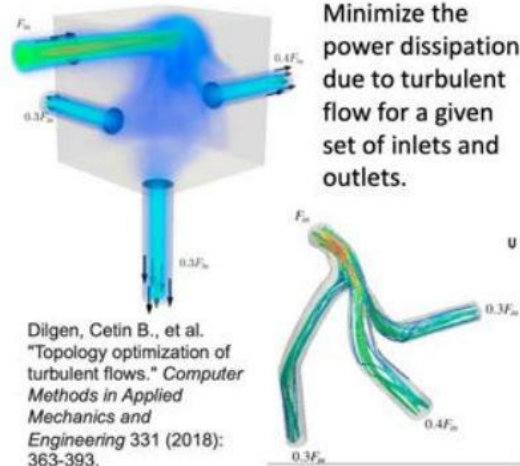
© Springer Nature Limited. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <https://ocw.mit.edu/help/faq-fair-use>.

Topology optimization with fluid dynamics



(A) High Re flow, velocity magnitude (B) Low Re flow, velocity magnitude
Switching flow channels for high vs. low viscosity

Zhou, Mingdong, et al. "Shape morphing and topology optimization of fluid channels by explicit boundary tracking." *International Journal for Numerical Methods in Fluids* 88.6 (2018): 296-313.



Minimize the power dissipation due to turbulent flow for a given set of inlets and outlets.

Dilgen, Cetin B., et al. "Topology optimization of turbulent flows." *Computer Methods in Applied Mechanics and Engineering* 331 (2018): 363-393.

Engineering optimization (structural "topology" optimization):

Find the physical structure that optimizes some objective (e.g. focusing light, minimizing drag, supporting weight, ...).



Topology-optimized 3D-printed hip replacement (Altair)

© Altair Engineering Inc. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <https://ocw.mit.edu/help/faq-fair-use>.



Topology-optimized 3D-printed seat bracket (General Motors)

© General Motors. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <https://ocw.mit.edu/help/faq-fair-use>.

Key point is that if you **have any complicated calculation** with lots of parameters, you can compute **gradient** (sensitivity) of a scalar output $g(u)$ with **respect to every parameter** with roughly **one additional calculation**.

Enabling factor for large-scale optimization in machine learning [g = loss function, u = network outputs, p = network weights & other parameters], statistics, finance, and **many other fields**.

© John Wiley & Sons, Inc. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <https://ocw.mit.edu/help/faq-fair-use>.

Courtesy Elsevier, Inc., <http://www.sciencedirect.com>.
Used with permission.

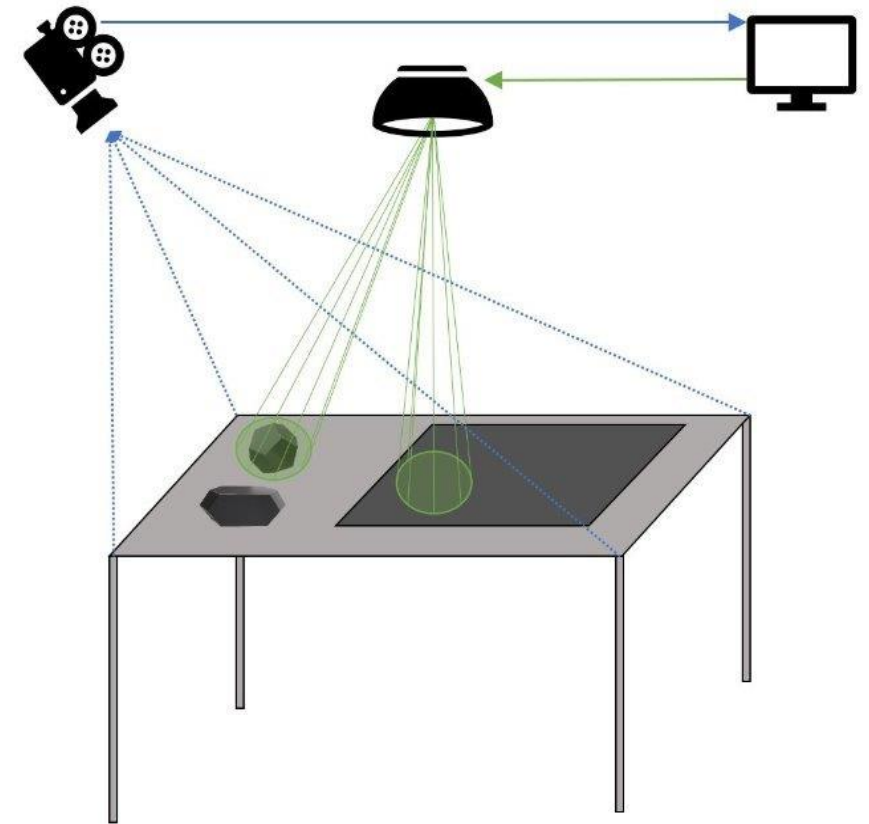
Le cas MECART

- Entreprise qui fabrique des salles blanches modulaires en acier
- Tout est fait sur mesure, tout est soudé
- Les soudeurs passent moins de 5% de leur temps à souder (!)



Assistance aux soudeurs via réalité augmentée non invasive

- Qu'est-ce que la réalité augmentée non invasive?
 - Information projetée sans matériel encombrant ou dérangeant (lunette, casque, etc.)
- Pourquoi la réalité augmentée non invasive?
 - Gains potentiels (rapidité dans la préparation et l'exécution, éviter les erreurs...)
 - Possibilité de projeter des instructions de montage
 - Possibilité de mise à jour si la pièce sur le plan de travail n'est pas la bonne



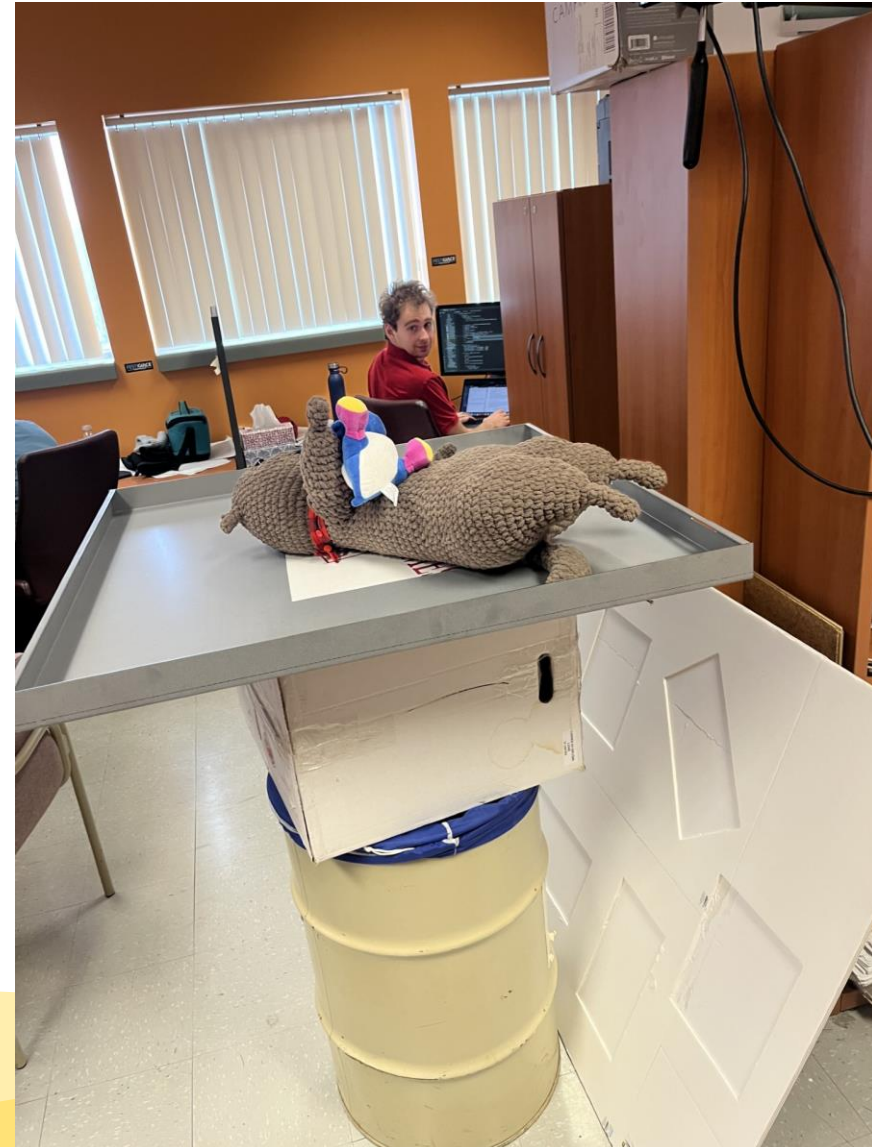
Qu'est-ce qu'on cherche à faire?

- Dans un premier temps, segmenter le panneau dans un nuage de points pour, plus tard, faire rapidement un ICP
- Va nous donner la position 6D du panneau en cours, nous permettant d'afficher les instructions pertinentes
- Temps réel est nécessaire
- Il y aurait plusieurs façons de faire, mais amusons-nous et allons-y par programmation différentiable!

Ce qu'on sait

- Dispose d'un nuage de points ($\sim 250\,000$ pts à 30 Hz)
- Doit faire la segmentation en moins d'un frame (< 0.03 secondes)
- Panneau est gros
- Panneau est // à la table (et je connais la normale de la table)
- Table bouge! (en dedans d'un certain intervalle)
- Occlusion!

Contexte expérimental



Idée générale

- Approximer le panneau par un plan (via régression linéaire)
- Optimiser itérativement des poids sur chaque points afin qu'une régression linéaire nous donne le plan sur lequel se trouve le panneau
- Il va falloir bien concevoir une fonction objectif selon nos contraintes géométriques (panneau est gros, panneau est // au sol, panneau est dans un certain intervalle)
- Filtrer les points proches du plan résultant pour obtenir le panneau

Faisons la fonction objectif

"For since the fabric of the universe is most perfect and the work of a most wise Creator, nothing at all takes place in the universe in which some rule of maximum or minimum does not appear."

- Leonhard Euler



Victoire?

- Non!
- C'est beaucoup trop lent
- Il faudrait que ça soit plus de dix fois plus vite!
- On pourrait sous-échantillonner le nuage de points, utiliser un ordi plus récent, etc.
- Mais ayons du plaisir et faisons une implémentation CUDA

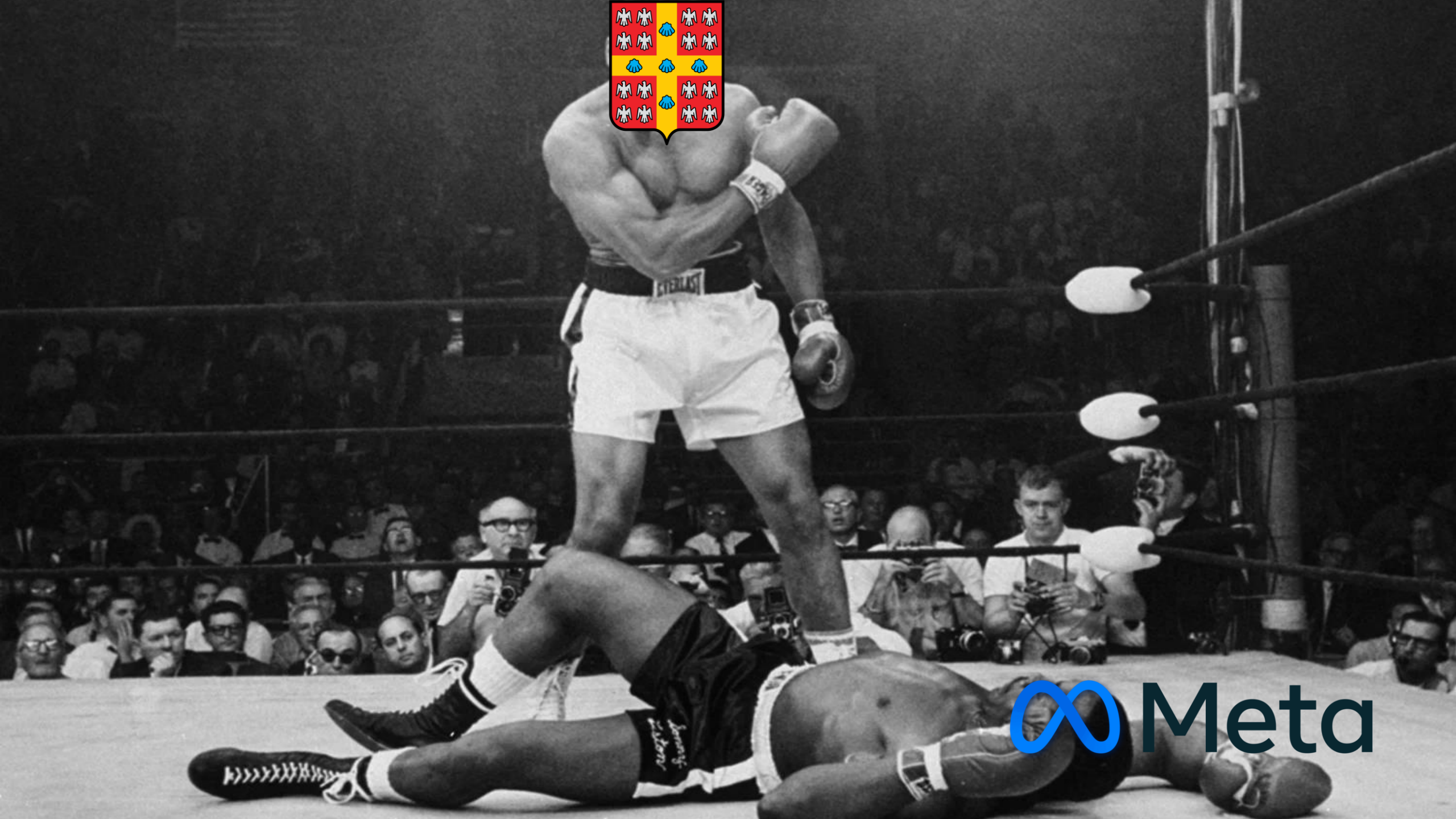
Quiz: Vitesse atteinte (ULaval (moi jeudi passé) v.s. Des années de Meta)

- a) Meta est au moins deux fois plus vite que nous
- b) Meta est autour de 20% plus vite que nous
- c) Égalité
- d) Nous sommes autour de 20% plus vite que Meta
- e) Nous sommes plus de deux fois plus vite que Meta

Un peu d'échelle



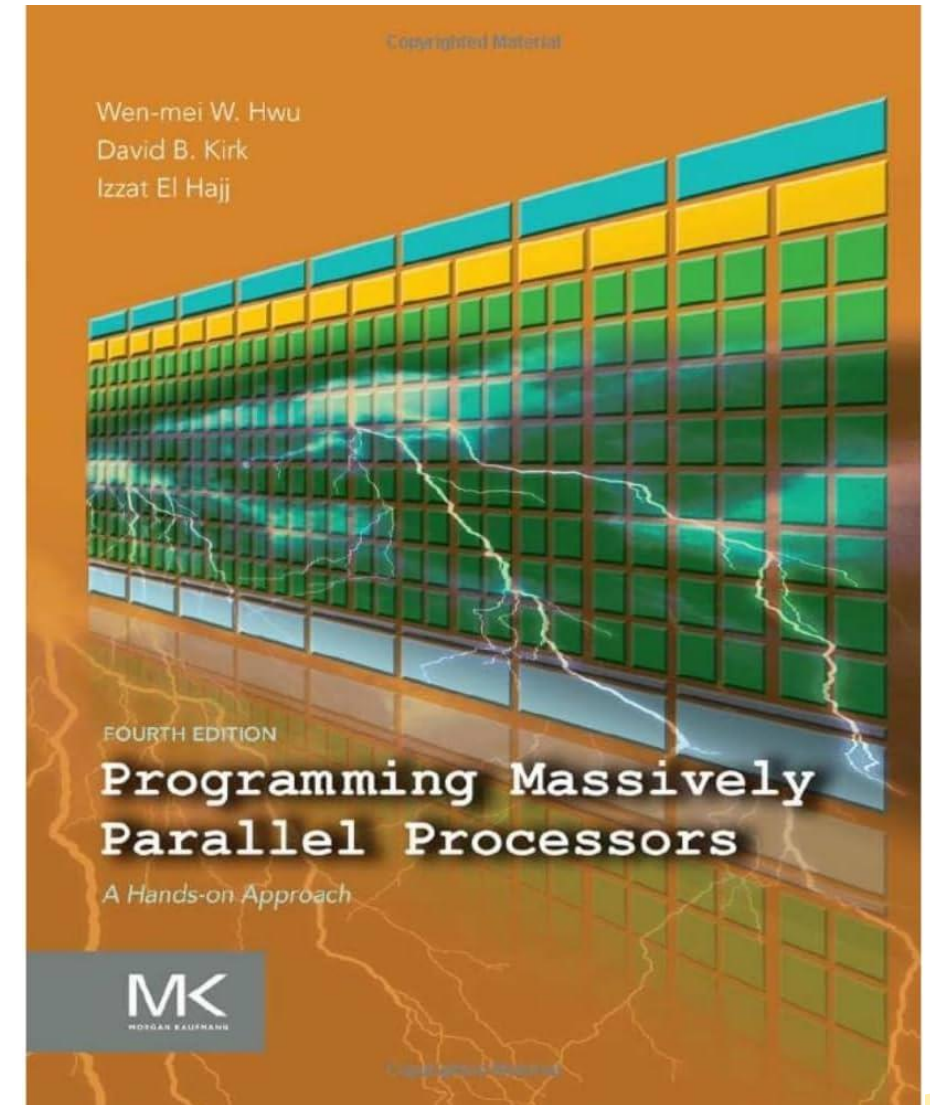




 Meta

C'était une fausse dichotomie

- Pytorch est un excellent outil de prototypage 😊
- Je ne comprends sincèrement pas pourquoi la très forte majorité des programmeurs ne s'intéresse pas aux ordinateurs 😞

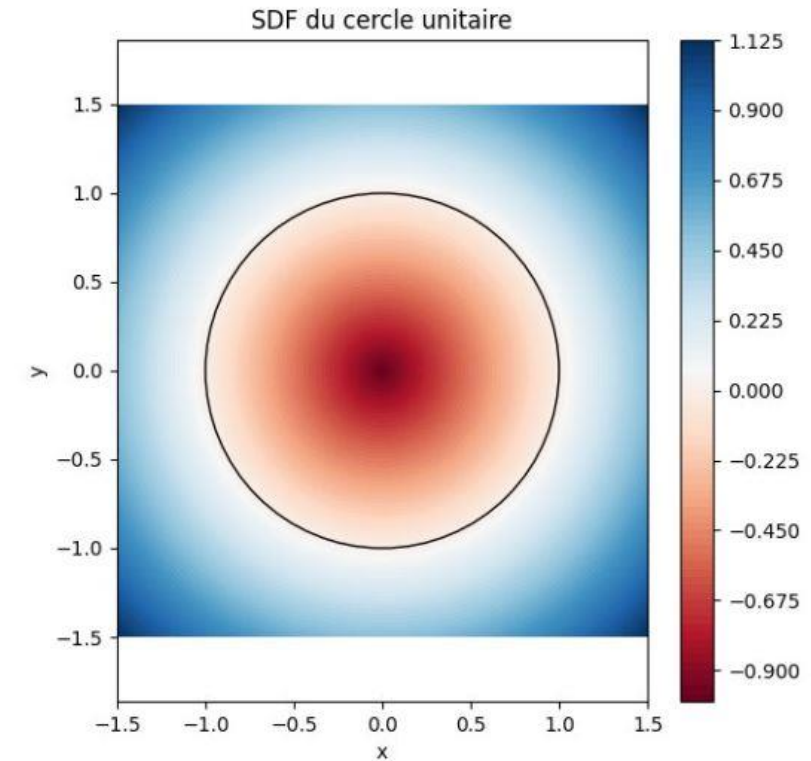


En résumé

- Écrire notre programme de façon différentiable nous a permis non-seulement de résoudre notre problème, mais de résoudre une catégorie de problèmes
- Notre programme différentiable peut servir de module dans un programme plus large
- Par exemple, utiliser un réseau de neurones en aval de notre programme pour prédire la position 6D du panneau (potentiellement comme guess initial pour un ICP)
- À l'entraînement, notre programme et le réseau de neurones serait optimisés conjointement à partir de données

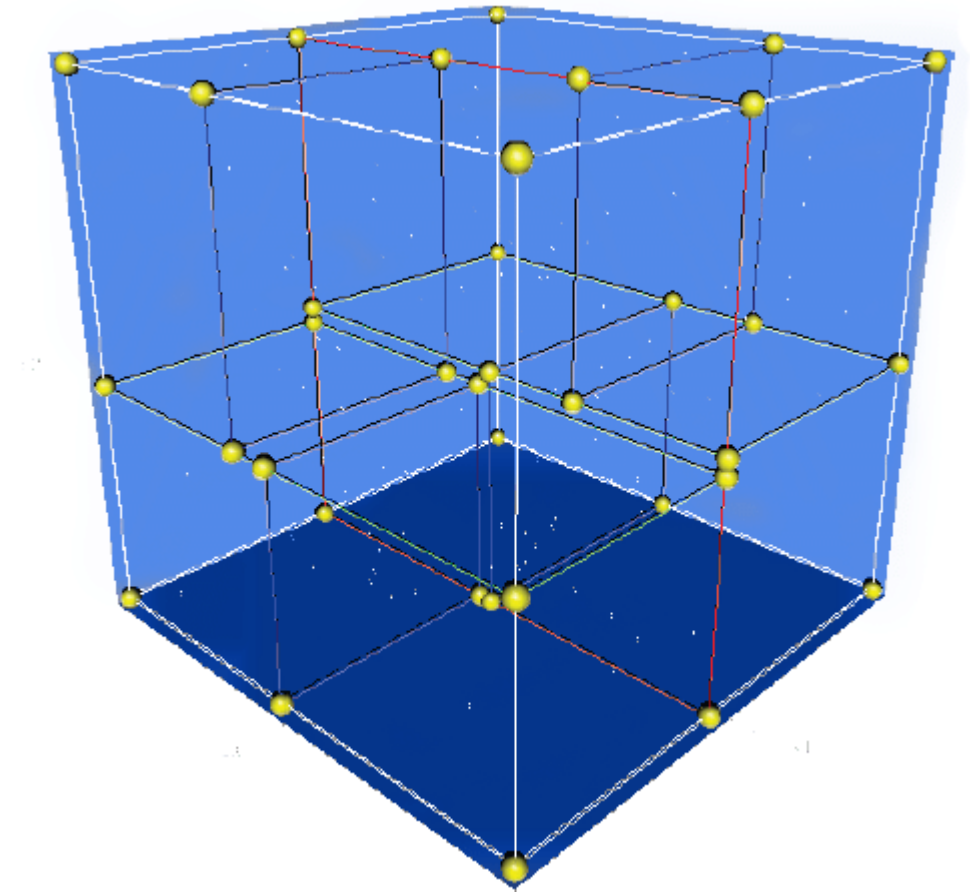
Structures de données différentiables pour données spatiales

- On en a déjà fait une!
- «Couche pleinement connectée d'un réseau de neurones» est juste une matrice
- Mais c'est quoi la différence entre une fonction est une structure de donnée?



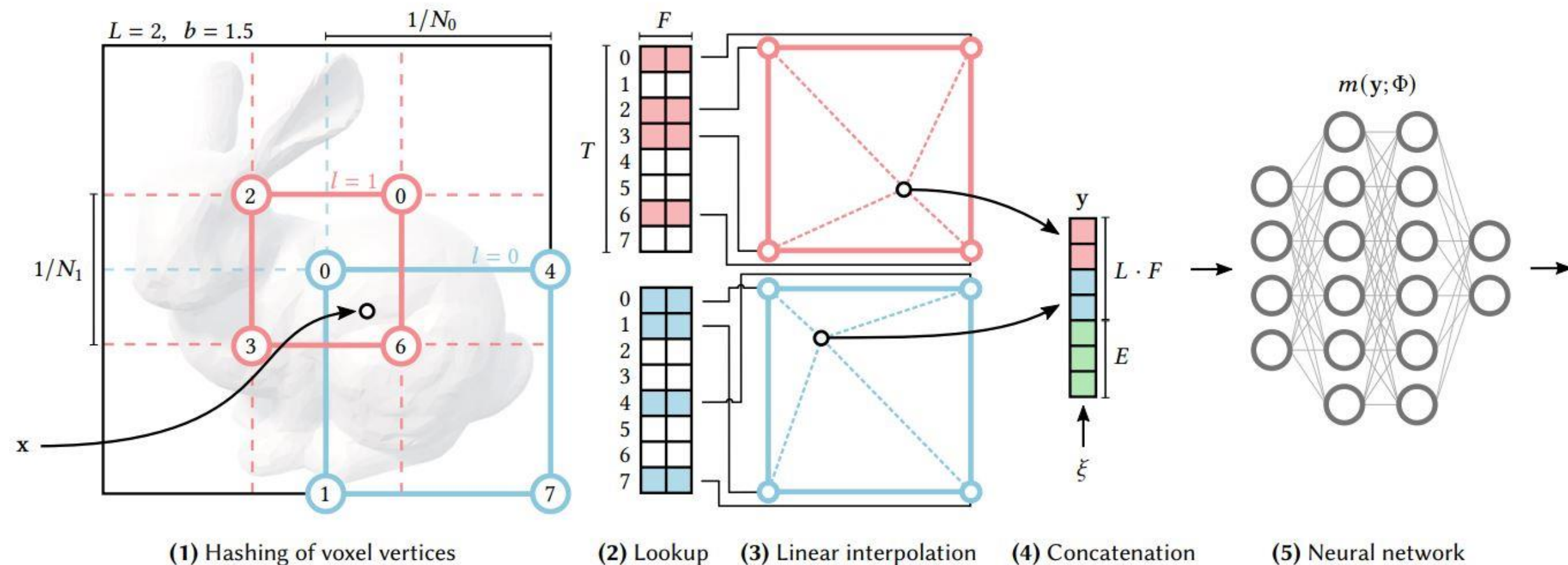
Est-ce qu'il y a mieux que des jeux sémantiques?

- Qu'en est-il de vraies structures de données utilisées pour de vrai de vrai pour des données spatiales comme des kd-tree?



Instant Neural Graphics Primitives with a Multiresolution Hash Encoding

102:4 • Müller et al.



Perspectives: différenciation implicite

« My powers are ordinary, only my application brings me success »
- Isaac Newton



Suite pour moi

- Besoin d'aide pour matérialiser ce papier pour IROS (j'aimerais ben gros ça aller en Chine)
- Manque pas grand-chose pour le papier, mais utiliser la différentiation implicite pour faire un ICP robuste basé sur le gradient de l'erreur des points et/ou utiliser l'algo en aval d'un réseau de neurones serait cool

Fast(er) Robust Point Cloud Alignment Using Lie Algebra

Jean-Thomas Sexton¹, Michael Morin², Philippe Giguère¹, Jonathan Gaudreault¹

Abstract—Point cloud alignment is a fundamental problem in robotics, with notable applications in visual odometry, autonomous navigation, mapping, and manipulation. Beyond robotics, it also finds important applications in related fields such as augmented and virtual reality, and computer vision. Iteratively Reweighted Least Squares methods are widely used for this task due to their robustness against noise and outliers. However, as the demand for real-time processing with larger datasets increases, computational efficiency becomes a growing concern. In this work, we present a novel approach for efficiently computing the optimal least squares update as a small perturbation in the Lie algebra of SE(3), the Lie group of rigid

popularity in robotics in the past two decades [3], [4]. In our context, it provides a framework for unconstrained optimization of rigid body motions, also known as the SE(3) Lie group (or SE(3) manifold, a Lie group being a group that is also a smooth manifold). Within this framework, it is not necessary to enforce constraints, as is the case when rotations are represented by matrices or quaternions, while also avoiding the problem of singularities, inherent to the representation of rotations with Euler angles.

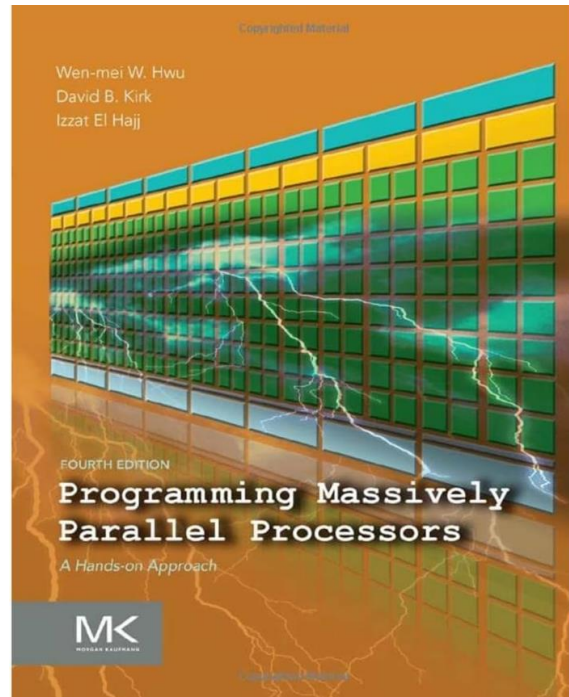
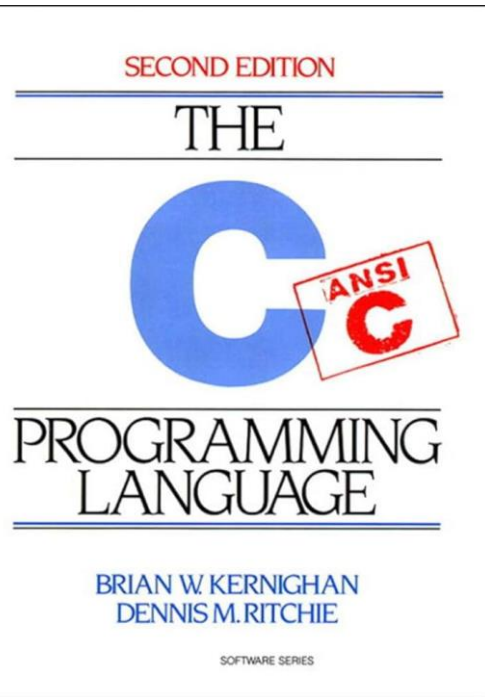
The use of Lie algebra has recently lead to practical

$$\xi^* = \left[\frac{1}{w} ((\sum_i w_i p'_i) \times \Omega + \sum_i w_i (q_i - p'_i)) \right]. \quad (42)$$

Bref

- La programmation différentiable nous permet d'optimiser nos programmes pour la réalité dans laquelle ils seront exécutés
- La programmation différentiable permet intrinsèquement une modularité exceptionnelle
- Le traitement des données spatiales se prêtent naturellement bien à la programmation différentiable

Ressources



The Elements of Differentiable Programming

Mathieu Blondel
Google DeepMind
mblondel@google.com

Vincent Roulet
Google DeepMind
vroulet@google.com

