

# Edition d'image de Poisson

Sébastien Boisgérault, MINES ParisTech

**Contexte.** Restaurer des portions d'images inconnues ou endommagées de façon visuellement crédible est un problème clé en édition d'images. Car une fois ce type de fonctionnalité de base disponible, d'autres en découlent ; ainsi pour dissimuler un bouton disgracieux sur une photo de visage, il suffit de désigner cette petite zone comme endommagée puis de laisser l'algorithme "réparer" la région.

La solution dite "de Poisson" à ce problème consiste à attribuer à chaque pixel dont la couleur est inconnue une couleur qui soit la moyenne des couleurs de ses voisins immédiats. Comme ces couleurs peuvent également être inconnues, la résolution du problème passe par l'inversion d'un système d'équations linéaires.

**Objectifs.** Ce projet implémentera la méthode dite "de Poisson" pour l'édition d'images. La réalisation d'un prototype qui interprète dans une image tous les points (purement) rouge comme des zones abimées et les remplace par la solution de l'équation de Poisson associée est relativement simple. Mais il existe ensuite de très nombreuses façons potentielles d'améliorer cette version initiale qui relèvent l'intérêt et le niveau de difficulté du projet ; notamment (indicatif) :

- En rajoutant des fonctionnalités. Par exemple en supportant l'utilisation de greffons spécifiques pour la réparation des zones endommagées plutôt qu'en utilisant uniquement les valeurs de l'image au bord de cette zone.
- En améliorant la performance de l'application. Le prototype initial sera vraisemblablement (très) lent, et ce d'autant plus que les zones endommagées seront grandes. Ce problème peut être attaqué de deux façons complémentaires : d'une part mathématiquement, en substituant aux méthodes d'inversion exactes de systèmes linéaires des méthodes approchées et itératives. Et d'autre part informatiquement, en profilant le code existant pour identifier les goulots d'étranglements, puis en adoptant des constructions Python/NumPy plus efficaces et/ou en introduisant avec Cython du code à la performance comparable au langage C.

**Technologies.** Python, NumPy, Python Imaging Library (PIL/Pillow), Cython (liste indicative).

**Pour en savoir plus.** "Poisson image editing", P. Pérez, M. Gangnet, A. Blake. ACM Transactions on Graphics (SIGGRAPH'03), 22(3):313-318, 2003., [http://www.irisa.fr/vista/Papers/2003\\_siggraph\\_perez.pdf](http://www.irisa.fr/vista/Papers/2003_siggraph_perez.pdf).