

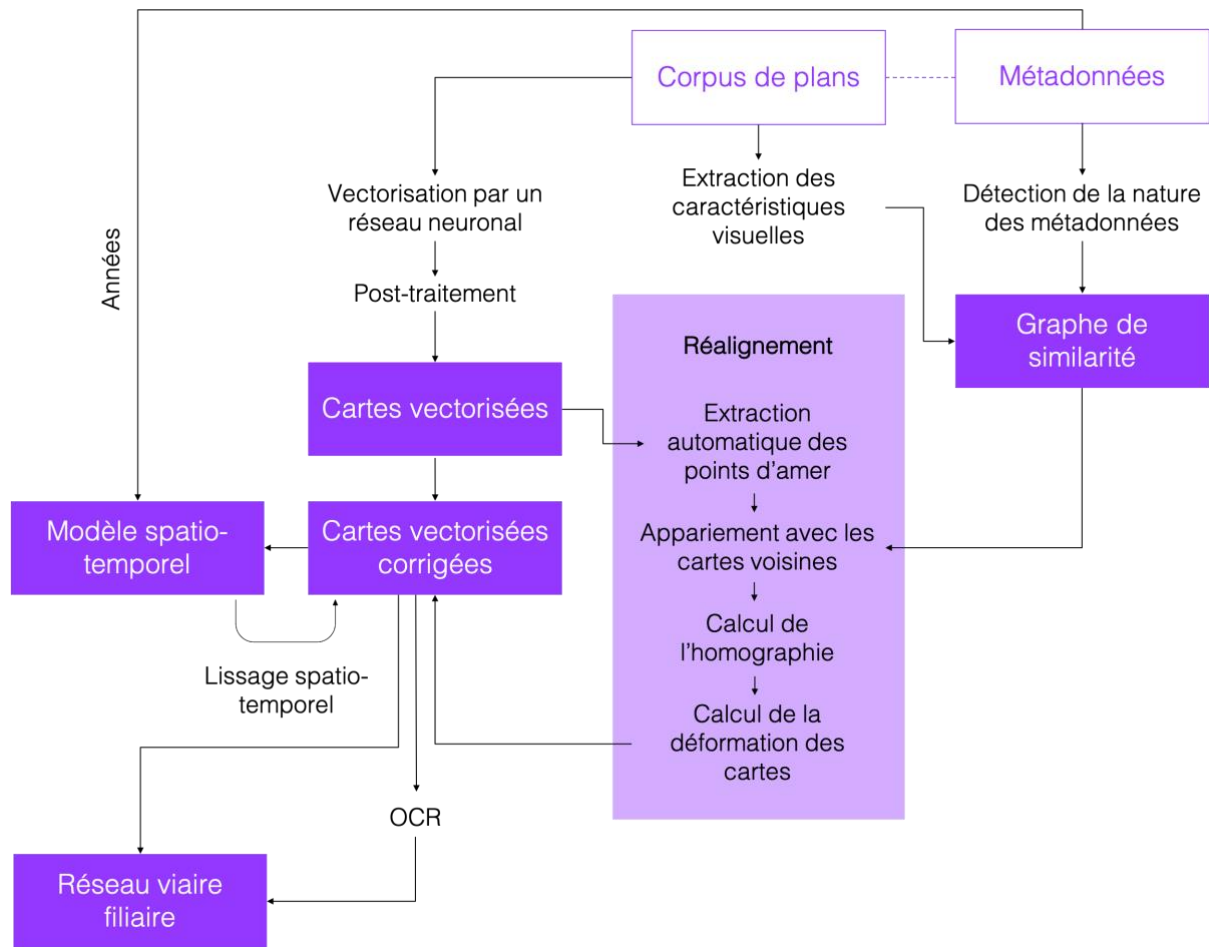
JADIS

Documentation technique
vulgarisée des expérimentations
abandonnées ou potentielles

Table des matières

TABLE DES MATIERES.....	1
VUE GLOBALE DE L'ALGORITHME	2
ANNOTATION DES DONNÉES	3
COULEUR	3
ENTRAÎNEMENT ET SEGMENTATION	4
RÉSEAU NEURONAL	4
ENTRAÎNEMENT	4
PRÉDICTION	5
VALIDATION	6
CRÉATION DE L'ANCRE.....	6
APPARIEMENT DES SURFACES D'AMER	6
EXTRACTION AUTOMATIQUE DES POINTS D'AMER (OBSOLÈTE)	6
RÉALIGNEMENT DES POINTS D'AMER.....	7
DÉFORMATION DES OBJETS-CARTES.....	9
QUANTIFICATION DE LA DÉFORMATION	9
CALCUL DU RÉSEAU DE SIMILARITÉ	9
CHAÎNAGE	9
GÉOLOCALISATION DES CARTES	10
ANCRAGE PRIMAIRE.....	10
ANCRAGE SECONDAIRE.....	10
OPTICAL CHARACTER RECOGNITION (OCR)	10
DÉTECTION DES SEGMENTS DE TEXTE	10
OCR	12
CROISEMENT DES RÉSULTATS AVEC LA BASE DE DONNÉES	12
RÉALIGNEMENT PAR SUPERPOSITION DES SURFACES	13
LIMITATIONS.....	14
LEVEL SETS.....	15
RÉALIGNEMENT DES GRAPHS	15
POST-TRAITEMENT (V1)	16
CRÉATION DU MASQUE VIAIRE	17
LISSAGE ET TRAITEMENT DE LA CARTE DE PROBABILITÉS	18
AUTRES APPROCHES.....	19
AUTRES APPROCHES TESTÉES ET ABANDONNÉES (PRINCIPALES)	19
AUTRES APPROCHES À TESTER.....	19
BIBLIOGRAPHIE	20

Vue globale de l'algorithme



Annotation des données

Couleur

Dans sa première version, l'annotation des données incluait de multiples labels, capables d'identifier l'utilisation du sol. Au total, 220 patches ont été annotés en couleur. Ces annotations exigeaient environ 1.5 fois plus de temps qu'une simple annotation du réseau viaire. C'est l'une des raisons pour lesquelles cette option a été écartée. L'autre raison est que le facteur déterminant dans le réaligement des cartes s'est finalement révélé être la forme des îlots, ainsi que le réseau viaire, plutôt que l'utilisation du sol à proprement parler.

De plus, la multiplication des labels implique systématiquement une complexification du problème de prédiction pour le réseau neuronal, et donc potentiellement une diminution de la qualité de la segmentation. Cependant, cette idée de labels multiples pourrait être réintroduite facilement, à condition de disposer d'un plus grand nombre de données d'entraînement, et présenterait un intérêt certain pour rendre les cartes numérisées intelligibles.



Six couleurs pures (les canaux rouge-vert-bleu RGB prennent les valeurs minimales de 0 ou maximales de 255) avaient été utilisées pour annoter les différents éléments du plan.

- En noir, l'extérieur de la carte (légende, bordure, arrière-plan, titre, annotations périphériques) ;
- En blanc, le réseau viaire, y compris les routes, les ponts et les chemins de fer ;
- En vert, les parcs, les forêts, les vergers ;
- En bleu, les cours d'eau, les citernes, les lacs, les canaux, les ports ;
- En rouge, les monuments, les murs d'enceinte, les forts, les églises ;
- En rose, les pâtés de maisons et l'environnement urbain construit (sauf monuments ou éléments appartenant aux catégories précédentes) ;
- En jaune, le non-bâti n'appartenant à aucune des catégories précédentes, comme les champs, les prés, les cours et les places distinctes du réseau viaire.

Entraînement et segmentation

Réseau neuronal

Deux autres variantes de Resnet50V1 : Resnet25 et Resnet100, ont également été testées.

Entraînement

De multiples tentatives d'entraînement ont été effectuées, pour optimiser les paramètres suivants :

- nombre d'époques d'apprentissage du réseau => optimal autour de 25 époques
- rotation maximale lors de l'augmentation => optimale à 2π
- augmentation des données par diminution maximale de l'échelle => optimale à un facteur $\frac{1}{2}$

À chaque fois, l'évaluation implique le set de patches de validation et les métriques de précision et de rappel, globales et pour chaque classe.

Au-delà de 25 époques, la performance diminue, ce qui est vraisemblablement dû à un effet de sur-apprentissage (*overfitting*). Il est probable que le nombre d'époques pourrait être avantageusement augmenté si le set d'entraînement était plus conséquent.

Un paramètre n'a pas pu être optimisé, en raison de la limitation de mémoire vive de la machine. Il s'agit de la taille de batch, fixée à 1 pour l'entraînement. Une augmentation modérée de la mémoire suffirait pour implémenter *minibatch* et simuler une taille de batch plus importante, alors qu'une augmentation significative de la mémoire vive serait nécessaire pour réellement augmenter la taille de batch.

Une masse d'entraînement convenable a été atteinte pour la géolocalisation des cartes. Cependant, la segmentation reste approximative. Une masse souhaitable pour améliorer les performances de segmentation se situerait probablement au-delà de 500, idéalement au-delà de 1000 patches d'entraînement, en accentuant la constitution du set d'entraînement sur les faiblesses de prédiction actuelles, en particulier :

- les cartes à petite échelle, dont la détection des rues est hasardeuse, et qui ne bénéficient pas de l'augmentation des données par échelle, voire qui en souffrent
- les cartes cadastrales à grande échelle
- les cartes dont les îlots sont représentés par une différence de texture, à l'exception peut-être des cartes hachurées type Andriveau, qui semblent déjà suffisamment représentées dans le set d'entraînement
- les cartes colorées récentes, en particulier lorsque le contraste rue/îlot est inhabituel (par exemple lorsque les îlots sont clairs alors que les rues sont foncées)

Une sous-division du corpus en fonction de la typologie figurative des cartes a été effectuée, afin de s'essayer à un entraînement plus spécifique pour différentes typologies de cartes. Pour ce, le set d'entraînement a été divisé selon deux critères :

- la représentation des îlots ; soit, dans l'ordre de prévalence,
 - Les îlots sont-ils (le plus souvent) texturés ? Si oui, la texture prévaut.
 - Si non, les îlots sont-ils (le plus souvent) colorés ? Si oui, la couleur prévaut.

- Si non, les îlots ne sont démarqués que par leur contour. La représentation est dite « blanche ».
- l'échelle (grande ou petite);

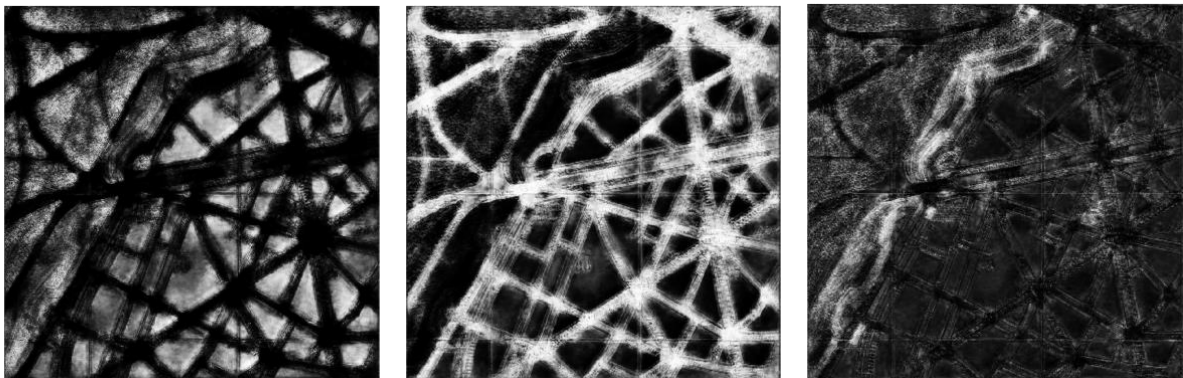
Nous obtenons donc 3 types de cartes : les cartes texturées, les cartes colorées (mais pas texturées) et les cartes ni texturées, ni colorées (« blanches »). Ces trois typologies s'appuyaient sur une observation empirique des prédictions par le modèle générique, et en particulier sur ses faiblesses. Ils s'appuyaient également sur la compréhension mathématique d'un neurone, pour lequel la texture, la couleur et les arêtes sont trois éléments très distincts.

Comme peu de carte texturées et colorées étaient représentées à une échelle de cadastre, ces deux catégories n'ont pas été séparées selon l'échelle. 4 catégories de cartes ont donc été retenues : T (texturées), C (colorées), B0 (blanches, grande échelle), B1 (blanches, petite échelle), et 4 réseaux de neurones entraînés.

Malheureusement, les résultats, tout en ne se révélant pas mauvais, en particulier pour les cartes dites « blanches », n'ont pas obtenu un score significativement meilleur que le modèle générique. Cela est sans-doute dû à la réduction de la masse d'entraînement impliquée par sa subdivision en 4 catégories. Des résultats intéressants pourraient cependant être obtenus avec une masse d'entraînement plus importante. La pertinence des catégories choisies doit également être investiguée.

Prédiction

Lorsque le set d'entraînement contient de nombreuses catégories (voir le chapitre *Annotation des données*), dhSegment renvoie en sortie une carte de probabilité, définissant pour chaque pixel la probabilité d'appartenir à l'une des catégories de régions.



Ci-dessus, la carte de probabilité d'une prédiction effectuée pour une image non-annotée du set de test. De gauche à droite, la probabilité pour chaque pixel d'appartenir à la région rose (pâtés de maisons), blanche (réseau viaire) et rouge (monuments et murs d'enceinte).

Validation

Sur un problème de prédiction trinaire, le premier réseau entraîné affichait une précision médiane de 85.5%. La version intermédiaire affichait une précision médiane de 93.01% (moyenne 77.18%), et un rappel médian de 86.22% (moyenne 82.94%).

Pour tous les réseaux, ce résultat s'améliore nettement après le post-traitement, pour atteindre une précision médiane de 94.59% (moyenne 90.49%), et un rappel médian de 94.11% (moyenne 87.02%), pour le réseau intermédiaire. Cependant, il faut relever que le problème de détection du réseau viaire est le plus complexe (rappel médian 75.47%).

Création de l'ancre

L'API Overpass renvoie une liste de tronçons de rue, avec une estimation de la catégorie de rues (primaire, secondaire, sentier, autoroute, etc.). Pour géolocaliser des cartes à plus petite échelle (e.g. départementale), ces métadonnées pourraient être utilisées pour réduire le niveau de détail de l'ancre, afin de la rendre plus proche des cartes étudiées et obtenir ainsi un meilleur réalignement.

Appariement des surfaces d'amer

Le nombre de tranches d'orientation a été optimisé pour atteindre 18, soit 20° par tranches. Apparemment, un nombre moins important de tranches obtient de moins bons résultats, probablement en raison d'un niveau de détail trop faible. Étonnamment, un nombre plus important de tranches obtient également de moins bons résultats. Cela est sans doute dû au « fléau de la dimension »¹ (*curse of dimensionality*), très fréquent lors de l'application de l'algorithme des K plus proches voisins (KNN). Ce fléau est causé par une perte de sens de la notion de voisinage, dans des espaces comprenant de très nombreuses dimensions (espaces de plus de 18 dimensions dans notre cas).

Un nombre de dimensions plus important pourrait potentiellement être utilisé s'il était suivi par une réduction dimensionnelle, par exemple basée sur l'analyse discriminante linéaire (*Fisher LDA*)². Cette dernière permettrait de ne conserver que les angles qui différencient le mieux les îlots les uns des autres et de ne pas prendre en compte les angles trop communs.

Extraction automatique des points d'amer (obsolète)

En général, pour résoudre un problème de géolocalisation, les points d'amer (*keypoints*) qui permettent de réaligner deux cartes sont indiqués manuellement sur les monuments (p.ex clochers des églises) ou sur le carroyage, la grille de coordonnées de la carte. En théorie, on peut remplacer cette approche par un plus grand nombre de points, placés automatiquement.

Cette approche a été testée sur les cartes brutes, mais sans succès. En revanche, elle est efficace sur l'image des graphes.

¹ Richard E. Bellman, *Adaptive Control Processes: A Guided Tour* (Princeton, New Jersey: Princeton University Press, 2015).

² Max Welling, 'Fisher Linear Discriminant Analysis', 2007, 4.

Différents types d'algorithmes permettent d'extraire les points d'amer, entre autres SIFT³ (*scale-invariant feature transform*), SURF (*speeded-up robust features*) et ORB (*oriented FAST and rotated BRIEF*). SIFT a obtenu de loin les meilleurs résultats sur le problème de réalignement des cartes. À noter que SIFT a fait l'objet d'un brevet, valide jusqu'au 7 mars 2020, ce qui n'en a donc pas fait un obstacle pour ce projet. Le principe général de SIFT consiste à identifier, à plusieurs échelles différentes, des caractéristiques géométriquement saillantes (angles, arêtes contrastées, surfaces vides) sur toute la carte. L'idée générale de SIFT a été conservée pour extraire les caractéristiques géométriques des îlots, mais adaptées pour correspondre aux objets géométriques de la carte. De plus, SIFT inclut la recherche automatique de points géométriquement intéressants, alors que les points géométriquement intéressants de la carte segmentée peuvent être directement extraits, et appariés par type.

Réalignement des points d'amer

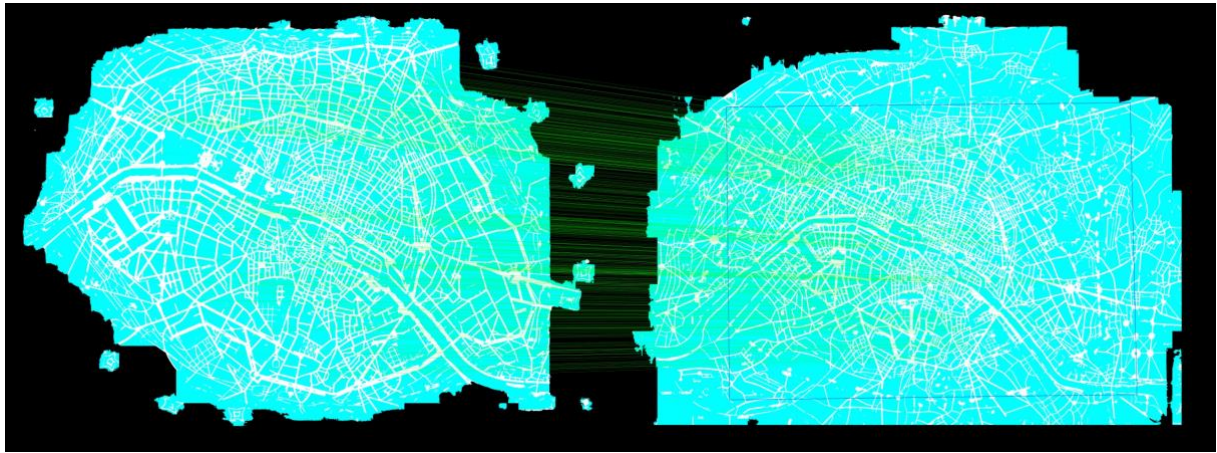
Cette étape a été testée avec un certain succès sur les images segmentées automatiquement. Cependant, la version implémentée, basée sur les surfaces, obtient de meilleurs résultats. Le réalignement des cartes par les points d'amer repose sur l'appariement de points extraits automatiquement par SIFT. Les points d'amer de l'image A sont appariés à leurs K plus proches voisins (KNN approx., K=2) de l'image B. La notion de voisinage n'est ici pas géographique, mais liée au contexte commun de la situation des deux points. Par exemple, deux points se situant à un coin de rue dont les rues se croisent à un angle très similaire seront considérés proches voisins. Comme pour l'appariement des surfaces, seuls les meilleurs appariements étaient conservés (ratio de Lowe).

Dans un second temps, l'homographie optimale et canonique entre les deux réseaux filaires était calculée. L'algorithme de RANSAC⁴ (*RANdom SAmple Consensus*) était appliqué, pour éliminer les valeurs aberrantes. Un seuil élevé était fixé, car les cartes d'origine contiennent des imprécisions et peuvent être déformées.

Ci-dessous, un exemple d'appariement deux cartes segmentées automatiquement, où les lignes vertes relient les points d'amer SIFT appariés. Le cadre bleu représente la position calculée de la première image par rapport à la seconde (bordure externe noire comprise).

³ David G. Lowe, 'Object Recognition from Local Scale-Invariant Features', in *Proceedings of the International Conference on Computer Vision* (presented at the International Conference on Computer Vision, Corfu, 1999), pp. 1–8 <https://www.academia.edu/20662687/Object_Recognition_from_Local_Scale-Invariant_Features> [accessed 12 November 2019].

⁴ Martin A. Fischler and Robert C. Bolles, 'Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography', *Communications of the ACM*, 24.6 (1981), 381–95 <<https://doi.org/10.1145/358669.358692>>.



Déformation des objets-cartes

Quantification de la déformation

Les erreurs pouvaient également être calculées sur les points d'amer, plutôt que sur les centroïdes des îlots. La méthodologie ne différait pas.

Calcul du réseau de similarité

Chaînage

La première version du réseau de similarité prenait la forme d'une chaîne. Le chaînage consistait, à l'aide des mêmes critères visuels et de proximité des métadonnées, à calculer un score de relation entre toutes les cartes et à aligner ces dernières sur une seule ligne, en commençant par les deux cartes se ressemblant le plus (i.e ayant le score de relation le plus élevé), puis en ajoutant à chaque fois, à l'extrémité, le plan ressemblant le plus au dernier plan ajouté. Pour des raisons de robustesse, cette ontologie a été abandonnée.

Ci-dessous, le début de la chaîne de cartes et un extrait central



Géolocalisation des cartes

Ancrage primaire

Dans certaines versions, les cartes segmentées, après extraction des points d'intérêt par SIFT, étaient toutes vectorisées (c'est à dire convertie au format filaire vectoriel) avant l'appariement. Cela augmentait sensiblement le nombre de cartes appariées avec succès mais diminuait également le nombre de points impliqués dans l'appariement. En effet, SIFT ne plaçait de points d'amer que sur les îlots, et plus sur le réseau viaire. En un sens, c'est sans doute le même phénomène qui explique que l'appariement par la géométrie des surfaces affiche un taux de réussite plus élevé. Semblablement, lorsque l'on inclut également les caractéristiques géométriques invariantes (IF) des intersections des rues dans la version finale de l'algorithme, le taux de réussite diminue.

Pour géolocaliser des cartes vectorisées dont les points d'amer ont été extraits automatiquement par SIFT, un nombre important de tentatives d'appariement à l'ancre doit être effectué (jusqu'à 50). En effet, si l'appariement des cartes au format vectoriel filaire semble plus facile, le nombre de points d'amer appariés est également moindre, ce qui diminue la précision de la géolocalisation. Parmi les tentatives réussies, celle dont le nombre de points d'appariement est maximum était conservé. Ce processus d'ancrage primaire permettait de géolocaliser directement environ 20% des cartes (contre plus de 60% pour la version basée sur les surfaces des îlots).

Ancrage secondaire

Dans un deuxième temps, les cartes qui n'avaient pas pu être appariées directement à l'ancre étaient appariées à leurs voisines ancrées, où le voisinage est déterminé par le réseau de similarité. Cet ancrage secondaire se faisait sur les cartes segmentées (non-filaires), car un deuxième appariement filaire diminuait trop fortement la précision.

Optical character recognition (OCR)

La reconnaissance optique des caractères permet principalement d'enrichir les métadonnées des cartes. Elle peut aussi contribuer, en comparant les réseaux, à corroborer l'alignement des cartes. Trois approches successives de détection des textes ont été testées.

Détection des segments de texte

Les trois approches ont été testées sur le même extrait de carte, qui comportait 26 éléments reconnaissables.

La première approche était indépendante de la segmentation automatique de la carte. Elle se basait sur un algorithme de détection automatique des segments de texte dénommé EAST⁵, appliqué directement sur l'image d'origine de la carte. Seuls 7 éléments reconnaissables ont pu être extraits grâce à cet algorithme. Quatre ont pu être lus par l'OCR, dont deux correctement (8% du total). Cette mauvaise

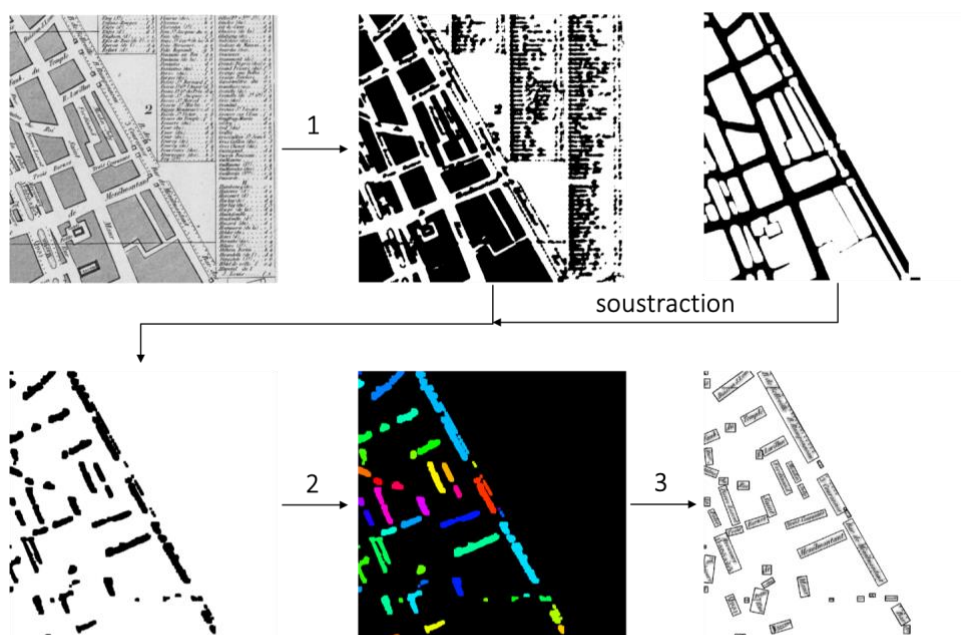
⁵ Xinyu Zhou and others, 'EAST: An Efficient and Accurate Scene Text Detector', *ArXiv:1704.03155 [Cs]*, 2017 <<http://arxiv.org/abs/1704.03155>> [accessed 12 November 2019].

performance explique la raison qui nous a poussé à abandonner l'idée d'un réaligement des cartes basé a priori sur l'OCR.

La deuxième approche considérait chaque segment de rue comme une zone d'intérêt potentielle. Elle était basée sur la segmentation automatique du réseau viaire. 12 éléments reconnaissables avaient pu être extraits adéquatement grâce à cet algorithme. 10 avaient pu être lus par l'OCR, dont 4 correctement (15% du total).

La troisième approche est celle qui a obtenu les meilleures performances. Pour des raisons de temps, elle n'a pas été incluse dans la version finale de JADIS, mais les résultats sont prometteurs et pourraient faire l'objet d'une recherche plus approfondie. Elle se basait simplement sur la différence de valeur entre le texte et l'arrière-plan (réseau viaire). Dans un premier temps, l'image était binarisée et floutée par un filtre Gaussien, puis binarisée à l'aide de l'algorithme de Otsu⁶. Intuitivement, cela revient à séparer les régions qui sont globalement plus foncées des régions globalement plus claires. Dans un deuxième temps, les îlots étaient dilatés puis soustraits à l'image binarisée. Les différentes composantes connexes étaient extraites et un rectangle circonscrit d'aire minimale y était ajusté. Grâce à cette méthode, la totalité des éléments reconnaissables avaient pu être extraits (26). 23 avaient pu être lus par l'OCR, dont 12 correctement (46% du total)

Extraction des segments de texte



1. Algorithme de Otsu
2. Extraction des composantes connexes
3. Ajustement du rectangle circonscrit

⁶ Nobuyuki Otsu, 'A Threshold Selection Method from Gray-Level Histograms', *IEEE Transactions on Systems, Man, and Cybernetics*, 9.1 (1979), 62–66 <<https://doi.org/10.1109/TSMC.1979.4310076>>.

OCR

L'OCR utilisée pour ces expérimentations est Tesseract, une bibliothèque de programmation libre développée par HP et Google. Des segments de texte (voir-ci-dessous) lui étaient présentés. Afin d'améliorer les résultats, chaque segment était présenté successivement à la version anglaise, puis à la version française de Tesseract. En effet, l'OCR anglaise est mieux entraînée, mais la version française est plus spécifique sur nos données. Des orientations différentes (+ ou - 10°) ont également été appliquées, pour maximiser les chances de reconnaître correctement le segment de texte, au cas où la détection de sa rotation, déduite de l'orientation du rectangle circonscrit, serait légèrement erronée. Pour correspondre au mieux aux données sur lesquelles l'OCR a été entraînée, la vignette a été redimensionnée, de sorte que la hauteur des majuscules se situe entre 19 et 45 pixels, et un cadre noir de 5 pixels a été ajouté.



Croisement des résultats avec la base de données

Chaque segment de texte était donc soumis 6 fois à l'OCR (2x3, pour chaque langue, et pour les trois orientations, -10°, 0°, 10°). Une étape de recherche expression régulière permettait de récupérer les mots. Cependant, le résultat était souvent bruité à cause des caractères qui n'ont pas été lus correctement par l'OCR. Il était donc nécessaire de croiser ces résultats avec la base de données des noms de rues, afin de retrouver la solution correcte.

Pour ce faire, lorsque la chaîne de caractère était suffisamment longue, la distance de Damerau⁷-Levenshtein⁸ était calculée avec tous les mots présents dans la base de données des noms de rues. La correspondance qui obtenait le meilleur score moyen était retenue.

Exemple : *GOBELINSEE* => GOBELINS (score de correspondance : 80%)

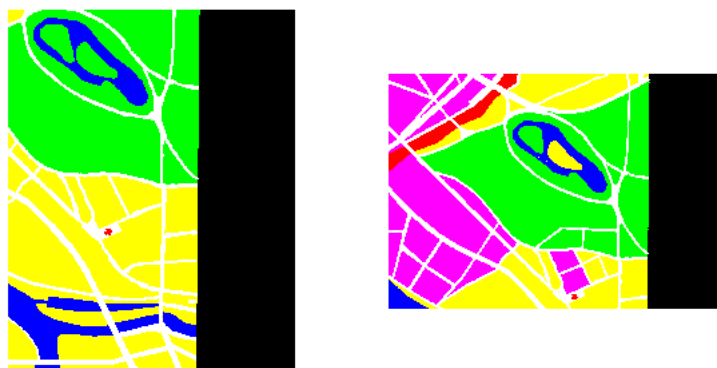
⁷ Fred J. Damerau, 'A Technique for Computer Detection and Correction of Spelling Errors', *Commun. ACM*, 7.3 (1964), 171–176 <<https://doi.org/10.1145/363958.363994>>.

⁸ V. I. Levenshtein, 'Binary Codes Capable of Correcting Deletions, Insertions and Reversals', *Soviet Physics Doklady*, 10 (1966), 707.

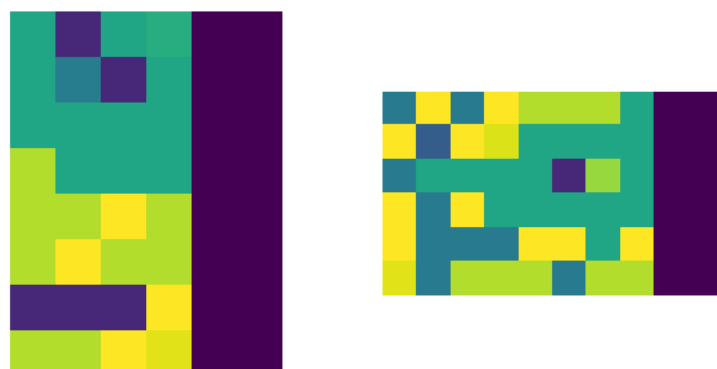
Réalignement par superposition des surfaces

Le principal avantage de la segmentation sémantique des cartes est la possibilité de calculer un score de réalignement par superposition des surfaces, via le calcul des aires communes (indice d'intersection des aires). Le principal désavantage est le nombre trop important de degrés de liberté. 1 pour la rotation, 2 pour la position et 1 pour l'échelle, ce qui explique pourquoi cette piste a été abandonnée. En effet, il est très difficile d'optimiser une fonction avec 4 degrés de liberté, d'autant plus qu'intuitivement, pour le problème de réalignement de cartes, on comprend bien que la fonction n'est pas du tout strictement convexe, ce qui signifie que de nombreux minimas locaux empêchent la convergence.

À partir de deux images qui se recouvrent partiellement, l'idée était de calculer la position d'une image par rapport à l'autre pour les réaligner. L'algorithme est robuste, car il fonctionne même lorsque les cartes contiennent des différences (erreurs d'annotation, transformations du tissu urbain entre deux cartes, etc.).

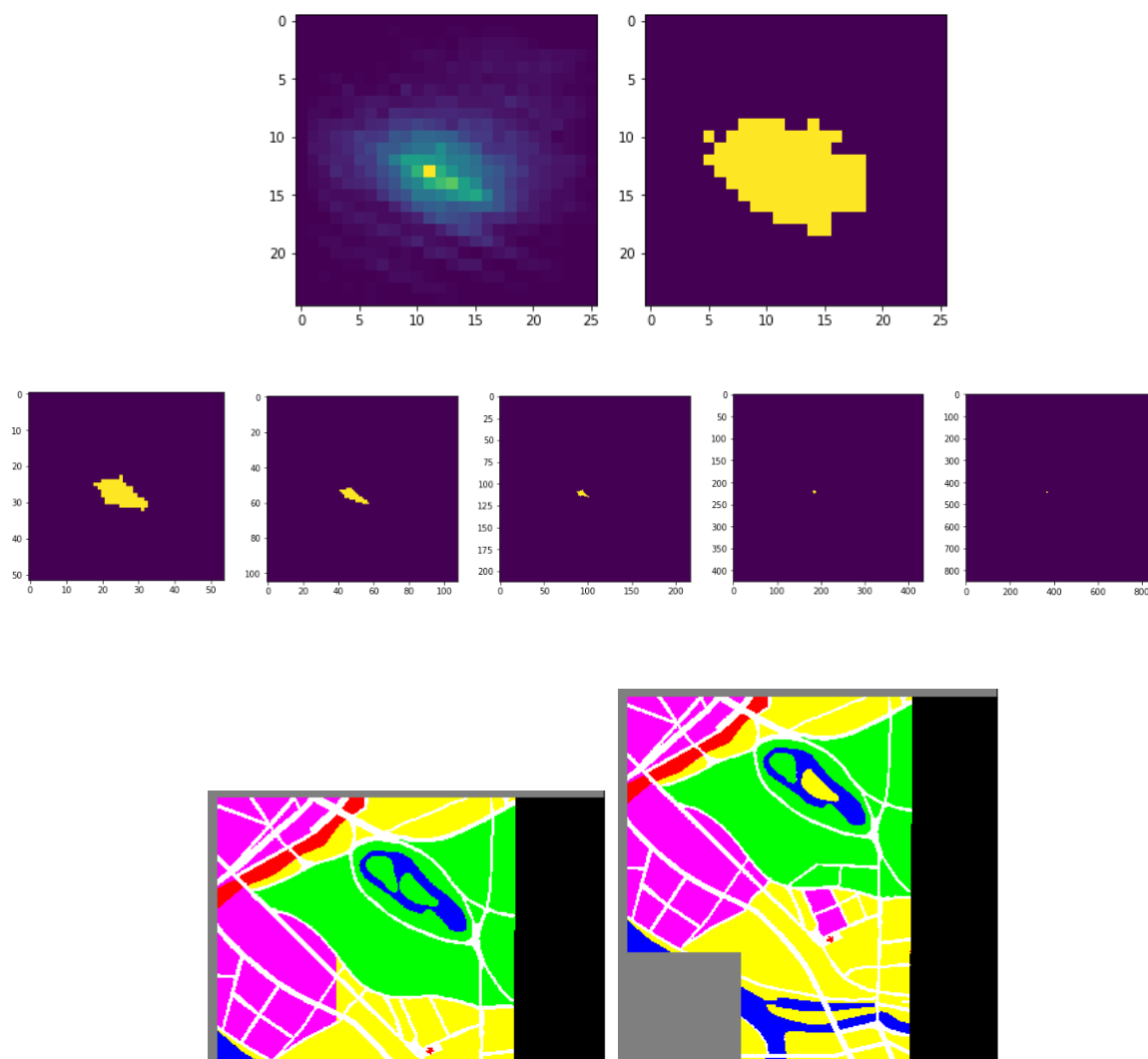


Dans un premier temps, la résolution des images est diminuée plusieurs dizaines de fois, comme si elles étaient observées à petite échelle. Chaque couleur est convertie en un code couleur numérique unique, sur un seul canal.



La réduction considérable du nombre de pixels permet de calculer, pour toutes les positions relatives possibles, un score (chaque pixel ayant une couleur identique au pixel sur lequel il est superposé augmentant le score de 1). Le quantile supérieur des pixels ayant obtenu le meilleur score est conservé, tandis que les autres pixels sont définitivement éliminés. L'opération est ensuite répétée plusieurs fois, en doublant la résolution à chaque étape et en ne considérant que les pixels ayant obtenu les meilleurs scores pour l'étape suivante. Ce processus itératif se poursuit, jusqu'à obtenir une précision de 1 pixel

qui permet de réaligner les deux cartes l'une par rapport à l'autre au pixel près. Le fait de commencer le réalignment à très basse résolution permet un gain de temps considérable.



Limitations

En plus de la latitude et de la longitude relative, l'angle de rotation et l'échelle doivent également être considérés pour pouvoir réaligner correctement les surfaces. Pour l'angle de rotation, une approche basée sur les histogrammes des HOG (voir explication sur le Chaînage des cartes) a été testée, sans grand succès.

Sur les cartes segmentées, la déduction de la rotation a été testée par l'histogramme des HOG. Cette approche n'a pas été un succès, mais elle a permis de mettre en évidence l'utilité des HOG pour identifier la similarité entre les images de cartes et également pour extraire les caractéristiques géométriques sur les cartes segmentées.

Level sets

La méthode des surfaces de niveau (*Level sets algorithm*⁹) est un algorithme qui permet de superposer des surfaces les unes sur les autres, et de les réaligner, comme si leurs régions correspondaient à des courbes de niveau topographiques. Hors topographie, cette méthode est également utilisée pour suivre le déplacement ou la transformation d'objets dans des vidéos. Elle pourrait aussi être appliquée à l'évolution du tissu urbain, où l'altitude de la montagne serait remplacée par l'axe du temps. Cela pourrait permettre, en comparant chaque carte avec ses voisines plus anciennes et plus récentes, de construire une carte spatio-temporelle agrégée de la ville, décrivant l'évolution la plus probable, au vu de toutes les sources étudiées, du tissu urbain au cours du temps.

Réalignement des graphes

L'extraction du réseau viaire des cartes peut être directement réaligné en appariant les rues (arêtes) et les intersections (nœuds) des différentes cartes. Ce réalignement basé sur les graphes nécessite une étape supplémentaire, puisque le réseau filaire doit être au préalable extrait des images vectorisées. La méthode robuste décrite par Costes dans sa thèse¹⁰ aurait par exemple pu être utilisée. Le réalignement des graphes de rues devrait également être robuste aux apparitions et aux disparitions de rues, ainsi qu'aux imprécisions. Il appartient aux problèmes de la théorie des graphes, où cette opération est connue sous le nom d'homéomorphisme partiel stable par composition.

Costes décrit aussi le processus de fusion des nœuds et des arêtes permettant de construire un STAG (*spatio-temporal aggregated graph*, ou graphe spatio-temporel agrégé), qui peut servir de référentiel pour l'étude de l'évolution du réseau viaire au cours du temps, sur la période d'étude.

Cette piste a été abandonnée car elle demande une vectorisation très performante (Costes a vectorisé ses cartes à la main), et qu'elle peut se révéler peu robuste à de grandes déformations de la carte. Elle serait cependant certainement plus efficace pour des cartes à petite échelle, comme cela a également été proposé très récemment par les étasuniens Howe et Weinman¹¹.

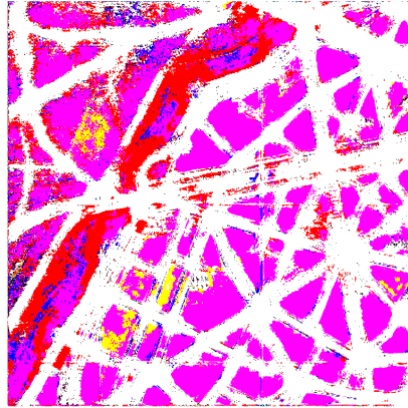
⁹ Stanley Osher and James A Sethian, 'Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations', *Journal of Computational Physics*, 79.1 (1988), 12–49 <[https://doi.org/10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2)>.

¹⁰ Benoît Costes, 'Vers la construction d'un référentiel géographique ancien : un modèle de graphe agrégé pour intégrer, qualifier et analyser des réseaux géohistoriques', 2016 <<https://tel.archives-ouvertes.fr/tel-01565850>> [accessed 12 November 2019].

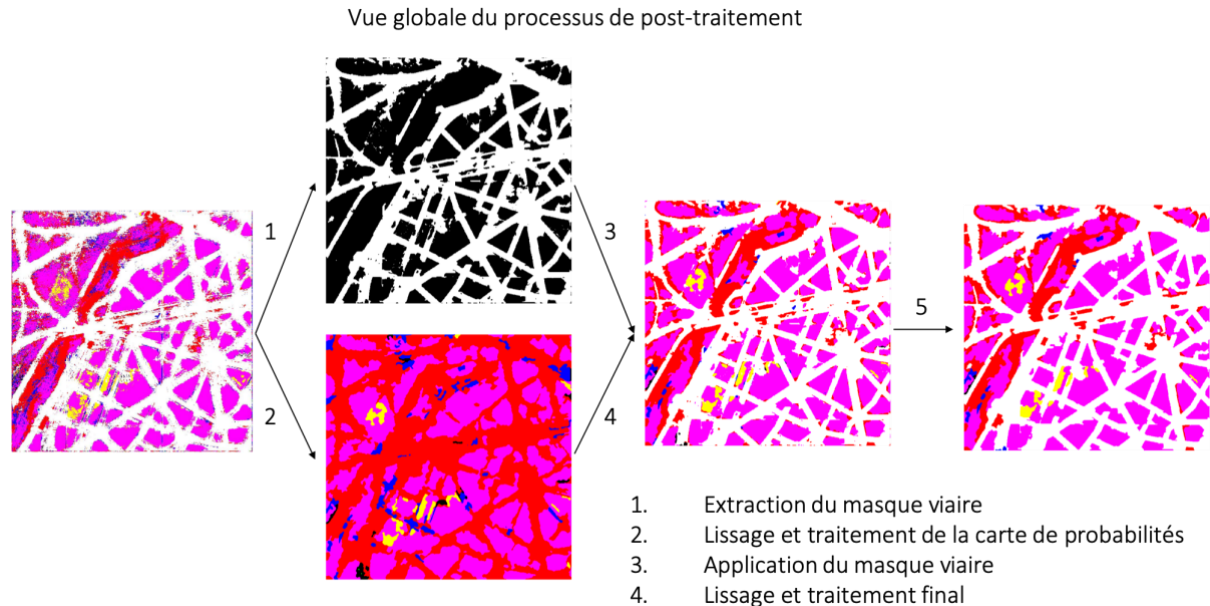
¹¹ Nicholas Howe, Jerod Weinman, and John Gouwar, 'Deformable Part Models for Automatically Georeferencing Historical Map Images' (presented at the SIGSPATIAL 19, Chicago, 2019), p. 4.

Post-traitement (V1)

Cette première version de post-traitement a été développée lorsque la segmentation était moins précise. Si l'on ne prend, pour chaque pixel, que le résultat le plus probable, on obtient la prédiction de sortie.



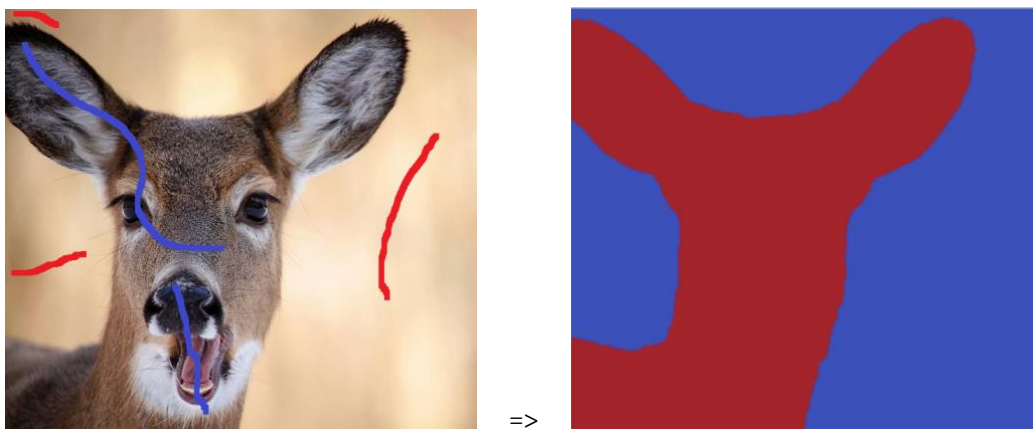
Il faut ensuite nettoyer cette prédiction pour éliminer le bruit inhérent. Dans notre cas, le bruit est nettoyé en appliquant quelques filtres de floutage successifs, suivi par un seuillage des couleurs, qui permettent ensemble de lisser le bruit. On remarque aussi que les contours des îlots sont imprécis. Une technique basée sur la connaissance de l'image originale permettra de les récupérer.



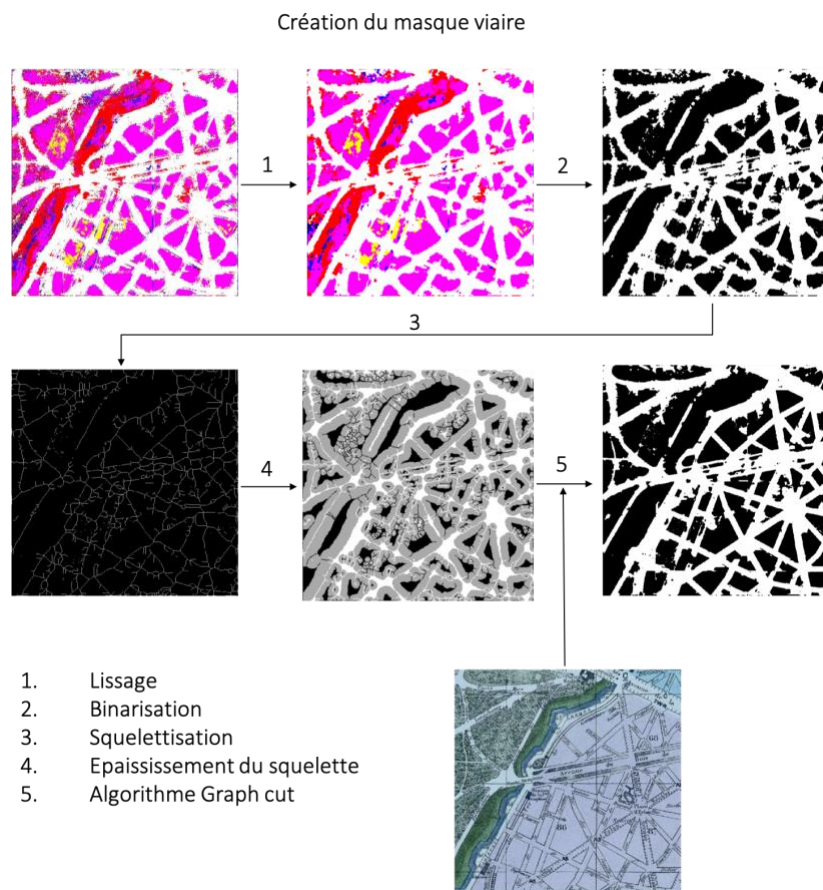
L'image de prédiction va donc être traitée séparément, pour ce qui concerne le réseau viaire d'une part et l'annotation des îlots d'autre part. À la fin du processus, ces deux voies de traitement sont rassemblées. Le résultat final est lissé et débarrassé des composantes connexes (îlots ou taches de même couleur) dont la surface est plus petite qu'un certain seuil (ici 400 pixels). Ces dernières sont fondues dans une composante voisine plus grande, selon un ordre de priorité défini pour toute l'image.

Création du masque viaire

L'un des deux voies de post-traitement a pour but la création d'un masque viaire, en se basant sur la connaissance de l'image de base. Le cœur de cette étape de post-traitement repose sur un algorithme appelé *graph cut*¹² (ou grab cut, selon la variante). *Graph cut* est un algorithme qui permet de trouver le découpage optimal entre deux plans d'une image, en se basant sur une annotation manuelle de ces plans. Dans notre cas, nous allons utiliser un artifice pour générer une annotation automatique des plans, basée sur la sortie du réseau neuronal. Nous allons donc récupérer les informations présentes dans l'image de base pour améliorer la qualité de la segmentation. Le découpage de *graph cut* repose en priorité sur les différences d'intensité entre les différentes régions de l'image et n'est donc performant que lorsque les contours des objets à découper sont nettement marqués, comme c'est le cas pour des cartes.



¹² Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, “GrabCut”: Interactive Foreground Extraction Using Iterated Graph Cuts’, in *ACM SIGGRAPH 2004 Papers*, SIGGRAPH ’04 (New York, NY, USA: ACM, 2004), pp. 309–314 <<https://doi.org/10.1145/1186562.1015720>>.



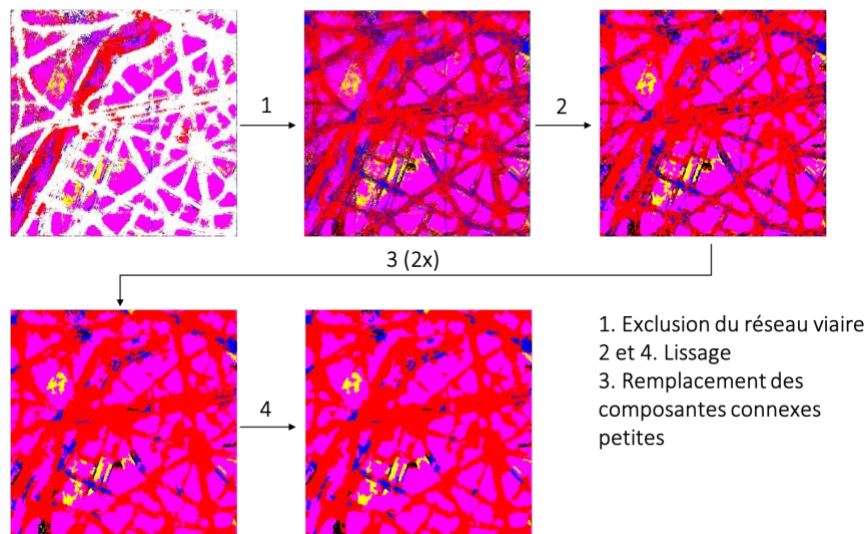
Pour générer automatiquement l'annotation de *graph cut*, l'image de base est lissée pour éliminer le bruit et les îlots sont séparés du réseau viaire dans une image binaire noir-blanc. Une représentation filaire est d'abord extraite, pour les îlots et pour le réseau viaire pour obtenir le « squelette » de ces derniers¹³. Les squelettes sont ensuite épaissis en leur adjoignant, pour chaque composante connexe, les pixels proches du centre. Ces deux groupes de squelettes épaissis (celui issu du réseau viaire et celui issu des îlots) sont utilisés comme annotation pour *graph cut*. L'algorithme *graph cut*, en se basant sur l'image de base, va donc établir le tracé de chaque façade à l'emplacement de l'arête la plus nette rencontrée entre le squelette de l'îlot et le squelette de la rue.

Lissage et traitement de la carte de probabilités

Puisque le réseau viaire a été extrait dans l'étape précédente, il peut être exclu de ce processus, qui vise à déterminer plus clairement l'attribution des différents îlots à l'une ou l'autre des catégories annotées. Cela passe principalement par un lissage de l'image, pour éliminer le bruit. Le lissage est composé à chaque fois par une étape de floutage suivi d'un seuillage des couleurs. En (2), cinq lissages successifs sont effectués, puis dix en (4). Mathématiquement, le floutage est obtenu par un filtre de convolution, qu'on pourrait intuitivement assimiler à un neurone unique et fixe. Les composantes connexes dont la surface est plus petite qu'un certain seuil (ici 200 pixels) sont ensuite rattachées à une composante plus grande, voisine (3), selon un ordre de priorité propre à l'image. Cette étape (3) est répétée deux fois.

¹³ Zicheng Guo and Richard W. Hall, 'Parallel Thinning with Two-Subiteration Algorithms', *Commun. ACM*, 32.3 (1989), 359–373 <<https://doi.org/10.1145/62065.62074>>.

Lissage et traitement de la carte de probabilités



Autres approches

Autres approches testées et abandonnées (principales)

- ⇒ Segmentation des cartes par la méthode des superpixels. Trop peu robuste, notamment vis-à-vis des annotations. Coût de calcul élevé. Trop de paramètres à ajuster manuellement
- ⇒ Thick mapping manuel des cartes en ajoutant des couches d'information mathématiques. Coût de calcul élevé. Moins pertinent que l'utilisation d'un réseau convolutionnel
- ⇒ Réalignement direct après binarisation par le filtre de Canny
- ⇒ Extraction automatique des îlots sur les cartes non-annotées avec la transformée de Hough
- ⇒ Evt. post-traitement et simplification des surfaces par l'algorithme de Douglas-Peucker
- ⇒ Océrisation des cartes avec handwriting-ocr

Autres approches à tester

- ⇒ Coupure de l'espace hors-carte (demande d'avoir préalablement segmenté les cartes avec dhSegment)
- ⇒ Estimation de l'échelle par HOG. Contagion des métadonnées d'échelle par similarité HOG
- ⇒ Calcul des points d'amer sur les cartes segmentées uniquement sur les monuments (binarisation des régions rouges)
- ⇒ Identification des monuments spécifiques sur les cartes segmentées et/ou sur les cartes originales par la transformée de Hough généralisée

Bibliographie

- Bellman, Richard E., *Adaptive Control Processes: A Guided Tour* (Princeton, New Jersey: Princeton University Press, 2015)
- Costes, Benoît, 'Vers la construction d'un référentiel géographique ancien : un modèle de graphe agrégé pour intégrer, qualifier et analyser des réseaux géohistoriques', 2016 <<https://tel.archives-ouvertes.fr/tel-01565850>> [accessed 12 November 2019]
- Damerau, Fred J., 'A Technique for Computer Detection and Correction of Spelling Errors', *Commun. ACM*, 7.3 (1964), 171–176 <<https://doi.org/10.1145/363958.363994>>
- Fischler, Martin A., and Robert C. Bolles, 'Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography', *Communications of the ACM*, 24.6 (1981), 381–95 <<https://doi.org/10.1145/358669.358692>>
- Guo, Zicheng, and Richard W. Hall, 'Parallel Thinning with Two-Subiteration Algorithms', *Commun. ACM*, 32.3 (1989), 359–373 <<https://doi.org/10.1145/62065.62074>>
- Howe, Nicholas, Jerod Weinman, and John Gouwar, 'Deformable Part Models for Automatically Georeferencing Historical Map Images' (presented at the SIGSPATIAL 19, Chicago, 2019)
- Levenshtein, V. I., 'Binary Codes Capable of Correcting Deletions, Insertions and Reversals', *Soviet Physics Doklady*, 10 (1966), 707
- Lowe, David G., 'Object Recognition from Local Scale-Invariant Features', in *Proceedings of the International Conference on Computer Vision* (presented at the International Conference on Computer Vision, Corfu, 1999), pp. 1–8 <https://www.academia.edu/20662687/Object_Recognition_from_Local_Scale-Invariant_Features> [accessed 12 November 2019]
- Osher, Stanley, and James A Sethian, 'Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations', *Journal of Computational Physics*, 79.1 (1988), 12–49 <[https://doi.org/10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2)>
- Otsu, Nobuyuki, 'A Threshold Selection Method from Gray-Level Histograms', *IEEE Transactions on Systems, Man, and Cybernetics*, 9.1 (1979), 62–66 <<https://doi.org/10.1109/TSMC.1979.4310076>>
- Rother, Carsten, Vladimir Kolmogorov, and Andrew Blake, "'GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts', in *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04 (New York, NY, USA: ACM, 2004), pp. 309–314 <<https://doi.org/10.1145/1186562.1015720>>
- Welling, Max, 'Fisher Linear Discriminant Analysis', 2007, 4
- Zhou, Xinyu, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and others, 'EAST: An Efficient and Accurate Scene Text Detector', *ArXiv:1704.03155 [Cs]*, 2017 <<http://arxiv.org/abs/1704.03155>> [accessed 12 November 2019]