

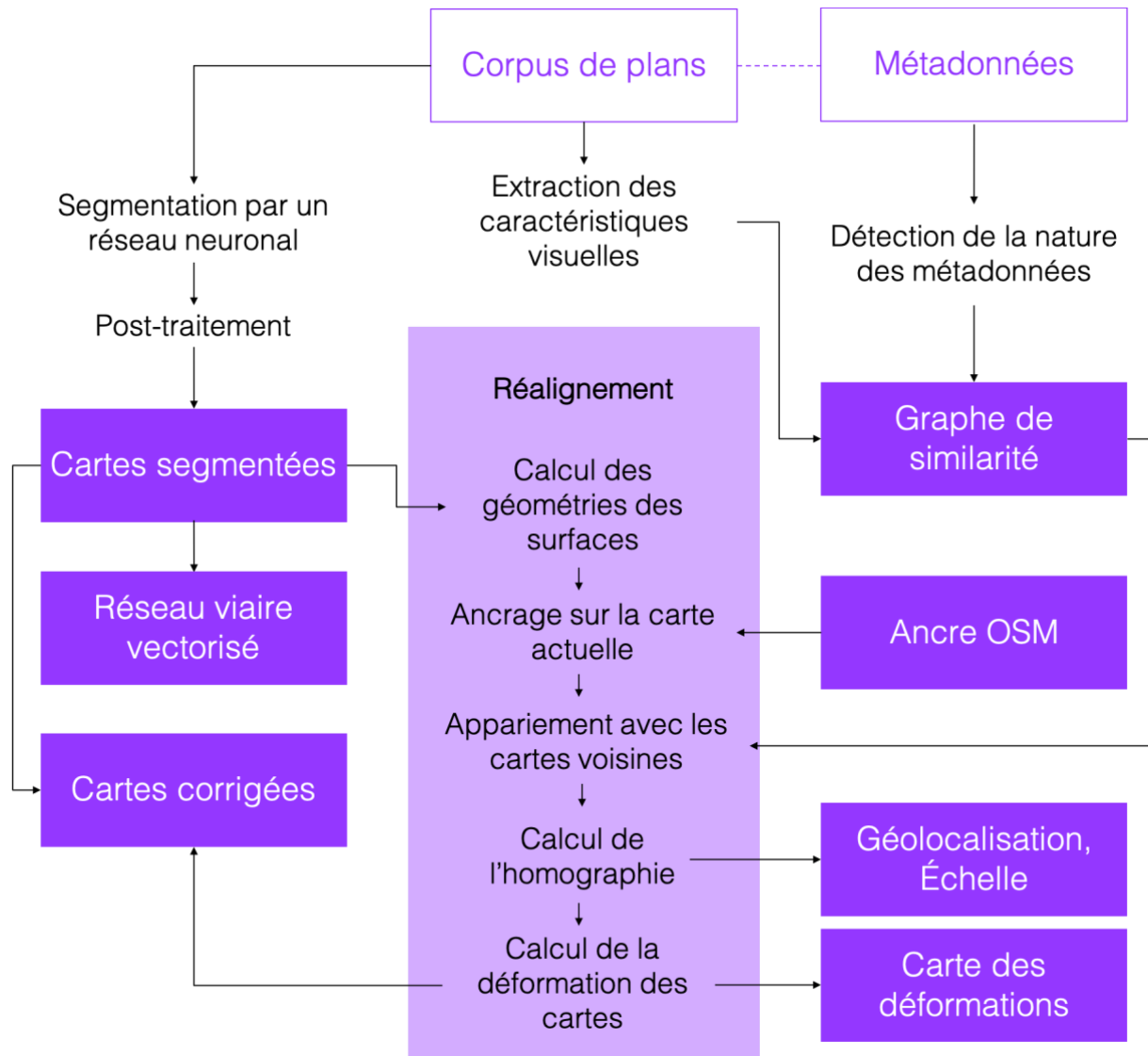
# JADIS

Documentation technique  
vulgarisée

# Table des matières

<b>TABLE DES MATIERES.....</b>	<b>1</b>
<b>VUE GLOBALE DE L'ALGORITHME .....</b>	<b>2</b>
PERFORMANCE.....	3
<b>DÉFINITIONS .....</b>	<b>3</b>
<b>CONSTITUTION DU CORPUS .....</b>	<b>5</b>
REQUÊTE SRU.....	5
RÉCUPÉRATION DES DONNÉES PAR GALICA API .....	5
NETTOYAGE DES DONNÉES.....	5
RÉDUCTION DU CORPUS .....	5
COLLABORATION AVEC LA BIBLIOTHÈQUE HISTORIQUE DE LA VILLE DE PARIS.....	7
<b>PRÉPARATION DES DONNÉES D'ENTRAÎNEMENT .....</b>	<b>7</b>
ANNOTATION .....	7
NETTOYAGE DES DONNÉES.....	7
<b>ENTRAÎNEMENT ET SEGMENTATION .....</b>	<b>8</b>
RÉSEAU NEURONAL .....	8
ENTRAÎNEMENT .....	8
PRÉDICTION .....	9
<b>POST-TRAITEMENT .....</b>	<b>11</b>
NETTOYAGE DU BRUIT.....	11
RÉDUCTION DES DÉFAUTS D'IDENTIFICATION DU CONTOUR DE LA CARTE .....	11
<b>VALIDATION .....</b>	<b>12</b>
<b>RÉSEAU VIAIRE FILAIRE .....</b>	<b>13</b>
EXTRACTION DU RÉSEAU VIAIRE .....	13
VECTORISATION DU RÉSEAU FILAIRE .....	13
<b>CRÉATION DE L'ANCRE.....</b>	<b>15</b>
<b>CALCUL DES SURFACES D'AMER .....</b>	<b>15</b>
<b>APPARIEMENT DES SURFACES D'AMER .....</b>	<b>16</b>
PROCESSUS D'ALIGNEMENT.....	19
CALCUL DES LONGITUDES ET LATITUDES .....	19
<b>DÉFORMATION DES OBJETS-CARTES.....</b>	<b>19</b>
QUANTIFICATION DE LA DÉFORMATION .....	19
TRANSFORMATION ÉLASTIQUE CORRECTIVE .....	19
<b>CALCUL DU RÉSEAU DE SIMILARITÉ .....</b>	<b>21</b>
MÉTADONNÉES .....	21
SIMILARITÉ DES HISTOGRAMMES HOG.....	21
RÉSEAU DE SIMILARITÉ .....	22
<b>BIBLIOGRAPHIE .....</b>	<b>23</b>

## Vue globale de l'algorithme



## Performance

Seules les étapes onéreuses en temps de calcul sont prises en compte dans ce tableau. Les étapes n'y figurant pas ont un temps de calcul considéré comme négligeable, à l'échelle du processus.

TRAITEMENT	TEMPS DE CALCUL
Extraction des caractéristiques visuelles	3,6 s par carte
Découpage et enregistrement des patches	6,3 s par carte
Prédiction par dhSegment	352 s par carte
Reconstitution	6 s par carte
Post-traitement	193 s par carte
Calcul du réseau viaire filaire	50 s par carte
Calcul des caractéristiques invariantes	50 s par carte
Ancrage ou réalignement	30 s par carte
Calcul de la déformation	6 s par carte
Correction de la déformation	5 s par carte

## Définitions

### Réseau neuronal, réseau de neurones :

Le réseau de neurone, tel que nous l'entendons dans ce travail, est un arbre de décision mathématique probabiliste. Il est basé sur de multiples couches de perception de l'information interconnectées.

### Apprentissage, entraînement [du réseau neuronal] :

Les très nombreux paramètres mathématiques du réseau sont optimisés par une phase d'apprentissage supervisé, visant à lui présenter de nombreuses fois des exemples de la tâche que nous souhaitons le voir accomplir. L'apprentissage est empirique. Le réseau neuronal adapte progressivement ses paramètres durant la phase d'apprentissage pour réduire le nombre d'erreurs qu'il effectue par rapport aux exemples d'entraînement.

### Annotation, label :

Le label est une « réponse » correspondant à une image spécifique. Le processus d'annotation consiste à apporter manuellement des réponses à un ensemble d'images de base, pour créer un ensemble d'exemples d'entraînement, composés de couples d'images associées chacune à un label.

### Test :

Phase durant laquelle un réseau neuronal entraîné est sollicité pour formuler une prédiction de réponse sur des images qui ne lui ont jamais été présentées auparavant.

**Evaluation, validation :**

Phase d'étude de la capacité d'un réseau neuronal à généraliser ce qu'il a appris sur des exemples de test

**Sur-apprentissage :**

Le sur-apprentissage est un phénomène qui a lieu lorsque le réseau neuronal se montre incapable de généraliser ce qu'il a appris lors de la phase de test, alors même qu'il est parfaitement capable de reproduire les exemples d'entraînement (apprentissage « par cœur »).

**Segmentation :**

Séparation, extraction des éléments d'une image ou d'une photographie.

**Vectorisation :**

Représentation mathématique de l'information.

**Sémantisation numérique :**

Donner un sens, rendre une information intelligible pour l'ordinateur.

**Canal :**

Couche d'information d'une image concernant une seule couleur. En général, les images possèdent trois canaux RGB/RVB : rouge, vert, bleu.

**Bruit :**

Fluctuation locale indésirable de l'image, due au grain du papier, à une impureté ou au bruit thermique de l'air lors de la numérisation.

**Réseau viaire :**

Réseau routier et voies de transport

**Filaire :**

Représentation de l'information sous la forme de lignes

**Voisinage :**

Proximité d'un élément ou d'une information dans un espace qui peut contenir de très nombreuses dimensions

**Patch :**

Extrait d'une image

## Constitution du corpus

Requête SRU

```
(dc.title all "Paris"
and (dc.type all "carte")
and (gallicapublication_date>="1760")
and ((bibliotheque adj "Bibliothèque nationale de France"))
and (provenance adj "bnf.fr")
&suggest=10&keywords=Paris)
```

Récupération des données par Gallica API

- étiquettes des métadonnées enregistrées : *coverage*, *date*, *description*, *identifier*, *language*, *publisher*, *relation*, *subject*, *source*, *format*, *title*

Nettoyage des données

- L'échelle est extraite de la colonne *description*, avec une expression régulière. Le dénominateur (e.g. 10'000 pour 1:10'000) est stocké dans une colonne à part.
- D'autres métadonnées sont nettoyées, ou extraites au format numérique (langue, géolocalisation, nombre d'items rattachés à l'ark pour les documents de plusieurs pages).
- La longitude et la latitude sont extraites et converties dans un format numérique, dans lequel les chiffres positifs indiquent le nord ou l'est, et les chiffres négatifs indiquent le sud ou l'ouest.

Réduction du corpus

- Lorsque l'échelle est présente, les documents dont l'échelle se situe en dehors des bornes 1:1000 et 1:100'000 sont retirés du corpus.
- Des critères automatiques sont appliqués pour restreindre le corpus :
  - *Paris* doit être contenu dans le sujet
  - La couverture doit contenir *France – Île-de-France – Paris* au moins
  - ni *environs*, ni *perspective* ne doivent être contenus dans le sujet ou dans la couverture
- Les données restantes sont inspectées manuellement et un dernier tri est effectué
- Le jeu de données final contient 882 entrées
- Un tri manuel diminue ce nombre à 772 cartes

Aperçu du jeu de données final :

	coverage	date	description	identifiant	language	publisher	relation	subject	source	format	title
0	France – Île-de-France – Paris	1937	Échelle(s) : Echelle:1:17 000	<a href="https://gallica.bnf.fr/ark:/12148/btv1b531212347">https://gallica.bnf.fr/ark:/12148/btv1b531212347</a>	[fre, français]	impr. Dufrénoy (Paris)	Notice du catalogue : <a href="http://catalogue.bnf.fr/...">http://catalogue.bnf.fr/...</a>	Paris	Bibliothèque nationale de France, département ...	[1 file : en bistre ; 92 x 60 cm, image/jpeg, ...	Paris
1	France – Île-de-France – Paris	1944	Échelle(s) : 1:17 500	<a href="https://gallica.bnf.fr/ark:/12148/btv1b53121232b">https://gallica.bnf.fr/ark:/12148/btv1b53121232b</a>	[fre, français]	Girard et Barrère (Paris)	Notice du catalogue : <a href="http://catalogue.bnf.fr/...">http://catalogue.bnf.fr/...</a>	Paris -- Plans de ville	Bibliothèque nationale de France, département ...	[1 file : en coul. ; 82 x 60 cm, image/jpeg, N...	Paris
2	France – Île-de-France – Paris	1945	Échelle(s) : 1:17 000	<a href="https://gallica.bnf.fr/ark:/12148/btv1b53121231w">https://gallica.bnf.fr/ark:/12148/btv1b53121231w</a>	[fre, français]	Girard et Barrère (Paris)	Notice du catalogue : <a href="http://catalogue.bnf.fr/...">http://catalogue.bnf.fr/...</a>	[Plans de ville, Paris -- Environs]	Bibliothèque nationale de France, département ...	[1 file : en coul. ; 82 x 60 cm, image/jpeg, N...	Paris

scale	ark	geolocalisation	n_items	latitude	longitude	filenames	fold
NaN	btv1b8444727g		8	None	None	maps/BnF /12148_btv1b8444727gf2.jpg	[1, 2, 3, 4, 5, 6, 7]
11100.0	btv1b53085231v		1	None	None	maps/BnF /12148_btv1b53085231vf1.jpg	[0]
10000.0	btv1b53085129n	E 2°21' / N 48°51'54"	1	[48.865]	[2.35]	maps/BnF /12148_btv1b53085129nf1.jpg	[0]

- Les images sont téléchargées grâce à l'API IIIF

## Collaboration avec la Bibliothèque Historique de la Ville de Paris

Dans le cadre d'une collaboration, 740 cartes ont été récupérées des collections numériques de la Bibliothèque Historique de la Ville de Paris (BHVP) pour enrichir le corpus. Ces cartes comprennent principalement les collections thématiques de la BHVP, ainsi que les Atlas municipaux et administratifs, sur une période s'étalant de 1787 à 1989. Les deux tiers du corpus sont ainsi constitués de plans à l'échelle 1: 5000, alors que les cartes du corpus BnF sont en moyenne plus proches d'une échelle 1: 20 000.

## Préparation des données d'entraînement

### Annotation

Au total, 320 patches de 1000x1000 pixels sont découpés aléatoirement parmi les images des cartes. Le but est d'obtenir un ensemble varié de données à annoter. L'annotation se fait ici simplement sur Photoshop.

Trois couleurs pures (les canaux rouge-vert-bleu RGB prennent les valeurs minimales de 0 ou maximales de 255) sont utilisées pour annoter différents éléments du plan :

- En noir, l'extérieur de la carte (légende, bordure, arrière-plan, titre, annotations périphériques) ;
- En blanc, le réseau viaire, en particulier les routes les ponts, mais pas le réseau ferroviaire ;
- En cyan, les îlots, murs, parcs, champs, monuments, cours d'eau, lacs, cours intérieures, et tout le reste.



### Nettoyage des données

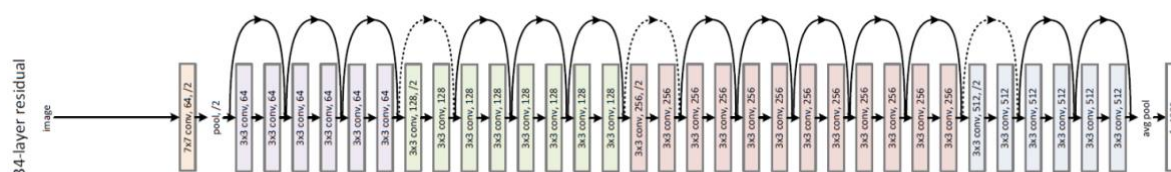
Les données annotées sont nettoyées automatiquement, pour s'assurer qu'elles ne contiennent que des couleurs pures et pour encoder ces couleurs dans un format sRGB. Un algorithme élimine aussi le bruit éventuel (composantes connexes dont la surface est très petite, en particulier les pixels isolés).



## Entraînement et segmentation

### Réseau neuronal

DhSegment<sup>1</sup> est un outil générique de segmentation des documents disponible en libre-accès sur Github. Il est programmé en Python et se base sur la bibliothèque d'apprentissage automatique profond (*deep learning*) de Google, appelée TensorFlow. DhSegment implémente un réseau neuronal convolutif (CNN) très populaire dénommé ResNet<sup>2</sup> (Resnet50V1 en l'occurrence), dont l'architecture est représentée ci-dessous. Le succès de Resnet est dû à un réseau très profond, permettant de disséquer très fortement le problème à résoudre, et surtout à une mémoire des couches plus concrètes, représentées sur le schéma par des flèches bondissantes.



### Entraînement

Pour fonctionner, le réseau neuronal convolutif doit être entraîné sur un set de données, et validé en parallèle sur un set différent. Cela permet une validation scientifique de l'entraînement, en s'assurant que le réseau est capable de généraliser avec succès ce qu'il a appris. Pour notre travail, le set d'entraînement est constitué de 300 patches annotés et le set de validation de 30 patches annotés. Chaque patch doit être constitué d'un label (annoté) et d'une image de référence (non-annotée).

Les données sont augmentées avant leur passage dans le réseau de convolution, c'est à dire qu'à partir d'un seul patch, on en crée plusieurs, de tailles différentes, et en leur appliquant des rotations ou des symétries différentes. Cette augmentation vise à forcer l'algorithme à adopter des critères de décision génériques et à l'empêcher d'apprendre par cœur la position des entités dans l'image (*overfitting*).

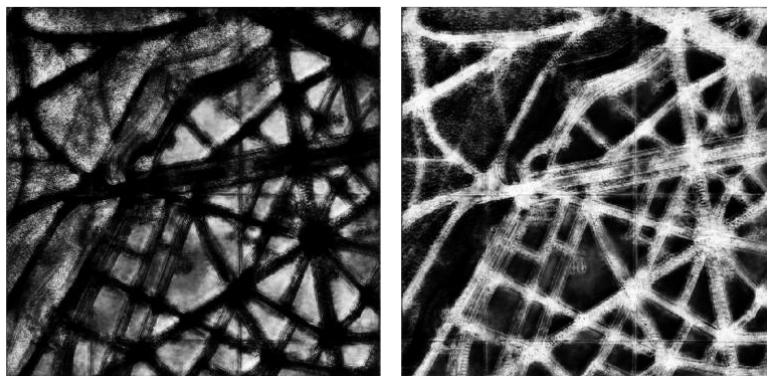
Le but de l'entraînement du réseau neuronal est de créer un arbre de décision complexe et automatique, en fonction de critères tels que la couleur, la texture ou le voisinage, permettant à l'ordinateur d'établir ensuite une prédiction d'annotation pour d'autres cartes. Dans notre cas, cette prédiction d'annotation correspond à une prédiction de segmentation, puisque l'annotation consistait à segmenter les cartes.

<sup>1</sup> Sofia Ares Oliveira, Benoit Seguin, and Frederic Kaplan, 'DhSegment: A Generic Deep-Learning Approach for Document Segmentation', in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)* (presented at the 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY, USA: IEEE, 2018), pp. 7–12 <<https://doi.org/10.1109/ICFHR-2018.2018.00011>>.

<sup>2</sup> Kaiming He and others, 'Deep Residual Learning for Image Recognition', *ArXiv:1512.03385 [Cs]*, 2015 <<http://arxiv.org/abs/1512.03385>> [accessed 12 November 2019].

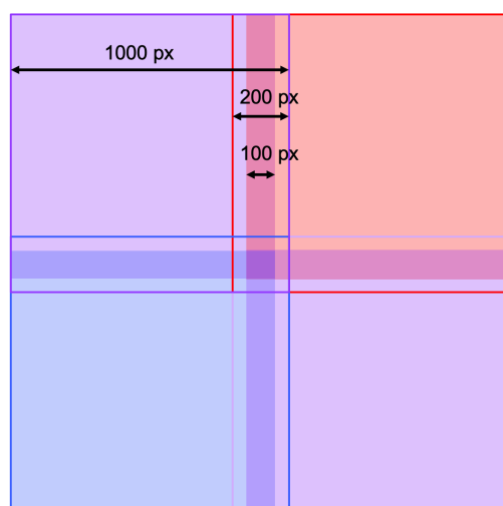
## Prédiction

dhSegment renvoie en sortie une carte de probabilité, définissant pour chaque pixel la probabilité d'appartenir à l'une des régions.

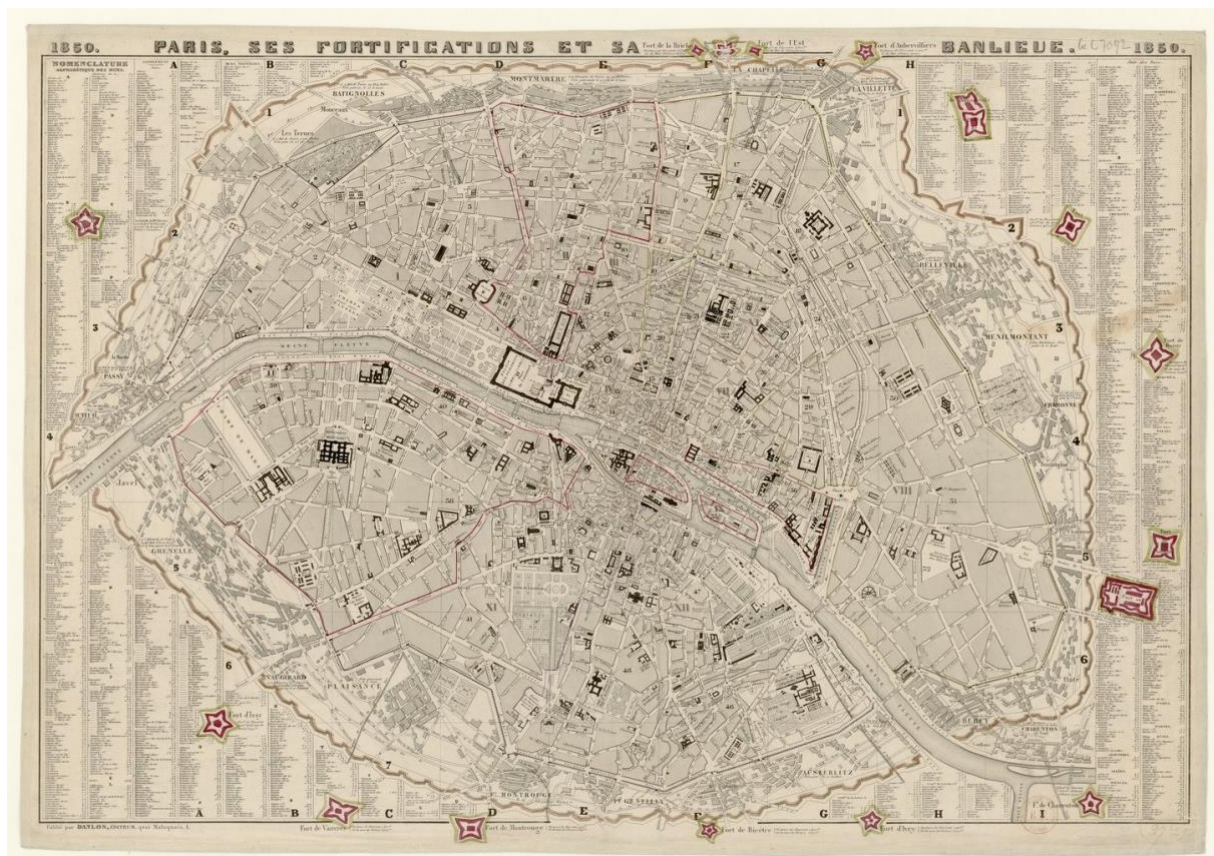


Ci-dessus, la carte de probabilité d'une prédiction effectuée pour une image non-annotée du set de test. À gauche, les zones claires représentent une forte probabilité que le pixel appartienne à un îlot, tandis qu'à droite ces mêmes zones claires représentent la probabilité d'appartenir au réseau viaire.

Comme l'entraînement est effectué sur des patches de 1000 x 1000 pixels, la prédiction doit être effectuée sur des patches similaires, pour donner de bons résultats. Par conséquent, pour appliquer dhSegment sur une carte de 10 000 x 8000 pixels, il est nécessaire de découper cette dernière en patches plus petits (de 1000 x 1000 pixels). Pour éviter l'effet de bord, les patches se recouvrent sur une tranche de 200 pixels de côté. Comme l'effet de bord rend la prédiction peu fiable sur les côtés, le patch suivant prend le relais sur les 50 derniers pixels et les 100 précédents font l'objet d'une prédiction conjointe sur 2 patches. Seuls les bords réels de l'image originale sont ainsi sujets à l'effet de bord



Voir en page suivante un exemple de la segmentation automatique d'une des cartes du corpus par dhSegment, après post-traitement.



Source gallica.bnf.fr / Bibliothèque nationale de France



## Post-traitement

### Nettoyage du bruit

Le post-traitement vise à nettoyer la prédiction du réseau de neurones. Au niveau rapproché, les erreurs sont principalement constituées par le bruit et les pixels isolés. Les pixels isolés sont retirés et l'image est lissée en appliquant dix fois un filtre de floutage très local (fenêtre de 3x3 pixels), suivi d'un seuillage, pour éliminer le flou.

Dans un deuxième temps, les petites composantes (jusqu'à une aire de 100 pixels) sont retirées. Pour des raisons de coût de calcul, cette deuxième étape est appliquée sur des patches découpés de l'image. L'image doit ensuite être reconstituée. Ce mode de procéder engendre un coût qualitatif, puisque le processus est sujet aux effets de bord. Pour retirer les petites composantes, on les soustrait de l'image, ce qui laisse un « trou ». Ce « trou » va ensuite être « bouché », dans un premier temps en dilatant localement les composantes connexes voisines. Intuitivement, on pourrait considérer que les surfaces voisines sont étendues sur la surface du trou. Le trou résiduel est ensuite refermé entièrement avec un algorithme de fermeture d'aire<sup>3</sup>.

### Réduction des défauts d'identification du contour de la carte

Une fois les bruits nettoyés, il est temps de s'intéresser aux défauts présents à l'échelle de la carte. Nous allons présumer pour cela que la carte n'est constituée que d'une seule image connexe et que la bordure est extérieure à celle-ci. Cette étape est nécessaire principalement parce que nous avons décomposé la carte en morceaux avant de la présenter au réseau de neurones. Si nous avions une puissance de calcul beaucoup plus importante, le réseau de neurone aurait pu être entraîné sur des cartes entières et il aurait normalement reconnu cette propriété inhérente à la bordure, de se trouver proche du bord de l'image. Dans notre cas, le réseau de neurone ne peut pas comprendre cette information car l'image telle qu'elle lui est présentée est déstructurée. Une alternative consisterait à présenter une vignette de la carte à un second réseau de neurones pour identifier la bordure automatiquement.

L'interface bordure/carte est lissée par une série de filtres de floutage moyens (15x15 pixels), suivis d'un seuillage. Les composantes connexes incluses dans les composantes de la bordure sont retirées. Finalement, les composantes de la bordure n'étant pas reliées à un coin de l'image sont retirées également. Ces dernières sont remplacées par la couleur majoritaire, au sein des composantes voisines.

---

<sup>3</sup> Luc Vincent, 'Grayscale Area Openings and Closings, Their Efficient Implementation and Applications' (presented at the EURASIP Workshop on Mathematical Morphology and its Applications to Signal Processing, Barcelona, Spain, 1993), pp. 22–27.

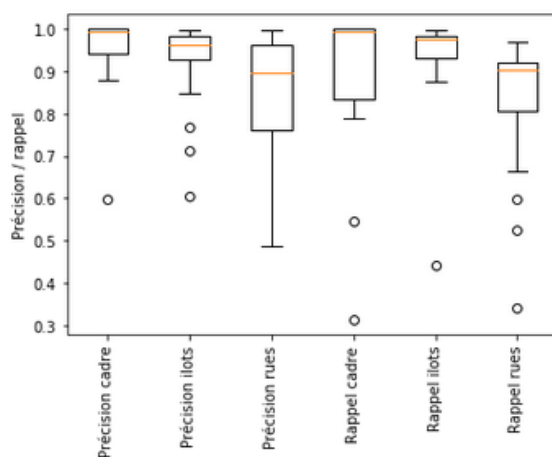
## Validation

Pour valider la qualité des différents modèles entraînés et des étapes de post-traitement, la précision et le rappel sont calculés sur un set de données de validation qui a été annoté, mais ne fait pas partie des données sur lesquelles le réseau de neurones a directement été entraîné. La précision est une métrique qui représente, pour chaque couleur de l'image prédite, la proportion de pixels qui ont été correctement classés par le processus automatique. Le rappel, à l'inverse, mesure la proportion de pixels qui ont été correctement classés par rapport au nombre de pixels de chaque couleur dans l'image annotée de validation.

Le dernier réseau entraîné obtient les résultats imprimés ci-dessous. On notera que les résultats médians sont souvent meilleurs que les résultats moyens. Cela signifie que plus de la moitié du corpus obtient de bons résultats. Cependant, une petite partie du corpus, environ 5-10%, obtient de mauvais résultats, ce qui tire la moyenne vers le bas. C'est la raison pour laquelle la médiane est plus adaptée pour mesurer l'efficacité de l'algorithme, tout en gardant à l'esprit qu'une petite partie des cartes ne peuvent pas être segmentées convenablement.

Pour reprendre les quelques éléments les plus importants, le rappel des rues (90.2%), signifie que 9 pixels de rue sur 10 sont détectés. La précision des îlots (96.4%), indique qu'un pixel sur 30 est classifié à tort comme faisant partie d'un îlot. L'extérieur de la carte est très bien identifié, seuls 1 pixel sur 200 n'est pas reconnu ou est mal classifié.

```
Précision cadre. Median: 99.427% Mean: 93.469%
Précision ilots. Median: 96.390% Mean: 92.904%
Précision rues. Median: 89.640% Mean: 85.340%
Total. Median: 94.614% Mean: 89.781%
Rappel cadre. Median: 99.494% Mean: 86.158%
Rappel ilots. Median: 97.693% Mean: 94.266%
Rappel rues. Median: 90.189% Mean: 83.852%
Total. Median: 93.248% Mean: 88.619%
```



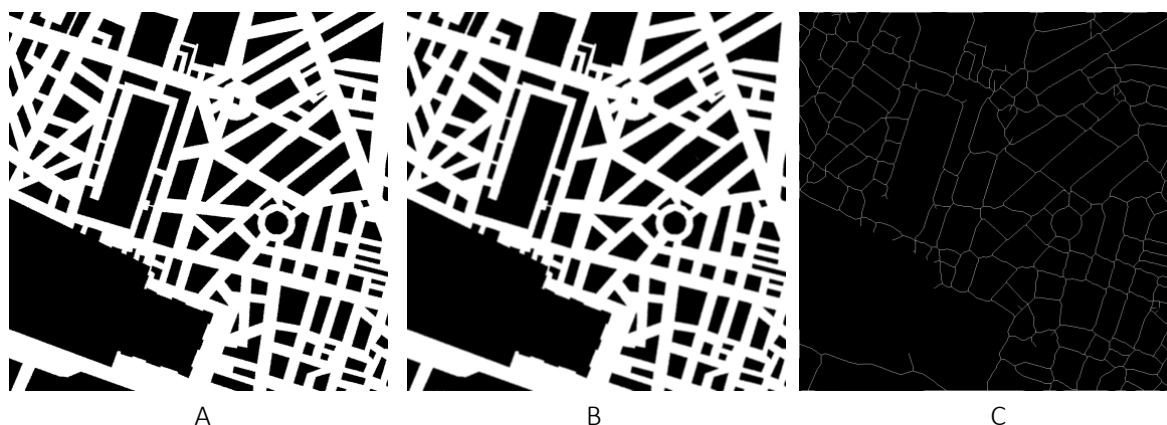
300 img, full auto aug, 25 epochs, post-treated, trained in one shot



## Réseau viaire filaire

### Extraction du réseau viaire

À ce stade, cette étape est testée sur une image annotée manuellement, comme preuve de concept. L'image est transformée en format binaire noir-blanc (A), avec le réseau viaire en blanc. Une paire de filtres de floutage, sans seuillage, permettent de lisser les contours (B). Dans un deuxième temps, l'image est squelettisée pour n'obtenir qu'un réseau filaire (C).

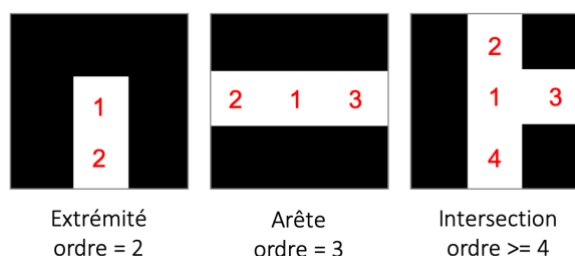


### Vectorisation du réseau filaire

La vectorisation automatique a été implémentée pour 4 usages :

- export .png pour la visualisation
- export .json pour l'utilisation sur internet ou par des algorithmes de programmation
- export .shp pour l'utilisation sur les SIG (QGIS)
- export .svg pour l'illustration, le partage ou la visualisation (Illustrator, Inkscape)

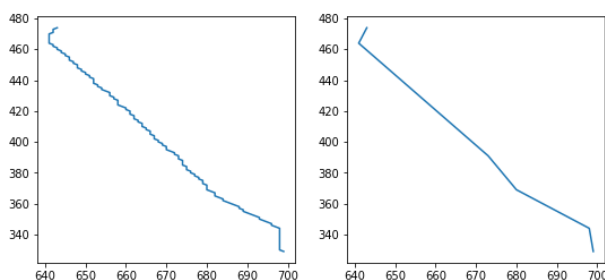
Pour établir cette vectorisation, un algorithme repère automatiquement les nœuds (intersections et extrémités) de l'image représentant le squelette. Pour chaque pixel, une fenêtre 3x3 du pixel et de ses voisins directs est considérée. Si dans cette fenêtre, deux pixels font partie du squelette, le pixel est un nœud et une extrémité terminale. Si 4 pixels ou plus font partie du squelette, le pixel est un nœud et une intersection. Si 3 pixels font partie du squelette, le pixel se trouve simplement le long d'une arête.



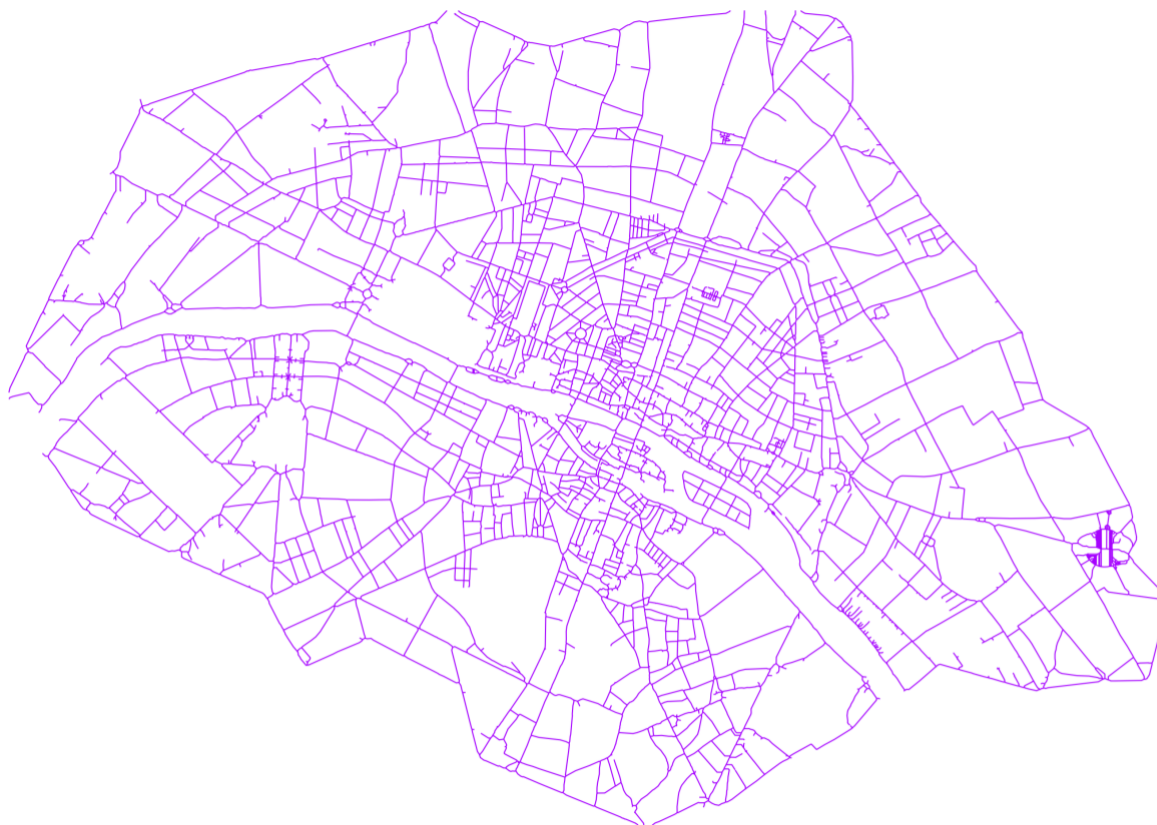
Le réseau vectoriel est ensuite construit en parcourant chaque nœud jusqu'à rencontrer un autre nœud, jusqu'à ce que tous les chemins aient été parcourus. Lorsque l'arête part d'un nœud terminal et qu'elle est plus courte qu'un certain seuil, elle est considérée comme un « poil » et est supprimée. Les réseaux

« poilus » sont formés lorsque des contours de rues, encore trop découpés malgré le lissage, sont interprétés à tort comme des arêtes du réseau.

Le réseau filaire est ensuite simplifié et allégé, grâce à l'algorithme de Douglas-Peucker<sup>4</sup> qui supprime le bruit dû à la granularité des pixels, et réduit la complexité de la courbe pour ne conserver que les changements d'orientation saillants (voir ci-dessous).



Ci-dessous, un exemple d'un réseau viaire filaire vectorisé automatiquement et exporté sous QGIS.



<sup>4</sup> David H Douglas and Thomas K Peucker, 'Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature', *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10.2 (1973), 112–22 <<https://doi.org/10.3138/FM57-6770-U75U-7727>>.

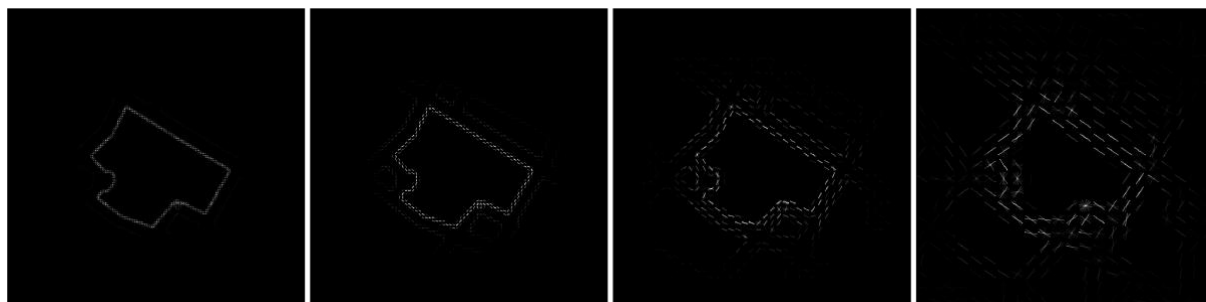
## Création de l'ancre

L'ancre est la représentation actuelle géolocalisée de la ville. C'est en réalignant les autres cartes sur l'ancre qu'il est possible de déduire la géolocalisation de ces dernières. L'ancre est générée automatiquement à partir du nom de la ville, entré dans les paramètres de l'algorithme. L'algorithme fait appel à l'API Overpass d'OpenStreetMap pour récupérer les coordonnées et les métadonnées des rues actuelles. Une carte approximative est générée, dans laquelle la largeur est estimée par rapport à la catégorie de la route sur OpenStreetMap. La carte est corrigée selon une projection pseudocylindrique « équivalente » (c'est-à-dire que les aires sont conservées), dite de Sanson-Flamsteed. Cette projection est centrée spécialement sur la ville, ce qui la rend également quasi-équidistante, a fortiori à grande échelle. Les surfaces ne sont donc pratiquement pas déformées. Cette projection équivaut à considérer que la Terre est plate localement, à l'emplacement de la ville.

## Calcul des surfaces d'amer

En général, pour résoudre un problème de géolocalisation, des 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.

Pour adapter ce paradigme aux cartes, nous allons considérer que chaque îlot est une surface d'amer potentielle. La raison pour laquelle les cartes ont dû être segmentées réside dans ce qu'elles ont été rendues intelligibles. La segmentation est une sémantisation numérique de la carte. Chaque bloc peut donc être extrait indépendamment, comme une composante connexe unique, et ses propriétés géométriques peuvent être calculées, puis comparées.

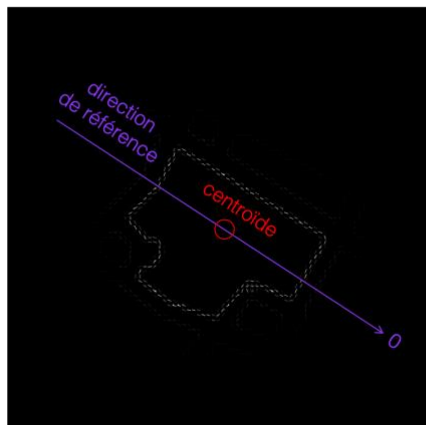


À nouveau, HHOG (double histogramme du gradient orienté) est utilisé pour transformer la géométrie en vecteur mathématique. Pour rendre les caractéristiques invariantes aux changements d'échelle ou de résolution, elles sont calculées pour 4 échelles différentes au sein d'une même carte avec un noyau variable, doublé à chaque itération. En théorie, cela permet donc d'apparier des cartes dont l'échelle ou la résolution varie d'un facteur 8. Par exemple, il serait possible d'apparier une carte 1:10 000 avec une carte 1:80 000. Evidemment, les paramètres de l'algorithme peuvent être modifiés pour ajouter une itération supplémentaire selon les besoins.

Pour rendre ces caractéristiques encore plus robustes, elles doivent être rendues invariantes à la rotation. Pour ce faire, la direction dominante du vecteur HHOG est identifiée et fixée comme direction



de référence, ou direction « zéro ». Grâce à cela, il sera possible d'apparier les îlots de deux cartes qui n'ont pas la même orientation par rapport au nord.



Les centres géométriques (centroïdes) des îlots sont évidemment stockés, pour permettre de calculer l'homographie, soit la formule de réaligement entre les deux cartes.

Au final, les données extraites pour chaque îlot ressemblent à cela :

- Centroïde (*coordonnées en pixel du centre géométrique de l'îlot*) : (2505, 5039)
- Direction (*par tranche de 20°, numérotées*) : 6
- Caractéristiques (*proportion normalisée des orientations*) : (3.34, 1.43, 0.01, 0.02, 0.00, 0.00, 0.03, 2.42, 0.00, 0.00, 0.32, 0.00, 0.00, 0.01, 0.89, 0.00, 1.53, 0.02)

## Appariement des surfaces d'amer

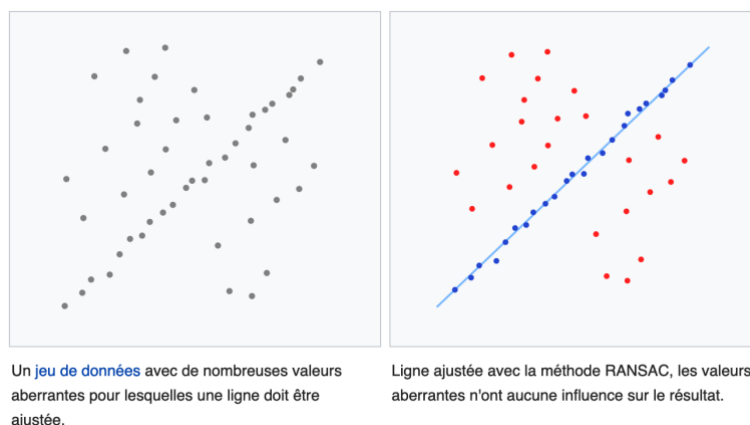
Le réaligement des cartes repose sur l'appariement de surfaces d'amer, soit des îlots. Les surfaces d'amer de l'image A sont appariées à leurs deux plus proches voisines de l'image B. Il faut s'imaginer le voisinage comme l'extension de la notion de voisinage géographique à 2 ou 3 dimensions, dans un espace à 18 dimensions (nombre dû aux 18 tranches de 20° d'orientations des HHOG), où la distance ne dépend plus du tout de la position géographique sur la carte, mais des caractéristiques géométriques des surfaces. La proximité de deux surfaces correspond donc à une mesure de leur similarité géométrique.

La raison pour laquelle les deux plus proches surfaces voisines sont relevées (et pas uniquement la plus proche), est que la forme de certains îlots est très commune. Par conséquent, il faut s'assurer que la géométrie de l'objet apparié est suffisamment atypique pour que l'appariement soit digne de confiance. Pour résoudre ce problème, il faut vérifier que l'appariement privilégié (*i.e. le meilleur*) est nettement meilleur que les suivants. On contrôle donc que la première surface voisine est au moins 5-25 % plus proche de la surface à appairer que la deuxième. Ce ratio, adaptable selon le problème, porte le nom de ratio de Lowe<sup>5</sup>.

<sup>5</sup> 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](https://www.academia.edu/20662687/Object_Recognition_from_Local_Scale-Invariant_Features)> [accessed 12 November 2019].

Puisque nous avons à chaque fois calculé les caractéristiques des îlots pour 4 échelles différentes, cette première partie est répétée pour chaque paire d'échelle possible (donc  $4 \times 4 = 16$  fois).

Dans un second temps, l'algorithme de Ransac<sup>6</sup> (*RAN*dOm *S*Ample *C*onsensus) est appliqué, pour éliminer les valeurs aberrantes. L'idée de l'algorithme de Ransac est de séparer le bon grain de l'ivraie en identifiant, parmi les nombreux appariements proposés, ceux qui font partie du plus grand ensemble cohérent. Un seuil de tolérance élevé est fixé (env. 150-250 pixels d'écart toléré), car nous savons que les cartes d'origine contiennent des imprécisions et peuvent être déformées.



Msm – Wikimedia

Lorsque les meilleurs appariements ont enfin été isolés, il est possible de calculer la fonction homographique, qui représente l'application projective entre les deux cartes, ou plus simplement, la formule de transformation entre une carte et l'autre. Cette formule correspond simplement, en fait, à la formule décrivant le plus grand ensemble cohérent identifié par Ransac. Dans notre cas, le champ de recherche est exceptionnellement restreint par deux conditions que nous allons imposer :

1. Les cartes sont des objets idéaux en deux dimensions
2. L'image de la carte B sur la carte A est rectangulaire, et conserve ses proportions

Un dernier problème subsiste cependant. Nous ne savons en effet pas si la transformation obtenue est correcte et rien dans l'algorithme de Ransac ni dans le ratio de Lowe ne permet d'identifier les erreurs. Par conséquent, nous allons devoir développer un score ou in spécialement pour notre problème, qui doit être en mesure de vérifier la cohérence de la transformation.

Puisque nous connaissons l'orientation initiale de tous les éléments de la carte A, et de même pour la carte B, et que la transformation entre les cartes peut impliquer une rotation, nous allons vérifier la cohérence de ces deux indices. Ainsi, s'il faut pivoter de  $20^\circ$  pour passer d'une carte à l'autre, les éléments appariés doivent aussi avoir été déplacés du même angle. Le pourcentage d'appariements cohérents est enregistré.

<sup>6</sup> 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>>.

D'autre part, les différences d'échelle entre îlots appariés doivent correspondre à la différence d'échelle entre la carte B et la projection de la carte A sur cette dernière.

Finalement, le nombre d'appariements doit être aussi grand que possible. Une pénalité est ajoutée, lorsque leur nombre est trop bas, agissant comme un régulariseur.

$$incoherence = orientation_{incoherence} + scale_{incoherence} + regularizer$$

$$orientation_{incoherence} = 1 - \left( \frac{\sum \mathbb{I}(\theta_i \equiv \alpha \pm 20^\circ)}{N_i} \right)^3$$

$\alpha$ , la rotation imposée par la transformation

$\theta_i$ , la différence d'orientation entre deux îlots appariés

$N_i$ , le nombre d'appariement conservés

$$scale_{incoherence} = \frac{1}{2} \left| \log_2 \left[ \frac{h}{h'} \frac{w}{w'} \right] - \bar{\delta}_i \right|$$

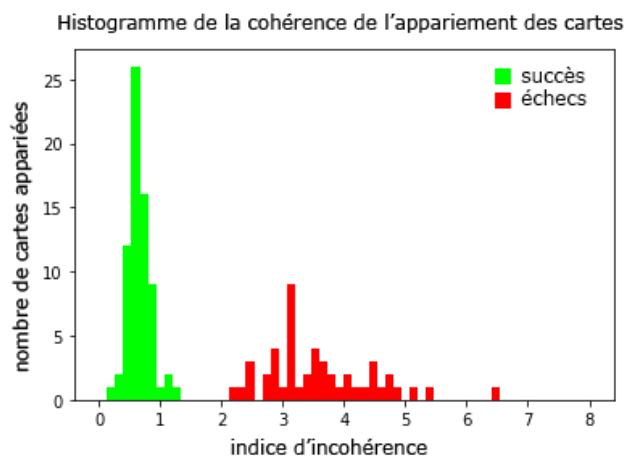
$h, w$ , la hauteur et la largeur de la carte B

$h', w'$ , la hauteur et la largeur de la projection de la carte A sur la carte B

$\delta_i$ , la différence d'échelle entre deux îlots appariés

$$regularizer = \frac{2}{\ln N_i}$$

$$\text{soit } incoherence = 1 - \left( \frac{\sum \mathbb{I}(\theta_i \equiv \alpha \pm 20^\circ)}{N_i} \right)^3 + \frac{1}{2} \left| \log_2 \left[ \frac{h}{h'} \frac{w}{w'} \right] - \bar{\delta}_i \right| + \frac{2}{\ln N_i}$$



En mesurant l'efficacité de l'indice de cohérence sur un set de 50 cartes, on remarque qu'il est efficace pour dissocier les cartes correctement appariées des échecs. On fixe un premier seuil à 1, en dessous duquel les cartes sont fiables pour un éventuel appariement secondaire, et à 1.5, au-delà duquel l'appariement est considéré comme un échec.

## Processus d'alignement

Dans un premier temps, chaque carte est réalignée à l'ancre. Si le score est bon (incohérence inférieure à 1), la carte obtient un statut d'ancre secondaire. Dans un deuxième temps, les cartes qui n'ont pas pu être réalignées à l'ancre initiale sont réalignées, par ordre de priorité avec les 10 ancres secondaires les plus proches d'elles dans le réseau de similarité. Le processus s'arrête dès que la carte à réaligner a été liée à au moins une ancre secondaire, ou lorsque les 10 tentatives ont échoué.

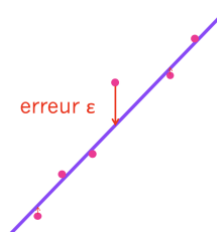
## Calcul des longitudes et latitudes

La position, en pixels, de la carte à géolocaliser, par rapport à l'ancre est convertie en longitudes et latitudes en inversant la formule utilisée pour la projection de l'ancre. Le résultat est un quadruplet de coordonnées, correspondant aux 4 coins de l'image de la carte.

## Déformation des objets-cartes

### Quantification de la déformation

Les objets cartes peuvent être déformés pour plusieurs raisons : imprécision lors des relevés, imprécision lors de la gravure, choix esthétique, grain du papier, déformation du support (humidité, plis, usure). Il est nécessaire de quantifier ces déformations, souvent locales, pour pouvoir dans un deuxième temps réaligner correctement les cartes entre elles. La quantification de la déformation peut se faire lors du processus d'alignement initial, et permet d'ajuster cet alignement.



Les erreurs verticales et horizontales sont calculées. L'erreur correspond à l'écart par rapport à l'homothétie (fonction de projection) calculée. Elle mesure l'écart entre la position théorique des points (ou des surfaces) et leur position effective.

### Transformation élastique corrective

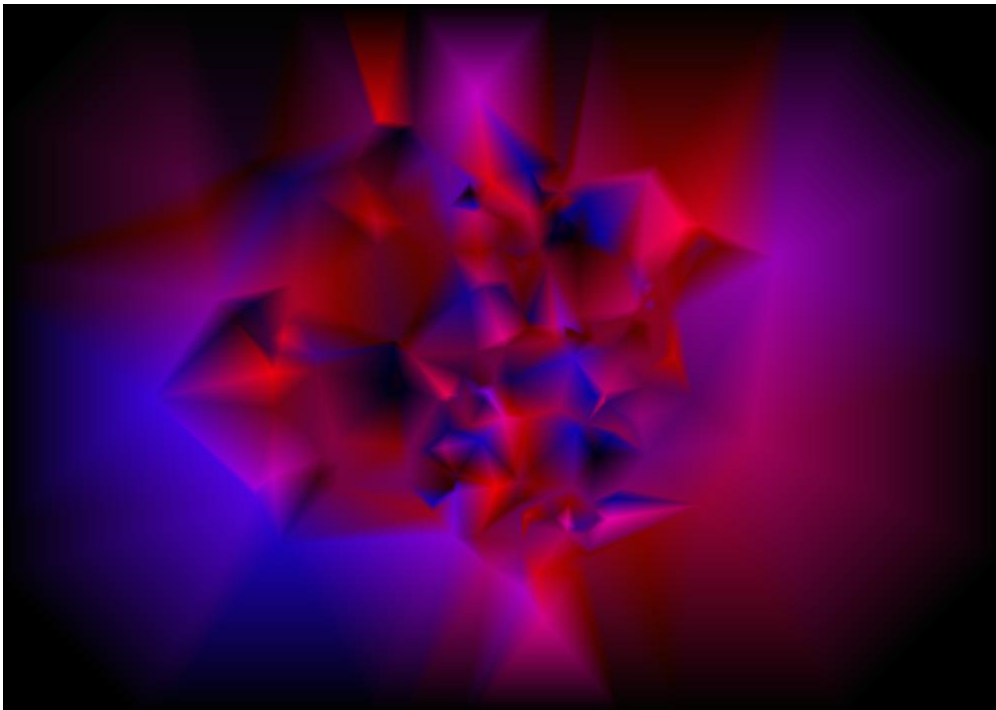
À terme, le but est de pouvoir superposer deux images de cartes pour les comparer, et notamment comparer l'évolution urbanistique de la ville. Dans cette optique, la possibilité de corriger la déformation d'une carte par rapport à une autre doit être implémentée, afin que chaque îlot se superpose correctement à son homologue.

Les points d'amer sont ancrés au cadre de l'image par des points automatiquement équirépartis, en moyenne tous les 100 pixels. Un maillage, composé de surfaces triangulaires ne s'intersectant jamais

est créé. Le champ des déformations ponctuelles (les points correspondent aux centroïdes des surfaces appariées), calculé précédemment, est étendu linéairement en tout point de l'image (i.e. pour chaque pixel), grâce à une équation linéaire définie pour chaque surface triangulaire. En clair, cela signifie que pour chaque surface appariée, une déformation est calculée et qu'entre deux surfaces, une transition linéaire permet de combler le manque d'informations.

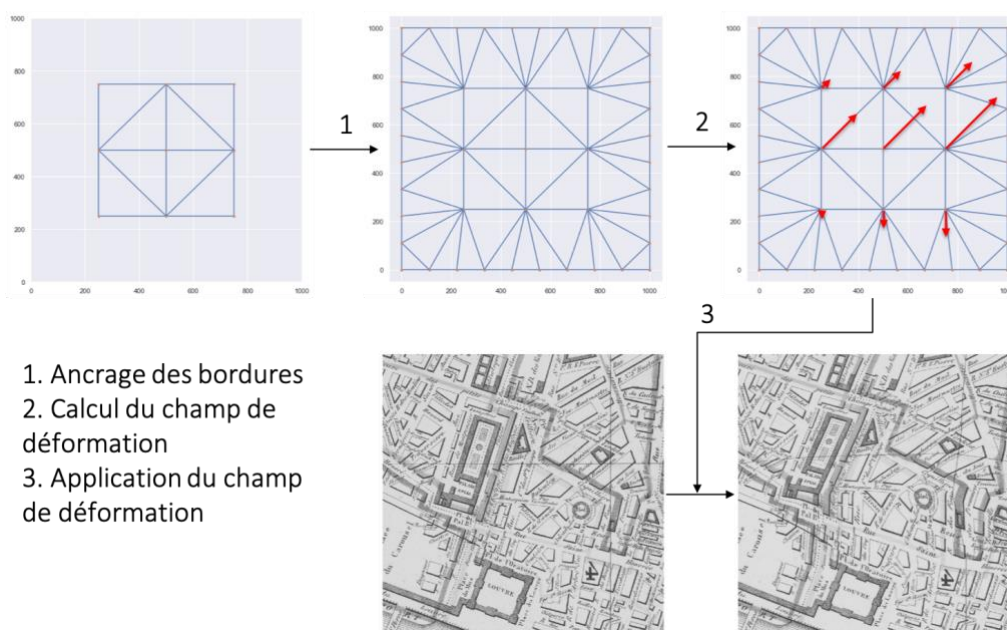
La transformation consiste ensuite simplement à appliquer le champ de déformation, c'est-à-dire à déplacer chaque pixel d'autant d'unités que calculé dans le champ. Le champ de déformation peut être compris intuitivement comme une liste 2D de la transformation à effectuer pour chaque pixel.

Cette carte étendue des déformations, qui peut être utilisée à des fins correctives, est stockée sous la forme d'une image. Le nombre de pixels de déplacement correspond à l'intensité des canaux de couleur bleus (= déformation verticale) et rouges (= déformation horizontale).



Pour prendre un exemple, un pixel A, de coordonnées (100, 100), se trouvant dans un triangle dont l'équation de déformation  $(dx, dy) = (ax + b, cy + d)$  a pour coefficients  $a = 0.2$ ,  $b = 5$ ,  $c = 0.1$ ,  $d = -15$  prendra une valeur de  $(dx, dy) = (0.2 \cdot 100 + 5, 0.1 \cdot 100 - 15) = (25, -5)$  dans le champ de déformation. Dans l'image finale, le pixel B situé en  $(100+25, 100-5) = (125, 95)$  prendra la valeur ( $\approx$  couleur) du pixel A de l'image d'origine.

## Déformation élastique corrective



## Calcul du réseau de similarité

L'intérêt de lier les plans réside dans la réduction du temps de calcul. En effet, le succès du réaligement d'une carte est plus grand avec des cartes qui lui ressemblent. Il est donc plus judicieux de réaligner chaque plan avec les X plans qui lui sont les plus proches qu'avec les tous les plans à chaque fois. Un gain de temps de l'ordre de 60x est facilement atteignable sur un corpus de 1500 cartes.

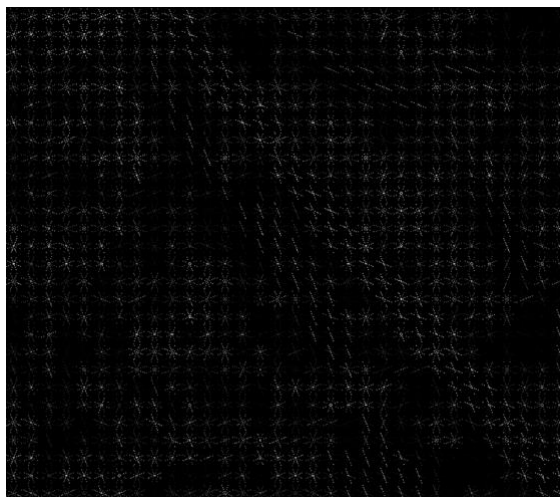
### Métadonnées

Les cartes sont réalignées selon deux types de critères : les métadonnées et la similarité des images. Pour les métadonnées, les critères retenus sont le titre, l'année, la maison d'édition, les sujets, l'échelle, la ville de publication et la langue. Pour éviter les égalités, un nombre arbitraire très petit est ajouté. Le poids des critères n'est pas équivalent. Pour les critères retenus, les poids sont respectivement de 4 pour le titre, 2 pour l'année et pour la maison d'édition, 1 par sujet commun, puis 0.5 pour les suivants.

### Similarité des histogrammes HOG

L'indice de similarité des images est obtenu par calcul de la corrélation (coefficient de Pearson) des histogrammes des HOG<sup>7</sup> (*histogram of oriented gradients* ou histogramme de gradient orienté). HOG est une représentation mathématique des images, visant à décrire l'orientation générale de l'image, en une série de points distribués régulièrement sur une grille. Cette méthode est utilisée notamment pour la reconnaissance des visages. Ci-dessous, le HOG (précision de 45°) d'une carte du XIXe, centré sur Tolbiac. L'orientation de la Seine est très clairement visible.

<sup>7</sup> Robert K. McConnell, 'Method of and Apparatus for Pattern Recognition', 1986 <<https://patents.google.com/patent/US4567610/en>> [accessed 12 November 2019].

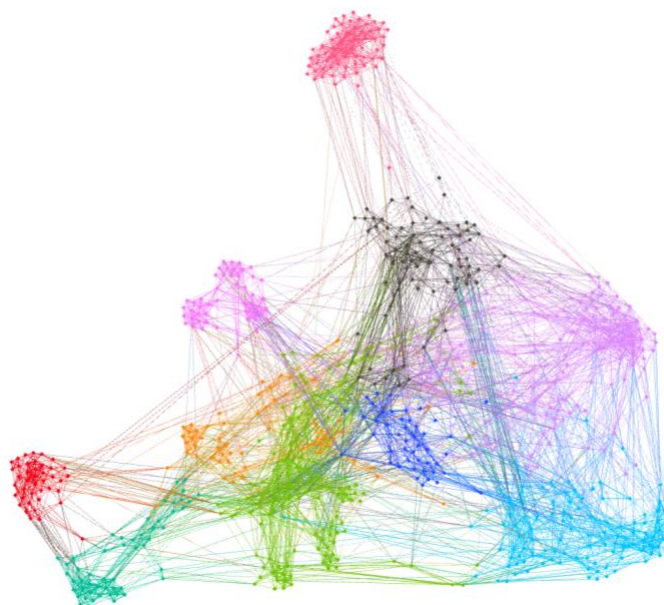


Faire l'histogramme de ce HOG consiste à sommer les orientations calculées uniformément sur toute l'image, pour chaque angle d'orientation (pour le projet, tous les  $2^\circ$ ). Cet histogramme est calculé pour chaque carte (non-annotée), puis comparé, pour trouver les cartes les plus ressemblantes. Pour le chaînage, un coefficient de 5 est attribué à cet indice de similarité.

### Réseau de similarité

Une version plus cohérente du chaînage a été implémentée sous la forme d'un réseau de similarité. Selon cette approche, chaque carte est liée à ses X plus proches voisines, à l'intérieur d'un graphe directionnel. Les relations peuvent donc ne pas être symétriques.

Le réseau de similarité du corpus peut aussi être exporté en format Gephi pour permettre au chercheur d'effectuer des méta-analyses du corpus. Ci-dessous, le graphe de similarité des cartes du corpus BnF, reliées à leur N=10 voisines les plus proches et classées par communautés par l'algorithme de Louvain<sup>8</sup>.



<sup>8</sup> Vincent D Blondel and others, 'Fast Unfolding of Communities in Large Networks', *Journal of Statistical Mechanics: Theory and Experiment*, 2008.10 (2008), P10008 <<https://doi.org/10.1088/1742-5468/2008/10/P10008>>.



Ci-dessous, trois cartes extraites d'un même cluster du réseau de similarité.



## Bibliographie

- Ares Oliveira, Sofia, Benoit Seguin, and Frederic Kaplan, 'DhSegment: A Generic Deep-Learning Approach for Document Segmentation', in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)* (presented at the 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY, USA: IEEE, 2018), pp. 7–12 <<https://doi.org/10.1109/ICFHR-2018.2018.00011>>
- Blondel, Vincent D, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre, 'Fast Unfolding of Communities in Large Networks', *Journal of Statistical Mechanics: Theory and Experiment*, 2008.10 (2008), P10008 <<https://doi.org/10.1088/1742-5468/2008/10/P10008>>
- 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>>
- Douglas, David H, and Thomas K Peucker, 'Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature', *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10.2 (1973), 112–22 <<https://doi.org/10.3138/FM57-6770-U75U-7727>>
- 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>>
- He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep Residual Learning for Image Recognition', *ArXiv:1512.03385 [Cs]*, 2015 <<http://arxiv.org/abs/1512.03385>> [accessed 12 November 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](https://www.academia.edu/20662687/Object_Recognition_from_Local_Scale-Invariant_Features)> [accessed 12 November 2019]
- McConnell, Robert K., 'Method of and Apparatus for Pattern Recognition', 1986 <<https://patents.google.com/patent/US4567610/en>> [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>>
- Vincent, Luc, 'Grayscale Area Openings and Closings, Their Efficient Implementation and Applications' (presented at the EURASIP Workshop on Mathematical Morphology and its Applications to Signal Processing, Barcelona, Spain, 1993), pp. 22–27
- 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]