

# Vectorisation stylisée de dessins d'un manuscrit copte du 14<sup>e</sup> siècle par Algorithme de *tracing*

Franck Jeannot

Montreal, Canada, Février 2017, I259, v1.1

---

## Abstract

One of the most important skill that helped Jean-François Champollion to decipher Egyptians hieroglyphics was to master the coptic language. Having a look to some coptic manuscripts, I had a closer look in ways to reuse some beautiful drawings from these manuscripts and identified solutions to render them in a vector outline **EPS (Encapsulated Postscript)** format, in order to be re-used in various  $\LaTeX$  articles with a neat layout. I especially used *tracing* algorithms and **Potrace**<sup>1</sup> tool.

*Keywords:* EPS, Potrace, tracing, Inkscape, bitmap, vector, Bezier curves, Courbes de Bézier, coptic, manuscript

---

## 1. Introduction

Les images peuvent être représentées sous forme de bitmap ou de *contour vectoriel*<sup>2</sup>. Une image bitmap représente une image sous la forme d'une grille de pixels. Un *contour vectoriel* décrit une image par une description algébrique de ses contours, typiquement sous la forme de *courbes de Bezier*<sup>3</sup>. L'exercice consiste, à partir d'un fichier image non vectoriel, par exemple au format JPEG, de créer un format vectoriel au plus proche de l'original, ici au format EPS,<sup>4</sup> grâce aux outils Potrace et Inkscape.

## 2. Inkscape et Potrace

**Inkscape**<sup>5</sup> est un logiciel libre de dessin vectoriel sous licence GNU GPL. Les versions récentes intègrent l'outil **Potrace** [1] qui est un moteur de vectorisation d'images matricielles créé

par Peter Selinger<sup>6</sup> utilisant un **algorithme de tracing**<sup>7</sup> de type « *polygon-based* ». L'algorithme Potrace vient de la contraction *polygone tracer*.

## 3. Résultats

### 3.1. Oiseau



L'oiseau vectorisé ci-dessous est réutilisable à l'infini, de n'importe quelle dimension, sans effet de pixellisation à l'agrandissement, c'est bien un *dessin vectoriel*, dans le cas échéant au format PS-Adobe-3.0 EPSF-3.0.



FIGURE (1): Oiseau vectorisé stylisé en tailles différentes. Ref page 9 du manuscrit source [2]

---

2. suite de points qui déterminent un contour  
3. courbes polynomiales paramétriques décrites pour la première fois en 1962 par Pierre Bézier  
4. <http://www.fileformat.info/format/eps/egff.htm>  
5. <https://inkscape.org>

---

6. <http://www.mathstat.dal.ca/~selinger/>  
7. <http://potrace.sourceforge.net/potrace.pdf>

### 3.2. Croix

La croix ci-dessous est vectorisée. On commence à être loin de l'original, mais l'effet de style est très intéressant. Dans ce cas-là, les raisons de forts changements depuis l'original sont essentiellement un graphisme original complexe et un faible contraste avec le fond (le papier du support ayant subi des dégâts) :



2- On extrait le dessin ciblé dans un format JPEG et on l'importe dans **Inkscape** (dans le cadre de cet article c'est la Version 0.91 qui est utilisée) en conservant la résolution (Image DPI) du fichier et en mode "Embed" et un import de type "*smooth rendering*".

3- On utilise l'outil *Potrace* intégré dans **Inkscape** via les menus *Path->Trace Bitmap*

4- On utilise Potrace en mode *Single Scan* et un *Seuil de luminosité* « **Brightness Cutoff** » avec threshold par défaut à **0.45**. On sélectionne l'image et valide (OK) et, important, on double-clique au centre de l'image afin de faire ressortir les paths/vecteurs individuels qu'on va sélectionner et extraire :



5- On sélectionne finalement les vecteurs créés sans le fond et les collent dans un nouveau fichier,

## 4. Description des étapes avec Inkscape

1- On part de l'original<sup>8</sup> (cas de l'oiseau stylisé) qui est issu d'un manuscrit Égyptien du 14<sup>e</sup> siècle [2] relatif à une nouvelle église copte<sup>9</sup> :

8. [http://vmr.bham.ac.uk/Collections/Mingana/Christian\\_Arabic\\_61\\_Coptic\\_2/Page\\_9/full/](http://vmr.bham.ac.uk/Collections/Mingana/Christian_Arabic_61_Coptic_2/Page_9/full/)

9. page scannée numéro 9, en marge, en haut à gauche [http://vmr.bham.ac.uk/Collections/Mingana/Christian\\_Arabic\\_61\\_Coptic\\_2/Page\\_4/viewer/](http://vmr.bham.ac.uk/Collections/Mingana/Christian_Arabic_61_Coptic_2/Page_4/viewer/)

on se débarrasse ainsi de la couleur de fond.

6- On obtient alors une image vectorielle dont il sera possible ensuite de faire quelques ajustements (typiquement suppression de taches et recolorisations de certaines parties éventuellement) :



## 5. Potrace

### 5.1. Exemple de traitement le plus simple en ligne de commande

En ligne de commande, à la différence de l'interface built-in d'Inkscape, on veillera à utiliser un fichier d'entrée au format **BMP**.

```
potrace-1.13.win64>
potrace.exe Coptic_2_oiseau2.bmp
-o Coptic_2_oiseau2.eps --eps
```

Un des inconvénients immédiats est que les anomalies diverses sont résiduelles. Le post-traitement manuel reste une nécessité.



FIGURE (2): Oiseau vectorisé sans post-traitement ni colorisation

Dans le cas de la génération en ligne de commande avec Potrace le format d'entête est :

```
!PS-Adobe-3.0 EPSF-3.0
%%Creator: potrace 1.13, written by \
Peter Selinger 2001-2015
...
```

A titre de comparaison, on peut vérifier que pour Inkscape, en fait, Potrace est lui-même utilisé avec l'outil **Cairo** :

```
!PS-Adobe-3.0 EPSF-3.0
%%Creator: cairo 1.14.1 \
(http://cairographics.org)
...
```

Il faut utiliser l'option `-cleartext` afin d'avoir le contenu du fichier EPS non compressé.

### 5.2. Eléments du fonctionnement de Potrace

Dans Potrace, un bitmap de taille  $w \times h$  est inclus dans un système de données cartésiennes où chaque pixel prend la place d'une unité carrée. Les pixels sont positionnés de telle façon que les coins des pixels (et pas leurs centres) sont à des points avec des coordonnées entières. L'origine du système de coordonnées est au coin *bas gauche* du bitmap. Les 4 coins des bitmaps ont pour coordonnées  $(0, 0)$ ,  $(0, h)$ ,  $(w, h)$ , et  $(w, 0)$ .

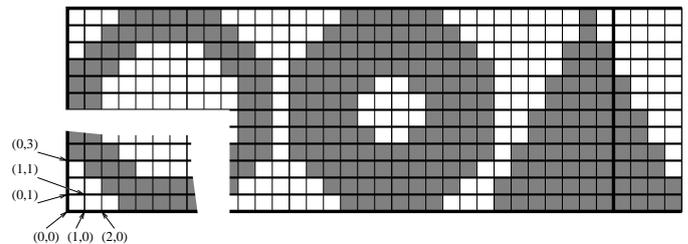


FIGURE (3): Représentation du système cartésien pour Potrace. Source : Potrace Library API 1.13

Potrace est disponible sous forme de bibliothèque externe (C) où l'entier non signé `potrace_word` comporte  $N$  bits.

Un bitmap de dimensions  $w \times h$  est divisé, du bas vers le haut, en  $h$  "scanlines" horizontales. Chaque "scanline" est divisée, de gauche à droite, en blocs de  $N$  pixels. Chacun de ces blocs de  $N$  pixels est stocké comme un simple `potrace_word`, avec les pixels les plus à gauche du bloc correspondant au bit le plus significatif du mot, et les pixels

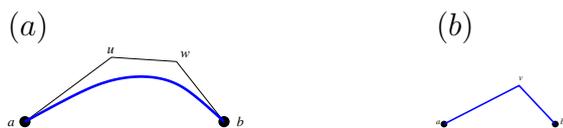


FIGURE (4): (a) Un segment de courbe de Bézier. (b) Un segment de coin

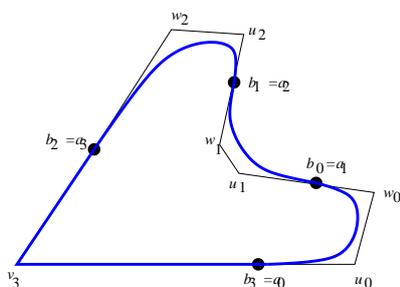


FIGURE (5): Une courbe fermée composée de 4 segments

les plus à droite correspondant aux bits les moins significatifs du mot. Les pixels “on” (ou “noir” ou “foreground”) sont représentés par la valeur 1. Les pixels “off” (ou “blanc” ou “background”) sont représentés par la valeur 0.

Au final, les pixels de coordonnées  $[i, j]$  sont accédés avec la formule suivante (C) :

```
pixel(i,j) = ((map + j*dy)[i/N] &
(1 << (N-1-i%N)) ? 1 : 0.
```

## 6. Courbes de Bézier et segments

Les courbes dans Potrace sont composées des deux types de segments suivants :

- **Segments de courbe de Bézier.** Une courbe de Bézier est généralement identifiée par un point de départ  $a$ , deux points de contrôle  $u$  et  $w$  et un point final  $b$ , tel que visible dans Figure 4(a).
- **Segments de coin.** Un segment de coin est identifié par deux lignes droites, une de  $a$  à  $v$ , et une de  $v$  à  $b$ , tel que visible dans la Figure 4(b).

Une courbe dans Potrace est une séquence de segments, de telle manière que le point final de chaque segment coïncide avec le point de départ du segment suivant. Toutes les courbes dans Potrace sont fermées et par conséquent le point final

d’un segment coïncide avec le point final du premier. La Figure 5 montre un exemple de courbe consistant en 4 segments : 3 courbes de Bézier et un segment de coin (ici les débuts et fins de segments sont symbolisés avec des “•” )

## 7. Autre exemple

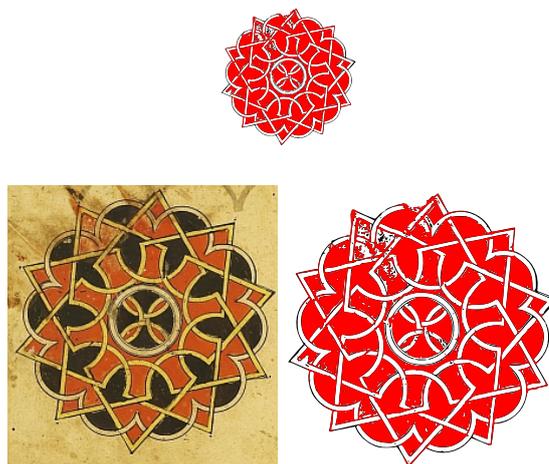


FIGURE (6): Vectorisation stylisée. Source page 204-page gauche de [2], la figure sur fond blanc est le résultat avec un remplissage intentionnel complet en rouge

## Références

- [1] P. Selinger., **Potrace : Transforming bitmaps into vector graphics.**  
URL <http://potrace.sourceforge.net>
- [2] **Christian Arabic 61 Coptic 2**, Vol. Collections : Mingana, ITSEE - JISC - University of Birmingham - United Kingdom, 14th century.  
URL [http://vmr.bham.ac.uk/Collections/Mingana/Christian\\_Arabic\\_61\\_Coptic\\_2/](http://vmr.bham.ac.uk/Collections/Mingana/Christian_Arabic_61_Coptic_2/)