

```
/*
 * Line Detection via Radon's Transform
 * 24 october 2018
 * Eliot Ragueneau
 * */

'use strict';

function pre_process(imp) { // Formate l'image 8-bit

    // Enlève le fond plus sombre que les objets d'intérêts
    IJ.run(imp, "Subtract Background...", "rolling=10");

    // Réduits le bruit (Pas très nécessaire)
    IJ.run(imp, "Median...", "radius=2");

    // Segmente automatiquement l'image pour extraire les cures dents
    uniquement
    IJ.setAutoThreshold(imp, "Default dark");
    Prefs.blackBackground = false;
    IJ.run(imp, "Convert to Mask", "");

    // Inverse les couleurs de sorte que le fond soit sombre et les cure
    dents blancs
    IJ.run(imp, "Invert LUT", "");

}

function radon_transform(imp) {
    // Récupère la plus grande dimension de l'image
    let w = imp.getWidth();
    let h = imp.getHeight();
    if (w < h) {
        w = h // w est donc la plus grande des dimensions
    }

    // Change le canvas en carré de côté w pour que l'image puisse tourner
    dans le canvas sans dépasser de celui-ci
    IJ.run(imp, "Canvas Size...", `width=${w} height=${w}
position=Center`);

    // Copie l'image actuelle
    IJ.run(imp, "Select All", "");
    IJ.run(imp, "Copy", "");

    // Rajoute une nouvelle slice (transforme donc l'image en stack)
    IJ.run(imp, "Add Slice", "");

    // Colle dans cette nouvelle slice l'image préalablement copiée
    IJ.run(imp, "Paste", "");
}
```

```

// Modifie le stack de sorte à avoir 360 copies de l'image
IJ.run(imp, "Scale...", `x=1.0 y=1.0 z=1.0 width=${w} height=${w}
depth=360 interpolation=None average process create`);

// Sélectionne le nouveau stack créé
let stack = IJ.getImage();

// ##### Rotation #####

for (let i = 0; i < 360; i++) {
    stack.setSlice(i + 1);
// Pour chaque slice du stack
    let deg = i / 2;
// Calcul le degré de rotation à appliquer
    IJ.run(stack, "Rotate... ", `angle=${deg} grid=0 interpolation=None
slice`); // Puis effectue la dite rotation
}

// ##### Calcul de la transformée de Radon #####

/*
 * Change le stack de sorte que la hauteur du nouveaux stack représente
les degrés de rotation (==> len(height) = 360)
 * et que chaque slice représente une ligne de pixel d'image
 * Concrètement :
 * Soit S0 = (x,y,z) un stack d'images de largeur x, de hauteur y, et
possédant un nombre z de slice:
 * Alors la résultante de l'opération suivante = S1 = (x,z,y)
 */
IJ.run(stack, "Reslice [/]...", "output=1.000 start=Top avoid");

// Sélectionne le résultat de la manipulation précédente
let top_view = IJ.getImage();

/*
 * Fait la somme de tous les stacks de sorte à ce que chaque pixel du
résultat corresponde à P(x, degré) = somme des valeurs sur y
 *
 * En effet, ainsi, la somme des valeurs de y sera maximale quand
l'angle seras tel que tous les pixels de la ligne seront alignées à la
verticale
 *     ligne verticale ==> pixels de la ligne tous à la même colonne x
==> somme(colonne x) = max
 * Par conséquent, il suffit de trouver les maxims de ces sommes pour
déterminer le nombre de lignes présentes dans l'image
 */
top_view = ZProjector.run(top_view, "sum");
top_view.show();

// Trouve les maxims de la transformée
let sum = IJ.getImage();
IJ.run(sum, "8-bit", "");
IJ.run(sum, "Enhance Contrast...", "saturated=0.3");
IJ.run(imp, "Find Maxima...", "noise=40 output=Count");

```

```
}

function post_process(imp) {

}

// MAIN
let imp = IJ.getImage(); // imp = 8-bit image (channel green dans
l'exemple)
pre_process(imp);
radon_transform(imp);
```