MAP586 PROJECT
DEPIXELIZING PIXEL ART

Quoc NGUYEN Wenjia FANG

18 mars 2022

DEPIXELIZING PIXEL ART

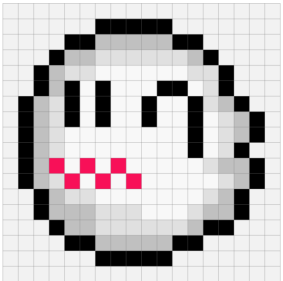


Figure: Exemple d'une image pixel art

Problématique : Les images pixel art présentent des défauts lorsqu'on les fait agrandir.

DEPIXELIZING PIXEL ART

Solution : Modifier la forme des cellules qui forment l'image pour qu'elles s'adaptent aux traits dans cette dernière.

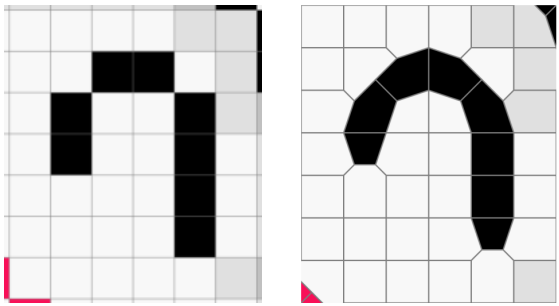


Figure: Modification de la forme de cellules à colorer

ALGORITHME

Etapes :

1. Graphe de similarité
2. Diagramme de Voronoi
3. Contour avec B-Spline

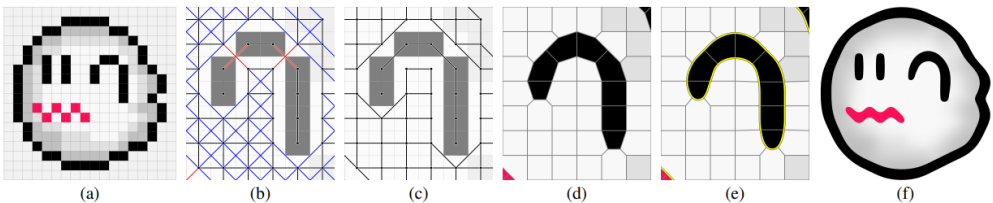
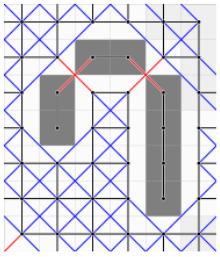
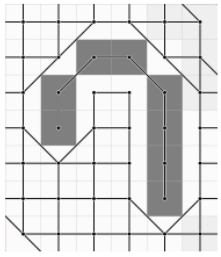


Figure: Procédé de l'algorithme

GRAPHE DE SIMILARITÉ

1. Relier les pixels de même couleur
2. Supprimer les croix
   - deux arêtes de même couleur: enlever directement les deux
   - deux arêtes de couleurs différentes: **trois heuristiques** pour contribuer un poids à chacun des deux arêtes (Curve, Sparse pixels, islands)





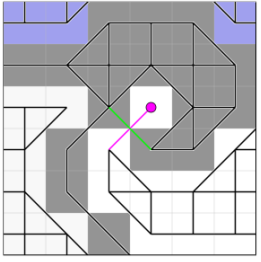
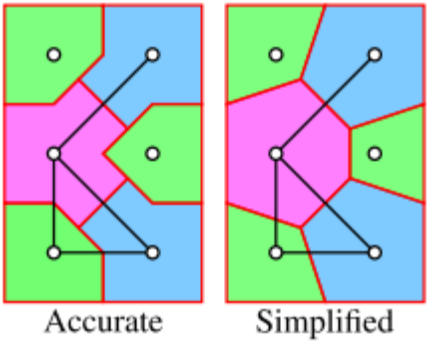
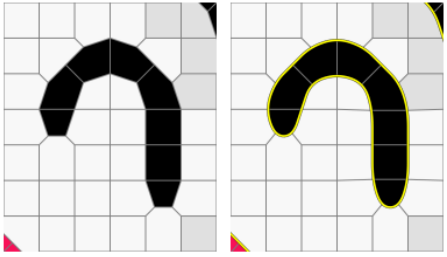


DIAGRAMME DE VORONOI



CONTOUR AVEC B-SPLINE



$$\mathbf{S}(t) = \sum_{i=0}^{m-n-1} b_{i,n}(t)\mathbf{P}_i$$

B-spline de degré 3, dont les fonctions de base sont:

$$b_{0,2}(t) = \frac{1}{2}(t^2 - 2t + 1), b_{1,2}(t) = \frac{1}{2}(-2t^2 + 2t + 1), b_{2,2}(t) = \frac{1}{2}t^2$$

STRUCTURE DE DONNÉE

Donnée à stocker et à manipuler :

- Graphe de similarité:
- Diagramme de Voronoi:
- Spline:

On utilise une classe, `Graph`, pour gérer les trois structures

CLASSE GRAPH

Un instance de classe Graph est initialisé par une image. L'information est stoqué dans le matrice `graph` dont l'élement est un `Node` qui est définié ci-dessous.

```
struct Node{

    std::vector<Direction> neighbors;
    signed int valence = 0;
    double weight = 0;
    std::map<Direction, std::pair<FPoint, FPoint>> edge_Voronoi;
};
```

Figure: Structure de Node

Le `Node` tient l'information de ses voisins ⇒ `graph` tiens l'information de la graph de simalarité.
Après avoir la graph de simularité, on en déduit le diagramme de Voronoi.

CLASSE GRAPH

Pour manipuler les Splines, on choisit les Points de controles. Ces point se trouvent aux sommets de Voronoi, un de deux extrémités d'un bord qui sépare deux régions différentes.

IMPLÉMENTATION

Figure: Caption

IMPLÉMENTATION

Graph de similarité

planarize()

Heuristic

simple_link()

Figure: Caption

IMPLÉMENTATION

Graphe de similarité	Diagramme Voronoi
planarize()	**voronoi_formation()**
Heuristic	
simple_link()	

Figure: Caption

IMPLÉMENTATION

Graphe de similarité	Diagramme Voronoi	B-Spline
planarize()	**voronoi_formation()**	**linkMainOutlines()**
Heuristic		
simple_link()		

Figure: Caption

IMPLÉMENTATION

Graphe de similarité	Diagramme Voronoi	B-Spline
planarize()	**voronoi_formation()**	**linkMainOutlines()**
Heuristic		linkMainOutlines() traverseGraph(point)
simple_link()		extractActiveGraph()

Figure: Caption

TESTS

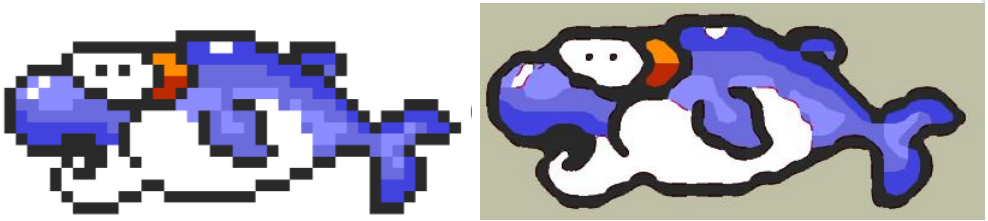


Figure: test dophine (image originale à gauche, résultat à droit)

CONCLUSION

- ça marche
- Beaucoup de point doivent être améliorés
  - règle heuristique
  - optimization
  - courbes de spline sont courts.

Merci pour votre attention !