

Generative Art via Grammatical Evolution

Présentation d'article

Hugo Mouton

Polytechnique Montréal

11 mars 2025



Table of contents

- 1 Introduction
- 2 Concepts clés
- 3 Le framework *GenerativeGl*
- 4 Résultats

- 1 Introduction
- 2 Concepts clés
- 3 Le framework *GenerativeGL*
- 4 Résultats

Contexte

- *Generative Art via Grammatical Evolution*, 2023
- Présentation de *GenerativeGl*, un framework expérimental de génération d'art
- Utilisation de l'Évolution Grammaticale (GE), technique issue des algorithmes génétiques
- Pas vraiment d'algorithme mais plutôt une combinaison de techniques artistiques et une optimisation des paramètres

1 Introduction

2 Concepts clés

- Art Génératif
- Évolution Grammaticale
- Sélection de génomes

3 Le framework *GenerativeGl*

4 Résultats

- 1 Introduction
- 2 Concepts clés
 - Art Génératif
 - Évolution Grammaticale
 - Sélection de génomes
- 3 Le framework *GenerativeGl*
- 4 Résultats

Techniques d'art génératif utilisées

Restriction à 7 techniques génératives :

- Stipple
- Cellular Automata
- Pixel Sorting
- Circle Packing
- Flow Field
- Drunkard's Walk
- Dithering

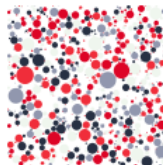
Techniques d'art génératif utilisées



(a) Stippled



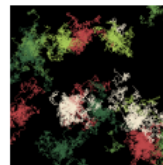
(b) Cellular Automata



(c) Circle Packing



(d) Flow Field



(e) Drunkard's Walk

Figure 1 – 5 des techniques utilisées

- 1 Introduction
- 2 Concepts clés
 - Art Génératif
 - Évolution Grammaticale
 - Sélection de génomes
- 3 Le framework *GenerativeGl*
- 4 Résultats

Qu'est ce que l'Évolution Grammaticale ?

- Issue des concepts de la programmation évolutive et des algorithmes génétiques
- Recherche d'une solution dans un espace contraint par des règles grammaticales établies
- Notion de *génom*e = mot de la grammaire choisie

Tracery

- Framework de génération de texte par application de règles grammaticales
- Symboles : les différentes techniques et leurs paramètres
- Mots/Génomes : suites de techniques ainsi que leurs paramètres
- Nombre limite de transformations dans le cas où il y a peu de symboles terminaux

```
rules = {  
    'ordered_pattern': ['#techniques#'],  
    'techniques': ['#technique#',  
                  '#techniques#,#technique#'],  
    'technique': ['stippledBG',  
                 'flowField'],  
    ...  
}
```

Figure 2 – Représentation Python de la structure de *Tracery*

Mutations de génomes

Dans *GenerativeGl*, 2 types de mutations sont autorisées :

- ❶ Sélection d'une technique à un indice aléatoire du génome et remplacement par une autre technique (ou une suite récursive de techniques)
`flowField('edgy', 0.1) → dither('halftone'),flowField('edgy', 0.1)`
- ❷ Modification des paramètres associés à la technique
`flowField('edgy', 0.1) → flowField('edgy', 0.025)`

Croisement de génomes

Une autre façon de créer de nouveaux individus est le croisement de génomes.
On choisit un symbole aléatoire dans chaque génome et on échange les 2 symboles.

Génome 1 : `dither('halftone'),flowField('edgy', 0.1)`

Génome 2 : `flowField('edgy', 0.025),flowField('edgy', 0.025)`

On échange les derniers symboles de chaque génome :

Nouveau génome 1 : `dither('halftone'),flowField('edgy', 0.025)`

Nouveau génome 2 : `flowField('edgy', 0.025),flowField('edgy', 0.1)`

- 1 Introduction
- 2 Concepts clés
 - Art Génératif
 - Évolution Grammaticale
 - Sélection de génomes
- 3 Le framework *GenerativeGl*
- 4 Résultats

Lexicase Selection

- Algorithme de sélection de parents dans les algorithmes génétiques
- Choix des parents à une certaine génération pour créer les individus enfants de la prochaine génération
- Algorithme de recherche de solutions dans un espace en utilisant un ensemble de fonctions objectif
- Domaines : programmation génétique, robotiques, géosciences, etc...

Phase de sélection

Déroulement d'une phase de sélection :

- ① On récupère un échantillon de la population
- ② On mélange les fonctions objectifs et on évalue les individus avec la première
- ③ Si un individu est meilleur que tous les autres, on le conserve, sinon, on évalue les individus égaux avec la seconde fonction objectif
- ④ On répète jusqu'à n'avoir qu'un individu ou s'il n'y a plus de fonctions objectif
- ⑤ Si il reste des individus égaux selon toutes les fonctions objectifs, un individu est choisi aléatoirement

- 1 Introduction
- 2 Concepts clés
- 3 Le framework *GenerativeGL*
- 4 Résultats

Le framework *GenerativeGl*

- Framework de création d'art génératif par évolution grammaticale
- Génération d'images plus satisfaisantes
- Contrôle de la génération en fonction des préférences de l'artiste

Architecture de *GenerativeGI*

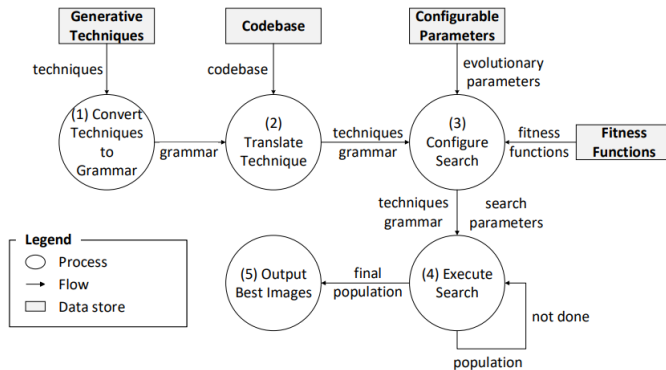


Figure 3 – Diagramme du fonctionnement de *GenerativeGI*

1 Introduction

2 Concepts clés

3 Le framework *GenerativeGl*

4 Résultats

Merci

Merci de votre attention !