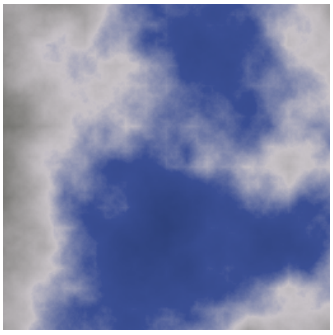
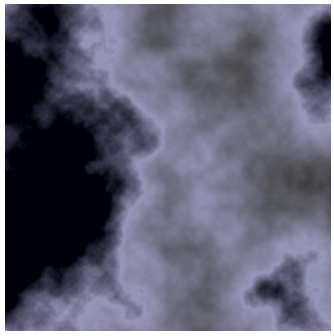
配列言語によるプラズマフラクタル生成アルゴリズムの実装と検討

中山敏宏 (工学部・アドバンス創造工学研修 2年) Oleg Kiselyov (情報科学研究科) 東北大学

導入



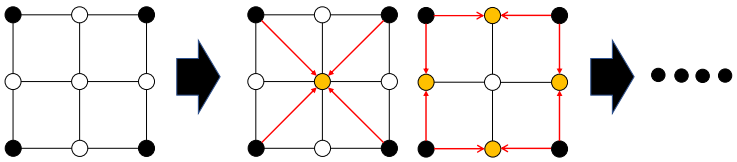
目的：様々なプラズマフラクタル生成
アルゴリズムの実装
方法：配列言語によってアルゴリズムを実装
意義：これらのアルゴリズムの本質が再帰的な
拡大であるという発見



プラズマフラクタル生成アルゴリズムと実装

・**例： bilinear interpolation**

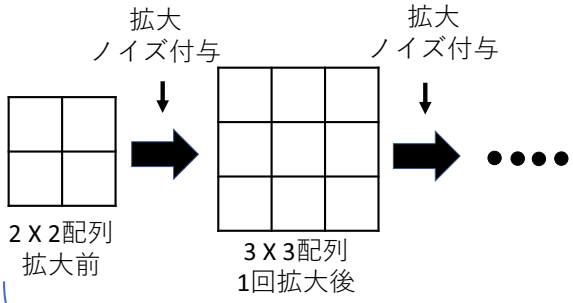
黄色の要素 = 平均値 + 乱数



※格子点は二次元配列を表現しており, 上図は3 X 3の配列の例

実装のアイデア

・**二次元配列のノイズを交えた拡大と捉えて実装**



配列言語によるプラズマフラクタル

・プラズマフラクタル生成アルゴリズム

ntimes n (**expander** scaler noisef)

```
type ('d, 'a) arr = Arr of 'd * ('d → 'a)

let expander : ((d2,int) arr → (d2,int) arr)      (*拡大を行う関数*)
                → float                            (*付与するノイズ*)
                → (d2,int) arr → (d2,int) arr =    (*拡大する配列*)

fun scaler noisef m →
let m2 = map (fimul noisef) m ▶ scaler in
let noise = Arr (rho m2, fun (i,j) →
                    if even i && even j then 0 else rand ())
in zip_with (+) m2 noise ▶ materialize2 0
```

Convolutionによる拡大

・mvert, mhorz, mdaigの3つのkernelによるconvolution

<bilinear interpolation>
・kernel

mvert(x2)

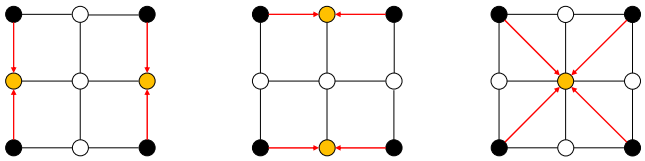
1
1

mhorz(x2)

1	1

mdiag(x4)

1	1
1	1



・拡張として他のアルゴリズムも同様に実装

<diamond-square algorithm>
・kernel

mvert(x8)

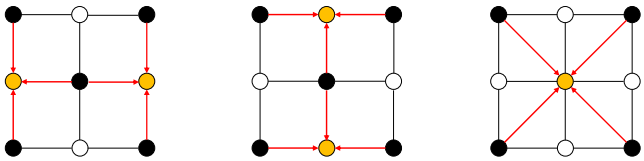
1	2	1
1	2	1

mhorz(x8)

1	1
2	2
1	1

mdiag(x4)

1	1
1	1



<bicubic interpolation>
・kernel =

-1	9	9	-1

X 1 / 16 で同様に実装

デモ

・配列を再帰的に拡張する回数
・付与するノイズの値
・拡大に使用するアルゴリズム

上記の要素を変化させながらいくつか画像を生成