

テクスチャ合成

ZHOU MENGTAO

G0321079

手法の説明

ピクセルベース全探索アルゴリズムのプログラムを基づいて多重解像度手法を実装しました。

Li-Yi Wei、Marc Levoy の論文「Fast Texture Synthesis using Tree-structured Vector Quantization」を参考にしてプログラムを書きました。

プログラムのアルゴリズムを簡単に説明します。以下の図のように書いています。前一段階の SSD を求める時、正方形の近隣画素群を使います。

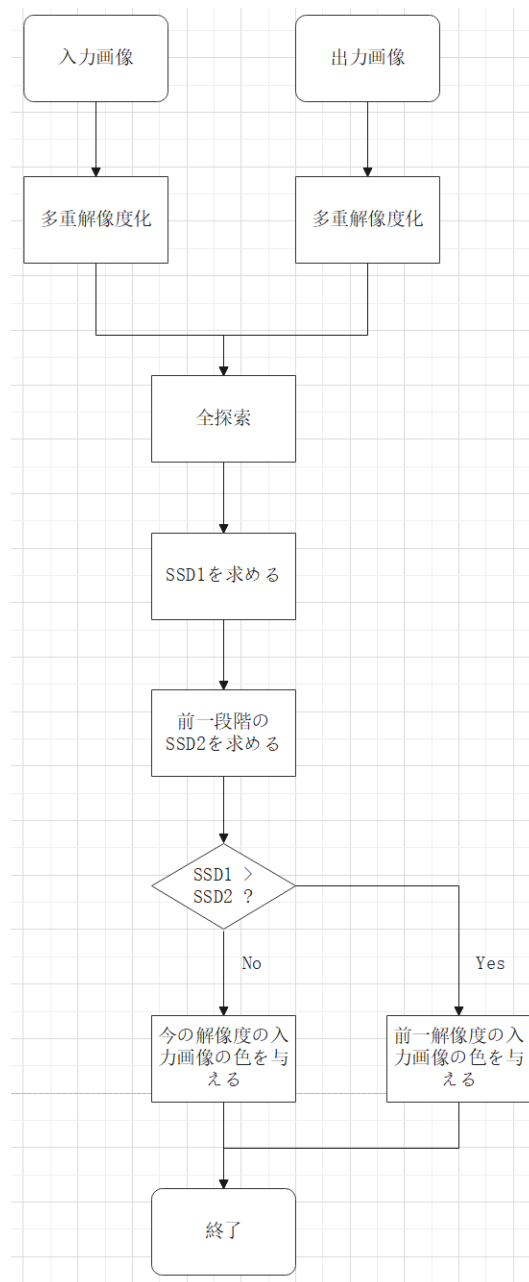


図 1 プログラムのアルゴリズム

多重解像度化のアルゴリズムは以下となります。

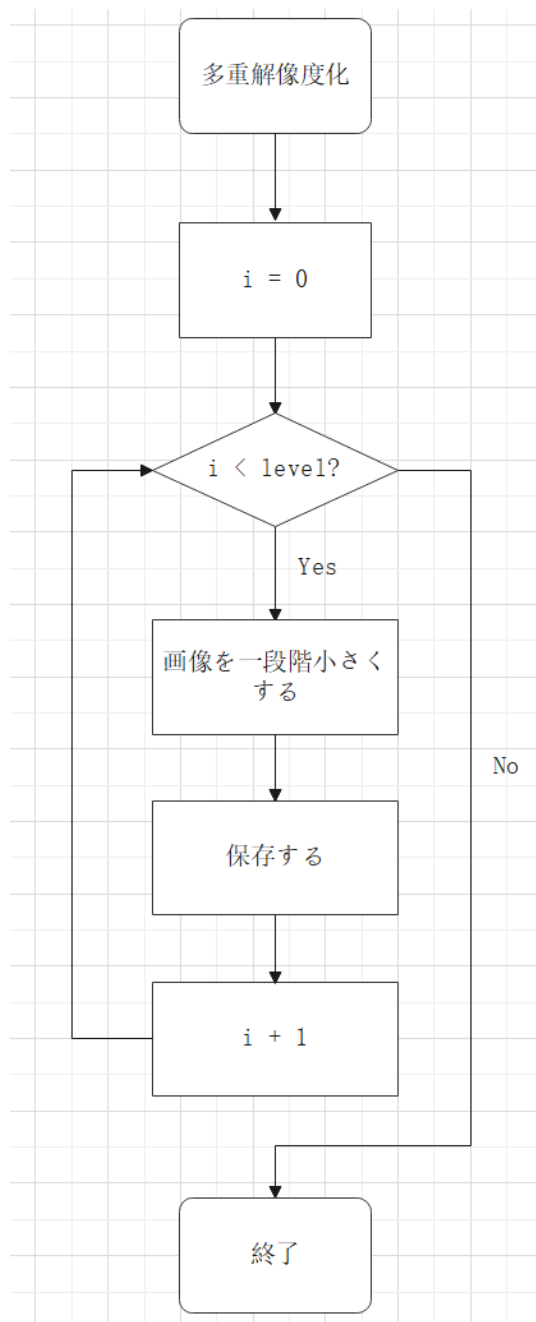


図2 多重解像度化のアルゴリズム

また、多重解像度画像を保存するために、vector データ構造を使います。

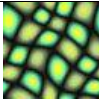
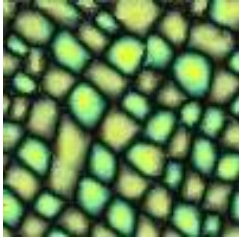
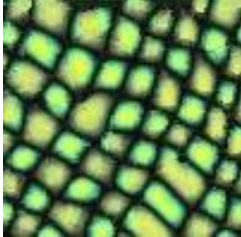
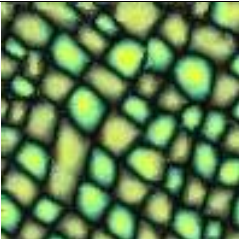
実験結果

三つのテクスチャを用いて実験を行いました。





入力画像の解像度は全て 64*64、出力画像の解像度は全て 100*100 です。

nbr は近隣画素群のサイズ、level は多重解像度の回数です。


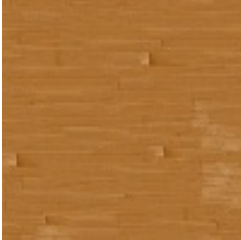

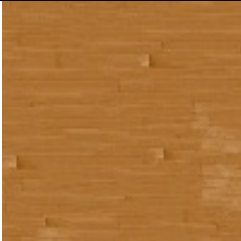
1、 サンプルテクスチャ

 入力画像	nbr = 3	nbr = 5
level = 3	 5 分 42 秒	 13 分 14 秒
level = 4	 6 分 19 秒	

2、 壁のテクスチャ

 入力画像	nbr = 3	nbr = 5
level = 3	 5 分 41 秒	 13 分 33 秒
level = 4	 5 分 47 秒	

3、床のテクスチャ

 入力画像	nbr = 3	nbr = 5
level = 3	 6 分 14 秒	 12 分 35 秒
level = 4	 5 分 39 秒	

考察

実験結果によると、テクスチャ合成はできました。近隣画素群のサイズを大きくすれば、合成結果の精度も高くなりましたが、計算時間が倍増しました。多重解像度の回数を増えても、計算時間は少しだけを増えましたが、合成結果の精度はあまり高くなりませんでした。その原因を探すのも今後の課題になります。今考えられる原因はアルゴリズムの問題かもしれません。

実験中に一番計算時間がかかったのは高い解像度のループでした。総時間は5、6分ぐらいでしたが、この中に小さい解像度の時間は数秒から数十秒で終わり、残る時間は全部最後の一つのループにかかりました。

開発環境

プログラミング言語：C++

パソコンのスペック：WINDOWS10, i7-9700K, RTX-2600s, 16GB。

実行環境：Visual Studio 2019 + OpenCV2.4.9