Trier University
of Applied Sciences

H O C H
S C H U L E
T R I E R

Texture Synthesis mit Image-Quilting

Modul: Tools- und Pluginprogrammierung

Dozent: Prof. Dr. Christof Rezk-Salama, Dipl.-Inf.

Vor- und Nachname: Imad-Eddine Abdessami

Abgabedatum: 08.01.2024

Inhaltsverzeichnis

1. Abstract

Ziel der Aufgabe ist es, große, visuell kohärente Texturen zu erzeugen, indem kleinere Texturfelder, die aus einem Eingabebild abgetastet wurden, nahtlos zusammengefügt werden, was wir Image-Quilting nennen.

2. Hintergrund

Die Motivation für dieses Projekt stammt aus dem Gebrauch, visuell überzeugende Texturen für verschiedene Anwendungen wie Computergrafik, Spieleentwicklung und digitale Kunst zu erstellen. Image Quilting bietet eine einzigartige Lösung für die Herausforderung, großflächige Texturen zu erzeugen.

Die allgemeinen Schritte des Image Quilting-Algorithmus sind:

Schritt 1 : Sampling (zufällige Platzierung)

- Der Algorithmus legt die Größe des Quadrats fest, nimmt zufällige Patches aus der Sample Image und platziert sie zufällig nebeneinander, um eine Output-Textur zu bilden.

Schritt 2 : Overlapping (Überlappung)

- An diesem Punkt definiert der Algorithmus die Größe der Überlappungsregion und die Fehlertoleranz, er sucht nach Patches in Sample Image, deren Überlappungsregion eine Summe der quadratischen Differenzen unter der Fehlertoleranz aufweist, und wählt dann zufällig einen dieser Patches aus, der der Output-Textur hinzugefügt wird.

Schritt 3 : Minimum Error Cut

- Der letzte Schritt besteht darin, die überlappenden Regionen entlang des Pfades des minimalen Fehlers auszuschneiden. Der Algorithmus findet Patches mit ähnlichen Überlappungsregionen wie in Schritt 2 beschrieben und berechnet dann das Kostenbild der überlappenden Region.

3. Programmierkonzept

Das Programm wurde mit Python implementiert und besteht aus 4 Module, ein “Textures” Verzeichnis und “Result” Verzeichnis :

1. main.py
2. texturing.py
3. synthesis_error.py
4. dpMinCut.py

2.1. main.py

Ich habe 2 main Module implementiert, das erste erlaubt es, ein beliebiges Bild von Ihrem Computer zu texturieren, und das zweite erstellt Texturen aus den gegebenen Samples im "src/Textures" Ordner, aber beide tun das Gleiche, da es als Hauptprogramm dient, um eine Textur unter Verwendung des Image Quilting Algorithmus zu erstellen. Es definiert Konstanten wie Blockgröße, Überlappungsgröße, Anzahl der Blöcke und Toleranzfaktor. Das Bild wird in das Double-Format konvertiert, und es wird eine Instanz der Klasse textureMain mit den angegebenen Parametern erstellt. Die Ausgabemaske wird generiert, und der Prozess der Texturerstellung wird mit der Methode createTexture eingeleitet. Die resultierende Textur wird

angezeigt, und eine Bilddatei wird in "src/Result" gespeichert. Der Benutzer kann das Eingabebild anpassen und Parameter einstellen, um mit verschiedenen Texturen zu experimentieren. Außerdem kann das Skript so modifiziert werden, dass der Benutzer das Eingabebild interaktiv in einem file Dialog auswählen kann.

2.2. texturing.py

texturing.py definiert eine Klasse, die einen Bild-Quilting-Algorithmus für die Textursynthese implementiert. Der Algorithmus erzeugt große Texturen durch das Zusammenfügen kleinerer Blöcke, die aus einem Eingabebild abgetastet werden, wie in "Hintergrund" erläutert. Zu den Methoden der Klasse gehören "generateOutputMask" zur Initialisierung der Ausgabebildmaske und "overlaps" zur Berechnung der Überlappungsbereiche zwischen Blöcken. Die Hauptmethode "createTexture" iteriert durch die Blöcke und synthetisiert die Textur auf der Grundlage der angegebenen Überlappungsbereiche und unter Verwendung einer Synthesefehlerfunktion. Sie verwendet dynamische Programmierung und minCut-Techniken aus dpMinCut.py zur Verfeinerung der Grenzen zwischen überlappenden Blöcken. Die sich daraus ergebende Textur wird schrittweise geformt, um visuelle Kohärenz zu gewährleisten.

2.3. synthesis_error.py

synthesis_error.py berechnet den Synthesefehler zwischen einem Ausgabebild-Slice und einem Eingabebild. Sie initialisiert eine Fehlermatrix mit Nullen und iteriert durch die Pixel im Eingabebild, wobei sie die entsprechenden Blöcke auf der Grundlage der angegebenen Überlappungsposition (links, oben oder Ecke) vergleicht. Für jedes Pixel wird der Eingabeblock extrahiert, der quadratische Fehler zwischen dem Ausgabe-Slice und dem Eingabeblock berechnet und die Fehlermatrix

aktualisiert. Die sich daraus ergebende Matrix stellt die Fehlerverteilung bei der Synthese dar und hilft dem Image Quilting Algorithmus bei der Auswahl optimaler Patches für die nahtlose Textursynthese.

2.4. dpMinCut.py

dpMinCut.py implementiert einen auf dynamischer Programmierung basierenden Min-Cut-Algorithmus, der die Grenzen zwischen sich überlappenden Bildblöcken verfeinert. Er berechnet die quadrierten Differenzen in den Farbkanälen, konstruiert eine Energiematrix und findet iterativ den optimalen Schnittpfad auf der Grundlage der kumulativen Energie. Die resultierende Schnittmatrix stellt die verfeinerte Grenze zwischen den Eingabe- und Ausgabeblöcken dar und verbessert die Qualität der Bildsteppung durch Minimierung visueller Artefakte.

4. Ergebnisse

Ergebnisse von „2001 Paper Image Quilting for Texture Synthesis and Transfer by Efros and Freeman“

Original Sample:

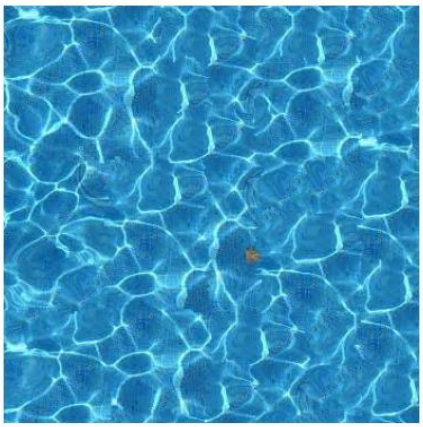
Random Texturing:

Overlap:

Cut Sample:

Meine Ergebnisse :

Textur 1:



Textur 2 :

Textur 3 :

Literaturverzeichnis

2001 Paper Image Quilting for Texture Synthesis and Transfer by Efros and Freeman