

1 Einleitung

Texturen sind ein fundamentales Mittel zur Erstellung von Modellen für die Darstellung in Bildern, Videos oder Spielen. Sie werden benötigt, um die Oberflächendetails eines Modells darzustellen ohne diese explizit über Geometrie oder Materialeigenschaften modellieren zu müssen [WLKT09]. Damit wird der Detailgrad eines Modells erhöht, ohne dabei den Detailgrad der zu Grunde liegenden Geometrie zu erhöhen. Das führt unmittelbar zu einer schnelleren Berechnung des darzustellenden Bildes und einer Einsparung von Speicherplatz.

Texturen zeigen die visuelle Oberfläche eines Modells mit sich wiederholenden Mustern [WLKT09]. Der Grad der Zufälligkeit dieser Wiederholung kann dabei üblicherweise für unterschiedliche Texturen variieren. Naturgegebene Texturen besitzen in der Regel auf Grund der Unvollkommenheit der Natur eine komplett zufällige Wiederholung von sich ähnlichen Mustern, wo hingegen Menschengeschaffenes oft auch eine deterministische Komponente besitzt. Ein Fliesenmuster besitzt zum Beispiel eine zufällige Wiederholung in den Mikrodetails seiner Fliesen, die Platzierung der Fliesen bzw. deren anliegenden Kanten unterliegen dagegen aber einem deterministischem Muster [WLKT09].

Die Erstellung von Texturen gilt allgemein anerkannt als sehr aufwändig [WLKT09]. Das zeichnet sich insbesondere dadurch wieder, dass es einen speziellen Berufszweig in der Film- und Spielebranche namens *Texture Artist* gibt, der sich ausschließlich mit der Aufgabe befasst, qualitativ hochwertige Texturen zu kreieren. Diese werden meistens über Handzeichnungen oder über handbearbeitete Fotografien erstellt. Handzeichnungen können zwar ästhetisch ansprechend sein, aber sie garantieren keinerlei Fotorealismus. Fotografien hingegen werden in einem beliebigem Grafikprogramm solange bearbeitet, bis sie den Anforderungen der Textur gerecht werden. Das führt in der Regel dazu, dass die Erstellung einer Textur sehr zeitaufwändig ist und sich dennoch auffällige Wiederholungen oder Unsauberkeiten in die Textur einschmuggeln können. Ferner ist nicht jeder Mensch ein Künstler und es ist äußerst schwierig wenn nicht gar unmöglich ohne Vorkenntnisse oder künstlerische Fähigkeiten qualitativ hochwertige Texturen zu erzeugen [WLKT09]. Das Verfahren der Textursynthese widmet sich diesem Problem.

der Satz ist
scheiße

andere einle
tungen auß
stateofthea
hinzufügen

2 Textursynthese

Die *Textursynthese* ist ein alternativer Ansatz für die Erstellung von Texturen, der es erlaubt auch ohne spezielle Vorkenntnisse qualitativ hochwertige Ergebnisse zu erzielen. Der Benutzer muss lediglich ein Beispielemuster und eventuell benötigte Konfigurationsparameter an die Textursynthese übergeben, die aus diesen Daten eine Textur synthetisiert [WLKT09]. Die resultierende Textur zeichnet sich dadurch aus, dass sie eine beliebige Größe annehmen kann, aber weiterhin sichtbare Ähnlichkeiten zu dem Beispielemuster aufweist. Die Textursynthese kümmert sich dabei automatisiert und im Idealfall ohne aufwändige Benutzereingaben um die Schwierigkeiten bei der Erstellung von Texturen. Sie berücksichtigt die visuellen Charakteristiken des Beispielemusters und vermeidet dabei zeitgleich auffällige Wiederholungen oder Unnatürlichkeiten in der synthetisierten Textur.

2.1 „Markov Random Fields“-Eigenschaft

Die „Markov Random Field“-Eigenschaft ist eine der populärsten Eigenschaften, die der Textursynthese zu Grunde liegt [WLKT09]. Sie motiviert die Metrik, die Verwendung findet, um die Ähnlichkeit zwischen dem Beispielemuster und der zu synthetisierenden Textur zu beschreiben [KEBK05]. Die „Markov Random Field“-Eigenschaft beschreibt dabei die Synthese als eine Aneinanderreihung von *lokalen* und stationären Prozessen [WLKT09]. Das bedeutet, dass jede Farbe eines Pixels in der Textur ausschließlich über die Pixel in seiner direkten räumlichen Nachbarschaft charakterisiert werden kann (lokal). Diese Charakterisierung ist dabei unabhängig von der Position des betrachteten Pixels (stationär) [KEBK05]. Die Intuition dieser Eigenschaft lässt sich anhand von Beispielen verdeutlichen. Einem Betrachter liegt ein Bild vor, welches er aber nur durch ein kleines bewegliches Fenster betrachten kann. Er sieht demnach nie das komplette Bild auf einmal, kann aber durch Bewegungen seines Fensters einzelne Bereiche des Bildes entdecken und erschließen. Das Bild ist dann stationär, falls das Bild unter verschiedenen Ausschnitten immer ähnlich zueinander erscheint (eine geeignete Fenstergröße vorausgesetzt). Das Bild ist lokal, falls jeder Pixel in der Mitte eines Fensters über die umliegenden Pixel bestimmt werden kann [WLKT09].

Basierend auf dieser Eigenschaft kann der Prozess der Textursynthese spezifiziert werden: Sei ein Beispielemuster gegeben. Dann lässt sich daraus eine Textur synthetisieren, sodass für jeden synthetisierten Pixel dessen räumliche Nachbarschaft zu mindestens einer Nachbarschaft im Beispielemuster ähnlich ist [WLKT09]. Die Größe

der betrachteten Nachbarschaften ist dabei in der Regel ein benutzerdefinierter Parameter. Die Nachbarschaftssuche basiert in der Regel auf der kleinsten quadrierten farblichen Abweichung

$$\min d(\mathbf{t}_i, \mathbf{x}_j) = \|\mathbf{t}_i - \mathbf{x}_j\|^2. \quad (1)$$

\mathbf{t}_i beziehungsweise \mathbf{x}_j beschreiben dabei die Farben einer Nachbarschaft der Größe $N \times N$ als Vektor um den Pixel i in der Textur T respektive um den Pixel j in dem Beispielemuster X . Für jede Nachbarschaft \mathbf{t}_i ist dann seine ähnlichste Nachbarschaft \mathbf{x}_j im Beispielemuster gefunden, wenn $d(\mathbf{t}_i, \mathbf{x}_j)$ minimal [KEBK05].

Aufgrund der Ähnlichkeiten zwischen lokalen Nachbarschaften im Beispielemuster und der Textur wird garantiert, dass die synthetisierte Textur Gemeinsamkeiten mit dem Beispielemuster aufweist [WLKT09].

2.2 Verfahren

Der Großteil an jüngst veröffentlichten Algorithmen zur Textursynthese basiert auf der „Markov Random Fields“-Eigenschaft [WLKT09]. Diese Verfahren lassen sich in der Regel einer von zwei Kategorien zuordnen: den *lokal wachsenden Verfahren* und den *global optimierenden Verfahren*. Lokal wachsende Verfahren synthetisieren die Textur nach und nach über einzelne Pixel oder Regionen [KEBK05]. Global optimierende Verfahren hingegen synthetisieren und optimieren die Textur iterativ als Ganzes auf Basis einer Zielfunktion [KNL⁺15]. Im Folgenden sollen drei Verfahren der beiden Kategorien näher betrachtet werden.

2.2.1 Pixelbasierte Textursynthese

Einer der ersten Ansätze der Textursynthese ist die *pixelbasierte Textursynthese* (vgl. [EL99]). Er fällt in die Kategorie der lokal wachsenden Verfahren. Sei ein Beispielemuster und eine Nachbarschaftsgröße gegeben, dann funktioniert die grundlegende Idee hinter diesem Algorithmus wie folgt:

Die zu synthetisierende Textur wird zuerst initialisiert, indem eine kleine beliebige Region des Beispielemusters in die Mitte der Textur kopiert wird. Der Algorithmus berechnet dann sukzessive die einzelnen Pixelfarben der Textur kreisförmig um die initiale Region von innen nach außen. Abbildung [illustriert dieses Verfahren](#). Die Farbe eines aktuell betrachteten Pixels p in der Textur wird dann über eine Nachbarschaftssuche im Beispielemuster ermittelt. Dazu wird zuerst ein Rahmen mit der Größe der Nachbarschaft um den Pixel p gelegt [\(\)](#) und alle bereits gesetzten Pixel in

abb

größe der
nachbarsch
im beispiel

diesem Rahmen gefunden. Auf Basis dieser gesetzten Pixel wird im Beispielemuster die ähnlichste Nachbarschaft auf Grundlage von (1) gefunden. Der Pixel p erhält dann die Farbe des Pixels in der Mitte der ermittelten ähnlichsten Nachbarschaft im Beispielemuster. Dieser Vorgang wird solange wiederholt, bis alle Pixel der Textur gesetzt sind.

Der Algorithmus [EL99] ist relativ einfach zu verstehen sowie zu implementieren [WLKT09]. Er ist außerdem benutzerfreundlich, da dieser nur einen Konfigurationsparameter, die Nachbarschaftsgröße, übergeben muss [EL99]. Die Wahl der Nachbarschaftsgröße ist jedoch nicht trivial. Fällt die Wahl der Nachbarschaftsgröße zu klein aus, so kann das Ergebnis zu zufällig wirken. Ist sie auf der anderen Seite zu groß, können sichtbare Wiederholungen entstehen oder es kommt zu Unnatürlichkeiten, da die Nachbarschaftssuche auf Grund ihrer Größe wohlmöglich keine geeigneten Kandidaten finden kann [WLKT09].

2.2.2 Regionsbasierte Textursynthese

Ein weiterer Vertreter der lokal wachsenden Verfahren und eine Erweiterung der pixelbasierten Textursynthese ist die *regionsbasierte Textursynthese*.

2.2.3 Texturoptimierung

Literatur

- [EL99] EFROS, Alexei A. ; LEUNG, Thomas K.: Texture Synthesis by Non-Parametric Sampling. In: *IEEE International Conference on Computer Vision*, 1999, S. 1033–1038
- [KEBK05] KWATRA, Vivek ; ESSA, Irfan ; BOBICK, Aaron ; KWATRA, Nipun: Texture Optimization for Example-based Synthesis. In: *SIGGRAPH '05*, 2005, S. 795–802
- [KNL⁺15] KASPAR, Alexandre ; NEUBERT, Boris ; LISCHINSKI, Dani ; PAULY, Mark ; KOPF, Johannes: Self Tuning Texture Optimization. In: *Computer Graphics Forum* 34 (2015), Nr. 2, S. 349–359
- [WLKT09] WEI, Li-Yi ; LEFEBVRE, Sylvain ; KWATRA, Vivek ; TURK, Greg: State of the Art in Example-based Texture Synthesis. In: *Eurographics 2009, State of the Art Report, EG-STAR*, 2009