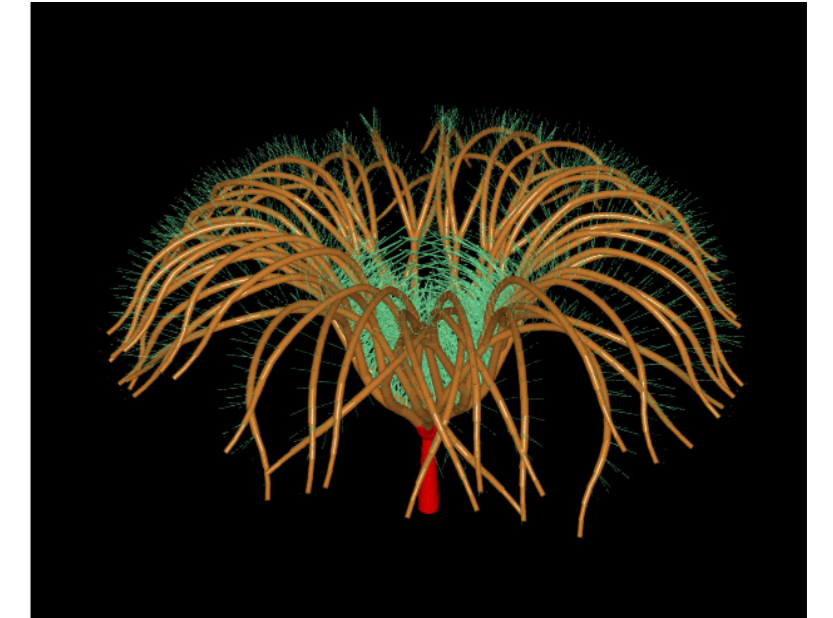


$$G = \langle \Sigma = \{a, b, c, d, S, \varepsilon\}, P = \{ S \rightarrow ab, a \rightarrow a|bc, b \rightarrow bda|d, c \rightarrow \varepsilon, d \rightarrow aba\}, \omega = S \rangle$$

Evolution einer 3D-Beschreibung aus Bildern mit Hilfe von Lindenmayer-Systemen

Diplomarbeit, vorgelegt von Jan Derer



Aufgabenstellung:

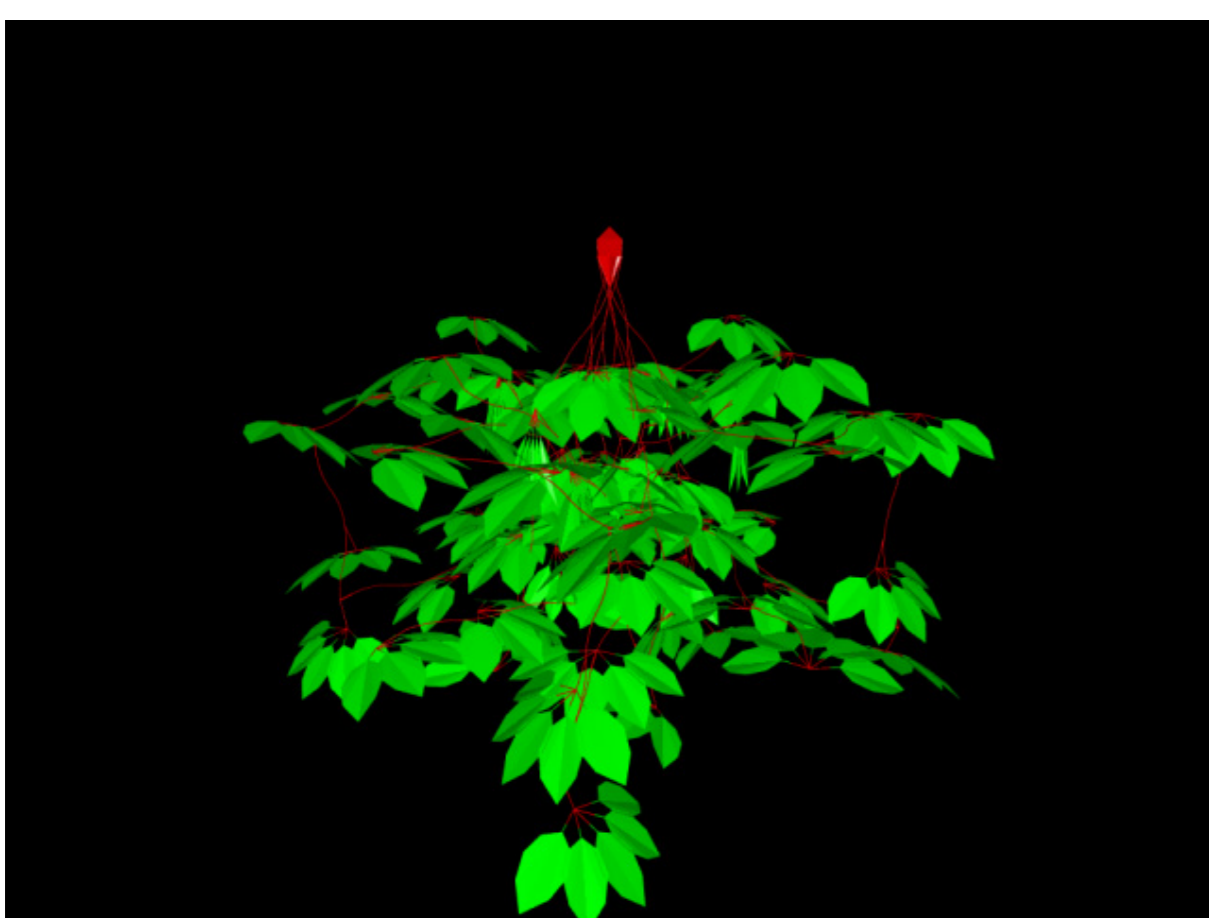
Ziel der Arbeit ist eine Untersuchung zur automatischen Erstellung von Objektbeschreibungen aus vorgegebenen Quellbildern.

Die Objektbeschreibung erfolgt in Form von L-Systemen, die passend zu Quellbildern erzeugt werden. Hierzu wird ein evolutionärer Prozess auf die L-Systeme angewendet. Fitnessfunktion sei die Übereinstimmung der Ansichten der erzeugten L-Systeme aus verschiedenen Kameraperspektiven mit den Originalbildern eines einfachen Gegenstandes.

Die Arbeit soll die theoretischen Grundlagen dieses Ansatzes darlegen (insbesondere Einordnung und Klassifizierung von L-Systemen und ihre Evolutionsmöglichkeiten), eine Systemarchitektur vorschlagen und in Teilen prototypisch implementieren. Bei der Implementation soll besonderer Wert auf die Weiternutzbarkeit der erstellten Module und die Definition externer Schnittstellen gelegt werden. Die Funktionsfähigkeit des Systems ist durch geeignete Teststellungen zu evaluieren.

Am Anfang war nicht das Wort ...

... sondern das L-System und sein Startaxiom. Die Lindenmayer-Systeme (kurz L-Systeme) sind Grammatiken und bestehen aus Produktionen, einem Alphabet sowie einem Startaxiom. Auch wenn es erscheint das Ähnlichkeiten zur Chomsky-Sprachfamilie existieren, gibt es doch wesentliche Unterschiede. Dazu zählt, dass zu jedem Ableitungsschritt alle Symbole eines Wortes abgeleitet werden und es keine Unterschiede zwischen Terminal- und Nichtterminalzeichen gibt. Um aus einem generiertem Wort eine 3D-Grafik entstehen zu lassen, muss das Wort entsprechend interpretiert werden. Diese Interpretationsvorschrift nennt sich Turtle-Interpretation. Dabei stelle man sich eine Schildkröte vor die auf bestimmte Kommandos reagiert, zum Beispiel steht das Kommando "F" für bewege die Schildkröte vorwärts und zeichne eine Linie. Aus der Abfolge der Turtle-Kommandos entsteht dabei eine 3D-Grafik.

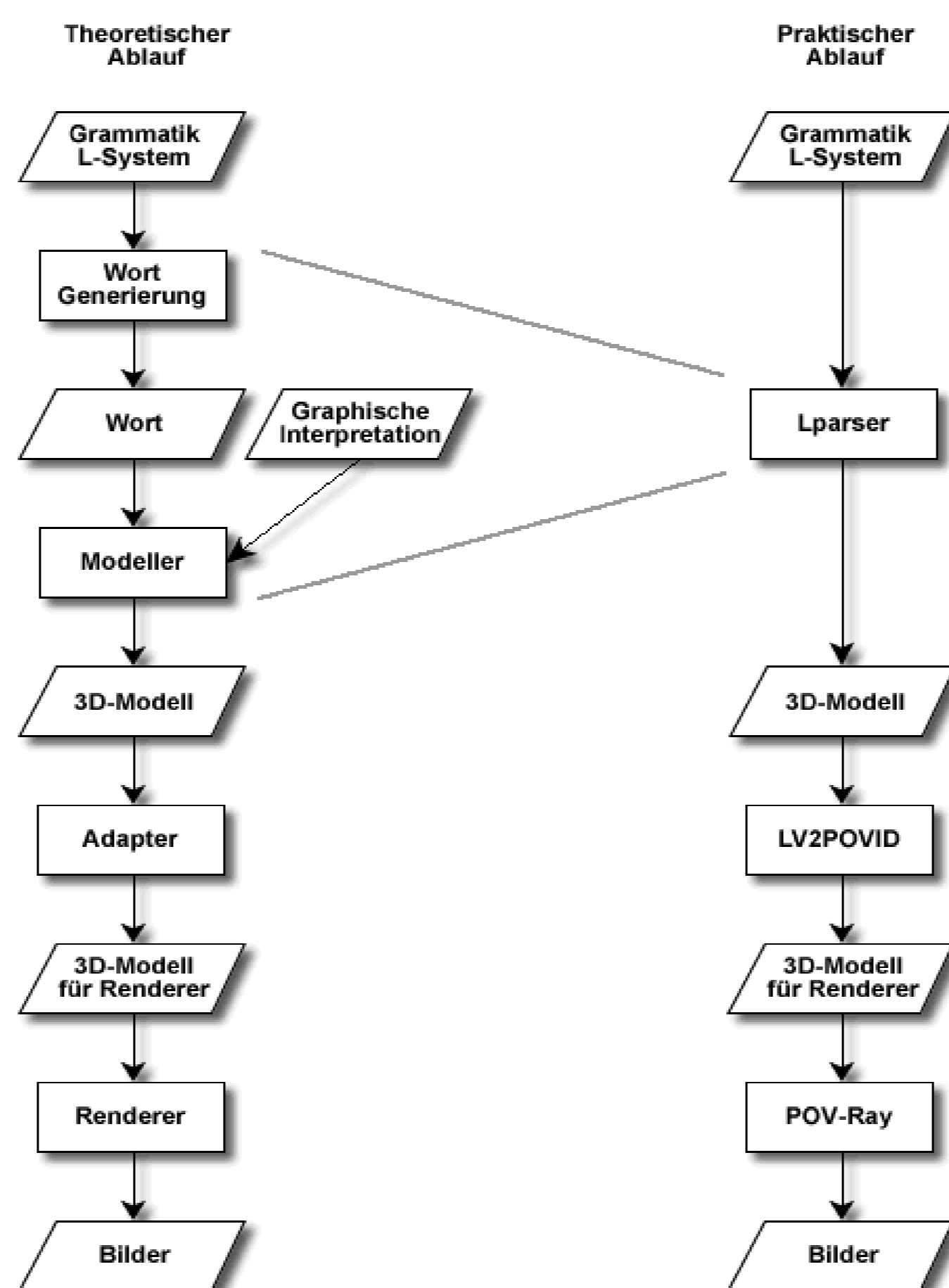


Das Bild entstand aus 14 Produktionen und 10 Ableitungsschritten. Zum Rendern der Grafik, wurde POV-Ray 3.5 für Windows verwendet.

Die Realisierung der Applikation:

Die Implementation aller Arbeitsschritte zur Erzeugung einer 3D-Grafik aus einem L-System und seinem Startaxiom, kann nicht in der Zeit einer Diplomarbeit realisiert werden. Daher wird auf existierende Programme zurückgegriffen. Es wird für die Wortgenerierung sowie für das Erzeugen der 3D-Szene das Programm Lparser verwendet. Als Ray-Tracing Programm wird POV-Ray 3.5, mit einem Hilfsprogramm namens QuietPOV, benutzt. Um die 3D-Szene, die Lparser generiert, in eine POV-Ray gerechte Beschreibung zu transformieren, wird das Programm LV2POVID verwendet.

Um dieses Programm in einer Evolutionsschleife laufen zu lassen, ist ein Programm entwickelt worden, welches die Programme bedient und stabil laufen lässt.



Es zeigt sich, dass einige verwendete Programme nicht zuverlässig arbeiten. Somit muss die entwickelte Applikation einige Korrekturen durchführen.

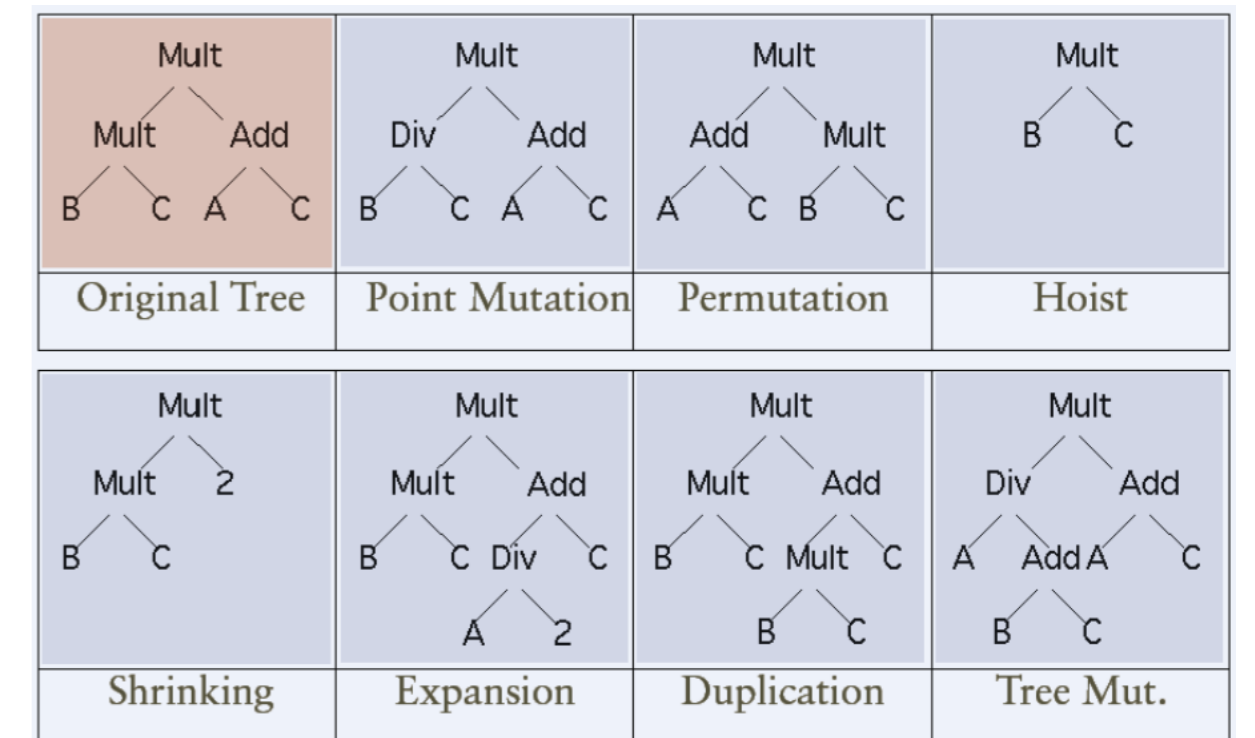
Lparser hinterlegt in der 3D-Szene keine Informationen zur Kamera- und Lichtquellenposition. Daher ist ein Algorithmus ausgearbeitet worden, der eine möglichst gute Kameraposition berechnet. Dieser Algorithmus platziert die Kamera relativ zur Ausdehnung des Objekts und die Lichtquelle relativ zur Position der Kamera. Über bestimmte Faktoren in einer Konfigurations-Datei, kann die Position der Kamera verändert werden und damit das Objekt aus allen Position betrachtet werden. Das gleiche gilt auch für die Position der Lichtquelle.

Weitere Korrekturen sind, das Prüfen des L-Systems auf Korrektheit und das richtige Positionieren der Kamera und der Lichtquelle in der POV-Ray Beschreibungsdatei.

Bei einem Performancetest zeigt sich, dass die entwickelte Anwendung nur 3% der gesamten Zeit für die Bilderzeugung einnimmt. Lparser benötigt 25% und POV-Ray über 50%.

Das Programm ist so entwickelt, dass eine Portierung auf ein anderes Betriebssystem wenig Zeit in Anspruch nehmen wird. Es wurde hauptsächlich ANSI C verwendet.

Zusätzlich zu dem kommandozeilenbasierten Programm für die Evolutionsschleife, wurde noch eine GUI entwickelt, mit der die Entwicklung von L-Systemen erleichtert wird.



Mutationsmöglichkeiten unter GP

Evolution von L-Systemen:

Neben der theoretischen Ausarbeitung zu den L-Systemen, deren verschiedene Ergänzungen, sowie deren Klassifikation und Interpretation, wird auch auf die Möglichkeiten der Evolution eingegangen. Insbesondere die Mutation in der Evolution wird vertieft erläutert.

Es zeigt sich dabei, dass L-Systeme häufig in Verbindung mit Genetischen Algorithmen verwendet werden. Dabei wird eine Produktion als Chromosom angesehen, deren einzelne Symbole mutieren können.

Seltener wird die Genetische Programmierung (kurz GP) für L-Systeme verwendet. Bei GP wird das komplette L-System als Chromosom betrachtet und in eine Baumstruktur überführt. Auf Basis dieser Baumstruktur werden die Mutationsoperationen angewandt.

Des Weiteren wird in dieser Arbeit auf die Anwendungsmöglichkeiten von L-Systemen und deren Evolution eingegangen. So kann die Bildverarbeitung herangezogen werden um Fotos von Pflanzen durch ein L-System nach zu modellieren, zu erkennen und zu klassifizieren.

Es kann soweit gegangen werden, dass dadurch Roboter Gartenarbeit verrichten, beispielsweise düngen und Unkraut entfernen.

Zusammenfassung und Ausblick:

Die Arbeit vermittelt in einer kompakten Form die theoretischen Grundlagen der L-Systeme sowie deren Klassifikation und Interpretation. Es geht weiterhin auf die Evolution von L-Systemen, insbesondere der Mutationsmöglichkeiten und auf deren Anwendung ein. Zusätzlich werden alternative Systeme vorgestellt und besprochen.

Im Rahmen der Diplomarbeit ist eine kommandozeilenbasierte Anwendung entstanden die aus LS-Dateien Bilder erzeugt und in einer Evolutionsschleife integriert werden kann. Weiterhin ist eine GUI für das Arbeiten mit L-Systemen entwickelt worden.

Da die Evolution eine zeitintensiver Prozess ist, wäre es wünschenswert das Programm Lparser performanter zu entwickeln.

Des Weiteren können die verwendeten Programme durch Eigenentwicklungen ersetzt werden um den Freiheitsgrad zu erhöhen. So hätte man die Möglichkeit nicht nur das L-System einer Evolution zu unterziehen, sondern auch beispielsweise die Interpretationsvorschrift.