

Merkmals-Visualisierung für Volumsdaten

Bakkalaureatsarbeit

Martin Haidacher
e0025416@student.tuwien.ac.at
Institut für Computergraphik und Algorithmen
TU Wien

18. Juni 2004



Bonsai-Datensatz gerendert mit den im Paper vorgestellten Methoden: (a) Depth of Field, (b) Focus+Context, (c) 3D Dithering, (d) Magic Lamp

Kurzfassung

In diesem Paper werden Methoden vorgestellt, welche das Direct Volume Rendering (DVR) unterstützen. Sie sollen vor allem dazu dienen, innere Strukturen und wichtige Details besser darzustellen. Die Methode *Depth of Field* adaptiert den Effekt der Tiefenunschärfe aus der Fotografie für die Volumsvisualisierung. Der Fokusbereich wird dabei scharf dargestellt und je nach Entfernung in Blickrichtung vom Fokuspunkt und Linsenstärke, wird der jeweilige Bereich unschärfer dargestellt. Die Methode führt dazu, dass der Benutzer schneller auf die wichtigen Teile des Datensatzes hingewiesen wird.

Die weiteren Methoden beschäftigen sich mit der Darstellung der inneren Struktur von Objekten. Die Methode des *Focus+Context* teilt dabei den Datensatz in einen Fokus und einen Kontextbereich. Objektteile, welche im Fokusbereich liegen, werden mittels DVR und den dazugehörigen Transferfunktionen gerendert. Der Rest wird mit einer nicht photorealistischen Methode gerendert, welche nur die Umrisse der Objekte darstellt. Bei *3D Dithering* wird der Inhalt von gleichmäßig verteilten Würfeln mit den vorgegebenen Transferfunktionen gerendert und vom dazwischen liegendem Rest werden nur die Konturen oder gar nichts dargestellt. Die letzte Methode, *Magic Lamp*, rendert Teile des Objektes mit DVR und den normalen Transferfunktionen nur dann, wenn deren Gradient nahezu parallel zu einer einstellbaren Richtung ist. Die restlichen Teile des Objekts werden wieder nur durch ihre Konturen dargestellt.

1 Einleitung

Diese Arbeit befasst sich mit Methoden, welche die Volumsvisualisierung effektiver machen. Unter der Volumsvisualisierung versteht man die Darstellung von Volumsdaten. Diese sind so aufgebaut, dass in einem regulären dreidimensionalen Gitter an jedem Kreuzungspunkt ein Dichtewert definiert ist (Voxelwert). Die Dichtewerte können durch Methoden wie die Computertomographie oder die Magnetresonanztomographie simuliert werden. Die Datensätze, welche in meiner Arbeit verwendet werden, stammen von der Computertomographie.

Die Grundmethode welche ich zur Visualisierung der Daten verwendet habe stammt von Marc Levoy [8]. Dabei wird für jedes Pixel ein Strahl in Blickrichtung geschossen. Wenn dieser Strahl auf den Datensatz trifft so werden an bestimmten Punkten mit gleich bleibendem Abstand (den so genannten Sample-Punkten), die zugehörigen Dichtewerte berechnet. Die berechneten Werte werden mittels Transfer-Functions in optische Eigenschaften transformiert. Die Transferfunktionen geben dabei an für welchen Dichtewert welcher Farb- und Opazitätswert verwendet werden soll. Die dadurch gewonnenen Farbwerte werden entlang eines Strahls gemäß ihrer Opazität akkumuliert. Ist die Gesamtopazität entlang eines Strahls so hoch, dass die dahinter liegenden Punkte nicht mehr nach vorne durchscheinen, dann ist der Farbwert für ein Pixel bestimmt. Durch Anwendung dieser Methode auf alle Pixel entsteht das Ergebnisbild. Diese Methode wird, mit manchen kleinen Abänderungen, sehr oft in der Volumsvisualisierung verwendet.

In den meisten Fällen sind aber nur gewisse Teile des gesamten Datensatzes von Interesse. Für die Diagnostik in der Medizin kann es notwendig sein, dass zum Beispiel ein Tumor genauer betrachtet werden muss. Um solche Teile darstellen zu können müssten sehr gute Transferfunktionen zur Verfügung sein. Wäre dies der Fall, so könnte man die unwichtigeren Teile durch niedrige Opazität ausblenden. Jedoch ist es sehr schwierig gute Transferfunktionen für solche Fälle zu finden. Es gibt Ansätze für die automatische oder halbautomatische Bestimmung von Transferfunktionen. In den Arbeiten von Taosong He, Gordon Kindlmann und James W. Durkin [3–5] sind solche Methoden beschrieben. In den meisten Fällen ist es aber trotzdem nicht möglich Transferfunktionen zu finden, welche dazu verwendet werden können spezielle Teile hervorzuheben. Dies kommt daher, dass die Dichte der gesuchten Teile oft nicht bekannt ist oder, häufiger, die zu betrachtenden Teile des Datensatzes in einem Dichtebereich liegen, in dem auch andere Teile des Datensatzes liegen. In diesem Fall ist

es nicht möglich mit geeigneten Transferfunktionen die Merkmale zu extrahieren, da es keinen Wertebereich gibt in dem nur die Teile von Interesse liegen.

Die eben genannte Problematik führt dazu, dass die wichtigen Teile eines Datensatzes durch herkömmliche Volumsvisualisierung, wie sie eben beschrieben wurde, nicht zufrieden stellend dargestellt werden können. In der Medizin kann es zum Beispiel sein, dass ein Tumor durch die Dichtewerte nicht vom umgebenden Gewebe unterschieden werden kann. Ist dies der Fall so könnte der Tumor nicht gut dargestellt werden, da er immer vom umgebenden Gewebe verdeckt würde und so nie eine freie Sicht auf den Tumor möglich ist.

In diesem Paper werden Methoden vorgestellt, welche Lösungen für solche und ähnliche Probleme bringen. Neben den Transferfunktionen werden dabei noch andere Parameter verwendet um den Farb- und Opazitätswert eines Sample-Punktes zu erhalten. Die erste vorgestellte Methode (*Depth of Field*) war der Einstieg in diese Thematik und dient vor allem dazu den Benutzer schneller auf bestimmte Details hinzuweisen. Die weiteren drei Methoden dienen dazu innere Strukturen besser darzustellen.

In den folgenden Kapiteln werden die Methoden zur Merkmals-Visualisierung noch genauer beschrieben. Zuvor werde ich aber einen Überblick über bestehende Arbeiten in diesem Gebiet geben.

2 Ähnliche Arbeiten

Es ist ein Schwerpunkt in der Volumsvisualisierung Methoden und Verfahren zu finden, welche Merkmale in einem Datensatz hervorheben. Die folgenden Arbeiten befassen sich mit dieser Aufgabe. Sie sind nur ein Auszug aus der Vielzahl von Ansätzen für diese Thematik. Sie dienen aber als Grundlage für die von mir beschriebenen Methoden.

Es gibt einige Ansätze welche sich diesem Problem angenommen haben. Es gibt Ansätze, welche auf segmentierte Daten angewendet werden können. Segmentierte Daten werden durch eine zusätzliche Datei beschrieben. In dieser Datei sind Informationen enthalten, welche einen einzelnen Datenpunkt zu einem Teil des Objekts zuordnet. Dadurch ergibt sich die Möglichkeit Teile eines Objektes separat darzustellen. Die Methode des Two-Level Rendering [2] ist eine der Methoden, die auf solche Daten angewandt wird. In dieser Arbeit wird erklärt wie durch Anwendung verschiedener Renderingtechniken auf unterschiedliche Segmente des Datensatzes interessante Teile hervorgehoben werden können. Durch passende Wahl der Renderingtechniken können mit dieser Methode innere Strukturen gut dargestellt werden. Das Problem an

dieser Methode ist jedoch, dass man dazu segmentierte Daten braucht. Es gibt noch keine zuverlässigen Verfahren welche diese Segmentierung automatisch durchführen können.

Für Datensätze welche keine Informationen zur Segmentation haben, gibt es auch eine Reihe von Ansätzen zur welche dazu dienen sollen Einblicke in die innere Struktur der Objekte zu gewinnen. Als Beispiel seien an dieser Stelle die Arbeiten von Jian-long Zhou [11, 12] genannt. In diesen Arbeiten werden je nach Position der Punkte verschieden Rendering-Techniken angewandt. Somit wird der Datensatz in einen Fokus- und einen Kontextbereich unterteilt. Für den Kontextbereich werden Nicht-Photorealistische-Renderingmethoden verwendet. Der Fokusbereich hingegen wird mit einer aufwendigeren Methoden gerendert.

Eine Einfachere Möglichkeit um innere Strukturen darzustellen ist das Clipping. Dabei wird der Datensatz mit einer Ebene geschnitten. Somit wird ein Teil des Volumens weggeschnitten. Beim Raycasting kann das Clipping ohne großen Aufwand eingebaut werden. Dabei werden die Sample-Punkte erst nach einer bestimmten Länge des Strahls zur Komposition des Pixelfarbwegs verwendet. Das heißt, dass Datenpunkte, die innerhalb dieses Abstands liegen nicht in das Ergebnisbild einfließen.

3 Methoden zur Merkmals-Visualisierung

Dieser Abschnitt beschreibt vier Methoden, welche der Extraktion von Merkmalen dienen. Die Reihenfolge, in welcher sie hier angeführt sind, entspricht ihrer Implementationsreihenfolge. Die vier unterschiedlichen Methoden sind für unterschiedliche Einsatzgebiete verwendbar. Jedoch werden alle vier in der Volumsvisualisierung verwendet.

In den folgenden Kapiteln (Implementation, Ergebnisse) wird erklärt wie die Methoden implementiert werden. Ausserdem werden noch weitere Ergebnisbilder diskutiert und die Methoden miteinander verglichen.

3.1 Depth of Field

Beim *Depth of Field* soll der gleichnamige Effekt aus der Fotografie in der Volumsvisualisierung simuliert werden. Dieser Effekt entsteht dadurch, dass in der Fotografie eine Linse verwendet wird. In Abbildung 1 ist ein Modell einer Linse abgebildet. Es ist zu erkennen, dass die Linse einen bestimmten Fokuspunkt hat. Gegenstände, die nahe an diesem Fokuspunkt

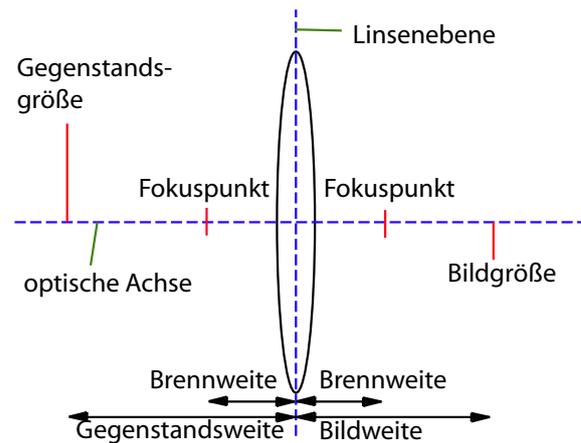


Abbildung 1: *Optische Linse.* Der Brennpunkt ist der charakteristische Wert einer Linse. Er bestimmt auch welcher Teil einer Szene scharf dargestellt wird.

positioniert sind, werden auch schärfer im Bild dargestellt. Je weiter ein Objekt von diesem Fokuspunkt entfernt ist, desto unschärfer wird es im Bild dargestellt. In der Fotografie wird der Fokusbereich und die Stärke des Effekts durch den Abstand von Linsen in einem optischen System und die Einstellung der Blende beeinflusst.

Den selben Effekt gibt es auch im menschlichen Auge. Das Auge besteht auch aus einer Linse, welche mittels Muskeln ihre Krümmung ändern kann. Je nach dem wie diese Muskeln angespannt sind wird der Fokuspunkt eingestellt. Sind die Muskeln entspannt, dann ist das Auge auf Fernsicht eingestellt.

In der Volumsvisualisierung wird üblicherweise von einem anderen Modell ausgegangen. Man nimmt an, dass alles im Fokusbereich liegt und somit alle Teile scharf dargestellt werden. In Abbildung 2 ist das theoretische Modell der Lochkamera dargestellt. Sie besitzt ein unendlich kleines Loch und ein unendlich empfindliches Photopapier. Durch das Loch ohne Linse gibt es keinen Fokuspunkt. Alle Teile einer Szene würden somit scharf dargestellt werden.

Dieses Modell ist in vielen Fällen ideal für die Volumsvisualisierung, da alle Teile scharf dargestellt werden und somit auch genau betrachtet werden können. Es ist das einfachste Modell. Ein scharfes Bild der ganzen Szene wird automatisch erzeugt, wenn das normale Raycasting-Verfahren angewendet wird.

Jedoch beinhaltet der Effekt des *Depth of Field* den Vorteil, dass mit seiner Hilfe interessante Teile hervorgehoben werden können. Wie in den Arbeiten von Robert Kosara [6, 7] erwähnt wird, ist dieser Effekt für den Menschen sehr intuitiv. Dies kommt eben daher, dass das menschliche Auge nach demselben Prinzip die Umgebung wahrnimmt. Es ist daher ein geeignete

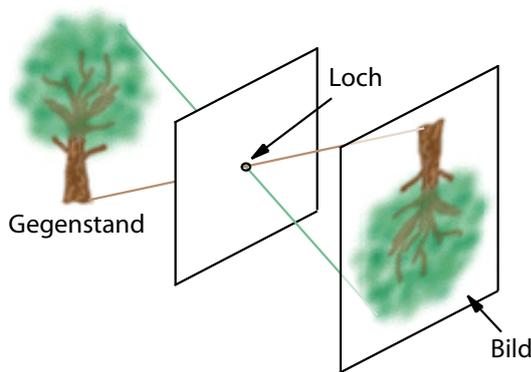


Abbildung 2: Theoretisches Modell einer Lochkamera. Objekte werden immer scharf dargestellt.



Abbildung 3: Schachspiel, das mit Semantic Depth of Field gerendert wurde. Der Blick des Betrachters sollte auf die scharf dargestellten Schachfiguren gelenkt werden. Quelle: [6]

ter Effekt um den Betrachter schnell auf die wichtigen Teile einer Szene zu lenken.

In der Arbeit von Robert Kosara [6] wurde der Effekt für *Semantic Depth of Field* verwendet. Die Teile einer Szene werden dabei nach ihrer Wichtigkeit eingestuft. Je essentieller dann ein Teil ist, desto schärfer wird es dargestellt. Abbildung 3 zeigt ein mit dieser Methode gerendertes Schachspiel. Dabei sind die Figuren schärfer, welche gerade bei einem Spielzug beteiligt sind. Die Methode des *Semantic Depth of Field* wird vor allem in der Informationsvisualisierung eingesetzt. Es müssen segmentierte Daten vorliegen um die Methode anwenden zu können.

Die hier vorgestellte Methode bezieht sich auf die Volumsvisualisierung. In der Volumsvisualisierung liegen die Daten meistens nicht in segmentierter Form vor, daher ist die Methode *Semantic Depth of Field* nicht anwendbar. Bei *Depth of Field* wird die

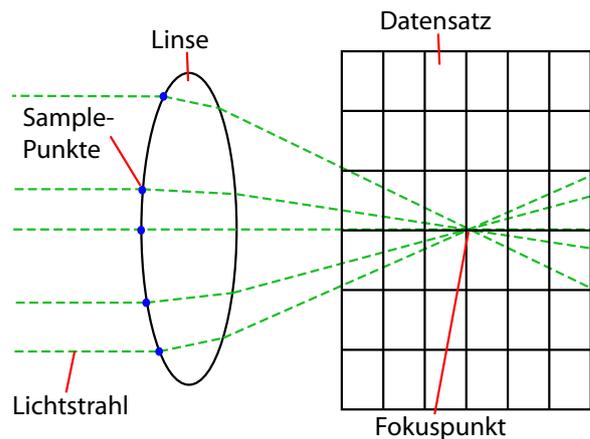


Abbildung 4: Prinzip der Implementierung von Depth of Field in Volumsdaten durch mehrere Sample-Punkte.

Unschärfe abhängig vom Fokuspunkt gesteuert. Diese Methode ist daher ein der Natur nachempfundenes Prinzip. Gleich dem menschlichen Auge oder einer Kamera, gibt es einen Fokuspunkt, in dessen Bereich die Szene am schärfsten dargestellt wird. Je weiter Teile der Szene vom Fokuspunkt entfernt sind, desto unschärfer werden sie dargestellt.

Wie zu Beginn dieses Kapitels schon kurz erwähnt wurde, produziert das Raycasting-Verfahren automatisch scharfe Bilder. Daher ist es notwendig das Raycasting-Verfahren zu adaptieren um den Effekt des *Depth of Field* erzeugen zu können.

Die von mir implementierte Methode funktioniert auch mit dem Raycasting-Verfahren. Es wird aber das Raycasting mehrmals auf den Datensatz angewendet und danach die einzelnen Ergebnisbilder zu einem Resultierenden vereint. Abbildung 4 zeigt das Prinzip wie die verschiedenen Raycasting-Bilder erzeugt werden. Es werden auf einer Linse Sample-Punkte ausgewählt und für jeden dieser wird ein Ergebnisbild mittels Raycasting erzeugt. Die Linse ist dabei nur symbolisch zu verstehen und soll nur einen Bereich angeben in dem sich die Sample-Punkte befinden. Ein Sample-Punkt entspricht einer bestimmten Blickrichtung auf den Datensatz. Die Strahlen beim Raycasting werden dabei pro Sample-Punkt in die eine vorgegebene Blickrichtung geschossen. Die Blickrichtung ist bei einer realen Linse von ihrer Krümmung und dem damit verbundenen Fokuspunkt abhängig.

Es gibt einige Parameter, welche den Effekt beeinflussen. Einer ist der Linsendurchmesser. Ist dieser groß so werden Teile, welche einen bestimmten Abstand zum Fokuspunkt haben, unschärfer dargestellt als bei einem kleineren Durchmesser. Der zweite Parameter ist der Fokuspunkt: dieser gibt an in welchem Punkt sich die Raycasting-Strahlen der verschiede-

nen Sample-Punkte treffen. Der Fokuspunkt definiert daher eine Ebene in der Szene, in welcher die Teile am schärfsten dargestellt werden. Der dritte Parameter der das Ergebnisbild beeinflusst ist die Anzahl der Sample-Punkte. Je mehr Sample-Punkte verwendet werden umso feiner ist die Abstufung im Ergebnisbild. Wird ein großer Linsendurchmesser gewählt, dann besteht die Notwendigkeit mehr Sample-Punkte zu verwenden, da gerade bei hoher Unschärfe die Ergebnisse der verschiedenen Sample-Punkte stark voneinander abweichen. Daher kann auch nur mit vielen Sample-Punkten ein gutes Bild erzeugt werden.

Die Bilder der einzelnen Sample-Punkte werden dann einfach überlagert und man erhält dadurch ein Ergebnisbild. In Abbildung 5 ist ein Leopardengecko zu sehen, der mit der Methode *Depth of Field* gerendert wurde. Die zwei Bilder wurden dabei mit einem unterschiedlichen Fokuspunkt gerendert. Im linken Bild ist der Fokuspunkt weiter vorne und im rechten ist er weiter hinten.

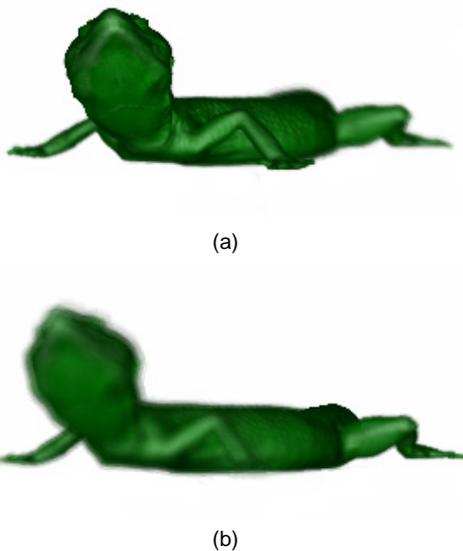


Abbildung 5: *Depth of Field* Ergebnis eines Leopardengecko-Datensatzes. (a) Fokuspunkt im Kopfbereich (vorne), (b) Fokusbereich am Schwanz (hinten)

Die Sample-Punkte werden zufällig ausgewählt. Dies ist aber nur für Einzelbilder sinnvoll. Bei Animationen werden die Sample-Punkte möglichst gleichmäßig auf der Linsenfläche verteilt. Dies ist notwendig, da die einzelnen Ergebnisbilder bei zufälligen Sample-Punkten zu stark voneinander abweichen würden und so die Animation nicht flüssig ablaufen würde.

Der Vorteil dieser Methode liegt darin, dass der Betrachter auf wichtige Teile einer Szene oder eines Da-

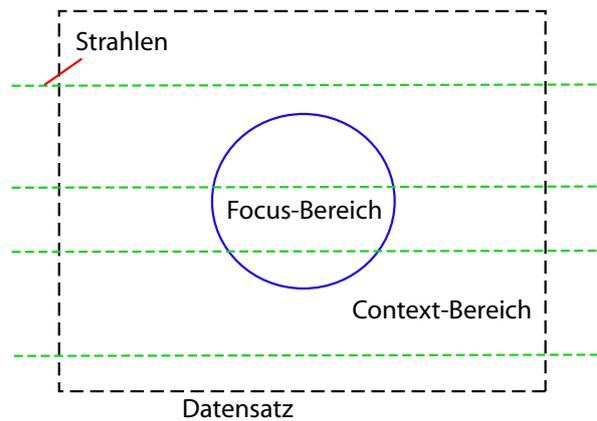


Abbildung 6: *Focus und Context Bereich*.

tensatzes gelenkt werden kann. Sie kann vor allem bei Visualisierungen zu Präsentationszwecke eingesetzt werden. Jedoch ist die Methode sehr aufwendig, da für ein Ergebnisbild ein Datensatz, je nach Anzahl der Sample-Punkte, öfters gerendert werden muss. Für Präsentationszwecke ist es aber in den meisten Fällen nicht so wichtig schnell das Ergebnis zu haben.

3.2 Focus+Context

Die zweite Methode der Merkmal-Extraktion, die ich vorstellen will, ist *Focus+Context*. Dies ist ein weit verbreitete Methode, welche vor allem in der Informationsvisualisierung vorkommt. Es gibt aber auch schon einige Methoden, welche *Focus+Context* auch auf die Volumsvisualisierung anwenden.

Das Prinzip von *Focus+Context* ist relativ simple. Der gesamte Bereich im Betrachtungsfeld wird in einen Focus-Bereich und in einen Context-Bereich unterteilt. Der Focus-Bereich ist von besonderem Interesse und wird deshalb auch aufwendiger und genauer dargestellt. Der Context-Bereich soll nur einen Überblick über den Rest der Daten geben.

In Abbildung 6 sieht man wie bei der hier vorgestellten Methode die Unterteilung von Focus-Bereich und Context-Bereich ist. Der Focus-Bereich ist eine Kugel und der Rest des Datensatzes ist einfach der Context-Bereich.

Um diese Einteilung in das Ergebnisbild einfließen zu lassen, werden die Strahlen beim Raycasting an jedem Sample-Punkt entlang des Strahls geprüft ob sie innerhalb der Focus-Kugel sind. Ist dies der Fall so wird eine andere Methode verwendet um den dortigen Farb- und Opazitätswert zu ermitteln als im Context-Bereich. In den folgenden zwei Unterpunkten wird erklärt wie dieser Farb- und Opazitätswert in den verschiedenen Bereichen berechnet wird.

Focus

Der Focus-Bereich wird, wie eben schon erwähnt, aufwendiger gerendert als der Context-Bereich. In ihm befinden sich die für den Betrachter interessanten Teile. Ich habe daher die von Marc Levoy [8] eingeführte Methode verwendet. Dabei wird jedem Datenwert mittels einer Transferfunktion ein Farb- und Opazitätswert zugeordnet. Zusätzlich wird noch der Gradientenbetrag verwendet um Bereiche mit wenig Änderung durchsichtiger darzustellen, als Bereiche mit großer Änderung. Dies erweist sich in den meisten Fällen als sinnvoll, da oft die Bereiche mit großer Änderung der Werte von mehr Interesse sind. Solche Bereiche sind zum Beispiel Objektgrenzen oder Kanten. Die Berechnung des Gradientenbetrages lässt sich von der Berechnung des Gradienten ableiten. Die Berechnung des Gradienten wird beim Context-Rendering-Verfahren noch genauer beschrieben.

Die Grenzen zum Context Bereich werden als Kombination der beiden Rendering-Methoden dargestellt. Dabei werden die zwei Bereiche kontinuierlich miteinander vermischt. Die Vermischung erfolgt mittels Interpolation zwischen den Farb- und Opazitätswerten der beiden Bereiche. Der Übergang von der Context- zur Focus-Renderingmethode erfolgt komplett im Focus-Bereich. Je näher ein Sample Punkt der Focus-Grenze ist, desto mehr Einfluss hat der Farb- und Opazitätswert der Context-Rendering-Methode.

Context

Der Context-Bereich dient hauptsächlich dazu um die Struktur des Objekts darzustellen. Er soll aber nicht die Sicht auf das Wesentliche verdecken. Wie auch in den Arbeiten von Penny Rheingans und Balazs Csefalvi [1, 9] nachzulesen ist, ist die beste Möglichkeit dies zu gewähren, wenn nur die Silhouetten der Objekte dargestellt werden. In den nächsten drei Unterpunkten wird mein Lösungsweg zur Extraktion der Silhouetten beschrieben.

Gradientenberechnung: Der Gradient an einer Stelle im dreidimensionalen Raum lässt sich mit dem Nabla-Operator berechnen.

$$\nabla f(x, y, z) = \left(\frac{\delta f}{\delta x}, \frac{\delta f}{\delta y}, \frac{\delta f}{\delta z} \right) \quad (1)$$

Die Formel 1 ist die exakte Lösung für den Gradienten. Dazu muss aber eine Funktion gegeben sein welche das Volumen angibt. In der Visualisierung bestehen die Datensätze aber aus Punktwolken und besitzen auch keine Funktion welche das Volumen beschreibt.

Im Falle von diskreten Werten, wie sie in der Volumvisualisierung vorkommen, muss man Methoden verwenden, welche die exakte Gradientenberechnung annähern. Es gibt eine Reihe von Methoden und Filtern welche diese Aufgabe lösen. Ich habe die Methode der zentralen Differenzen verwendet.

Bei dieser Methode wird die Berechnung des Nabla-Operators einfach auf diskrete Werte übertragen.

$$\nabla f(x_i, y_i, z_i) = \begin{pmatrix} \frac{1}{2}f(x_{i-1}, y_i, z_i) - \frac{1}{2}f(x_{i+1}, y_i, z_i) \\ \frac{1}{2}f(x_i, y_{i-1}, z_i) - \frac{1}{2}f(x_i, y_{i+1}, z_i) \\ \frac{1}{2}f(x_i, y_i, z_{i-1}) - \frac{1}{2}f(x_i, y_i, z_{i+1}) \end{pmatrix} \quad (2)$$

In dieser Formel 2 sieht man dass zur Bestimmung der einzelnen Komponenten die Differenz der zwei angrenzenden Werte in den drei Hauptachsrichtungen verwendet wird. Dies ist eine grobe Annäherung an die partiellen Ableitungen in der exakten Formel.

Es gibt noch eine Reihe aufwendigerer Filter, welche mehr als nur sechs benachbarte Werte in die Gradientenberechnung mit einbeziehen. Jedoch steigt dadurch auch der Rechenaufwand. Für *Focus+Context* hat sich die Methode der zentralen Differenzen an den Ergebnissen als effizient erwiesen.

Betonung der Grenzen: Das Ergebnis der Gradientenberechnung kann nun verwendet werden um die Opazität so zu steuern, dass die Grenzen der Objekte stärker zur Geltung kommen als gleich bleibende Regionen. Jeder Skalarwert bekommt durch die Transferfunktion schon eine bestimmte Opazität und einen bestimmten Farbwert zugewiesen. Der Farbwert wird durch die Gradienten Extraktion nicht verändert. Der Opazitätswert wird wie folgt verändert:

$$o_g = o_v \left(k_{gc} + k_{gs} \left(\frac{\|\nabla f\|}{\|\nabla_{max}\|} \right)^{k_{ge}} \right) \quad (3)$$

In dieser Formel ist o_v der Opazitätswert, welchen das Voxel nach der Transferfunktion besitzt. ∇_f ist der Wert des Gradienten an der betrachteten Stelle. ∇_{max} ist der maximale Wert eines Gradienten, welcher im betrachteten Datensatz auftreten kann. Die Division der beiden Gradientenbeträge bewirkt, dass steile Übergänge mehr Opazität haben als weniger scharfe Kanten.

Die Koeffizienten k_{gc} , k_{gs} und k_{ge} beeinflussen die Darstellung der Grenzen. k_{gc} steuert den Einfluss der Originalopacity. Ist k_{gc} im Vergleich zu k_{gs} klein, so werden fast ausschließlich die Grenzen dargestellt und keine Zwischenräume. Der Koeffizient k_{ge} beeinflusst die Schärfe der Kanten. Ist er hoch so werden die Kanten feiner.

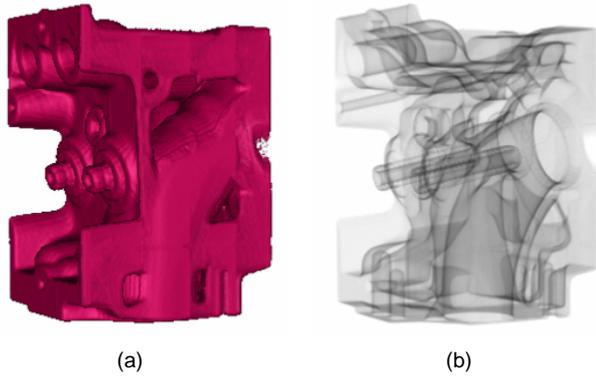


Abbildung 7: Maschinenblock, (a) mit einer einfachen Transfer Function, (b) mit Betonung der Grenzen

In Abbildung 7 sieht man das Ergebnis bei Anwendung der obigen Formel. Für das in (b) erzeugte Bild wurden die Koeffizienten auf $k_{gc} = 0$, $k_{gs} = 1$ und $k_{ge} = 0.7$ gesetzt. Im nächsten Abschnitt wird nun noch beschrieben wie man diese Grenzen in Abhängigkeit der Blickrichtung darstellt.

Betonung der Silhouetten: Wie man in Abbildung 7 (b) sieht, geht durch die Betonung der Grenzen die Tiefeninformation verloren. Diese ist in der Abbildung (a) noch gut ersichtlich. Allerdings bekommt man einen besseren Einblick in die inneren Strukturen eines Objektes. Um diesen Einblick noch zu verstärken ist es von Vorteil, nur die Silhouetten von Objekten im Datensatz darzustellen. Somit werden störende Grenzen, die nicht zur Silhouette gehören, nicht dargestellt. Diese Teile sind relativ unwichtig, da man sie durch die fehlende Tiefeninformation schwer zuordnen kann.

Die Formel für die Berechnung der Silhouetten lautet:

$$o_s = o_g \left(k_{sc} + k_{ss} (1 - \text{abs}(\nabla_{f_n} * v))^{k_{se}} \right) \quad (4)$$

o_g ist dabei die berechnete Opazität von zuvor. Das bedeutet es werden nur die zuvor berechneten Grenzen weiter modifiziert. Es gibt auch Ansätze, in denen statt der Opazität der Grenzberechnung o_g nur die zuvor verwendete Opazität der Transfer Function o_v verwendet wird. Dies hat den Effekt zur Folge, dass feinere Kanten mehr zur Geltung kommen. Dadurch verliert man allerdings Informationen über die Dichte des Materials, da dadurch der Voxelbetrag gar keinen direkten Einfluss auf die Opazität an einer Stelle hat. ∇_{f_n} ist der normierte Gradient (auch Normale genannt) und v ist die Blickrichtung. Das skalare Produkt der beiden ist dem \cos des Winkels zwischen den Vektoren gleichzustellen. Das bedeutet, dass Bereiche in denen der Gradient eher orthogonal zur Blickrichtung steht, stärker berücksichtigt werden.

Die Koeffizienten k_{sc} und k_{ss} steuern den Übergang der Kanten. Ist k_{ss} im Vergleich zu k_{sc} hoch so werden nur Silhouetten dargestellt. Umgekehrt werden Grenzen, die nicht zur Silhouette gehören, auch undurchsichtiger dargestellt. Wenn $k_{sc} = 1$ und $k_{ss} = 0$ dann entspricht das Ergebnis genau dem in Abbildung 8 (b) dargestellten Bild.

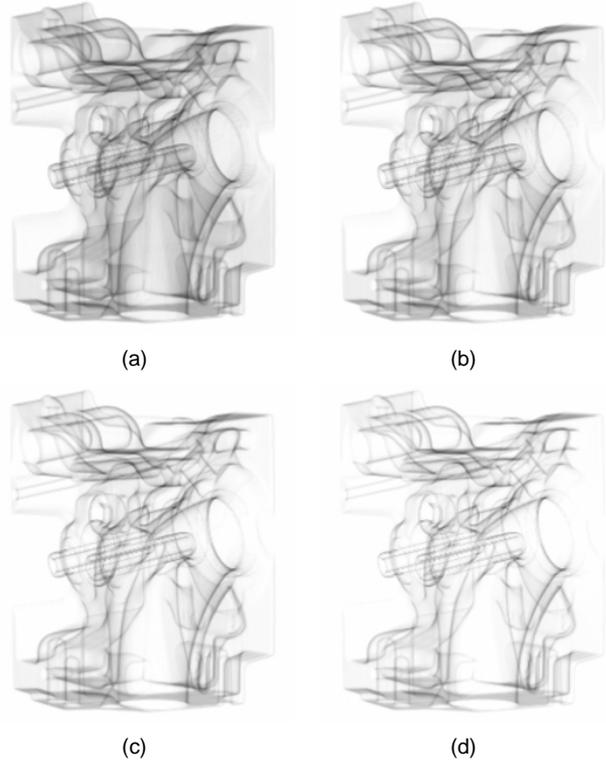


Abbildung 8: Maschinenblock mit $k_{sc} = 0$ und $k_{ss} = 1$, (a) $k_{se} = 0.5$, (b) $k_{se} = 1$, (c) $k_{se} = 1.7$, (d) $k_{se} = 2.5$

k_{se} steuert die Schärfe der Silhouette. Je höher k_{se} desto feiner werden die Silhouetten dargestellt. In Abbildung 8 sieht man den schon in Abbildung 7 verwendeten Maschinenblock in einer Darstellung mit betonten Silhouetten. Die vier Bilder wurden mit unterschiedlichen Werten für k_{se} gerendert.

Bei der Implementation werden noch Parameter verwendet, um die Opazität des Context-Bereichs in Abhängigkeit zum Abstand vom Focus-Bereich zu steuern. Der Context wird daher in der näheren Umgebung des Focus-Bereich stärker dargestellt als etwas weiter davon entfernt.

Um ein Ergebnisbild durch die *Focus+Context* Methode zu erhalten, wird für jeden Sample-Punkt entlang eines Strahls ein Farb- und Opazitätswert entsprechend seiner Lage (entweder Focus-, Context-Bereich oder eine Interpolation der beiden im Übergangsbereich) ermittelt. Die Werte entlang eines Strahl werden danach mittels des Kompositionsver-

fahren, wie es auch in der Arbeit von Marc Levoy [8] erwähnt ist, zu einem resultierende Farbwert für das entsprechende Pixel vermischt.

In Abbildung 9 ist das Ergebnis der *Focus+Context* Methode zu sehen. Der Focus-Bereich ist allen vier Bildern leicht zu erkennen und wird auch nicht von davor liegenden Teilen verdeckt, da im Context-Bereich nur die Silhouetten dargestellt werden.

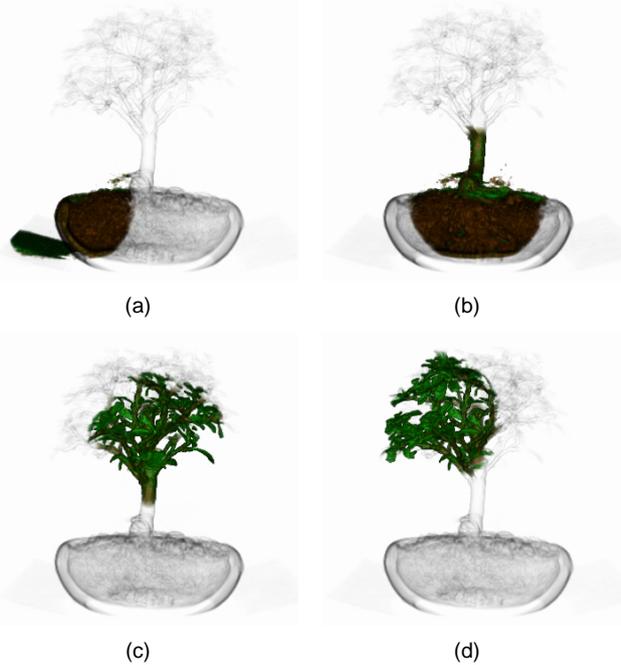


Abbildung 9: *Bonsai Datensatz einer Computertomographie. Die Position des Focus-Bereichs in den vier Bildern ist unterschiedlich.*

Der Einfluss des Transferkoeffizienten ist in Abbildung 10 abgebildet. In Bild (a) ist der Koeffizient höher als in Bild (b). Durch einen kleineren Transferkoeffizienten wird der äußere Bereich des Focus-Bereich transparenter dargestellt und gibt so den Blick auf den inneren Bereich frei.

Der Vorteil von *Focus+Context* besteht darin, dass man unwichtige Teile mehr oder weniger mit Hilfe des Context-Bereichs ausblenden kann, aber dennoch einen Überblick über diese Teile hat. Diese Methode ist daher in der Volumsvisualisierung sehr gut einsetzbar. Zum Beispiel müssen in der medizinischen Visualisierung bestimmte Teile eines Körpers genauer betrachtet (zum Beispiel ein Tumor) werden, während man mit dem Context nur sehen will, wo dieses Teil im Körper liegt. Es gibt aber noch eine Reihe anderer Anwendungen, bei denen ein solcher Effekt wünschenswert ist.

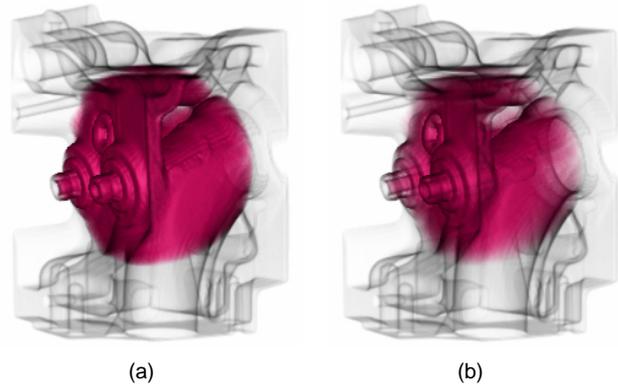


Abbildung 10: *Maschinenblock mit Focus+Context gerendert. In Bild (a) ist der Übergang zwischen den zwei Bereichen stärker als in Bild (b).*

3.3 3D Dithering

Die dritte Methode, auf welche ich in diesem Paper eingehen möchte, ist das *3D Dithering*. Es handelt sich dabei um eine Abwandlung des aus der Druckerei bekannten Dithering. Beim Dithering, auch Dithern genannt, werden Schwarz-Weiß- oder Farbgrafiken "weichgezeichnet". Man erreicht dies durch die Simulation von Farben und Graustufen über Punktmuster, die zusätzliche Zwischentöne erzeugen können. Es werden mehr Farbwerte dargestellt, als das Ausgabegerät eigentlich wiedergeben kann. Das führt zwar zu einer gewissen Unschärfe der Darstellung, hat aber den Vorteil, dass das menschliche Auge die einzelnen Farbwerte der Bildpunkte nicht erkennt, sondern nur fließende Übergänge zwischen den Farben oder Grauwerten.

In Abbildung 11 ist ein Beispielbild, welches mit einem solchen Dithering-Verfahren dargestellt wurde. Durch die Größe und den Durchmesser der Dithering-Punkte wirken Teile im Bild heller und andere dunkler. Das menschliche Auge nimmt nicht die einzelnen Punkte wahr sondern den gesamten Farbton. Dadurch lässt sich trotz der einfachen Darstellung das eigentliche Bild leicht erkennen.

Das Dithering wird, wie bereits erwähnt, in der Druckerei eingesetzt. Das hat den Grund, dass durch dieses Verfahren Farben eingespart werden können, da Farben durch Dithering-Muster mit vorhandenen Farben erzeugt werden können. In der Volumsvisualisierung bringt dieser Vorteil eigentlich nichts, da am Bildschirm alle Farben vorhanden sind. Die Idee meines Ansatzes ist jedoch durch die Darstellungen eines Datensatzes in Punkten (Quadern) einen besseren Einblick in die inneren Bereiche des Datensatzes zu bekommen.

Ein weiterer Einflussfaktor für meinen Ansatz war die Arbeit von Ivan Viola [10]. In dieser Arbeit ist eine



Abbildung 11: Beispiel eines Bildes das mit Dithering dargestellt wird (Quelle: Stefan Roettger, Universität Stuttgart).

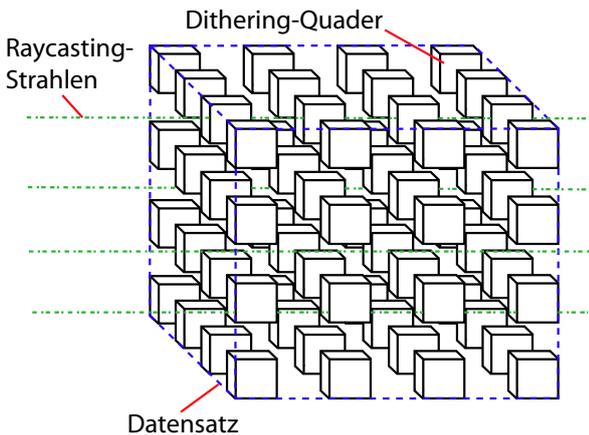


Abbildung 12: Schematische Darstellung der 3D Ditheringmethode.

Methode beschrieben in der Teile des Objektes nach ihrer Wichtigkeit eingestuft werden. Weniger wichtige Teile werden durchsichtiger dargestellt, um die wichtigeren Teile hervorzuheben. In einem Ansatz wird die Durchsichtigkeit mittels "Screen-Door" gesteuert. Dabei wird nur ein Gitter des unwichtigeren Teil dargestellt. Dazwischen liegen durchsichtige Flächen, durch welche die Sicht auf dahinter liegende Teile freigegeben wird.

In der hier beschriebenen Methode des 3D Dithering habe ich diese beiden Ansätze vereint. Die Idee dabei ist, den Datensatz in "Punkte" zu unterteilen. Im Dreidimensionalen werden solche "Punkte" als Quader verwirklicht. Diese Quader werden gleichmäßig im Datensatz verteilt. In Abbildung 12 sieht man eine schematische Darstellung dieser Idee.

Die Dithering-Quader kennzeichnen Bereiche in denen eine andere Renderingmethode verwendet wird. Liegt ein Sample-Punkt eines Strahl in einem solchen Quader, dann wird der Farb- und Opazitätswert nach der im vorherigen Kapitel beschriebenen Methode von Marc Levoy [8] ermittelt. Die Farb- und Opazitätswerte des dazwischen liegenden Freiraums werden mit dem im Kapitel *Focus+Context* beschriebenen Verfahren zur Darstellung des Context ermittelt. Der Farbwert des resultierenden Pixels wird wieder mittels Komposition ermittelt.

Die Verwendung der Context-Rendering-Methode für den Freiraum hat sich als sehr sinnvoll erwiesen. Die Silhouetten dienen der Zuordnung der Dithering-Quader. Es ist somit einfacher die Lage des Dithering-Quaders im Datensatz zu erkennen. Versuche in denen der Freiraum einfach als durchsichtig angenommen wurde, haben sich deshalb nicht bewährt.

Der Nutzen der 3D Dithering Methode soll es sein einen Überblick über das Innere eines Objekts zu bringen. Dadurch dass der Freiraum mit der Context-Methode gerendert wird, ist dieser relativ durchscheinend und gibt so den Blick auf Dithering-Quader im Inneren des Datensatzes frei.

Um beim Raycasting einen Sample-Punkt einem Dithering-Quader oder einem Freiraum zuzuordnen, verwende ich eine Maske. Diese Maske hat dieselbe Größe wie der Datensatz. Sie wird in einem Preprocessing-Schritt mit Werten befüllt. An Stellen, an denen ein Dithering-Quader liegt, stehen andere Werte als an Stellen an denen sich ein Freiraum befindet.

Während des Rendering-Verfahrens wird für jeden Sample-Punkt zusätzlich zu den Daten auch der Wert in der Maske ermittelt. Dies geschieht mittels der "Nearest-Neighbor"-Methode. Dabei wird einfach der Wert des Punktes verwendet, welcher dem Sample-Punkt räumlich am nächsten ist. Steht an dieser Stelle ein Wert, welcher einem Dithering-Quader zugeschrieben wird, so wird der Farb- und Opazitätswert mit den Transferfunktionen ermittelt ansonsten wird er mit der Methode zum Rendern des Context ermittelt.

Die Maske wird einmalig für den Datensatz ermittelt und danach nicht mehr geändert. Das heißt, dass bei einer Transformation des Datensatzes (Drehung, Skalierung) sich die Position oder Ausrichtung der Dithering-Quader nicht ändert. Somit kann der Datensatz gedreht werden und man erhält dadurch einen bessere Einsicht auf alle Dithering-Quader. Um den Speicheraufwand für die Maske klein zu halten reicht es aus einen Wert als Dualwert abzuspeichern. Mehr wird nicht gebraucht, da es nur die Unterteilung in Dithering-Quader und Freiraum gibt.

In Abbildung 13 ist ein Ergebnisbild der 3D Dithering Methode zu sehen. Die Rotation des Datensatzes

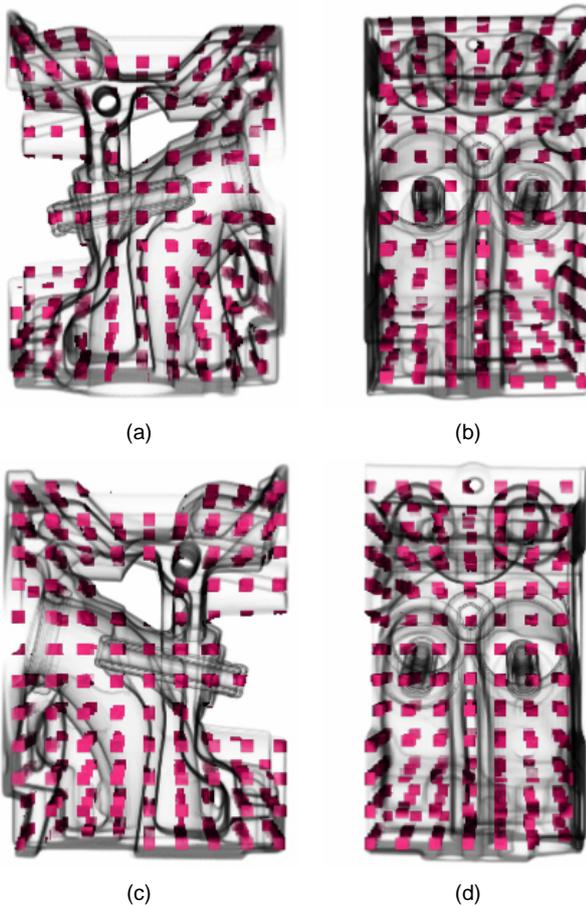


Abbildung 13: Maschinenblock gerendert mit der 3D Dithering Methode. Die vier Bilder unterscheiden sich in ihrer Orientierung. Angefangen bei Bild (a) wird jedes weitere Bild um 90° weiter gedreht.

in den vier Bildern ist unterschiedlich. Durch das perspektivische Rendering hat man so einen guten Einblick auf die inneren Dithering-Quader.

Ich habe in meiner Implementation einige Parameter verwendet, welche das Dithering-Muster verändern. Einerseits lässt sich die Größe der Dithering-Quader variieren. Wird dieser Parameter erhöht so werden weniger Quader dargestellt. Größere Quader bieten einen gröberen Einblick in die Daten. Das kann dazu verwendet werden um einen groben Überblick über das Innere eines Datensatzes zu bekommen. Ein weiterer Parameter bestimmt die Distanz zwischen den Dithering-Quadern. Je größer dieser Parameter ist, desto einfacher sind die inneren Quader zu sehen. Jedoch wird dadurch mehr vom Datensatz nur mit Context-Methode gerendert.

Es ist notwendig, dass diese Parameter wählbar sind, da sonst Teile, welche in den Zwischenräumen liegen, nie mit der Rendering-Methode für den Dithering-Quader gerendert würden.

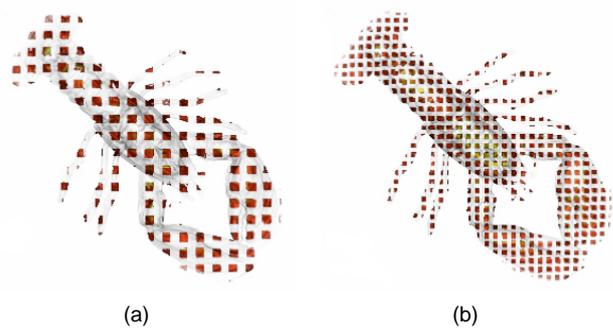


Abbildung 14: Hummer gerendert mit der 3D Dithering Methode. In Bild (a) sind die Dithering-Quader und der Abstand zwischen ihnen größer als in Bild (b).

Die Abbildung 14 zeigt einen Hummer-Datensatz, welcher mit der 3D Dithering Methode gerendert wurde. Das linke Bild wurde dabei mit höheren Werten für die Distanz und Größe der Dithering-Quader gerendert. Es ist gut erkennbar, dass die Silhouetten im Freiraum einen guten Überblick über den gesamten Datensatz bieten.

Die beschriebene Methode eignet sich besonders gut zur Entdeckung interessanter Teile in einem Datensatz. Wenn eine "vernünftige" Transferfunktion vorhanden ist, kann man relativ schnell interessante Teile an deren Farbe erkennen. Mit 3D Dithering sticht dann der Dithering-Quader, welcher den interessanten Teil schneidet, heraus. Dessen Lage ist somit einfach zu erkennen. Nach der Ermittlung der Lage kann man auf die im vorherigen Kapitel vorgestellte Focus+Context Methode umsteigen und eben den Fokusbereich auf die im 3D Dithering ersichtliche Position setzen.

3.4 Magic Lamp

Wie bei Focus+Context und bei 3D Dithering geht es auch darum die Daten in einen Focus und einen Context Bereich zu unterteilen. In diesen beiden Ansätzen und auch in allen anderen gängigen Ansätzen (siehe Kapitel ähnliche Arbeiten) wird der Focus-Bereich und der Context-Bereich immer räumlich getrennt. Das heißt es das die Renderingmethode, ob Focus oder Context, abhängig ist von der Lage des jeweiligen Sample-Punktes am Strahl.

Die Magic Lamp Methode ist ein neuer Ansatz in dem der Context-Bereich und der Focus-Bereich nicht räumlich von einander getrennt werden. Sie soll einen neuen, anderen Einblick in den Datensatz bringen.

Der Name Magic Lamp wurde deshalb gewählt, weil die Einteilung in Focus-Bereich und Context-Bereich abhängig von einer bestimmten Richtung (genauer: die Richtung der Normalvektoren) ist. Die Annahme ist daher, dass aus einer Richtung eine Art Lampe strahlt.

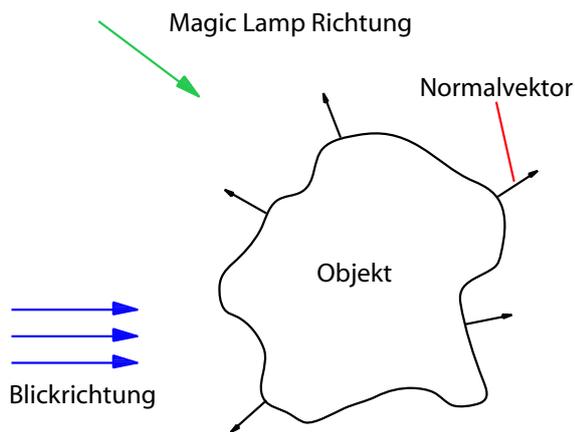


Abbildung 15: Schematische Darstellung der Magic Lamp Methode.

Da diese Lampe allerdings nicht die Eigenschaften einer normalen Lampe aufweist kommt der Name *Magic Lamp*. Bei einer normalen Lichtlampe werden die angestrahlten Teile je nach Lage zur Lichtquelle heller oder dunkler dargestellt. In der hier vorgestellten Methode bestimmt die "Lampe" nicht die Helligkeit der angestrahlten Teile sondern die für diesen Teil verwendete Rendering-Methode. Zusätzlich zur *Magic Lamp* gibt es aber noch eine normale Lichtquelle, welche unabhängig von der *Magic Lamp* die Szene bestrahlt.

Die Idee an dieser Methode ist Objektgrenzen oder Kanten, welche in die Richtung der "Lampe" gerichtet sind, hervor zu heben. Der Vorteil zu bestehenden Lösungen welche Focus- und Context-Bereiche unterteilen ist, dass bei dieser Methode die jeweiligen Teile unabhängig von ihrer Position im Datensatz hervorgehoben werden. Durch geeignete Wahl der Lampenrichtung können so gewünschte Teile des Datensatzes hervorgehoben werden. Die Sicht auf diese Teile ist auch nicht verdeckt, da der restliche Daten mit einer Context-Rendering-Methode gerendert werden.

In Abbildung 15 ist eine schematische Darstellung der *Magic Lamp* Methode dargestellt. Die Umrandung des Objekts soll die Objektgrenzen darstellen. Die Richtung des Normalvektors der Objektgrenze und die Richtung der "Lampe" beeinflussen dabei die Wahl der Rendering-Methode.

Die Position der Lampe kann verändert werden, damit die Teile eines Objekts im Focus-Bereich sind, welche für den Betrachter von Interesse sind. Es reicht aber auch nur die Richtung aus der die "Lampe" strahlt anzugeben. Dies reicht deshalb aus, weil bei dieser Methode davon ausgegangen wird, dass die *Magic Lamp* unendlich weit entfernt ist und daher die Strahlen parallel verlaufen. Deshalb muss nur eine Richtung angegeben werden, da die anderen parallel dazu verlaufen.

Der Focus-Bereich wird wieder, wie in vorhergehenden Kapiteln, mit der Methode von Marc Levoy [8] gerendert. Vom Rest werden nur die Silhouetten gerendert (siehe Kapitel *Focus+Context*). Das Ergebnisbild wird durch die Komposition der Farb- und Opazitätswerte entlang der Strahlen beim Raycasting ermittelt.

Mit Hilfe des Normalvektors wird die Rendering-Methode für einen Sample-Punkt gewählt. Der Normalvektor ist der normierte Gradient. Der Gradient eines Sample-Punkts wird mit der Zentralen-Differenzen-Methode ermittelt. Der Normalvektor kann aus dem Gradienten gewonnen werden, wenn dieser durch seine Länge dividiert wird.

Die eigentliche Klassifikation erfolgt mit folgender Formel:

$$\cos \varphi = v_{ml} \times n \quad (5)$$

v_{ml} ist dabei ein normierter Vektor, welcher in die Richtung der Strahlen der *Magic Lamp* zeigt. n ist der Normalvektor. Das skalare Produkt \times liefert bei normierten Vektoren den \cos des dazwischen liegenden Winkels. Ist der $\cos \varphi$ gleich 0 dann stehen die beiden Vektoren orthogonal aufeinander. Ergibt das skalare Produkt 1 so zeigen beide Vektoren in die gleiche Richtung. Der für diese Methode relevante Wert der Gleichung ist aber -1 . Dann zeigt der Normalvektor genau gegen den Vektor der *Magic Lamp*. Das heißt das die Objektgrenze genau in Richtung der "Lampe" gerichtet ist.

Um die Rendering-Methode für einen Punkt zu wählen muss noch ein Bereich in der Umgebung von -1 gewählt werden. Liegt das Ergebnis des skalaren Produkts in diesem Bereich, dann wird der Wert des Punkts mit den normalen Transferfunktionen in einen Farb- und Opazitätswert transformiert. Da der \cos nicht kleiner als -1 sein kann braucht der Bereich nur von $[x, -1]$ reichen. Je größer dieser Bereich ist desto mehr Punkte werden mit den normalen Transferfunktionen gerendert. x soll dabei aber nicht größer oder gleich 0 sein. Ansonsten würden zu viele Punkte klassifiziert. Die restlichen Teile des Objekts, welche nicht in diesem Bereich liegen, werden nur als Silhouetten dargestellt.

Durch die Normierung kommt es auch dazu, dass betragsmäßig kleine Gradienten nicht mehr von betragsmäßig großen Gradienten unterschieden werden können. Dies führt dazu, dass mehr Teile im Klassifikationsbereich liegen und dadurch auch mehr Teile des Objekts mit den normalen Transferfunktionen gerendert werden. Dies reduziert den Einblick in den Datensatz, da mehr Teile verdeckt werden. In den meisten Fällen sind betragsmäßig kleine Gradienten auch von wenig Interesse, da sich an diesen Stellen die Daten kaum ändern und daher nur kleine Objektgrenzen sind oder auch nur Unstetigkeiten im Datensatz

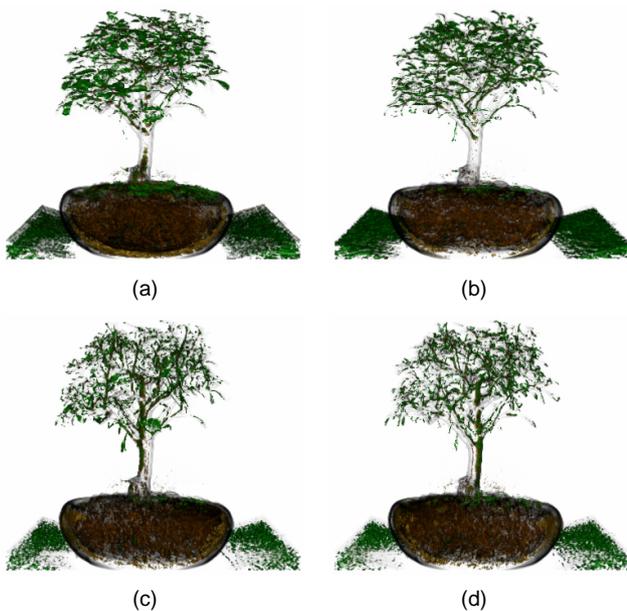


Abbildung 16: *Bonsai-Datensatz gerendert mit der Magic Lamp Methode. In Bild (a) ist die Richtung aus der die Magic Lamp "strahlt" eher oben, bei (b) unten, bei (c) strahlt sie von links und bei (d) von rechts.*

auftreten. Damit diese Stellen nicht klassifiziert werden, empfiehlt es sich einen Grenzwert einzuführen, welcher betragsmäßig kleine Gradienten herausfiltert. Dieser Grenzwert ist abhängig von dem maximalen Wert, welcher in einem Datensatz auftreten kann. Aus diesem Wert kann man einen maximalen Gradienten berechnen. Es hat sich in Versuchen herausgestellt, dass das 0.2...0.25-fache des Betrags vom maximalen Gradienten ein guter Grenzwert ist. Dabei ist die Wahl des Koeffizienten abhängig von der Stärke der Objektgrenzen in einem Datensatz.

Für die Gradienten, welche unter diesen Grenzwert fallen wird der Normalvektor auf 0 gesetzt. Dadurch ergibt das skalare Produkt $v_{ml} \times n$ auch 0. Damit werden diese Teile nicht klassifiziert und daher mit der Silhouetten-Methode gerendert.

In Abbildung 16 ist ein Ergebnis dieser Methode abgebildet. Es zeigt den Bonsai-Datensatz, welcher viermal mit unterschiedlicher Position der *Magic Lamp* gerendert wurde. Am Stamm des Bonsai ist leicht zu erkennen, dass nur die Objektgrenzen bei der Klassifikation berücksichtigt werden. Im Bild (a) ist ersichtlich wie viele Blätter in Richtung der Sonne gerichtet sind, da dort die *Magic Lamp* von oben auf den Bonsai "strahlt".

Der Übergang zwischen den beiden Bereichen ist in den meisten Fällen sehr kantig. Das kommt daher, dass die Gradientenberechnung mit der Methode der zentralen Differenzen nur einen Annäherung an den

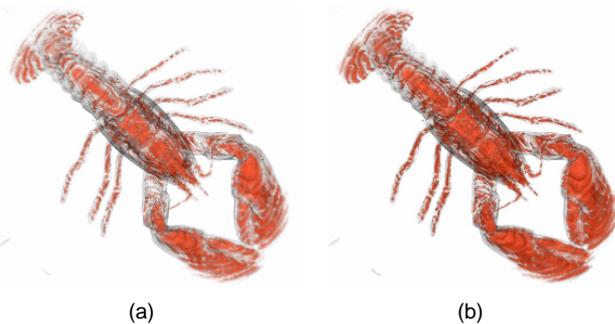


Abbildung 17: *Hummer gerendert mit unterschiedlichen Einstellungen für den Abweichungswinkel. In Bild (a) beträgt der Winkel nur 12°, in Bild (b) beträgt er 25°.*

eigentlichen Gradienten ist. Ein weiterer Faktor dafür ist, dass sich der Gradient zwischen zwei benachbarten Sample-Punkten stark ändern kann. Aus diesen Gründen sehen die Kanten an den Übergängen sehr ausgefranst aus.

Um dem entgegen zu wirken hat es sich bewährt einen Übergangsbereich zwischen den beiden Bereichen zu machen. Dieser Übergangsbereich gestaltet sich so, dass der Wechsel vom einen Bereich in den anderen nicht mehr sprunghaft sondern kontinuierlich verläuft. Dadurch verschwinden die ausgefranst Kanten am Ergebnisbild. Der Übergang wurde so gelöst, dass in diesem Übergangsbereich die Farb- und Opazitätswerte der beiden Bereiche interpoliert werden. Die Interpolation beginnt an der Klassifikationsgrenze der beiden Bereiche. Dort ist der Einfluss des Farb- und Opazitätswert der normalen Transferfunktion am größten. Je mehr der Winkel zwischen dem Normalvektor und dem Vektor der *Magic Lamp* abweicht, desto größer wird der Einfluss der Farb- und Opazitätswerte der Silhouetten-Rendering-Methode.

Der Parameter welcher die Größe des Übergangsbereichs angibt ist frei wählbar. Ebenso ist auch der Parameter wählbar, welcher die Abweichung des Normalvektors vom Vektor der *Magic Lamp* angibt frei wählbar. Mit diesen beiden Parametern kann das Aussehen des Ergebnis stark verändert werden.

Abbildung 17 zeigt ein Rendering-Ergebnis von einem Datensatz eines Hummers. Die zwei Bilder wurden mit unterschiedlichen Parametern für den Abweichungswinkel zur Klassifikation der Bereiche gerendert. In Bild (b) ist der Bereich größer und daher sind auch mehr Teile des Hummers mit der normalen Transferfunktion gerendert.

4 Implementation

In diesem Abschnitt werde ich kurz auf die Implementation des Programms eingehen, welches die Bilder der eben genannten Methoden erzeugen kann. Ich werde auch kurz die Probleme erwähnen welche während der Implementation aufgetreten sind.

Das Programm wurde in C++ erstellt. Die Programmierumgebung, welche ich dafür verwendet habe war Visual Studio 6. Zusätzlich wurde noch VTK (Visualization Toolkit von Kitware, <http://www.vtk.org>) verwendet. Die Oberfläche für das Programm ist mit Qt von Trolltech (<http://www.trolltech.com>) gestaltet. Der Grund warum ich Qt verwendet habe ist die Möglichkeit das VTK Ergebnis relativ einfach in Qt darstellbar sind. Ein weiterer Grund ist auch die Portierbarkeit von Qt.

Alle vier Methoden basieren auf dem gleichen Raycasting Prinzip. Dabei wird ein Farb- und Opazitätswert für jeden Sample-Punkt ermittelt. Die ermittelten Farben werden dann entlang jedes Strahls nach ihrer Opazität zu einer resultierenden Farbe vermischt. Das passiert von vorne nach hinten und wird wenn eine bestimmte Gesamtopazität erreicht wird abgebrochen.

Da das Raycasting bei allen Methoden gleich ist, hängt das Ergebnis nur von der Transformation der Werte der Sample-Punkte auf den Farb- und Opazitätswert ab. Die Transformation erfolgt mit einer Transferfunktion, in der für jeden Wert im gesamten Wertebereich ein Farb- und Opazitätswert festgelegt ist. Je nach Methode wird eine unterschiedliche Transferfunktion an einer Stelle benutzt oder die Opazität zusätzlich verändert (zum Beispiel durch den Gradienten).

Zur Lösung dieser Aufgabe habe ich die Raycasting-Klasse von VTK so modifiziert, dass je nach Wahl der jeweiligen Methode andere Werte berechnet werden. Das heißt, dass eine Klasse ausreicht um alle Methoden zu rendern. Der Vorteil der somit entsteht ist, dass dadurch auch jede beliebige Kombination der vier Methoden möglich ist.

Für die Komposition der Sample-Punkte entlang eines Strahls habe ich die vorgefertigte Klasse von VTK verwendet. Das Ergebnis wird in einem Qt-Widget ausgegeben. Um das Ergebnisbild in einem Qt-Widget ausgeben zu können habe ich die Klassen von Matthias König (<http://www.matthias-koenig.net/vtkqt>) verwendet, die ein VTK-Fenster in einem Qt-Widget darstellen.

Abbildung 18 zeigt einen Screenshot meines Programms. Links kann man in den verschiedenen Karteikarten die Einstellungen für die einzelnen Methoden vornehmen. Mit Hilfe der Check Boxen wird die zu verwendende Methode für das Rendern ausgewählt.

Die während der Implementation auftretenden Pro-

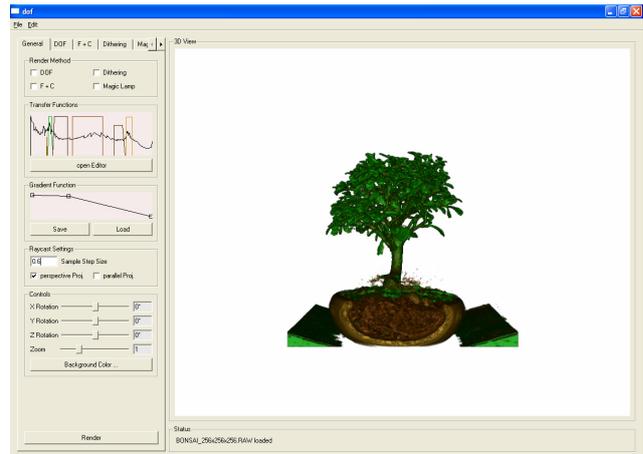


Abbildung 18: Screenshot des Programms.

bleme waren hauptsächlich programmier-technischen Ursprungs. Es gab vor allem Schwierigkeiten mit den VTK Klassen. Daher hab ich einige Klassen umgeschrieben, um sie verwenden zu können. Zu Beginn der Implementation trat auch das Problem auf das VTK Fenster in einem Qt-Widget darzustellen. Mit Hilfe der Klassen von Matthias König konnte dieses Problem allerdings gelöst werden.

Der Arbeitsaufwand beträgt geschätzte 380 Stunden für die Implementation. Wobei aber ein Großteil dieser Zeit für das Einlernen von C++, VTK und Qt gebraucht wurde.

5 Ergebnisse

In diesem Abschnitt sind einige Ergebnisbilder der vier vorgestellten Methoden abgebildet. Sie wurden alle mit dem von mir erstellten Programm gerendert. Die Bilder sollen dazu dienen die Vorteile der einzelnen Methoden hervor zu heben. Alle verwendeten Datensätze stammen aus der Computertomographie.

Depth of Field

Die zwei Abbildungen 19 und 20 zeigen zwei Ergebnisse der *Depth of Field* Methode. Der Datensatz in Abbildung 19 stammt von einem menschlichen Kopf. Die Größe des Datensatzes ist $184 \times 256 \times 170$. Die Datenwerte bewegen sich zwischen 0 und 4096. In Bild (a) wurde der Fokuspunkt am vorderen Ende (Kiefer) gesetzt, in Bild (b) im Bereich der Ohren.

Abbildung 20 zeigt einen Motorblock. Die Größe des Datensatzes ist 256×128 . Der Wert eines Datenpunkts kann zwischen 0 und 256 liegen. In dieser Abbildung ist in Bild (a) der Fokuspunkt am vorderen Ende und in Bild (b) am hinteren Ende des Objekts.

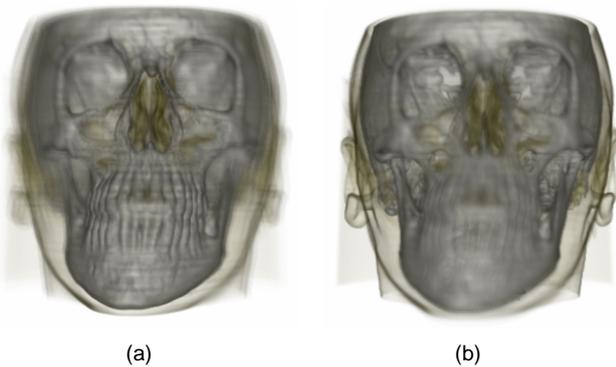


Abbildung 19: Datensatz eines Kopfs. Der Fokuspunkt in (a) und (b) ist unterschiedlich.

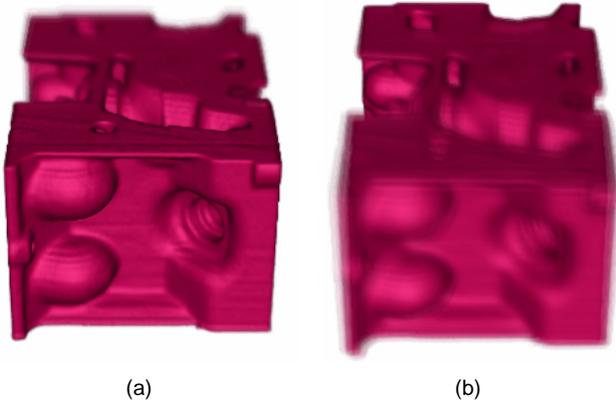


Abbildung 20: Motorblock gerendert mit Depth of Field und zwei unterschiedlichen Fokuspunkten.

Focus+Context

Für die Ergebnisbilder in Abbildung 21 wurde der gleiche Datensatz den menschlichen Kopfs verwendet wie bei der *Depth of Field* Methode. Bild (a) zeigt das Ergebnis des normalen Rendering-Verfahrens nach Marc Levoy [8]. Bild (b) hebt den Ansatz der Wirbelsäule hervor. Durch die *Focus+Context* Methode ist dieser Bereich sehr gut freigestellt. Auch das Drehen des Datensatzes verdeckt diesen Bereich nicht, da immer nur die Silhouetten der restlichen Teile dargestellt werden

In Abbildung 22 ist selbiger Effekt mit einem anderen Datensatz dargestellt. Es wurde der Datensatz eines Teetopfs verwendet. Die Größe des Datensatzes beträgt $256 \times 256 \times 178$. In diesen Teetopf wurde noch ein Hummer platziert. In Bild (a) ist der Datensatz mit der normalen Rendering-Methode dargestellt. Die Außenwände des Teetopfs verhindern den Blick auf den Hummer. In Bild (b) ist der Datensatz mit der *Focus+Context* Methode dargestellt. Der Hummer in der Mitte des Teetopfs ist hier gut zu erkennen.

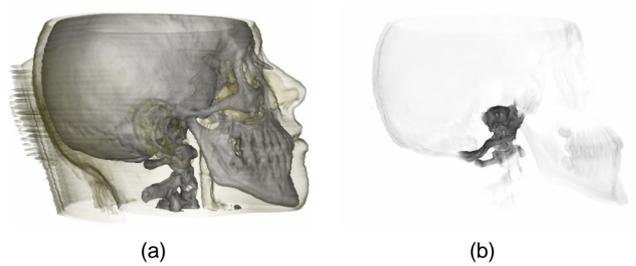


Abbildung 21: Abbildung zeigt einen Kopf. Das Bild (a) wurde im Gegensatz zu Bild (b) ohne *Focus+Context* gerendert.

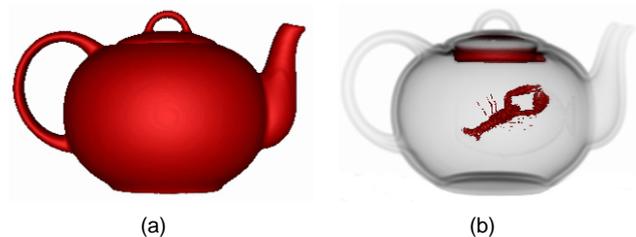


Abbildung 22: Die zwei Bilder zeigen einmal den Teetopf mit der *Focus+Context* Methode gerendert und einmal ohne.

3D Dithering

Um ein Ergebnisbild für diese Methode zu erhalten wurde der Datensatz eines Leopardengeckos verwendet. Der Datensatz hat die Ausmaße von $512 \times 512 \times 88$, wobei der Abstand in z-Richtung 2.8 mal größer ist als in den beiden anderen Hauptrichtungen. Bild (a) wurde mit der normalen Rendering-Methode erstellt. Bild (b) wurde mit der *3D Dithering* Methode erstellt. Im mittleren bis hinteren Bereich des Geckos fällt ein rötlicher Dithering-Quader auf. Dieser schneidet ein inneres Organ. Somit kann man die Position dieses Organs lokalisieren und es möglicherweise mit der *Focus+Context* Methode darstellen.

Magic Lamp

Für die Abbildung 24 wurde der Datensatz des Teetopfs verwendet. Die *Magic Lamp* "strahlt" von rechts oben auf das Objekt herab. Da nur die Teile hervorgehoben werden, deren Normalvektor in Richtung der *Magic Lamp* zeigt, ist auch ein Teil der Innenwand des Teetopfs zu sehen. Dies ist mit herkömmlichen Rendering-Methoden kaum zu erreichen.

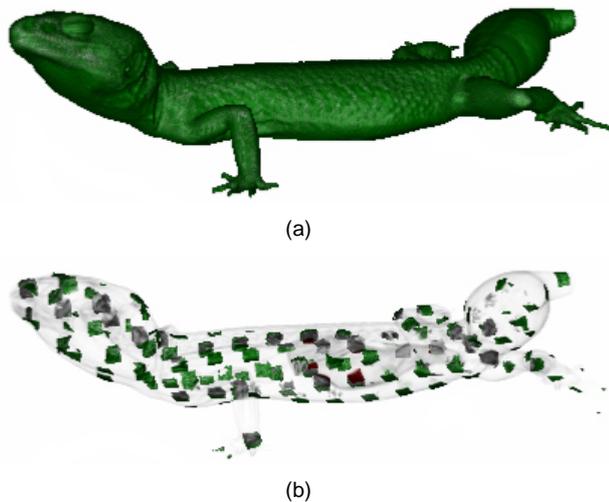


Abbildung 23: Leopardengecko, welches einmal normal und einmal mit 3D Dithering gerendert wurde.



Abbildung 24: Abbildung zeigt Teetopf, welcher einmal mit normaler und einmal mit Magic Lamp Rendering-Methode erstellt wurde.

6 Schlussfolgerungen

Während der Materialsuche für diese Arbeit hat sich herausgestellt, dass es bereits eine große Anzahl von Arbeiten über Merkmals-Visualisierung gibt. Dabei kommen die verschiedensten Ansätze vor. Doch alle haben eines zum Ziel: sie wollen einen besseren Einblick in die Daten geben. Dies ist zurzeit einer der Schwerpunkte in der Visualisierung. Besonders in der Informationsvisualisierung war es seit langem von großem Interesse Teile eines Datensatzes zu betonen, damit dieser genauer betrachtet werden kann. Mittlerweile ist dieser Ansatz auch in der Volumsvisualisierung von großem Interesse. Diesen Trend belegt auch die große Anzahl von Arbeiten, welche über dieses Thema verfasst wurden und werden.

Das größte Problem aller Ansätze ist noch eine gute Qualität bei hoher Geschwindigkeit zu erreichen, denn die meisten Methoden sind relativ aufwendig was den Rechenaufwand betrifft. Es wäre aber in vielen Anwendungen wünschenswert interaktive Frameraten zu

erreichen.

Durch die rasante Entwicklung der Computer-Hardware ist es aber durchaus möglich in einigen Jahren auch aufwendigere Methoden mit einem akzeptablen Zeitaufwand zu rendern.

Meine Arbeit ist auch im Umfeld der Merkmals-Visualisierung positioniert. Es gibt einige Einsatzgebiete, in denen die vier von mir vorgestellten Methoden eingesetzt werden können. Da relativ wenige Datensätze frei zugänglich sind war es auch schwierig Datensätze für bestimmte Einsatzgebiete zu erproben. Mit den wenigen Datensätzen konnte ich aber dennoch einige Einsatzgebiete herausfiltern.

Die Methode *Depth of Field* unterscheidet sich von den anderen drei beschriebenen Methoden in dieser Arbeit. Sie bietet nämlich keinen zusätzlichen Einblick in die inneren Strukturen von Daten. Es ist jedoch erwiesen, dass das menschliche Auge sehr gut scharfe von unscharfen Teilen unterscheiden kann und dass der Fokus schnell auf scharfe Teile gerichtet wird. Daher fällt der Blick des Betrachters schnell auf den wesentlichen Teil in der *Depth of Field* Methode. Diesen Vorteil könnte besonders für Präsentationszwecke ausgenutzt werden. Der Nachteil ist allerdings der hohe Berechnungsaufwand. Ein Datensatz wird mehrmals gerendert, um zu einem Ergebnis zu gelangen. Werden Bilder aber für Präsentationszwecke gerendert, spielt das kaum eine Rolle, da dort die Rendering-Geschwindigkeit eher bedeutungslos ist.

Focus+Context ist die wohl effektivste Methode, welche in dieser Arbeit vorgestellt wird. Sie kann überall dort eingesetzt werden, wo die Position des interessanten Bereichs bekannt ist. Dies kann zum Beispiel in der Medizin der Fall sein, wenn eine bestimmte Region des Körpers genauer untersucht werden soll. Die Teile, welche im Context-Bereich liegen, werden nur durch deren Silhouetten dargestellt und verdecken daher die Sicht auf den Focus-Bereich nicht.

Wenn die Lage des Bereichs der betrachtet werden soll nicht bekannt ist, dann kann das *3D Dithering* Abhilfe schaffen. Da der Datensatz in viele kleine Quader geteilt wird ist die Wahrscheinlichkeit groß, dass einer dieser Quader das gesuchte Objekt schneidet. Ist dies der Fall, so kann mittels *Focus+Context* dieser Bereich genauer betrachtet werden.

Einen ganz anderen Einblick in die Daten liefert die *Magic Lamp* Methode. Bei ihr bestimmt im Gegensatz zu *Focus+Context* und *3D Dithering* nicht die Position eines Punktes die Rendering Methode, sondern dessen Ausrichtung und der Gradientenbetrag. Ein mögliches Einsatzgebiet wäre etwa die Biologie. Dort könnte mit dieser Methode die Fläche von Blättern eines Baums, welche in eine bestimmte Richtung zeigen, ermittelt werden. So kann bestimmt werden aus welchem Winkel die Sonne scheinen müsste, damit am

meisten Fläche bestrahlt wird. Es sind aber noch eine Reihe anderer Einsatzgebiete für diese Methode möglich. Zum Beispiel ist sie auch gut dazu geeignet die Innenseite von Objekten darzustellen.

In den letzten drei Methoden spielt die Context-Darstellung eine wichtige Rolle. Dabei hat die in dieser Arbeit vorgestellte Methode der Silhouetten-Extraktion als sehr einfach aber ausreichend genau erwiesen. Die Darstellung der Silhouette hat in allen Anwendungen einen wesentlichen Vorteil bezüglich der Übersichtlichkeit gebracht.

Es gibt noch viele Möglichkeiten die Methoden zu verfeinern und zu beschleunigen. Das Programm zum Erstellen der Ergebnisse ist so gemacht, dass auch alle möglichen Kombinationen der vier Methoden gerendert werden können. Dadurch könnten möglicherweise neue Methoden entstehen, welche die Vorteile der einzelnen vereinen.

Meiner Meinung nach ist die Merkmals-Extraktion in der Volumsvisualisierung ein sehr interessantes und umfangreiches Thema ist. Es sind zurzeit schon eine ganze Reihe von guten Ansätzen vorhanden. Wenn die Computer-Hardware noch um einiges besser wird, werden diese Methoden mit Sicherheit zum alltäglichen Werkzeug in der Volumsvisualisierung.

Literatur

- [1] Balazs Csebfalvi, Lukas Mroz, Helwig Hauser, Andreas König, and Eduard Gröller. Fast visualization of object contours by non-photorealistic volume rendering. *Computer Graphics Journal*, 20(3), 2001.
- [2] Helwig Hauser, Lukas Mroz, Gian-Italo Bischi, and Eduard Gröller. Two-level volume rendering. *IEEE Transaction on Visualization and Computer Graphics*, 10(5), 2001.
- [3] Taosong He, Lichan Hong, Arie Kaufmann, and Hans Peter Pfister. Generation of transfer functions with stochastic search techniques. *Proceedings IEEE Visualization 1996*, pages 227–234, 1996.
- [4] Gordon Kindlmann. Semi-automatic generation of transfer functions for direct volume rendering. Technical report, Cornell University, Ithaca, NY, 1999.
- [5] Gordon Kindlmann and James W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. *IEEE Symposium on Volume Visualization*, pages 79–86, 1998.
- [6] Robert Kosara, Silvia Miksch, and Helwig Hauser. Semantic depth of field. *IEEE Symposium on Information Visualization 2001*, 2001.
- [7] Robert Kosara, Silvia Miksch, Helwig Hauser, Johann Schrammel, Verena Giller, and Manfred Tscheligi. Useful properties of semantic depth of field for better f+c visualization. 2001.
- [8] Marc Levoy. Display of surface from volume data. *IEEE Computer Graphics and Applications*, 8:29–37, 1987.
- [9] Penny Rheingans and David Ebert. Volume illustration: Nonphotorealistic rendering of volume models. *IEEE Transaction on Visualization and Computer Graphics*, 7(3), 2001.
- [10] Ivan Viola, Armin Kanitsar, and Eduard Gröller. Importance-driven volume rendering. Technical report, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2004.
- [11] Jianlong Zhou, Anreas Döring, and Kalus D. Tönnies. Distance transfer function based rendering. *IEEE TCVG Symposium on Visualization*, pages 1–9, 2004.
- [12] Jianlong Zhou, Manfred Hinz, and Kalus D. Tönnies. Focal region-guided feature-based volume rendering. *IEEE First International Symposium on 3D Data Processing Visualization and Transmission*, pages 87–90, 2002.