Karlsruhe Institute of Technology

Intelligente Industrieroboter

Implementierung eines Clusteringbasierten Verfahrens zur Visualisierung von Volumenmodellen

written by

Lukas Diewald

At the Department of Computer Science
Institute for Anthropomatics and Robotics (IAR) -
Intelligent Process Control and Robotics (IPR)

First reviewer: Prof. Dr.-Ing. habil. Björn Hein
Second reviewer: Prof. B

August 14, 2018

Todo list

Executive Summary

English abstract.

Keywords: *Keywords, of, my, Thesis*

Contents

1. Introduction

As an useful aid in all scientific work following book is recommended: [**?**]

Rewrite this seciton

-transferfunktionen was ist das? -ventrikelsystem -medizinische bildverarbeitung -ct daten

2. State of the art

visualisierung farbe etc.

Das Gebiet der Transferfunktionen ist bereits weit erforscht und es existieren viele unterschiedliche Methoden und Herangehensweisen um medizinische Daten abhängig von verschiedenen Problemstellungen passend darzustellen.

begründung für methodenwahl

Dabei kann man einerseits zwischen dem Level an Automation eines Systems unterscheiden. Hierbei gibts es vollautomatische Verfahren, bei denen keine Interaktion mit den Benutzer von Nöten ist, semiautomatische Verfahren bei denen der Benutzer noch an gewissen Stellschrauben drehen kann um das Ergebnis zu beeinflussen und manuele Verfahren, bei denen der Anwender mehr oder minder auf sich alleine gestellt ist.Transferfunktionene unterscheiden sich weiterhin in ihrer Dimensionalität. Es gibt ein- zwei- und allgemein -mehrdimensionale Transferfunktionen. Desweiteren sind die Grundlagen auf denen die Berechnungen ruhen teils völlig verschieden. Manche Verfahren basieren auf den Intensitätswerten oder deren Änderung im gegebenen Volumen, andere auf der Größe der Features die für den Nutzer von interesse sind. Wieder andere sind Bildbasiert oder wenden Machinelearning an um zum gewünschten Ziel zu gelangen.

besser formulieren

In diesem Abschnitt wird ein Überblick über die unterschiedlichen Vorgehensweisen von Transferfunktionen gegeben. Dabei werden diese im Folgenden für die Übersicht in die Kategorien Eindimensionale, Zweidimensionale, Räumlichbasierte,Machinelearning, Bildbasierte, Clusteringbasierte Verfahren geteilt.

übersicht zu oft

Dabei ist jedoch zu beachten, dass die Kategorien lediglich der Übsicht diese Abschnitts dienen. Die Kategorien hätten anders gewählt werden können. Desweiteren können manche Verfahren nicht eindeutig in die hier genannte Unterteilung eingeteilt werden, da sie mehrere Klassen auf sie zutreffen.

2.1. Eindimensionale Verfahren

Die einfachste Form der Transferfunktionrm, sind die eindimensionalen Transferfunktionen. In diesen, wird nur der Intensitätswert der Voxel in Betracht gezogen. Abgesehe von den niedrigen Berechnungszeiten, sind diese jedoch aus mehreren Gründen suboptimal. Medizinischen Daten werden gemessen und haben deshalb meist ein Rauschen, was die genaue Darstellung erschwert. Weiterhin sind die Intensitätswerte verschiedener Bereiche nah beieinander oder gar gleich und damit sind eindimensionale Transferfunktionen unpraktisch um verschiedene Materialien kenntlich zu machen. Trotzdem sind eindimensionale Transferfunktionen weit verbreitet und werden oft benutzt.

2.2. Zweidimensionale Verfahren

[1] SG-TF

Die Arbeit von Shouren Lan [2] befasst sich mit der Verbesserung von 2D Transferfunktionen die auf Skalarwerten und Gradienten(SG-TF) basieren. Genauer geht es darum das Problem vom Auftauchen von Überlappungen von Bereichen die nicht zusammen gehören zu vermeiden.
Dabei wird im Paper zwischen 3 verschiedenen Arten von Strukturen unterschieden:

- (i) Strukturen die keine andere Struktur berühren
- (ii) Strukturen die keine andere Struktur berühren, jedoch nah an einer andern liegen
- (iii) Strukturen die andere Strukturen berühren

Wenn der Benutzer eine Region ausgewählt hat, werden zunächst alle Strukturen in dem Bereich klassifiziert und kleine Fragmente entfernt. Durch verschiedene Algorithmen werden Strukturen der Klassen (ii) durch Erosion, Dilatation, Aufteilen und neu zusammenfügen von einander getrennt. Strukturen der Klasse (iii) werden durch eine weitere niedrig dimensionale Transferfunktion getrennt. Anschließend werden durch das Aufteilen entstehende Löcher mit Hilfe von Dilatation gestopft. Als letztes wird durch eine Transferfunktion den Strukturen entsprechende Farben und Intensitäten zugewiesen.

2.3. Räumlichbasierte Verfahren

In der Arbeit von Wesarg und Kirschner [3, 4] wird das "Stucture-Size-Enhanced Histogram" vorgestellt.
Dafür muss für jeden Voxel die "structure size estimators" berechnet werden. Dies geschieht, indem betrachtet wird, wie viele Schritte in die Richtung der 26 Nachbarn des Voxel gemacht werden kann. Ein Schritt kann gemacht werden, wenn der Intensitätswert des erreichten Voxels nicht mehr als ein gegebener Parameter abweicht. Die Schrittweiter startet mit einem Voxel und verdoppelt sich mit jedem gemachten Schritt bis hin zur Hälft der Größe des gegebenen Volumens. Für ein besseres Ergebnis wird nur der kleinere Wert von zwei entgegengesetzten Richtungen gespeichert. Die Akkumulation alle Werte ergibt die Größe der Struktur. Aus diesem und dem Intensitätswert wird ein zweidimensionales Histogram erstellt.

Es gibt diverse Regiongrowingverfahren um Strukturen von interesse hervozuheben.
Beipsielsweise wird im Paper von Huang [5] ein solches Regiongrowingverfahren vorgestellt. Der Benutzer kann einen Punkt von interesse im Volumen wählen, den sogenannten "seed". Es werden alle 26 Nachbarn des "seeds" besucht und anhand einer Kostenfunktion, die den entsprechenden Wert des besuchten Voxels und des Seedvoxels vergleicht, entschieden ob sie zu der Region dazugehören oder nicht. Sind sie Teil der Struktur werden auch ihre Nachbarn besucht und alle passenden Voxel zu der Region hinzugefügt. Diese Vorgang wiederholt sich so lange bis alle Voxel gefunden wurden, oder ein anderes internes Abbruchkriterium erfüllt wurde. Es stehen dem Anwender 3 verschiedene Kostenfunktionen bereit. Die erste Funktion bezieht sich auf die Intensitätswerte der Voxel, die Zweite auf die Länge der jeweiligen Gradienten und bei der Dritten werden die Gewichte der Voxel verglichen, die vorher vom Benutzer definiert werden müssen. In einem Nachbearbeitungsschritt ist es anschließend noch möglich unpassende Elemente zu entfernen, wenn beispielsweise eine ganze weitere Struktur auch visualisiert wird, da sie über eine kleine Brücke von ein bis zwei Voxeln mit der eigentlich gesuchten Struktur verbunden ist.
Da das Regiongrowing sehr zeitaufwändig ist, kann der Anwender auswählen, dass er zunächst nur eine gewisse Teil der Region sich errechnen lässt.

Correa und Ma zeigen in ihrer Arbeit [6] einen Ansatz, der auf der relativen Größe der zu visualisierenden Features basiert.
Dazu benutzen sie den sogenannten "scale-space", welcher für das Volumen berechnet

wird, um anschließend eine auf Größe basierende Transferfunktion anzuwenden. Diese mappt Farbe und Okklusion zu den entsprechenden Größen der Features des Volumens.
In einem weiteren Paper [7] beschreiben die beiden Forscher ein Verfahren, dass auf der Okklusion der Voxel basiert.
Dafür betrachten sie die Umgebung einzelner Voxel und berechnen abhängig davon, die Okklusion. Die Ergebnisse werden in einem zweidimensionalen Histogram gespeichert in Kombination mit den Intensitätswerten der Voxel. Durch die Umgebungsokklusion der Voxel ist es leicht mit einer auf dem Histogram basierenden Transferfunktion unterschiedliche Materialien mit gleichen Intensitätwerten zu unterscheiden.
Eine weitere Arbeit [8] von Correa und Ma beschäftigt sich mit Transferfunktionen abhängig von der Sichtbarkeit einzelner Voxel.
Die Sichtbarkeit jedes Voxels wird abhängig vom Sichtpunkt auf das Volumen berechnet, indem die Opazität vom Standpunkt der Kamera bis hin zum Voxel akkumuliert wird. Anschließend wird ein Histogram über die Sichtbarkeitswerte erstellt. Auf diesem kann der Benutzer eine Transferfunktion erstellen, bei der er ein direktes Feedback über die Darstellung erhält. Um das gewünschte Ergebnis zu erhalten, wäre jedoch ein sehr genaues Einstellen dieser Funktion von Nöten, was der User nur schwer umsetzten kann. Deshalb wird eine Energiefunktion erstellt, die es zu minimieren gilt. Damit wird das Problem wie bei [9] zu einem Optimierungsproblem, welches mithilfe von progressiver Suche gelöst werden kann. Der Anwender muss dafür lediglich ein Opazitätsfunktion angeben, die seine gewünschten Visualisierungsziele beschreibt. Hierbei kann er auch aus vorgefertigten Funktionen wählen.
Diese Histogramme des vorangegangen Papers wurde von Correa und Ma in einer erweiterten Arbeit [10] nochmals verbessert. Sie führten multidimensionale Sichtbarkeitshistogramme, die beispielsweise auch die Gradientenlänge in Betracht ziehen. Desweiteren stellen sie zwei Methoden vor zur Berechnung von Sichtpunkt unabhängigen Sichtbarkeitshistogrammen. Zum einen ein Omni-direktionales Sichtbarkeitshistogramm, bei dem die Sichtbarkeit von allen möglich Sichtpunkten berechnet wird. Und ein Raidales Sichtbarkeitshistogramm, bei dem radiale Strahlen verwendet werden. Dazu wird das kartesische in ein sphärisches Koordinatensystem umgerechnet.

Im Vergleich zu Huan [5] benutzt Chen in seiner Arbeit [11] nicht nur ein seedbasiertes Verfahren, sondern fügt noch ein sketchbasiertes Verfahren davor ein.
Anfangs wählt der Anwender eine Reihe an Intensitätswerten im Histogram, die für ihn interessant sind. Danach kann er direkt im Volumen eine Region von Interesse einzeichnen und markieren. Das Program schneidet im Anschluss, alle Teile des Volumens außerhalb der gewählten Region weg. Jetzt kann der Nutzer wie im vorrig vorgestllen Verfahren seinen "seed" setzen.
Dies erleichtert dem Benutzer die Anwendung, da er schneller zu seinem Punkt von Interesse gelangt ohne vorher durch diverese Querschnittsbilder iterieren zu müssen. Desweiteren ist es Zeitsparrend für den User, falls er sich nicht genau mit dem Datensatz und der zu Visualisierenden Region auskennt.

2.4. Machinelearning Verfahren

Die Arbeit von Tzeng [12] benutzt Machinelearning um interessante Strukturen darzustellen. Hierbei wird ein Neuronales Netz und eine Support Vecotr Machine benutzt.
Als Input für das Verfahren kann der Benutzer im Volumen mit zwei verschiedenen Farben Regionen anmalen und damit markieren. Mit der einen Farbe markiert der Nutzer

die Stellen von Interesse, die er hervorgehoben haben möchte, mit der anderen Farbe Stellen, die ihn explizit nicht interessieren. Das Programm nimmt im Anschluss die Intensitätswerte, Länge der Gradienten und Intensitätswerte der Nachbarn aller markierter Voxel als Input um eine sinnvole Segmentierung zu finden. Das Ergebnis wird dem Anwender in Form einer farbigen Darstellung gezeigt, bei der er abhängig von der Farbe der Regionen sieht wie ähnlich sie den angemalten Voxeln sind. Gefällt dem Benutzer das Ergebnis noch nicht, so kann er durch weiteres einfärben von Regionen das Ergebnis verbessern bis das gewünschte Resultat erreicht wird.

Soundararajan stellt ein Verfahren vor [13], bei dem der Anwender auch direkt im Volumen markieren kann, welche Gebiete für ihn von interesse sind.
Dabei wird überwachtes maschinelles Lernen angewendet, um aus dieser Eingabe eine probabilistische Übertragungsfunktion abzuleiten. Dabei ist es wichtig, dass das ausgewählte Machinelearningverfahren wie gesagt eine probabilistische Klassifikation erlaubt. Desweiteren muss es nicht nur zwischen zwei verschiedenen Klassen unterscheiden können sondern multiple Klassen unterstüzten.
In dem Paper wird das Verfahren mit fünf verschiedenen Machinelearningverfahren getestet und erklärt wie weit mit diesen das gegebene Verfahren umsetztbar ist. Es wurden "Gaussian Naive Bayes", "k Nearest Neighbor", "Support Vector Machines", "Random Forests" und "Neural Networks" getestet.

2.5. Bildbasierte Verfahren

Eine weitere Art Transferfunktionen anzuwenden, sind Verfahren, die auf ?einem Bild? basieren. Hier hat der Benutzer die Möglichkeit, mit dem Programm zu interagieren. Dies ist für einen unerfahrenen User intuitiver und er kann durch ausprobieren ein gewünschtes Ergebnis erzielen.

Fang stellt in seinem Paper [14] ein Verfahren vor, bei dem die Transferfunktioneine eine Abfolge von verschieden 3D Bildverabteitungsverfahren ist, deren Parameter vom Benutzer angepasst werden können.

Es gibt auch Methoden, bei denen ein Hilfswerkzeug zum Einsatz kommt. Zum Beispiel kann der Benutzer im Paper von Reitinger: [15] sich gezielt Bereiche des Volumens hervorheben lassen, indem er sie mit einem Stift auswählt. Hierbei wird die Intensiät des ausgewählten Punktes als auch der räumliche Abstand zum Stift in Betracht gezogen.

Wu und Qu stellen in ihrer Arbeit [9] ein intuitives Verfahren zum verändern von Features von Transferfunktionen vor.
Der Benutzer lädt zwei verschiedene direct volume rendered images(DVRIs) in das Program. Hierbei wird ihm die jeweilige Visualisierung und Transferfunktion angezeigt. Der User kann entscheiden ob er gewisse Features der beiden DVRIs zu einem verschmelzen, aus beiden mischen oder einzelne löschen möchte. Die Zusammenführung geschiet im Anschluss mithilfe einer Energiefunktion, die die Ähnlichkeit zweier Bilder beschreibt. Das Zusammenführen wird somit zu einem Optimierungsproblem undzwar dem minimieren der Energiefunktion. Dieses Problem kann mithilfe eines stochastischen Suchalgorithmus gelöst werden. Der Anwender bekommt am eine Visualisierung mit allen gewünschen Features.

2.6. Clusteringbasierte Verfahren

Sereda baut seine Arbeit [16] auf den von Serlie [17] vorgestellten LH-Histogramen auf und zeigt wie man mit ihnen Objekte klassifizieren kann.
Die Berechnung eines LH-Histograms ist eine Methode zur Erkennung von Kanten, unter der Verwendung von Low- und High-Werten. Dabei werden die Voxel in zwei verschiedenen Kategorien eingeteilt. Es gibt Voxel, die innerhalb eines Materials liegen und welche, die an der Grenze zweier Materialien liegen. Ist ein Voxel innerhalb, so sind seine LH-Werte gleich. Grenzvoxel hingegen haben unterschiedliche Low- und High-Werte, wobei disese die Intensitätswerte der beiden Materialien, zwischen denen die Grenze verläuft, beschreiben.
Bei der Berechnung des Histograms wird als erstes getestet, ob der betrachtete Voxel an einer Grenze liegt. Ein Punkt liegt innerhalb eines Materials, wenn die Länge des Gradienten kleiner als ein gewisses epsilon (bei MRT-Daten) oder gleich null(bei CT-Daten ist). In diesem Fall wären die Low- und High-Werte der Intensitätswert des Voxels. Ist dies jedoch nicht der Fall, wird in Richtung(für die High-Werte) und entgegengesetzter Richtung(für die Low-Werte) des Gradientens schrittweise integriert. Dies stoppt sobald ein Material gefunden wurde. Dies wird für jeden Punkt im Volumen berechnet und danach aus allen LH-Werten ein Histogram erstellt.
Zur Visualisierung benutzt Sereda eine dreidimensionale Transferfunktion. Diese nimmt die beiden LH-Werte als auch die Gradientenlänge, aus dem Grund, dass vor allem Voxel nah an der Grenze interessant sind und diese dadurch hervorgehoben werden, als Parameter entgegen.
Weiterhin verwenden die Forscher Regiongrowing um Strukturen zu erkennen. Dabei basiert die Kostenfunktion auf dem LH-Histogram. Dies ist deutlich besser als Kostenfunktionen, die auf dem Intensitätswert und der Gradientenlänge basieren, da Kanten trotz Überlappungen besser erkannt werden können.

besser schreiben

In einer späteren Arbeit [18] stellt Sereda ein hierarschies Clusteringverfahren vor. Hierbei werden in einer Menge von Clustern immer die zwei gemerged, die sich bei dem ausgewählten Vergleichsverfahren am ähnlichsten sind. Es wird eine Kombination aus zwei solcher Vergleichsverfahren vorgestellt. Zum einen wird die räumliche Nähe in betracht gezogen, bei der gezählt wird, wie viele direkte Nachbarn zwei Cluster besitzen. Zum anderen wird die Nähe im LH-Raum untersucht. Als Startcluster dienen hierbei die Kästchen des LH-Histograms. Die einzelnen Cluster bekommen für die Visualisierung am Ende eine zufälligen Farbwert zugewiesen.

Das Paper von Binh P. Nguyen [19] stellt ein clusteringbasiertes Verfahren um für den Benutzer interessante Gebiete hervozuheben.

altes paper auch einbinden

Zunächst wird in einem Vorverarbeitungsschritt die Gradienten des Volumens berechnet. Hierzu wurde Hong's Methode [20] verwendet. Im Anschluss daran wird anhand der Gradienten die LH-Werte mithilfe von Heuns Methode, ein modifizierte Euler Methode, ermittelt.Hierbei wird desweiteren eine Gewichtung abhängig von der zurückgelegten Strecke bei der Interpolation für den Low- bzw. High-Wert errechnet. Aus den LH-Werten und deren Gewichten wird anschließend ein LH-Histogramm erzeugt.
Dem Benutzter steht dann ein zwei stufiges und ein drei stufiges Clusteringverfahren zur Auswahl, wobei die ersten beiden Clusteringschritte die Selben sind. Im ersten Clusteringschritt wird im LH-Raum mithilfe von "Meanshiftclustering" geclustert. Es wird für jeden LH-Wert alle Werte gefunden, die in einem Kreis mit einem Radius von 7% - 9%

des maximalen LH-Wertes um den ursprünglichen Punkt liegen. Anschließend wird der neue durchschnittliche Mittelpunkt von allen Punkten im Cluster ermittelt. Der Vorgang wiederholt sich der Mittelpunkt zwischen zwei Iterationen nur minimal ändert. Der komplette Vorgang wird für jeden Punkt im LH-Histogram wiederholt und in Folge werden Cluster, deren Mittelpunkt nah beieinander liegen zu einem Cluster gemerged.
Der zweite Clusteringschritt wird auf die Cluster des ersten Schittes angewendet. Hierbei wird auch "Meanschiftclustering" verwendet. Diesmal wird jedoch räumlich, also abhängig von der Position im Volumen, geclustert. Desweiteren werden die Parameter für den Suchradius und Distanz zweier Mittelpunkte damit sie gemerged werden angepasst.
Als Ergebnis der ersten zwei Schritte erhält man Cluster mit Voxeln die ähnliche LH-Werte haben, als auch räumlich nah beieinander liegen.
Im optionalen dritten und letzten Clusteringschritt wird hieraschisch geclustert. Hierbei wird für jeden Cluster die paarweise Nähe zu jedem anderen Cluster errechnet. Anschließend werden hieraschich immer die zwei Cluster, die sich am nähsten sind, zu einem gemerged, solange bis nur noch ein Cluster existiert. Hierbei speichert das Programm jeweils welche Cluster wann miteinander gemerged wurden. Der Benutzer kann im Anschluss entscheiden wie viel Cluster er haben möchte. Abhängig davon, wird das hierarschiche Clustern umgekehrt und die Cluster werden wieder getrennt, bis die gewünschte Anzahl an Clustern erreicht ist.

3. Methods

Die Implementierung dieser Arbeit orientiert sich an dem zweistufigen Clusteringverfahren aus der Arbeit von Nguyen [19].

Die hier vorgestellte Implementierung teilt sich in zwei verschieden Programme auf. Zum einen ein Programm, dass Volumedaten lädt, verarbteit und das Ergebnis als binäre Datei abspeichert und zum anderen ein Unityprogramm, das für die Visualisierung von den binären Volumendaten zuständig ist. Das Modul zum Laden und Speichern von Volumendaten, sowie das Unityprogramm zum rendern von Volumendaten sind in Vorarbeiten des IPRs entstanden und werden in dieser Arbeit teilweise vorgestellt und verwendet.

christians paper

Anfangs liegen die CT-Daten von den Volumen als Schnittbilder im DICOM Format vor. Diese werden mithilfe der MITK Workbench zu einer einzelnen Datei im .nrrd Format umgewandelt, welche vom Programm eingelesen werden können. Da CT-Daten Werte im negativen Bereich haben, die für das Verfahren hinderlich sind, werden die Daten um den kleinsten Wert verschoben, sodass das Minimum des Volumens bei null liegt.

Zunächst müssen die LH-Werte des Volumen berechnet werden, dafür müssen die Gradienten aller Voxel bestimmt werden. Wie im Paper beschrieben wurde in dieser Arbeit auch Hong's Methode [20] dafür gewählt. Diese ist ein Approximationsbasiertes Verfahren zur Berechnung von Gradienten eines Volumens vorgestellt.
In Hong's Verfahren wird zur Berechnung die lokale 4x4x4 Nachbarschaft hinzugezogen. Hierbei ist zu beachten, dass der Gradient für einen Punkt nicht direkt berechnet werden kann. Der Gradient kann immer nur zwischen zwei Punkten berechnet werden, da er die Veränderung der Werte angibt. Deshalb liegt er im Falle eines dreidimensionalen Volumens im Zentrum eins Würfels, der von 8 benachbarten Voxeln aufgepsannt wird, wie man in Figure 1 sehen kann. Die Knoten sind hierbei die Voxel des Volumens in denen ein Intensitätswert gespeichert ist.

richtig bild zitieren u. evtl kleiner

Die Funktionen für die Intensitätswerte wird im Paper mit:
$f(x, y, z) = Ax^2 + By^2 + Cz^2 + 2Fyz + 2Gzx + 2Hxy + 2Ix + 2Jy + 2Kz + D$
approximiert. Da der Gradient die Ableitung der Intensitätsfunktion ist, erhält man den dreidimensionalen Gradientenvektor n, indem die Funktion ableitetet wird:
$n = (Ax + Gz + Hy + I, By + Fz + Hx + J, Cz + Fy + Gx + K)$.
Um den Gradienten zu Berechnen müssen die Parameter A,B,C,E,F,G,H,I,J,K berechnet werden. Dies geschieht mithilfe der Methode der kleinsten Quadrate.

methode der kleinsten quadrate genauer beschreiben

Dies wird für jeden Voxel im Volumen berechnet. Als Ergebnis des Verfahrens kommt ein Volumen, in dem alle Gradienten gespeichert sind, heraus. Jedoch sind die Punkte um eine halbe Voxellänge verschoben. Desweiteren ist die Dimension dieses Volumens in jeder Achse um eins kleiner als das vorher gegebene Intensitätsvolumen. Da für die Berechnung der LH-Werte jedoch der Gradient und der Intensitätswert an einer Stelle im Volumen bekannt sein muss, wurden die Intensitätwerte auf das Volumen der Gradienten umgerechenet. Dies geschieht durch eine einfach Interpolation indem von allen 8 Nachbarn eines Punktes der Intensitätwert geachtelt wurde und aufaddiert. Hierbei gehen Information verloren....

abbildung für interpolation und beschreiben was

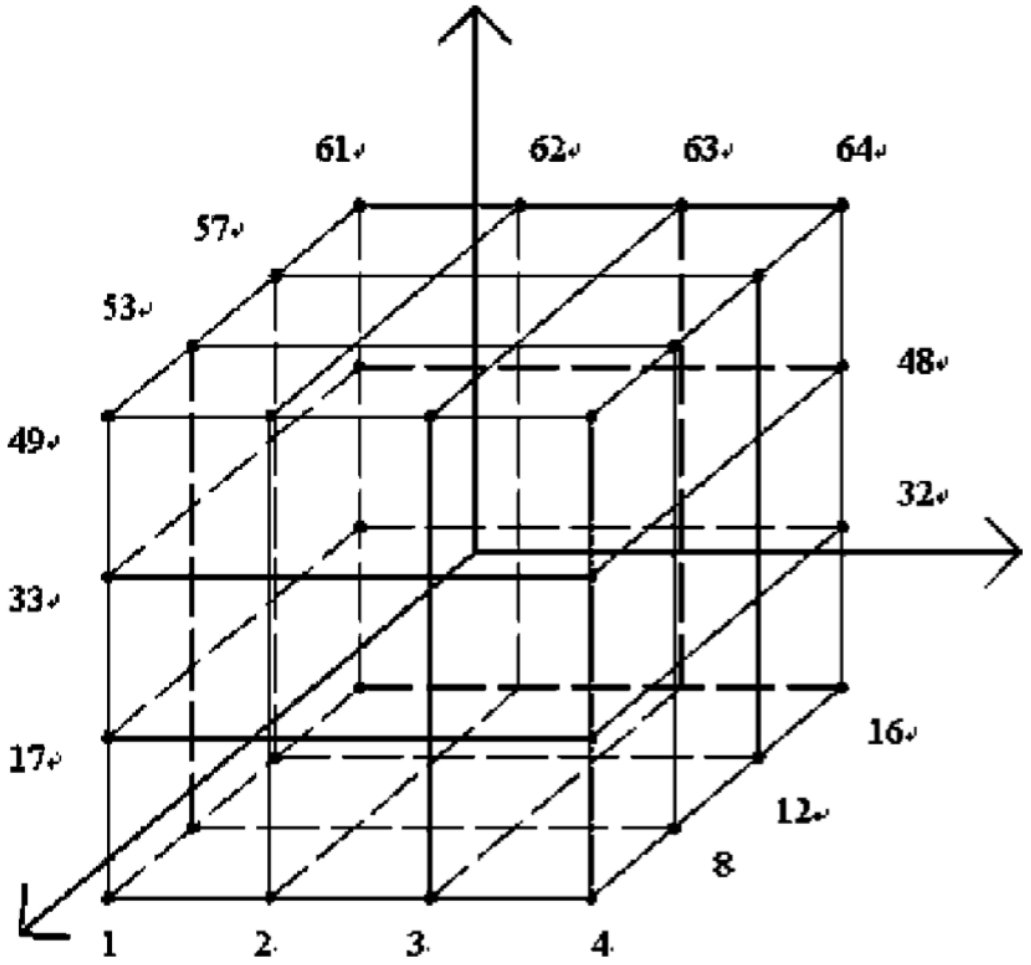

Figure 1: Darstellung der lokalen 4x4x4 Nachbarschaft

Nachdem die Gradienten berechnet sind und die Intensitätswerte passend umgerechnet wurde, folgt die Berechnung der Low- und High-Werte. Dazu wird in Richtung der Gradienten integriert. Hierfür wurde wie von Nguyen auch Heun's Methode, eine modifizierte Euler Methode verwendet. Die hierzu benutzte Formel lautet:
$u_{i+1} = u_i + \frac{1}{2}d(\nabla f(u_i) + \nabla f(u_i + d\nabla f(u_i)))$
Hierbei sind u_i und u_{i+1} die Positionen des aktuellen, beziehungsweise des nächsten Voxels. $\nabla f(x)$ beschreibt den normalisierten Gradienten für die High-Werte und den normalisierten inversen Gradienten für die Low-Werte an Stelle x . d steht für die Schrittweite (ein Voxel). Da das Verfahren in dieser Arbeit auf CT-Daten angewendet wird, ist das Abbruchkriterum der Integration, dass ein Gradient mit Länge null gefunden wird. Ist dies der Fall wir der Intensitätswert dieses Voxels als Ergebnis für den Low- beziehungsweise High-Wert des Startvoxel festgelegt.

Anschließend wird ein LH-Histogram über alle berechneten Werte erstellt. Die x-Achse sind hierbei die Low- und die y-Achse die High-Werte. Deren Reichweite geht von null bis zu den jeweiligen Maxima der Werte.

Auf diesem Histogram findet der erste Clusteringschritt statt. Es wird ein Meanshiftclustering verwendet. Dazu wird zuerst eine Bandweite und ein Threshold gewählt, welche die Sensitivität des Clusterings bestimmen. Danach wird das Clustering für jeden Punkt

im LH-Histogramm wie folgt durchgeführt.
Es werden alle Punkte die innerhalb des Radius der Bandbreite um den Startpunkt liegen gespeichert. Diese Punkte bilden nun den gefundenen Cluster. Von diesem Cluster wird der neue Mittelpunkt, der jeweilige Mittelwert der beiden Koordinaten, berechnet. Um diesen Punkt wird erneut mit selben Radius alle Punkte die bisher nicht zu dem Cluster gehören gesucht und hinzugefügt. Dies geschieht solange, bis der Abstand des neu kalkulierte Mittelpunkt zum Alten weniger als der Threshold mal die Bandweite ist. Nachdem das Clustering für jeden Punkt im Histogramm beendet ist, werden jene Cluster deren Mittelpunkte eine Distanz kleiner als die Hälfte der Bandweite zueinander haben miteinander zu einem einzigen Cluster verschmolzen.

Da die Intensitätswerte im Gehirn sehr nah beieinander liegen, wäre es nicht sinnvoll wie von Nguyen beschrieben über das komplette Histogramm mit einer Bandweite von 7% - 9% des maximalen LH-Wertes, das Maximum aller Low- und High-Werte, zu clustern. Das Gehirn würde dabei als ein paar wenige, sehr große Cluster erkannt werden. Diese Herangehensweise mag praktibale zum Darstellen von Knochen, des kompletter Gehirns oder anderen Bereiche sein, entspricht jedoch nicht dem Ziel dieser Arbeit, das Ventrikelsystem kenntlich zu machen und zu visualisieren. Aus diesem Grund wurde die Bandbreite des Clustering auf 0,1% des maximalen LH-Wertes gesetzt. Da so eine Bandbreite sehr Rechenaufwendig ist und bekannt ist, dass das Ventrikelsystem einen Intensitätswert um die ... hat, wurde desweiteren nur im Intensitätswertbereich von 1025 bis 1075 geclustert, um die Rechenzeiten gering zu halten.Weiterhin wurde nicht jeder Punkt des Histograms besucht, sondern eine Schrittweite von fünf Kasten festegelegt. Dies geschah aus der Beobachtung heraus, dass für zwei oder drei direkt nebeneinanderliegende Kasten im LH-Histogramm meist der gleiche Cluster als Ergebnis berechnet wurde. Diese wurden im letzten Schritt dann ohnehin zu einem einzigen Cluster verschmolzen, was die Berechnung jedes einzelnen Kastens unnötig machte. Diese Maßnahmen dienen ausschließlich dem Zweck Strukturen im Gehirn bessers unterscheiden zu können, da sie so in verschiedne LH-Cluster eingeteilt werden und damit unterscheidbar sind. Wenn das Verfahren für andere Ziele genutzt werden soll, können die Parameter auf die von Nguyen vorgeschlagenen Werte gesetzt werden.

intensitätswert ventrikel, vllt orginial werte nehmen und in implementierung verschiebun vorstellen

Als nächstes werden die LH-Cluster erneut mit Meanshiftclustering geclustert. Diesmal jedoch anhand ihrer räumlichen Informationen im Volumen. Im Paper von Ngujen wird hierzu kein Wert für die Bandbreite vorgeschlagen. In dieser Arbeit hat sich eine Bandbreit von 10 und eine minimale Distanz vom alten zum neuen Mittelpunkt von 0,01 als zielführend erwiesen.

allgemein meanschift-clustering beschreiben

Nachdem alle Cluster erzeugt wurden, werden ihnen zufällige IDs von eins bis zur Anzahl an Clustern zugeteilt. Anschließend wird ein Volumen erstellt, mit den Dimensionen des ursprünglichen Intensitätsvolumens, dass nur mit nullen als Werte gefüllt ist. Warum es diese Dimension haben muss, wird später erklärt. Danach wird durch alle Cluster iteriert, und jeder Voxel in das neu erstellte Volumen an der jeweiligen Position eingetragen. Dabei ist der eingetragene Wert immer die ID des aktuellen Clusters. Nachdem alle Cluster abgearbeitet wurden, besteht das Volumen aus ausschließlich nullen und den IDs der Cluster an den passenden Stellen.

Dieses Volumen wird als eine binäre Datei abgespeichert und in Unity geladen. Hier kann der Anwender sich entweder die Form von einzelnen IDs anschauen oder eine Reichweite

von IDs. Die gewählten IDs werden rot markiert und sich deutlich vom restlichen Volumen zu unterscheiden. Mithile der Darstellung muss der Nutzer anhand der Form die Cluster erkennen, die zum Ventrikelsystem gehören, und deren IDs notieren.

Hat er dies getan kann er im ... Programm die binäre Datei mit den IDs, die ursprüngliche Intensitätsvolumendatei laden und eine die Liste der passenden IDs als Parameter übergeben. Die ausgewählten IDs werden zu einem Cluster gemerged und in das Intensitätsvolumen übernommen. Dort erhalten sie, da der maximale Wert von CT-Daten bei ...(4400) liegt, den Wert 5000. Dieser Wert erhöht das Maximum für die Darstellung der Daten nur gering, ist dafür aber im Volumen sonst niergends enthalten. Das Ergebnis dieses Merges wird wieder als binäre Datei gespeichert.

Als letzten Schritt kann nun der Benutzer die gemerged Datei erneut in Unity laden. Hier kann er sich dann das Volumen normale abhängig von den Grauwerten dargestellt anzeigen lassen, mit dem Ventrikelsystem in rot, oder einer anderen ausgewählten Farbe darin.

letzten schritt genauer

4. Concept

Formulierung

Das hier folgende Kapitel beschäftigt sich mit dem Softwaredesign der Implementierung dieser Arbeit. Da diese jedoch, wie schon erwähnt, auf einer Vorarbeit des IPR basiert, war es nicht möglich, die Programmiersprache frei zu wählen. Aus diesem Grund ist es möglich, dass diese Implementierung in einer anderen Sprache besser lösbar ist. In dieser Arbeit wurde das Verfahren, da Unity als Programm für die Visualisierung gewählt wurde, in *c sharp* umgesetzt.

Die interne Speicherung des Volumens wurde mit der generischen Klasse *Volume* umgesetzt.

attribute funktionen etc. von volumenklasse in UML

Die Interaktion des Benutzers mit dem Programm findet über eine Kommandozeile statt. Hierbei hat der Anwender die Befehle Load, Dump, Resample, Info, Write, LHHistogram, ClusterVolume und MergeCluster zur Auswahl. Jedes Modul hat dabei seine eigene Syntax die mithilfe eines Help Befehls angezeigt werden kann. Die Funktionen der Anweisungen entsprechen dem Namen der Befehle. So lädt Load bespielsweise eine .nrrd oder binäre Datei, Dump und Write speichern das geladene Volumen als binäre oder .nrrd Datei ab, Info gibt Informationen über das aktuell geladene Volumen zurück und Resample lässt den Anwender die Größe des Volumens verändern. Ruft man den Dump Befehl mit einem "u" auf, so wird das Volumen vor dem Speichern zum Typ "unsigned int" gecastet, indem wie im vorigen Kapitel beschrieben die Werte verschoben werden.

Im Folgenden werden hauptsächlich die Module LHHistogram, ClusterVolume und MergeCluster erläutert, da diese im Laufe dieser Arbeit entstanden sind. LHHistogram berechnet hierbei lediglich die Gradienten und LH-Werte und erstellt eine .csv Datei, in der das LH-Histogram gespeichert wird. ClusterVolume kalkuliert das Gleiche, führt hinterher jedoch noch die beiden Clusteringschritte aus. Das Modul MergeCluster hingegen dient der Verschmelzung der gewünschten IDs, und dem ursprünglichen Volumen. In Figure 2 sieht man einen Überblick über den Aufbau der Implementierung.

Damit das Verfahren funktioniert, muss das geladene Volumen als "unsigned int" vorliegen. Der Ablauf der Berechnung startet in der statischen *Gradient* Klasse. Diese ist eine Implementierung des Verfahren von Hong [20]. Sie nimmt für ihre Berechnung als Parameter ein Volumen von Integern entgegen und gibt ein Volumen mit dem generischen Datentyp *FVector3* zurück. Diese ist eine Hilfsklasse zur Darstellung von dreidimensionalen Vektoren.

Als nächstes wird die Funktion *LHValueVolume* der statischen Klasse *LHValues* aufgerufen. Diese berechnet für das gegebene *FVector3* Volumen und den dazugehörigen Intensitätswerten die LH-Werte für jeden Voxel. Diese gibt die Methode in Form eines Volumens vom Typen "Tuple<float,float>" zurück.

Hat der Benutzer das Modul LHHistogram aufgerufen, entsteht im Anschluss das LH-Histogramm in der Klasse *LHHistogramCSV* und wird von ihr als .csv Datei in einem vom Anwender angegebenen Pfad abgespeichert. Diese kann der Benutzer anschließend in dem Pythonscript *PlotHelper* laden und sich Visualisieren lassen. Hierbei ist zu beachten, dass das Histogramm gebildet wird, indem die Häufigkeit des Vorkommens eines LH-Wertpaares im Volumen im jeweils dazu passenden Kästchen gespeichert wird. Dies

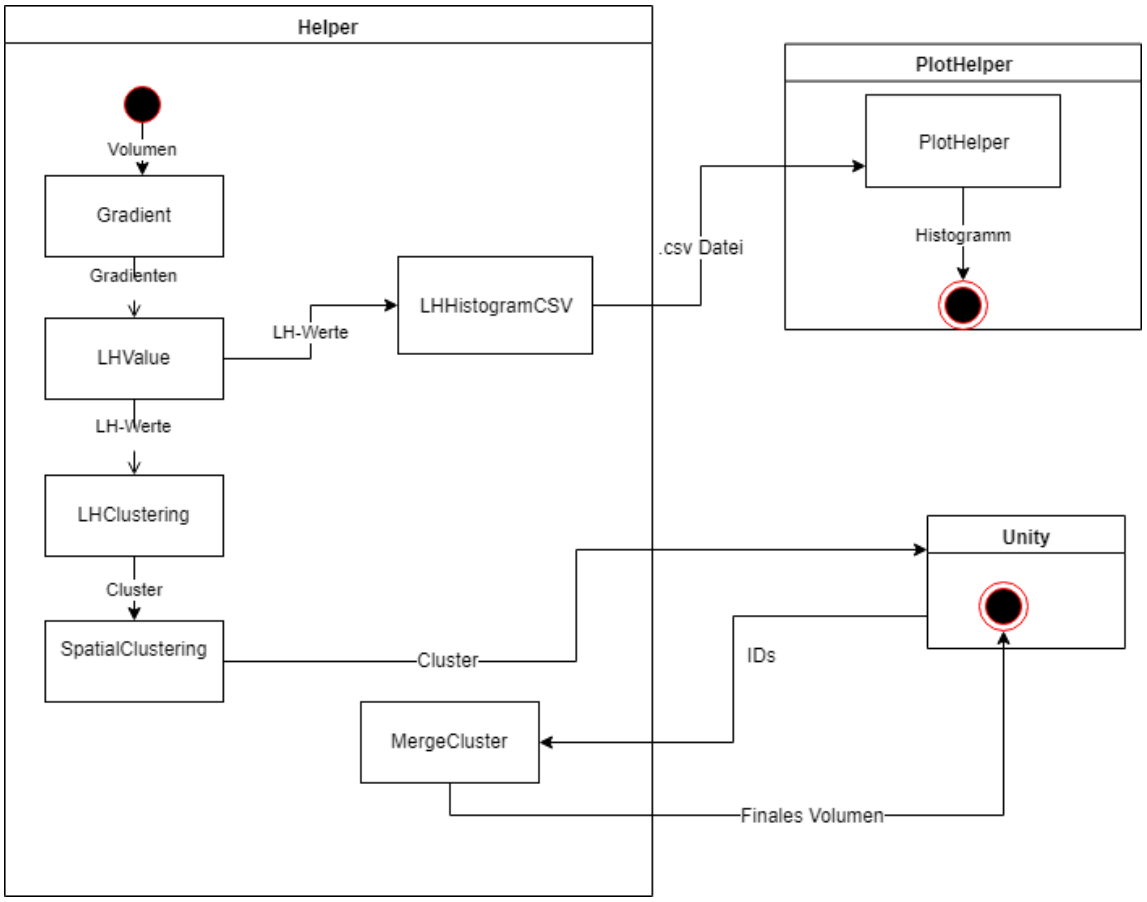

Figure 2: Überblick über das Programm

ist simpler als das im Paper von Nguyen [19] benutzte Erstellen des Histogramms abhängig von einer für jeden Voxel berechneten Gewichtung. Da die Arbeit an der Implementierung zeitlich beschränkt war, wurde diese Gewichtung, die einzig und allein einer genaueren Darstellung des für das Verfahren irrelevante LH-Histogramm dient, vernachlässigt. Die Gewichtung ist für das Clustering belanglos, da dort ein Histogramm wie oben beschrieben, abhängig von der Häufigkeit der LH-Werte verwendet wird.

Wurde jedoch das ClusterVolume Modul aufgerufen, wird mit den beiden Clusteringschritten fortgefahren. Die Berechnung die in der *LHClustering* Klasse geschieht nimmt das Volumen mit den LH-Werten entgegen und rechnet es aus Performancegründen in ein ein Histogramm um. Dieser Schritt könnte gespart werden, wenn die Methode *LHValueVolume* direkt ein Histogramm als Rückgabewert liefern würde. Als Ergebnis der Clusteringfunktion *ComputeLHClusters* wird eine Liste der Cluster zurückgegeben. Ein Cluster besteht aus einer Liste von *IVector3*.Diese Hilfsklasse beschreibt wie *FVector3* Vektoren, jedoch können die Werte nur ganze Zahlen annehmen, folglich eigenen sie sich gut, wie hier verwendet, zum Beschreiben von Positionen im Volumen.Die Cluster werden nur als Liste der räumliche Information der Punkte gespeichert, da für den nächsten Clusteringschritt lediglich diese Information benötigt wird.

5. ImplementationImplementation

...

Please add some figures

6. Results

Die folgenden Zeitmessungen wurde alle auf einem Computer mit einem 3.70GHz Intel Core(TM) i7-8700K CPU mit 32GB RAM ausgeführt. Um die Berechnungszeit des Systems messen zu können, wurde die Berechnung des gesamten Clusteringverfahrens und die des LH-Histogramms mit drei verschieden großen Volumen durchgeführt. Diese stammen alle von den gleichen CT-Daten ab und wurden lediglich mit dem Resamplemodul verkleinert. Es war geplant, noch ein viertes Volumen zum Vergleich hinzuzuziehen, jedoch war es aus einem unbekannten Fehler leider nicht möglich die Verfahren mit einem gevierteltes Volumen durchzuführen. Desweiteren funktioniert die gesamte Berechnung nicht für das ganze Volumen, da es vermutlich zu viele Daten für die aktuelle Implementierung sind.

Die Berechnungszeit hängt stark von der Größe des Eingabevolumens ab. Die ist in Table 1 sehr gut zu erkennen.

Volumengröße	LH-Histogramm $[s]$	Komplettes Verfahren $[s]$
Halbes Volumen (256x101x256)	30	50
Dreiviertel Volumen (384x151x384)	90	380
Ganzes Volumen (512x201x512)	225	-

Table 1: Überblick über die Berechnungszeiten der verschiedenen Volumengrößen

Hierbei ist wichtig zu beachten, dass die Zeit zur Berechnung der LH-Histogramme, im gleichen Umfang Zeit bei der Kalkulation des gesamten Verfahrens benötigt. Zieht man also diese Berechnungszeit von der Gesamtzeit ab, erhält man die Zeit, die die beiden Clusteringschritte benötigen. Eine interessante Beobachtung dabei ist, dass die Berechnung der LH-Histogramme abhängig von der Anzahl der Pixel gesehen grob gleich schnell abläuft. Das halbe Volumen hat eine Gesamtpixelzahl von ungefähr 6,6 Millionen, das dreiviertel Volumen von zirka 22,2 Millionen und das ganze Volumen von grob 52,6 Millionen Pixeln. Wird die Anzahl an Pixeln die pro Sekunde bearbeitet werden für diese drei Volumen berechnet so ist zu beobachten, dass kein großer Unterschied zu bemerken ist. Das Halbe bearbeitet etwa 220 Tausend, das Dreiviertel ungefähr 247 Tausend und das Ganze 234 Tausend Pixel pro Sekunde. Der kleine Unterschied lässt sich einerseits durch feste Berechnungen die unabhängig von der Pixelzahl Zeit benötigen und andererseits über nicht ganz genauen Messungen. Folglich kann man sagen, dass die Berechnung der LH-Werte in etwa linear mit der Anzahl an Eingabepixeln wächst.

gradienten zeit und lh zeit getrennt erwähnen

Auf der anderen Seite kann man jedoch auch sehen, dass die beiden Clusteringschritte mit zunehmender Eingabegröße deutlich langsamer werden. Das Clustering des halben Volumens dauerte 20 Sekunden und hat damit eine Verarbeitungsrate von zirka 330 Tausend Pixeln pro Sekunde. Hingegen dauert es beim dreiviertel Volumen 290 Sekunden und erreicht damit gerade einmal einen Rate von 76 Tausend Pixeln pro Sekunde. Es braucht also 14,5 Mal so lange an Zeit für die 3,3 fache Anzahl an Pixeln.

In dieser Arbeit wurden hauptsächlich Volumen mit einer Auflösung von 256x101x256 Pixeln verwendet... -> ergebnisse zeigen

7. Discussion

...

8. Conclusion

...

References

[1] J. Kniss, G. Kindlmann, and C. Hansen, "Multidimensional transfer functions for interactive volume rendering," *IEEE Transactions on visualization and computer graphics*, vol. 8, no. 3, pp. 270–285, 2002.

[2] S. Lan, L. Wang, Y. Song, Y.-p. Wang, L. Yao, K. Sun, B. Xia, and Z. Xu, "Improving separability of structures with similar attributes in 2d transfer function design," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 5, pp. 1546–1560, 2017.

[3] S. Wesarg and M. Kirschner, "Structure size enhanced histogram," in *Bildverarbeitung für die Medizin 2009*. Springer, 2009, pp. 16–20.

[4] S. Wesarg, M. Kirschner, and M. F. Khan, "2d histogram based volume visualization: combining intensity and size of anatomical structures," *International journal of computer assisted radiology and surgery*, vol. 5, no. 6, pp. 655–666, 2010.

[5] R. Huang and K.-L. Ma, "Rgvis: Region growing based techniques for volume visualization," in *null*. IEEE, 2003, p. 355.

[6] C. Correa and K.-L. Ma, "Size-based transfer functions: A new volume exploration technique," *IEEE transactions on visualization and computer graphics*, vol. 14, no. 6, pp. 1380–1387, 2008.

[7] ——, "The occlusion spectrum for volume classification and visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1465–1472, 2009.

[8] C. D. Correa and K.-L. Ma, "Visibility-driven transfer functions," in *Visualization Symposium, 2009. PacificVis' 09. IEEE Pacific*. IEEE, 2009, pp. 177–184.

[9] Y. Wu and H. Qu, "Interactive transfer function design based on editing direct volume rendered images," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 5, pp. 1027–1040, 2007.

[10] C. D. Correa and K.-L. Ma, "Visibility histograms and visibility-driven transfer functions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 2, pp. 192–204, 2011.

[11] H.-L. J. Chen, F. F. Samavati, M. C. Sousa, and J. R. Mitchell, "Sketch-based volumetric seeded region growing." in *SBM*, 2006, pp. 123–129.

[12] F.-Y. Tzeng, E. B. Lum, and K.-L. Ma, "An intelligent system approach to higher-dimensional classification of volume data," *IEEE Transactions on visualization and computer graphics*, vol. 11, no. 3, pp. 273–284, 2005.

[13] K. P. Soundararajan and T. Schultz, "Learning probabilistic transfer functions: A comparative study of classifiers," in *Computer Graphics Forum*, vol. 34, no. 3. Wiley Online Library, 2015, pp. 111–120.

[14] S. Fang, T. Biddlecome, and M. Tuceryan, "Image-based transfer function design for data exploration in volume visualization," in *Visualization'98. Proceedings*. IEEE, 1998, pp. 319–326.

[15] B. Reitinger, C. Zach, A. Bornik, and R. Beichel, "User-centric transfer function specification in augmented reality," 2004.

[16] P. Sereda, A. V. Bartroli, I. W. Serlie, and F. A. Gerritsen, "Visualization of boundaries in volumetric data sets using lh histograms," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 2, pp. 208–218, 2006.

[17] I. Serlie, R. Truyen, J. Florie, F. Post, L. van Vliet, and F. Vos, "Computed cleansing for virtual colonoscopy using a three-material transition model," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2003, pp. 175–183.

[18] P. Sereda, A. Vilanova, and F. A. Gerritsen, "Automating transfer function design for volume rendering using hierarchical clustering of material boundaries." in *EuroVis*, 2006, pp. 243–250.

[19] B. P. Nguyen, W.-L. Tay, C.-K. Chui, and S.-H. Ong, "A clustering-based system to automate transfer function design for medical image visualization," *The Visual Computer*, vol. 28, no. 2, pp. 181–191, 2012.

[20] D.-h. Hong, G.-m. Ning, T. Zhao, M. Zhang, and X. Zheng, "Method of normal estimation based on approximation for visualization," *Journal of Electronic Imaging*, vol. 12, no. 3, pp. 470–478, 2003.

Appendix

A. First Appendix Section

ein Bild

Figure A.1: A figure

...