



FAKULTÄT FÜR
INFORMATIK

Institut für Simulation und Grafik

Abschlussarbeit

zur Erlangung des akademischen Grades

Bachelor of Science (B.Sc.)

Konzepte zur Tiefendarstellung von augmentierten Patientendaten in stereoskopischer Laparoskopie

Vorgelegt von	Kai Bornemann
Matrikelnummer	201587
am	26. August 2016
Erstgutachter	Jun.-Prof. Dr. Christian Hansen
Betreuer	Julian Hettig (M. Sc.)

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Stellen sind als solche kenntlich gemacht. Alle Bilder und Skizzen wurden selbst erstellt und nicht aus anderen Quellen entnommen.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Magdeburg, den 26. August 2016

(Kai Bornemann)

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	5
2.1	Medizinische Grundlagen	5
2.1.1	Die Leber	5
2.1.2	Laparoskopie	7
2.2	Augmented Reality	9
2.3	Tiefenhinweise	12
2.3.1	Okulomotorische Tiefenhinweise	12
2.3.2	Monokulare Tiefenhinweise	13
2.3.3	Binokulare Tiefenhinweise	15
2.4	Stand der Technik	16
3	Konzeption	19
3.1	Anforderungsanalyse	19
3.1.1	Technische Anforderungen	19
3.1.2	Wahrnehmungsspezifische Anforderungen	20
3.1.3	Anwendungsspezifische Anforderungen	21
3.2	Virtual Window	23
3.3	Farbdistanzkodierung	24
3.4	Illustrative Visualisierung	26
4	Implementierung	29
4.1	Entwicklungsumgebung	29
4.1.1	Coin3D	30
4.1.2	OpenGL Shading Language	31
4.2	Umsetzung der Konzeptentwürfe	32
4.2.1	Virtual Window	33

4.2.2	Farbdistanzkodierung	35
4.2.3	Illustrative Visualisierung	37
4.2.4	Kombination der Konzepte	39
5	Ergebnisse	41
5.1	Virtual Window	41
5.2	Konturen	45
5.3	Farbdistanz	46
5.4	Toon-Shading	48
5.5	Kombination der Konzepte	49
6	Zusammenfassung	53
A	Anhang: Laparoskopische Werkzeuge	57
	Abbildungsverzeichnis	63
	Literaturverzeichnis	65

1 Einleitung

Laparoskopie ist eine vielversprechende Alternative zur konventionellen Chirurgie im Bauchbereich, da sie eine Reduktion des Traumas für den Patienten bedeutet und eine schnellere Genesung ermöglicht [1]. Diese Vorteile gehen jedoch auch mit Problemen einher. Dem Chirurgen werden wichtige sensorische Fähigkeiten, wie das taktile Feedback, die Tiefenwahrnehmung und ein großes Sichtfeld genommen [2]. Ein Ansatz, um einige dieser Nachteile zu reduzieren, ist Augmented Reality. Durch eine Überlagerung von Patientendaten auf das Laparoskopiebild sollen wichtige Hinweise zur Navigation und Lagebestimmung vermittelt werden, sodass der Eingriff schneller, präziser und weniger schwierig für den Chirurgen gestaltet werden kann. Eine zentrale Rolle spielt hierbei die korrekte Darstellung der Tiefe der augmentierten Daten. Abbildung 1.1 zeigt die Problematik einer falschen Tiefendarstellung, die es zu lösen gilt.

Im Rahmen dieser Bachelorarbeit sollen verschiedene Konzepte zur Darstellung von augmentierten Patientendaten für stereoskopische Laparoskopie entwickelt werden, welche einem Chirurgen bei laparoskopischen Lebereingriffen unterstützen. Der Fokus liegt hierbei auf der Visualisierung von Tiefenhinweisen und der räumlichen Orientierung in den augmentierten Daten.

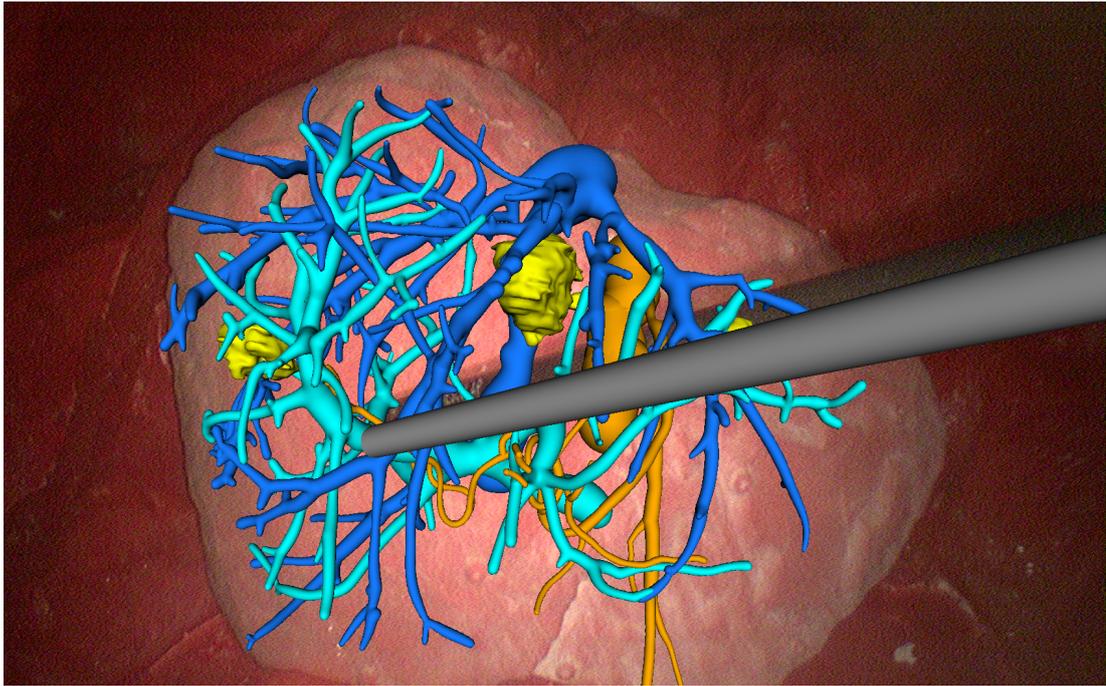


Abbildung 1.1: Der aktuelle Stand der Visualisierung der augmentierten Patientendaten auf ein laparoskopisches Bild. Die einfache Überlagerung der virtuellen Lebergefäße (blau, türkis und orange) und Tumore (gelb) lässt keine Rückschlüsse der Tiefe zu und wirken deshalb fehlplatziert vor der Leber. Es werden Konzepte gesucht, um diese Problematik zu lösen.

Die vorliegende Arbeit untergliedert sich folgendermaßen: **Kapitel 2** befasst sich mit den Grundlagen, die zum Verständnis der Arbeit relevant sind. In den medizinischen Grundlagen werden die Themen Leber und Laparoskopie erläutert. Anschließend wird Augmented Reality beschrieben. Die wahrnehmungspsychologischen Grundlagen der Tiefenwahrnehmung werden dem Leser in dem darauffolgenden Abschnitt nähergebracht. Das Kapitel schließt mit einer Übersicht an verwandten Arbeiten ab.

Kapitel 3 befasst sich mit dem Entwurf von Konzepten zur Darstellung von augmentierten Patientendaten, die auf Grundlage einer Anforderungsanalyse entwickelt wurden. Die Anforderungen umfassen dabei technische, wahrnehmungsspezifische und anwendungsspezifische Gesichtspunkte. Im Zuge der technischen Anforderungen wird zudem das verwendete laparoskopische Navigationssystem beschrieben.

Kapitel 4 erläutert die Implementierung der vorgestellten Konzepte. Hierbei wird die verwendete Entwicklungsumgebung beschrieben und ein Fokus auf Coin3D und die OpenGL Shading Language gelegt.

In **Kapitel 5** werden die Ergebnisse der Konzepte dargestellt. Es werden ebenfalls verschiedene Parameterkonfigurationen der einzelnen Konzepte erläutert und diskutiert. Anschließend werden einige Konzepte miteinander kombiniert und auf ihre Anwendbarkeit geprüft.

Kapitel 6 schließt die Arbeit mit einer Zusammenfassung und Erkenntnissen, die in dieser Arbeit gewonnen wurden, ab. Im anschließenden Ausblick werden Probleme, die es noch zu lösen gilt, diskutiert.

2 Grundlagen

In diesem Kapitel werden alle relevanten Grundlagen vermittelt, die zum Verständnis der Bachelorarbeit notwendig sind. Hierbei werden medizinische, wahrnehmungspsychologische und informationstechnische Aspekte behandelt. Abschließend wird ein allgemeiner Überblick über verwandte Arbeiten und den aktuellen Stand der Technik gegeben.

2.1 Medizinische Grundlagen

Dieser Abschnitt befasst sich mit den medizinischen Grundlagen. Zuerst wird die Funktion und der Aufbau der Leber erläutert. Anschließend werden häufige Lebertumore und ihre Bildungsarten beschrieben. Danach wird dem Leser die Laparoskopie nähergebracht. Hier werden Ablauf, Werkzeuge und verschiedene Arten Tumore zu entfernen, erläutert.

2.1.1 Die Leber

Die Leber ist die größte Drüse im menschlichen Körper [3]. Sie ist ein braunrotes innerhalb der Bauchhöhle gelegenes Organ, welches bei einem erwachsenen Menschen zwischen 1.400 g und 1.600 g wiegt. Ihre Oberfläche ist glatt, spiegelnd und feucht. Sie liegt im Schutz des Brustkorbes [4], wobei sich zwei Drittel ihrer Masse in der rechten und ein Drittel in der linken Körperhälfte befinden. Dementsprechend werden diese beiden Teile als linker und rechter Leberlappen bezeichnet. Die Leber ist für verschiedene Stoffwechselforgänge zuständig. Sie produziert beispielsweise Gallenflüssigkeit, welche für die Verdauung von Fetten unentbehrlich ist. Weiterhin entgiftet die Leber schädliche körpereigene und

körperfremde Substanzen, die über die Galle und die Nieren ausgeschieden werden. Beim Abbau von Aminosäuren entsteht toxisches Ammoniak, welches über die sogenannte Harnstoffsynthese von der Leber entgiftet wird. Zudem speichert die Leber Energie in Form von Glykogen, welches bei niedrigem Blutzucker in Glukose umgewandelt werden kann, um den Körper mit Energie zu versorgen [3]. Während der Embryonal- und Fetalphase in der menschlichen Entwicklung ist die Leber sogar in der Lage, Blut zu bilden. Nach der Geburt verliert die Leber diese Fähigkeit wieder. Die Leber besitzt im Gegensatz zu anderen inneren Organen doppelte Gefäßanschlüsse. Die Zufuhr von nährstoff- und sauerstoffreichem Blut geschieht über die Pfortader (auch Portalvene) und über die Leberarterie. Für den Abtransport von Stoffen sind die Gallengänge und die Lebervenen verantwortlich [4].

Die Leber lässt sich nach Couinaud in acht Segmente unterteilen (siehe Abbildung 2.1). Diese Einteilung richtet sich nach den Verzweigungsmustern der Pfortader, der Leberarterie und der Gallengänge. Durch diese Einteilung der Leber in autonome Segmente kann relativ sicher eine Resektion (siehe Abschnitt 2.1.2), also das Entfernen von (krankhaftem) Gewebe, an den einzelnen Segmenten vorgenommen werden, ohne den Blutfluss zu anderen Segmenten abzuschneiden [4, 5].

Lebertumore

Als Tumore bezeichnet man im Allgemeinen eine Neubildung von abnormalem Körpergewebe, in Folge einer fehlenden Regulation von Zellwachstum. Hierbei unterscheidet man gutartige (benigome) Tumore und bösartige (maligne) Tumore. Gutartige Tumore verdrängen umliegendes Gewebe, breiten sich jedoch nicht über die Grenzen von Nachbargewebe aus. Bösartige Tumore (auch Malignome genannt) wachsen hingegen invasiv in umliegendes Gewebe ein und zerstören es. Zudem bilden Malignome häufig Tochtergeschwülste, die Metastasen genannt werden. Im Volksmund werden Malignome als Krebs bezeichnet [6]. Die häufigsten primären malignen Lebertumore sind das sogenannte hepatozelluläre Karzinom und das cholangiozelluläre Karzinom. In 80 bis 90 Prozent aller Fälle geht dem hepatozellulärem Karzinom eine Leberzirrhose voraus [7]. Eine Leberzirrhose entsteht, wenn gesundes Lebergewebe durch Bindegewebe ersetzt wird, wodurch die Leber eine knotenhafte Struktur erhält. Dies ist häufig die Folge einer dauerhaften Lebererkrankung wie der Fettleber. Eine Fettleber entsteht, wenn die Leber

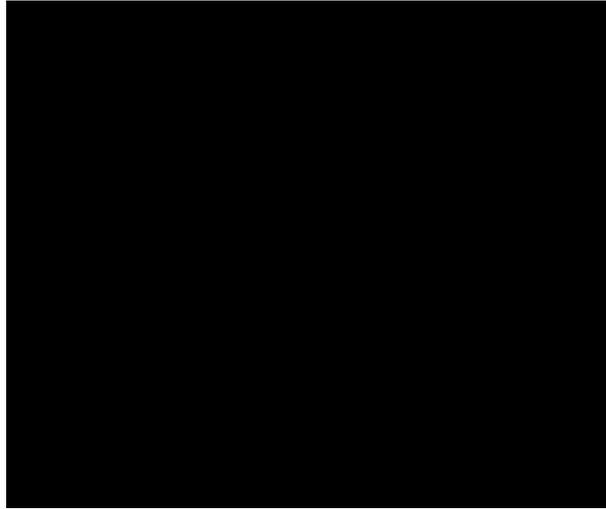


Abbildung 2.1: Die Einteilung der Leber in acht Segmente nach Claude Couinaud. Die Segmente sind im Uhrzeigersinn auf jeder Seite durchnummeriert. Lebervenen sind blau dargestellt und die Pfortader und ihre Äste violett [4].

damit überfordert ist, gesundheitsschädliche Stoffe zu entgiften. Die Leber nutzt dann fast den kompletten Sauerstoff, der ihr zur Verfügung steht, zur Neutralisierung dieser Stoffe. Als Folge kann die Leber das Fett nicht mehr verbrennen und lagert es ein [8].

Je nach Charakteristik der Lebererkrankung wird ein chirurgischer Eingriff nötig. Dieser erfolgt häufig durch Entfernung von krankem Gewebe durch Resektion oder durch Ablation (siehe Abschnitt 2.1.2). Beide Ansätze lassen sich minimal-invasiv mit einem laparoskopischen Eingriff durchführen und werden im nächsten Kapitel näher erläutert.

2.1.2 Laparoskopie

Die Laparoskopie ist eine Alternative zur konventionellen offenen Chirurgie. Bei der offenen Chirurgie wird die Bauchdecke des Patienten geöffnet, um eine freie Sicht auf den Operationsort zu erhalten [9]. Bei einer Laparoskopie bleibt die Bauchdecke geschlossen. Stattdessen werden über kleine Einschnitte in den Bauch chirurgische Werkzeuge eingeführt, um die Operation durchzuführen. Eine Kamera zeigt dabei das Innere der Bauchhöhle. Durch dieses minimal-invasive Vorgehen wird das Trauma des Patienten reduziert und eine schnellere Genesung

ermöglicht. Jedoch wird hierdurch das Sichtfeld und der Tastsinn des Chirurgen eingeschränkt [1, 10]. Abbildung 2.2 zeigt einen schematischen Vergleich zwischen einem laparoskopischen und einem offenen chirurgischen Eingriff. Im Folgenden wird das grundsätzliche Vorgehen bei einem laparoskopischen Eingriff beschrieben und welche Werkzeuge hierfür benötigt werden.

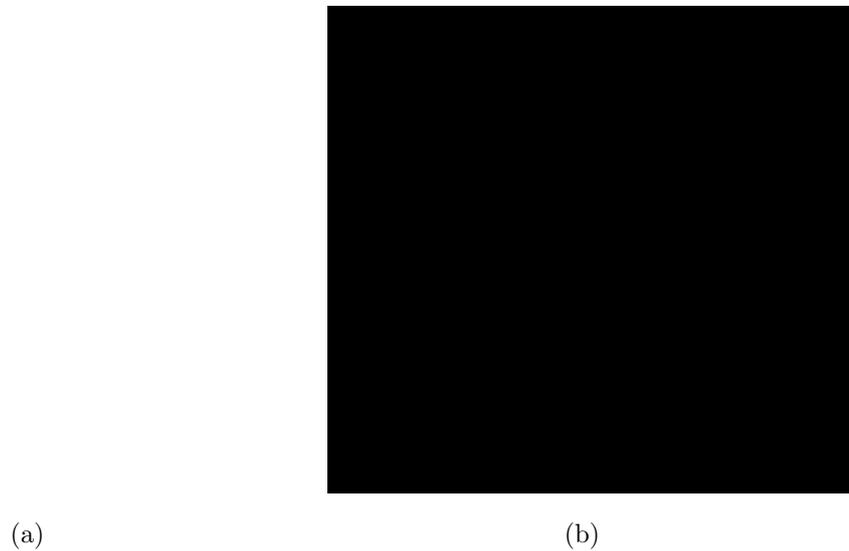


Abbildung 2.2: Abbildung (a) zeigt einen laparoskopischen Eingriff mit drei Trokaren und einer Optik. Im Vergleich dazu ist in Abbildung (b) ein offener chirurgischer Eingriff zu sehen (Bildquelle: [11]).

Vorgehen

Nachdem der Patient vorbereitet wurde, verschafft sich ein Chirurg mithilfe der Veress-Nadel (siehe Abbildung A.1) einen Zugang zu dem Abdominalbereich des Patienten. Anschließend wird ein Pneumoperitoneum geschaffen. Es ist dafür zuständig, die Bauchdecke von den Organen zu heben, um die Einführung von Werkzeugen zu erleichtern und ein größeres Operationsfeld zu schaffen. Meist wird hierfür Kohlenstoffdioxid insuffliert. Es ist unbrennbar, wird rasch resorbiert und ist relativ preiswert [10, 12]. Nach der Schaffung des Pneumoperitoneums platziert der Chirurg Arbeitstrokare (siehe Abbildung A.2). Die Hülsen dieser Trokare werden als Zugangspunkte für weitere Werkzeuge genutzt. Durch die Einführung einer Optik, auch Laparoskop genannt, in den Bauchraum wird die Sicht auf das

Operationsfeld ermöglicht. Im Anhang A werden die oben genannten Werkzeuge bei einer Laparoskopie genauer beschrieben.

Resektion

Die Resektion beschreibt das Entfernen eines (bösartigen) Gewebeteiles durch Wegschneiden. Hierbei spielt vor allem ein ausreichender Sicherheitsabstand zum Tumorgewebe eine wichtige Rolle. Wenn zu wenig vom Tumor wegschnitten wird, bleibt Tumorgewebe zurück. Gleichzeitig ist es das Ziel so wenig wie möglich vom gesunden Gewebe zu entfernen [13]. Die Leberresektion eignet sich zur Entfernung von Lebertumoren und Metastasen, vor allem in den Segmenten II. bis VI. (siehe Abschnitt 2.1.1). Tumore, die unter der Leberoberfläche liegen, werden in der Regel nur mit dem Einsatz intraoperativer, bildgebender Verfahren durchgeführt [12]. Die Leber kann sich innerhalb eines Jahres selbst nach einer 80 prozentigen Geweberesektion zu ihrer vollen Größe und Stoffwechselfunktion erholen [14].

Ablation

Die Ablation in der minimal-invasiven Chirurgie beschreibt Techniken zur Nekrotisierung von bösartigem Gewebe, vor allem Metastasen, durch Zuführung von Hitze oder Kälte. Hierfür wird ein spezielles Werkzeug, der Ablator, an den Zielort gebracht und je nach Größe und Form der Metastasen mehrmals benutzt. Die Zerstörung des Gewebes breitet sich dann je nach Ablator zylinder- oder kugelförmig vom Einsatzort aus. Um bei der Ablation tieferliegendes Gewebe zu zerstören, wird ebenfalls auf computergestützte Systeme zurückgegriffen. Häufig ist die betroffene Stelle nach Entfernen des Werkzeugs bereits blut trocken. Die Ablation stellt zudem eine mögliche Alternative bei Nichtdurchführbarkeit der Resektion dar [12].

2.2 Augmented Reality

Während Augmented Reality (AR) bereits als „a form of virtual reality where the participant’s head-mounted display is transparent, allowing a clear view of the

real world “definiert wurde [15], ist eine offenere und technologieunabhängigere Definition von AR sinnvoll. Die bekannteste Definition von AR wurde 1994 von Milgram [15] durch das *Reality Virtuality Continuum* beschrieben (siehe Abbildung 2.3). Er definiert AR als einen Teilbereich eines Kontinuums auf dessen Extrempunkten eine komplett virtuelle Welt und eine ausschließlich reale Welt liegen. Jegliche Verschmelzung beider Extrema wird als Mixed Reality definiert. AR und Augmented Virtuality (AV) sind verschiedene Ausprägungen von Mixed Reality. AR ist hierbei näher dem Extrem der realen Welt zuzuordnen, wohin gegen AV auf der virtuellen Seite zu finden ist. Demnach beschreibt AR im Allgemeinen die Erweiterung der Realität mit künstlichen, virtuellen Inhalten [16]. Diese Definition schließt jedoch auch digital gerenderte Filmszenen ein. Deshalb charakterisiert Azuma [17] AR durch folgende drei Punkte:

1. AR ist eine Verschmelzung von Realität und Virtualität.
2. AR erlaubt echtzeitfähige Interaktion.
3. AR hat eine dreidimensionale Registrierung.

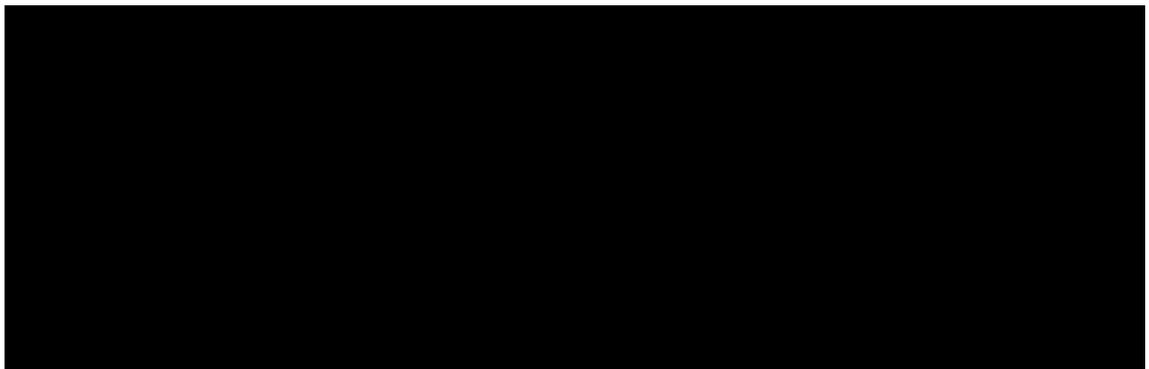


Abbildung 2.3: Die Verschmelzung von Realität mit Virtualität ergibt eine Mixed Reality. AR ist eine Ausprägung von Mixed Reality, welche eher der Realität zuzuordnen ist (Quelle: [15].)

Um eine reale Szene mit virtuellen Inhalten zu augmentieren, wird das prinzipielle Vorgehen von AR Visualisierungen nach Dörner et al. [16] in folgende fünf Schritte unterteilt (durch Preim und Botha [5] und Furht [18] ergänzt):

1 Videoaufnahme: Der erste Schritt erfolgt in der Regel durch eine Aufnahme der Umgebung, die augmentiert werden soll. Je nach Einsatzbereich werden

verschiedene Kameras verwendet. Die Aufnahme liegt dann in den meisten Fällen als Videostream vor, sodass in Echtzeit auf neue Bilder reagiert werden muss.

2 Tracking: Im nächsten Schritt wird die Position eines virtuellen Objektes relativ zur Kamera im Raum bestimmt (getrackt). Dieser Schritt liefert eine Transformation des Koordinatensystems des Betrachters in das der virtuellen Umgebung. Mögliche Ansätze sind hierbei: GPS basiertes Tracking, optisches Tracking von (reflektierenden) Markern oder Mustern, sowie sensorbasiertes Tracking.

Im Folgenden wird optisches Tracking mit Markern näher erläutert, welches in der medizinischen AR am weitesten verbreitet ist. Beim optischen Tracking mit Markern werden die zu augmentierenden, realen Objekte mit Markerkugeln versehen. Die Kugeln werden verwendet, um die sechs Freiheitsgrade eines Objektes im Raum zu bestimmen. Die Marker können hierbei aktiv oder passiv sein. Aktive Marker strahlen (nicht sichtbares) Licht ab, welches durch das System erfasst werden kann. Aktive Marker benötigen eine Kabelverbindung, weshalb diese Art von Tracking in einem Operationsraum selten genutzt wird. Passive Marker reflektieren hingegen (nicht sichtbares) Licht, welches vom System in den Raum gestrahlt und von ihm wieder erfasst wird. Optisches Tracking ist hierbei anfällig für Störfaktoren wie schlechte Lichtverhältnisse und Verdeckungen des Markers.

3 Registrierung: Bei der Registrierung wird die Pose eines virtuellen Objektes im realen Raum verankert. Diese verankerte Pose bleibt selbst dann erhalten, wenn der Betrachter seine Lage im Raum verändert. Basis für diesen Schritt ist die Ergebnistransformation des Trackings. Eine Art der Registrierung ist die punktbasierende Registrierung. Hierbei werden Punkte im virtuellen Raum mit Punkten im realen Raum in Bezug gebracht. Anschließend werden die virtuellen Punkte durch eine Fehlerminimierung an den realen Punkten ausgerichtet.

4 Darstellung: Nachdem ein virtuelles Objekt einen Platz in der Aufnahme der Realität gefunden hat, kann es abhängig von seiner Position, seiner Transformation und der Kameraperspektive wiedergegeben werden. In diesem Schritt findet die eigentliche Augmentierung statt, indem das virtuelle Objekt (sofern erwünscht) einen nahtlosen Übergang mit den Objekten der

Szene eingeht. Die Auflösung, die Schärfe und die Beleuchtung des virtuellen Objektes sind wichtige Parameter für eine gute Überlagerung.

5 Ausgabe: Im letzten Schritt kann das augmentierte Bildmaterial über ein Display ausgegeben werden. Hierbei stehen je nach Einsatzbereich verschiedene Möglichkeiten zur Verfügung, beispielsweise über einen Computerbildschirm, das Display eines Smartphones oder ein Head-Mounted Display.

2.3 Tiefenhinweise

Tiefenhinweise (englisch: depth cues) bezeichnen Informationsquellen, welche im Netzhautbild oder außerhalb von diesem zu finden sind, die einen Rückschluss auf die Tiefe bzw. die Entfernung eines Objekts zum Betrachter erlauben [19]. Tiefenhinweise lassen sich in folgende Hauptarten unterteilen: okulomotorische, monokulare und binokulare Tiefenhinweise, welche nachfolgend näher erläutert werden [20].

2.3.1 Okulomotorische Tiefenhinweise

Okulomotorische Tiefenhinweise entstehen durch die Eigenwahrnehmung der Augenmuskulaturspannung und die Stellung der Augen selbst. Die beiden Hauptmechanismen dieser Tiefenwahrnehmung sind Akkommodation und die Konvergenz. Okulomotorische Tiefenhinweise funktionieren in einer virtuellen AR Umgebung nicht.

Akkommodation

Damit Objekte scharf gesehen werden können, müssen ihre Lichtstrahlen fokussiert auf die Retina einfallen. Der entscheidendste Faktor bei dieser Fokussierung ist der Krümmungsgrad der Linse. Je näher sich ein Objekt befindet, desto mehr spannen sich die Ziliarmuskeln an, um die Linse zu krümmen und die Lichtstrahlen gebündelt auf die Netzhaut zu projizieren (siehe Abbildung 2.4). Dieser Vorgang der Scharfstellung wird als Akkommodation bezeichnet [19, 20].



Abbildung 2.4: Akkommodation. Um nahe Objekte zu fokussieren, krümmt sich die Linse, damit die einfallenden Lichtstrahlen eines Punktes gebündelt auf der Netzhaut abgebildet werden können (Bildquelle: [21]).

Konvergenz

Konvergenz bezeichnet die zueinander gerichtete Stellung der Augen, die durch Anspannung der Augenmuskulatur herbeigeführt wird. Je näher die Augen durch Fokussierung auf ein Objekt konvergieren, desto näher wird das Objekt wahrgenommen.

2.3.2 Monokulare Tiefenhinweise

Als monokular werden Tiefenhinweise bezeichnet, die bereits unter Verwendung von nur einem Auge erfasst werden können. Sie sind deshalb auch schon auf zweidimensionalen Bildern erfassbar.

Verdeckung

Wenn ein Objekt A ein Objekt B teilweise oder komplett verdeckt, so muss sich Objekt B hinter dem verdeckenden Objekt A befinden, es ist also weiter entfernt vom Betrachter (siehe Abbildung 2.5). Verdeckung wird auch als Okklusion bezeichnet [19]. Die Verdeckung ist der effektivste und eindeutigste einzelne, relative Tiefenhinweis in der menschlichen Wahrnehmung [22].

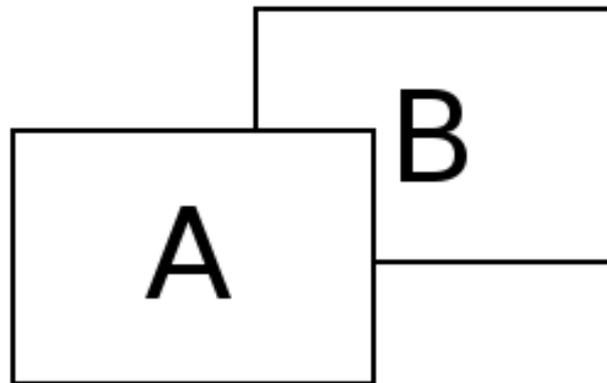


Abbildung 2.5: Obwohl die Abbildung keine komplexen Tiefenhinweise beinhaltet, weiß der Betrachter durch die Verdeckung, dass sich Rechteck A vor Rechteck B befinden muss. Der Abstand zwischen beiden Objekten ist jedoch nicht abschätzbar.

Schatten

Schatten können dazu beitragen, die räumliche Positionierung und damit die Tiefe der Objekte, die Schatten werfen, festzustellen (siehe Abbildung 2.6).

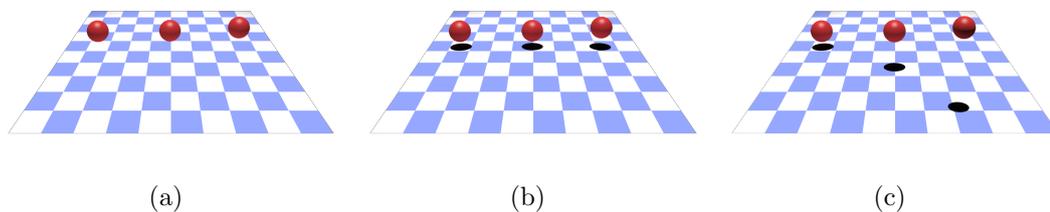


Abbildung 2.6: Welche Kugel befindet sich wo? Erst durch den Schatten kann eine korrekte Einschätzung der Tiefe erfolgen.

Relative Höhe

Gleichartige Objekte, die im Gesichtsfeld weiter oben liegen, aber noch unter dem Horizont liegen, wirken weiter entfernt. Für Objekte über dem Horizont gilt der umgekehrte Zusammenhang: Objekte im unteren Gesichtsfeld wirken weiter entfernt.

Relative Größe

Gleichartige Objekte, die größer auf dem Netzhautbild erscheinen, sind weniger weit vom Betrachter entfernt.

Texturgradient

Elemente einer Fläche mit ähnlichem Abstand (wie z.B. Blumen auf einer Wiese), wirken dichter gepackt, je weiter sie sich in den Raum erstrecken (siehe Abbildung 2.7).

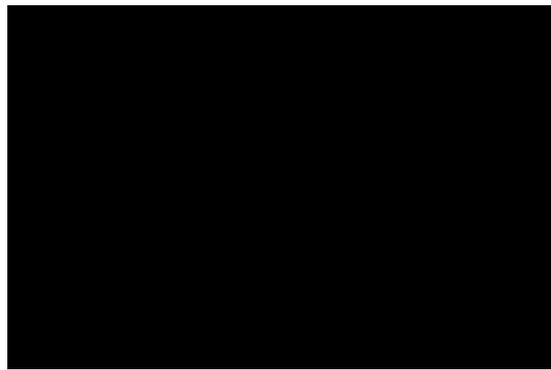


Abbildung 2.7: Texturgradient: die zunehmend dichter gepackten Blumen verstärken die Tiefenwahrnehmung (Bildquelle: [23]).

Bewegungsparallaxe

Dieser Effekt kann bei der Fortbewegung (z.B. mit einem Auto oder zu Fuß) beobachtet werden. Hierbei scheinen Objekte, die sich nah am Betrachter befinden, sich schneller zu bewegen, als Objekte, die in einer größeren Entfernung liegen. Dieser Effekt wird Bewegungsparallaxe genannt [20].

2.3.3 Binokulare Tiefenhinweise

Binokulare Tiefenhinweise können nur unter der Verwendung beider Augen erfasst werden. Sie beruhen auf Informationen, die aus den Unterschieden des linken und rechten Netzhautbildes errechnet werden können. Diese Unterschiede kommen dadurch zustande, dass die beiden Augen aus unterschiedlichen Perspektiven

dieselbe Szene erfassen [19]. Beidäugiges Tiefensehen wird auch stereoskopisches Sehen oder Stereopsis genannt [20].

Querdisparität

Grundlage des stereoskopischen Tiefensehens ist die Querdisparität, welche den Unterschied zwischen dem linken und rechten Retinabild beschreibt. Um diesen Effekt zu erklären, betrachtet man korrespondierende Punkte auf der Netzhaut. Würde man eine Retina auf die andere legen können, dann würde ein fokussierter Punkt im Raum auf denselben Punkt auf der Netzhaut abgebildet werden. Punkte, die bei gleichem Fokuspunkt ebenfalls auf korrespondierende Netzhautorte abgebildet werden, liegen auf einem gedachten Kreis (Horopter). Alle anderen Punkte im Raum werden auf unterschiedliche (disparate) Punkte auf der Netzhaut abgebildet. Je weiter ein Punkt vom Horopter entfernt liegt, desto größer ist der Abstand zwischen den beiden projizierten Orten auf der Netzhaut und desto größer ist der Unterschied zwischen dem linken und dem rechten Netzhautbild. Dieser Unterschied wird als absolute Disparität bezeichnet [20].

2.4 Stand der Technik

In diesem Abschnitt werden verwandte Arbeiten beschrieben. Wie bereits in Abschnitt 2.3 dargestellt, existieren verschiedene Tiefenhinweise, die in bestimmten Situationen gut oder nur teilweise funktionieren [22]. Hierbei hat sich vor allem die Verdeckung als sehr effizient herausgestellt.

Kruijff et al. [24] überprüften das ursprünglichen Paper von Drascic und Milgram [25] von 1996, welches Probleme von AR diskutierte. Die meisten dieser Herausforderungen wurden durch neue Hardware gelöst. Das Problem der korrekten Tiefenwahrnehmung bleibt jedoch erhalten.

In der Arbeit von Nicolau et al. [26] wird der aktuelle Stand der Technik in Bezug auf AR in der Laparoskopie beschrieben. Hierbei werden das Vorgehen, die Methoden und Werkzeuge, sowie die Prinzipien und Vor- und Nachteile eines laparoskopischen Navigationssystems erläutert. Jedoch werden keine weiteren Konzepte zur Verbesserung von überlagerten, augmentierten Daten beschrieben.

Schlussendlich werden die Hauptprobleme von AR aufgezählt: eingeschränkte Tiefenhinweise und Organdeformationen, die durch Organbewegungen und die Interaktion des Chirurgen mit dem Organ verursacht werden. Sie verhindern eine korrekte Überlagerung der augmentierten Daten in einem realen Szenario. Hier ist weitere Entwicklung nötig.

Bichlmeier et al. [27, 28] entwickelten das Virtual Window (siehe Abbildung 2.8) und verglichen dieses mit der Darstellung von augmentierten Daten, die transparent oder volumetrisch überlagert wurden. Das Virtual Window verdeckte augmentierte Daten außerhalb einer vorgegebenen Geometrie, sodass der Eindruck entsteht, dass das Fenster Einblick in diese gibt. Es setzte sich gegen transparente und volumetrische Überlagerungen durch. Das Fenster wurde dabei auf der Oberfläche des Patienten verankert und wurde mit einem stereoskopischen Head-Mounted-Display und einem Trackingsystem realisiert.



Abbildung 2.8: Abbildung (a) zeigt ein Phantom mit augmentierter Wirbelsäule, die durch eine einfache Überlagerung deplatziert wirkt. Auf Abbildung (b) wird die Wirbelsäule teilweise vom Virtual Window verdeckt und erweckt den Eindruck einer korrekten Tiefe [28].

Um das Tiefenproblem in der medizinischen AR anzugehen, entwickelten Choi et al. [29] zwei Konzepte. Erstens einen dynamischen, nahtlosen Übergang von AR und Virtual Reality (VR) und zweitens die Visualisierung des minimalsten Abstandes zwischen der Spitze eines Werkzeugs und der augmentierten Geometrie (siehe Abbildung 2.9). Zusätzlich wurde der Abstand als Zahl dargestellt. Auf diese Art und Weise lies sich die Position eines Werkzeugs im Raum gut

einschätzen. Die Berechnung des minimalsten Abstandes einer hochauflösenden Geometrie zu einem Punkt kann jedoch rechentechnisch aufwändig sein.



Abbildung 2.9: Das Konzept der Darstellung des minimalsten Abstandes von der Spitze eines chirurgischen Werkzeugs zur augmentierten Geometrie [29].

Ein Ansatz zur Lösung des Problems der Organbewegungen wurde von Haouchine et al. [30] beschrieben. Ihr Ansatz beinhaltete eine Diskretisierung der augmentierten Daten und eine physikalische Simulation dieser, welche durch Informationen auf laparoskopischen Aufnahmen berechnet wurden (siehe Abbildung 2.10). Die Performance ihres Ansatzes ist jedoch nur bei geringer Auflösung echtzeitfähig, sodass das Problem weiterhin relevant bleibt.

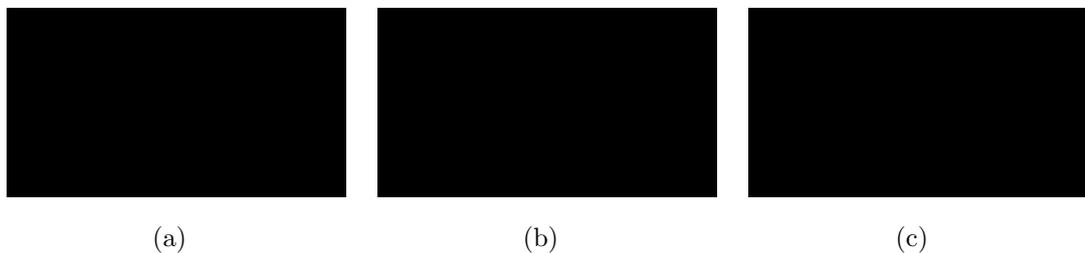


Abbildung 2.10: Auf Abbildung (a) sind die Featurepunkte einer Leber zu sehen. Das in Abbildung (b) dargestellte diskretisierte Modell der Leber konnte durch diese Punkte deformiert werden. Abbildung (c) zeigt die Überlagerung der deformierten, virtuellen Leber mit ihren Gefäßen [30].

3 Konzeption

In diesem Kapitel werden Konzepte zur Lösung der Aufgabenstellung entwickelt und vorgestellt, die anhand einer Anforderungsanalyse ausgewählt wurden.

3.1 Anforderungsanalyse

Die Anforderungsanalyse dient der Definition von Rahmenbedingungen, die für eine Augmentierung von virtuellen Patientendaten für stereoskopische Laparoskopie notwendig sind. Diese umfassen technische, wahrnehmungsspezifische sowie anwendungsspezifische Anforderungen.

3.1.1 Technische Anforderungen

Die technischen Anforderungen dienen der Überprüfung der Machbarkeit auf einem vorgegebenen System. Außerdem legen sie Kriterien an ein zu implementierendes Konzept fest.

Als Basis für die Konzepte wird das Navigationssystem CAS-One Liver (siehe Abbildung 3.1) der Firma CAScination [31] (Bern, Schweiz) verwendet. Die Hardware umschließt dabei eine NVIDIA Quatro K2000 Grafikkarte, einen Intel(R) Core(TM) i7-4790K Prozessor mit acht Kernen und vier Gigahertz (GHz), sowie einen 16 Gigabyte (GB) Arbeitsspeicher. Das Navigationssystem erfasst laparoskopische Werkzeuge durch optisches Tracking (siehe Abschnitt 2.1.2) und ist in der Lage das Bild des Laparoscops auf einem Monitor auszugeben und mit präoperativ gewonnenen Patientendaten zu überlagern. Die Interaktionsmöglichkeiten mit dem System werden über Touchscreen und die verwendeten laparoskopischen Werkzeuge realisiert. Dementsprechend sollte die Interaktion mit den

Konzepten und ihren Parametern auf wenige Arbeitsschritte beschränkt sein. Eine weitere wichtige Anforderung ist die der Echtzeitfähigkeit. Echtzeitfähigkeit bedeutet, dass die Berechnung der augmentierten Bildern innerhalb eines bestimmten Zeitintervalls geschehen muss, damit ein Mensch diese als verzögerungsfrei wahrnimmt. Da die Definition von Echtzeit kein konkretes Zeitintervall nennt und stattdessen von der jeweiligen Anwendung abhängt, ist das hier limitierende Zeitintervall 30 Bilder pro Sekunde, dies entspricht ca. 33,33 Millisekunden pro Bild. Wenn diese Anforderung nicht erfüllt ist, agiert das System verzögert und der Chirurg sieht veraltete Bilddaten, was in einem realen Szenario sehr störend sein kann.



Abbildung 3.1: Das verwendete CAS-ONE Liver System (Bildquelle: [32]).

3.1.2 Wahrnehmungsspezifische Anforderungen

Die Anforderungen an die wahrnehmungsbezogenen Aspekte befassen sich vor allem mit der Frage, ob ein Konzept von einem Menschen richtig interpretiert werden kann und somit hilfreich ist. Es befasst sich auch mit der Frage, ob ein Konzept die Kapazitäten der menschlichen Wahrnehmung berücksichtigt und in sinnvollem Maße unterstützt.

Die Anzahl an Informationseinheiten, die ein Mensch bewusst in jedem Augenblick wahrnehmen kann, ist auf ca. sieben beschränkt. Diese Informationseinheiten werden auch Chunks genannt [33]. Dies bedeutet, dass die Anzahl der augmentierten Informationen auf ein geringes Maß beschränkt sein sollte, um die Aufmerksamkeit eines Chirurgen nicht unnötig zu strapazieren [34]. Eine weitere Einschränkung der menschlichen visuellen Wahrnehmung ist, dass sie in nur einem relativ geringen Maße dazu fähig ist, verschiedene Grau- bzw. Farbwertabstufungen wahrzunehmen. Dies hat vor allem auf Konzepte und Visualisierungen Auswirkungen, die Farben beinhalten. Zusätzlich ist der Mensch in der Lage, Farbunterschiede in dunklen Bereichen besser wahrzunehmen als in hellen Bereichen, sodass eine lineare Farbskala nicht-linear wahrgenommen wird. Zusätzliche Arbeit an Farbskalen ist nötig, sofern dieser Effekt relevant ist. Außerdem kann das menschliche visuelle System Blautöne schlechter wahrnehmen als Rot- oder Grüntöne. Dies bedeutet, dass viele Blauabstufungen vermieden werden sollten, wenn erwünscht ist, dass ein Chirurg anhand von diesen Entscheidungen treffen soll [5].

3.1.3 Anwendungsspezifische Anforderungen

Die letzte Kategorie der Anforderungen ist die der Anwendungsspezifität. Hier wird vor allem auf die Rahmenbedingungen der Nutzung der Konzepte in einem konkreten Anwendungsfall eingegangen. Es befasst sich auch mit der Frage, ob ein Konzept dazu beiträgt den Anwendungsfall effektiver durchzuführen oder ob es diesen erschwert. Die zu entwickelnden Konzepte sollen an die beiden Anwendungsfälle Ablation und Resektion angepasst sein (siehe Abschnitt 2.1.2).

Bei einer Ablation werden Lebertumore und -metastasen und umliegendes Gewebe durch ein heißes laparoskopisches Werkzeug zerstört. Hierbei ist es unerlässlich, das umliegende Gewebe nach Möglichkeit zu schützen und gegebenenfalls eine andere Operations- oder Zugangsart zu wählen. Hieraus ergibt sich eine der wichtigsten Anforderungen an ein Konzept für die Ablation: die Gefahrenstufe für umliegendes Gewebe sollte hervorgehoben werden. Der Blick auf augmentierte Daten ist in der näheren Umgebung erwünscht.

Bei einer Resektion hingegen, bei der Lebertumore und -metastasen durch Wegschneiden von Gewebe entfernt werden, ist eine unaugmentierte oder zumindest eine nur leicht augmentierte Sicht wichtig, da der Chirurg zielgenau am realen

Bild arbeiten muss. Jedoch ist es wünschenswert, die Strukturen in der weiteren Umgebung zu sehen, damit der Chirurg sich orientieren kann und genau weiß, wo das Resektionsziel liegt.

Da während eines laparoskopischen Eingriffs Komplikationen auftreten können, die situationsabhängige Reaktionen des Chirurgen erfordern, sollten die Konzepte jederzeit flexibel anpassbar oder deaktivierbar sein. Dies beinhaltet beispielsweise schnelle Veränderungen verschiedener Parameter der Visualisierung, um den Chirurgen in jedem Fall optimal unterstützen zu können. Außerdem sind Konzepte zu bevorzugen, die Rückschlüsse auf reale Informationen zulassen, beispielsweise das Ablesen von millimetergenauen Entfernungen statt des Ablesens von subjektiven Begriffen wie nah und fern. Ebenso sollten verstellbare Parameter domänenspezifische Einheiten verwenden (z.B. Millimeter statt generischer Einheiten). Da das verwendete System stereoskopische Bilder verwendet, sind zudem Konzepte vorzuziehen, die in dreidimensionalen Bildern anwendbar sind.

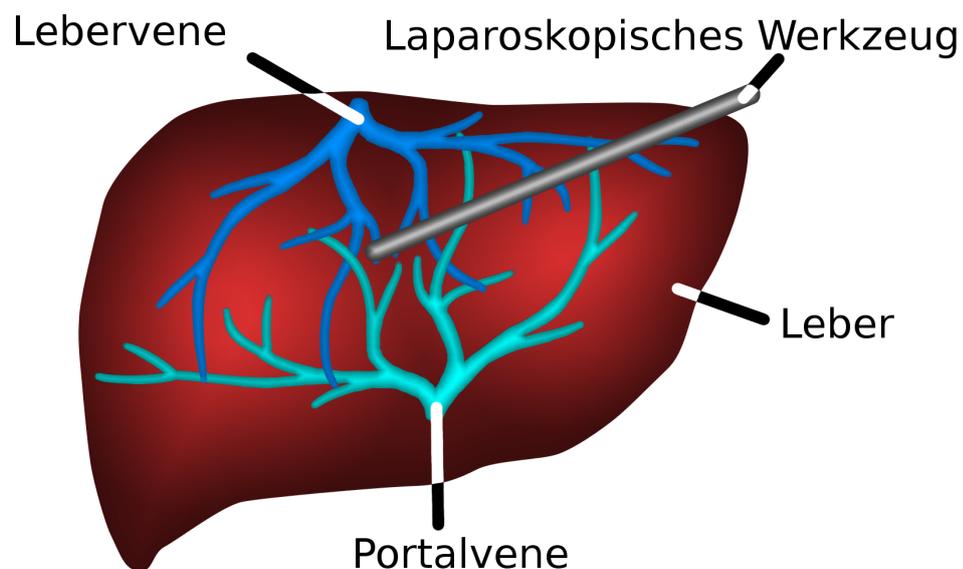


Abbildung 3.2: Die nachfolgenden Konzepte werden an diesem Entwurf gezeigt.

Die braunrote Leber wird von ihren virtuellen, augmentierten Gefäßen (in blau dargestellt) überlagert und wirken, als lägen sie vor der Leber. Ein laparoskopisches Werkzeug ist in grau dargestellt.

3.2 Virtual Window

Ein großes Problem von AR ist, dass das aufgenommene Bild nur noch wenige Rückschlüsse auf Tiefeninformationen zulässt und dass simpel überlagerte, augmentierte Daten keine angemessene Tiefenwahrnehmung entstehen lassen können [5] (siehe Abbildung 2.8(a)).

Das erste Konzept ist an das Virtual Window von Bichlmeier et al. [28] angelehnt (siehe Abschnitt 2.4). Hierbei wird ein künstliches Fenster erzeugt, das die Sicht auf die augmentierten Daten erlaubt. Außerhalb des Fensters sind die Daten nicht betrachtbar. Je nach Definition des Fensters ergibt sich hieraus bereits ein starker Tiefeneindruck, da die Verdeckung der zuverlässigste Tiefenhinweis ist (siehe Abschnitt 2.3.2). Durch die Verwendung des Fensters wirkt es, als lägen die augmentierten Daten hinter dem realen Bild und erlauben so eine verbesserte Wahrnehmung der Tiefenrelation (siehe Abbildung 2.8(b)).

Das Virtual Window soll im Anwendungsfall der Laparoskopie in folgenden Punkten erweitert bzw. modifiziert werden, um die oben genannten Anforderungen zu erfüllen:

Interaktivität: Die Position des Virtual Windows soll in Echtzeit an die Position getrackter Werkzeuge anpassbar sein. Dies erlaubt es dem Chirurgen, zu jedem Zeitpunkt die gewünschten augmentierten Daten zu betrachten und unnötige Informationen auszublenden.

Form: Das Virtual Window soll kreisförmig gestaltet werden. Zudem soll die Größe des Fensters anpassbar sein, um dem Chirurgen entweder zu helfen eine allgemeine Übersicht über die Daten zu erlangen oder sich fokussiert auf einen bestimmten Ort zu konzentrieren.

Verhalten: Die Stärke der Überlagerung der augmentierten Daten soll innerhalb und außerhalb des Fensters jederzeit steuerbar sein. Dies hat zudem den großen Vorteil, dass Daten im Fenster sowohl angezeigt als auch ausgeblendet werden können, was sich direkt auf die Anforderungen der Anwendungsfälle Ablation und Resektion übertragen lässt.

Das Virtual Window dient also erstens dazu, die Tiefe der augmentierten Daten besser einzuschätzen. Zweitens reduziert es die Fülle an virtuellen Informationen, die gleichzeitig dargestellt werden und ermöglicht so einen besseren Überblick.

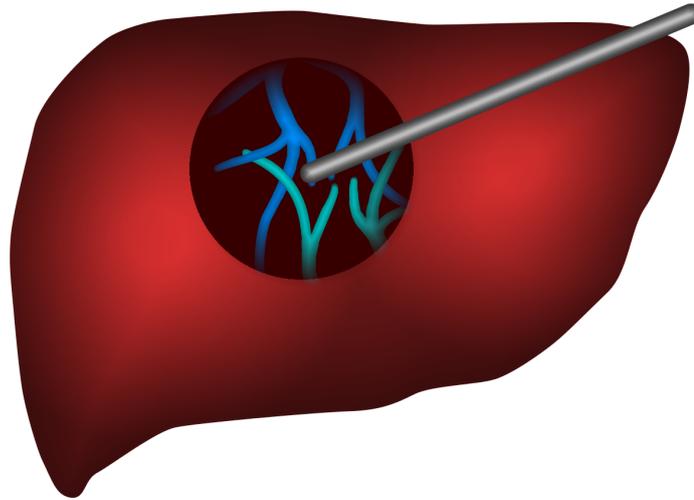


Abbildung 3.3: Das Konzept des Virtual Window auf die Leber und ihre Gefäße angewandt. Das Fenster gibt Einblick in das Innere der Leber. Hierfür wird die Position des Fensters in Relation zur Position des Werkzeugs gebracht.

3.3 Farbdistanzkodierung

Eine geläufige Visualisierungstechnik, um quantitative Daten darzustellen, ist die Abbildung von diesen auf eine visuelle Repräsentation etwa durch Isolinien oder Farbkodierungen [5]. Für dieses Konzept soll die Abbildung von Distanzen auf Farbe gewählt werden. Die quantitativen Daten, die hier visualisiert werden sollen, sind die Entfernungen der Spitze eines laparoskopischen Werkzeugs zu jedem Punkt der augmentierten Daten. Dazu müssen die Koordinatensysteme des virtuellen Werkzeugs und der augmentierten Daten übereinstimmen (siehe Abschnitt 2.2). Die Entfernung der dreidimensionalen Punkte kann dann durch ein Entfernungsmaß berechnet werden. Das hier verwendete Distanzmaß ist die euklidische Distanz d , welche sich im dreidimensionalen Raum für zwei beliebige Punkte p_0, p_1 folgendermaßen berechnet:

$$d(p_0, p_1) = \sqrt{(p_{1.x} - p_{0.x})^2 + (p_{1.y} - p_{0.y})^2 + (p_{1.z} - p_{0.z})^2} \quad (3.1)$$

Anschließend kann diese Entfernung auf Farbe abgebildet werden. Hierbei spielt es eine entscheidende Rolle, dass die gewählte Farbskala eine Interpretation des

Menschen auf die zugehörige Entfernung zulässt, dass also zwei Farben durch ein bestimmtes Kriterium miteinander verglichen werden können. Geringe Distanz kann beispielsweise auf heiße Farbstrahlungen abgebildet werden und hohe Distanz auf kalte (siehe Abbildung 3.4).

Ein wichtiger Diskussionspunkt ist die Art des Farbverlaufs. Während ein stetiger Farbverlauf, wie in Abbildung 3.4(a) dargestellt, die maximale Informationsdichte enthält, wird gerade in der chirurgischen Anwendung oft eine diskrete Darstellung bevorzugt (siehe Abbildung 3.4(b)), da chirurgische Entscheidungen während eines Eingriffs ebenfalls diskreter Natur sind [5]. So lassen sich beispielsweise Risikobereiche bilden, die eindeutig durch bestimmte Farben charakterisierbar sind. Zum Beispiel ist eine simple und verständliche Abbildung hierbei eine Farbskala, die sich an einer Ampel orientiert. Rot deutet auf hohe, gelb auf mittlere und grün auf niedrige Risikobereiche hin. Hieran kann ein Chirurg direkt die Gefahrenstufe für die umliegenden Strukturen während eines Eingriffs ablesen.

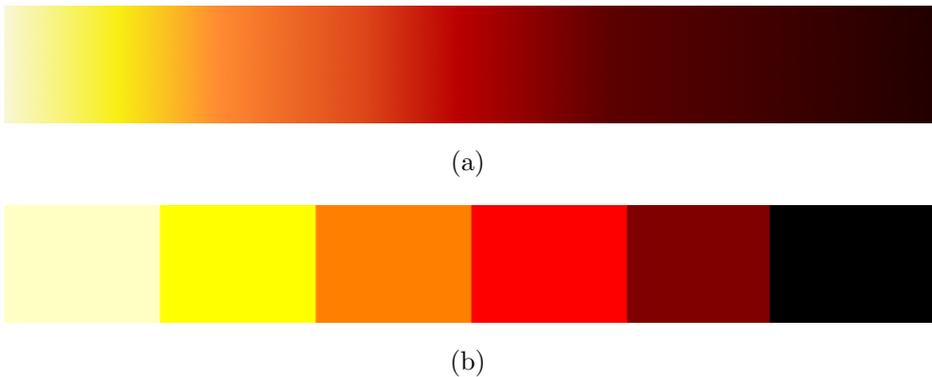


Abbildung 3.4: Die Hitzefarbskala simuliert die Erhitzung eines metallischen Körpers, welcher dadurch sichtbare Strahlung abgibt. Schwarz bedeutet hierbei, dass der Körper nicht glüht. Dann erhitzt er sich und glüht rot, um dann schließlich über gelb in weiß überzugehen. Abbildung (a) zeigt eine kontinuierliche Hitzefarbskala und auf Abbildung (b) ist eine diskrete und äquidistante Hitzefarbskala zu sehen.

Dieses Konzept der Farbdistanzkodierung dient dem Chirurgen sowohl als Orientierungshilfe zur Navigation eines Werkzeugs, als auch zur Abschätzung von Risikobereichen, die bei einer Ablation Schäden erleiden könnten.

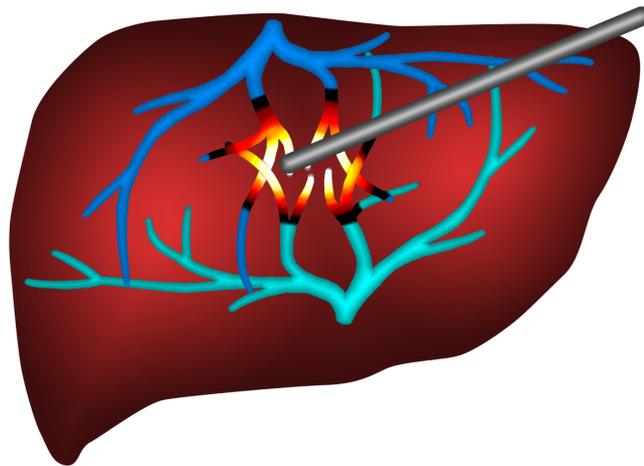


Abbildung 3.5: Das Konzept der Farbdistanzkodierung berechnet die Entfernung von umliegenden augmentierten Lebergefäßen zur Spitze des Werkzeugs und bildet die Entfernung auf eine Farbskala ab.

3.4 Illustrative Visualisierung

Bei der konventionellen Darstellung von dreidimensionalen Objekten werden diese beleuchtet und verdecken wichtige Teile der realen Sicht auf das zu operierende Gewebe. Eine Möglichkeit dieses Problem zu umgehen, ist die Darstellung einer abstrakteren Art der Visualisierung der augmentierten Daten [5]. Ein solches abstraktere Rendering wird Non-photorealistic Rendering (NPR) genannt.

Konturen: Feature Lines beschreiben im Allgemeinen die Darstellung von Merkmalen eines Objektes durch Linien (siehe Abbildung 3.6). Konturen sind eine Art von Feature Lines, die die Umrisse und Orte mit hoher Krümmung als Linien darstellt. Durch diese Art des Renderings wird weniger vom Kamerabild des Laparoscops verdeckt und erhöht so die freie Sicht auf reale Daten. Jedoch verlieren die augmentierten Daten auf diese Art ihren dreidimensionalen Eindruck und damit reduziert sich ihre Tiefenwahrnehmung. Eine Lösung ist es, die dargestellten Linien mit Tiefeninformationen zu versehen, um dem Chirurgen zusätzliche Orientierung zu geben. So kann die Tiefe zum Beispiel auf Transparenz und Strichstärke abgebildet werden [35].

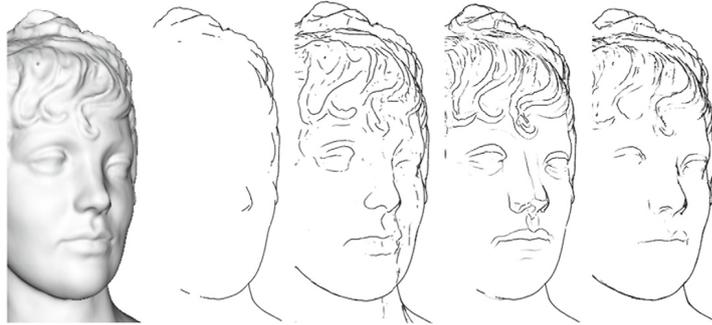


Abbildung 3.6: Vergleich von konventionellem Shading und verschiedenen Feature Lines. Von Links nach Rechts: Silhouetten, suggestive Konturen, Hügel- und Tallinien und scheinbare Furchen (Quelle: [36]).

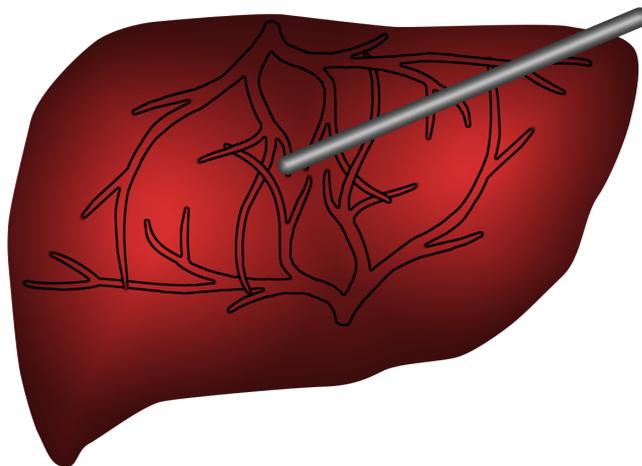


Abbildung 3.7: Das Konzept der Konturen. Der Tiefeneindruck durch Eigenverdeckung der Strukturen bleibt erhalten. Ansonsten sollte dieses Konzept mit weiteren Konzepten kombiniert werden, um seine Effektivität zu erhöhen.

Toon-Shading: Um das Problem der relativ hohen visuellen Komplexität durch Beleuchtung zu vermeiden, kann bewusst auf reales Shading verzichtet und auf nicht-realistisches Shading zurückgegriffen werden. Beim sogenannten Toon-Shading wird der übliche kontinuierliche Helligkeitsverlauf durch die Beleuchtung auf wenige Stufen diskretisiert und kann anschließend mit Konturen überblendet werden, um das Objekt vom Hintergrund abzuheben (siehe Abbildung 3.8). Durch diese Art des Renderings reduzieren sich die Details eines Objektes, was zu einer Abnahme der Komplexität der Szene allgemein führen kann.



Abbildung 3.8: Konventionelles Rendering und Toon-Shading im Vergleich (Bildquelle: [37]).

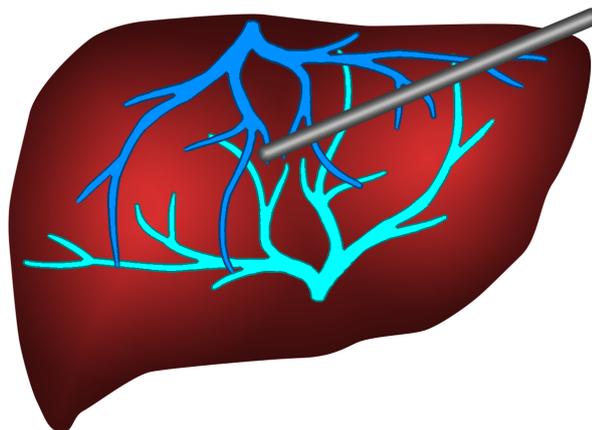


Abbildung 3.9: Das Toon-Shading reduziert den Farbverlauf der Lebergefäße, der durch Beleuchtung entsteht.

4 Implementierung

Dieses Kapitel umfasst alle Aspekte für die Implementierung der in Kapitel 3 vorgestellten Konzepte. Zuerst wird die verwendete Entwicklungsumgebung, mit der die Konzepte umgesetzt wurden, beschrieben. Hierbei wird eine genauere Erläuterung von Coin3D und der OpenGL Shading Language (GLSL) gegeben. Anschließend wird die konkrete Implementierung des Virtual Windows, der Farbdistanzkodierung, sowie den beiden illustrativen Techniken Konturen und Toon-Shading beschrieben. Die Umsetzung wird mit Hilfe von Beschreibungen, Pseudocode und Bildern erläutert.

4.1 Entwicklungsumgebung

In diesem Abschnitt werden die Entwicklungsumgebung, die verwendeten Bibliotheken und Frameworks beschrieben, mit deren Hilfe die Konzepte umgesetzt werden. Als integrierte Entwicklungsumgebung (IDE) wird Microsofts Visual Studio in der Version 2013 mit der Programmiersprache C++ genutzt [38]. Als Grafikbibliothek zur 3D-Visualisierung wird Coin3D in der Version 3.1.3 eingesetzt [39]. Um Einfluss auf das Rendering durch Coin3D zu nehmen, werden Shader in GLSL geschrieben [40]. Zur Interaktion mit der Visualisierung der Konzepte werden Benutzerschnittstellen mit der Bibliothek Qt in der Version 5.3.2 umgesetzt [41]. Durch das verwendete CAS-One Liver Navigationssystem (siehe Abschnitt 3.1.1) stehen verschiedenen Leberdatensätze mit Tumoren und Gefäßen, sowie registrierte laparoskopische Aufnahmen mit getrackten Werkzeugen zur Verfügung.

4.1.1 Coin3D

Coin3D ist eine high-level 3D-Grafikchnittstelle, die auf OpenGL basiert. Als solche vereinfacht sie den Umgang mit OpenGL und den Aufwand, um eine 3D-Szene zu erstellen. Zur Erstellung einer solchen Szene nutzt Coin3D Knoten. Diese Knoten enthalten verschiedene Informationen und sind die Bausteine des sogenannten Szenengraphen. Der Szenengraph ist eine gerichtete, hierarchische Datenstruktur, die durch Aneinanderreihung von Knoten entsteht. Es existieren Knoten, die beispielsweise Flächen, Modelle, Lichter, Kameras und ihre zugehörigen Eigenschaften enthalten. Der Szenengraph wird in den häufigsten Fällen genutzt, um die Daten, die die Knoten beinhalten, zu visualisieren. Abbildung 4.1 zeigt beispielhaft einen Szenengraphen, der eine beleuchtete Kugel mit bestimmten Materialeigenschaften rendert. Die Knoten des Szenengraphen werden dabei in pre-order traversiert. Beim Traversieren führt jede Art von Knoten eine spezifische Aktion durch. So rendert beispielsweise der Knoten *SoSphere* eine Kugel mit gegebenen Parametern. Es gibt außerdem Knoten, die den Zustand des OpenGL Zustandsautomaten verändern. Coin3D unterstützt in der Version 3.1.3 Vertex- und Fragment-Shader. Die Tabelle 4.1 zeigt die wichtigsten verwendeten Knoten und gibt eine kurze Beschreibung.

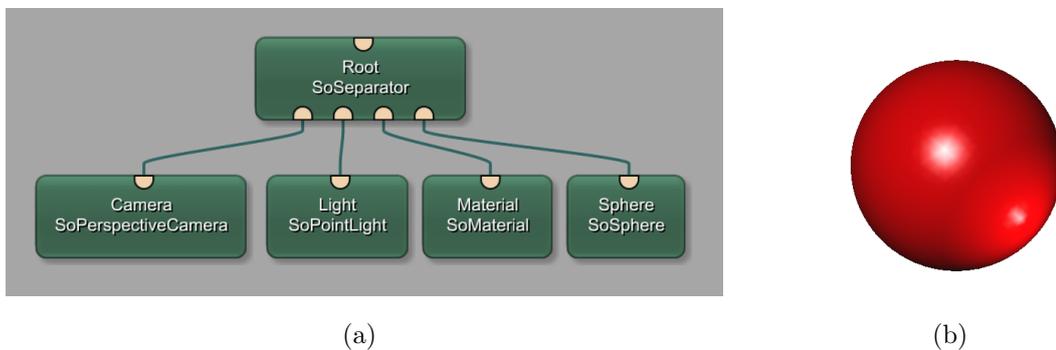


Abbildung 4.1: Abbildung (a) zeigt einen beispielhaften Szenengraphen, der eine Kamera, eine Punktlichtquelle, ein Material und eine Kugel beinhaltet. Auf Abbildung (b) ist die gerenderte Kugel, welche durch den Szenengraphen definiert wurde, dargestellt. Die Bilder wurden mit der Software MeVisLab [42] erstellt.

Tabelle 4.1: Übersicht über einige Coin3D Knoten [39].

Knotenname	Beschreibung
<i>SoNode</i>	Abstrakte Klasse, von der alle Knoten erben.
<i>SoGroup</i>	Ein Knoten, an den weitere Knoten angehängt werden können. Veränderungen am Zustand von OpenGL bleiben nach Traversierung erhalten.
<i>SoSeparator</i>	An diesen Knoten können ebenfalls Unterknoten angehängt werden. Er speichert beim Traversieren den aktuellen Zustand von OpenGL und stellt ihn nach erfolgreichem Traversieren wieder her.
<i>SoCamera</i>	Beinhaltet die Transformationen, die zum Rendering nötig sind.
<i>SoShaderProgram</i>	Beinhaltet einen Vertex- und einen Fragment-Shader. Diese werden beim Traversieren gebunden.
<i>SoShaderParameter</i>	Mit diesem Knoten können Uniform-Variablen an Shader übergeben werden.
<i>SoTexture</i>	Beinhaltet eine Textur, die beim Traversieren gebunden wird.
<i>SoSceneTexture2</i>	Zeichnet einen gegebenen Szenengraphen in eine Textur und bindet sie.

4.1.2 OpenGL Shading Language

GLSL ist eine auf C basierte high-level Programmiersprache, die es ermöglicht, Teile der Grafikpipeline zu programmieren, um Einfluss auf das Rendering und damit auf das Ergebnis der Darstellung von virtuellen 3D-Objekten zu nehmen. GLSL stellt zudem Klassen für Matrizen und Vektoren, sowie Operationen auf diesen bereit [43]. Dabei besteht ein GLSL Programm aus mindestens einem Vertex- und einem Fragment-Shader, die kompiliert und gelinkt werden. Zusätzlich existieren in der aktuellsten Version von GLSL Geometry-, Tessellation- und Compute-Shader. Im Folgenden werden Vertex- und Fragment-Shader näher beschrieben, da sie zur Umsetzung der Konzepte verwendet werden. Der Vertex-Shader wird für jeden zu zeichnenden Vertex aufgerufen. Ein Vertex ist ein Punkt im 3D-Raum mit verschiedenen Eigenschaften wie z.B. Farbe und Normale. Er ist meist Teil eines größeren geometrischen Modells. Typische Aufgaben des Vertex-

Shaders sind die Transformationen von Vertices, Normalen und Texturkoordinaten und die Berechnung dieser, sowie Beleuchtung. Zusätzlich können beliebige abgeleitete Eigenschaften berechnet und an den Fragment-Shader weitergegeben werden [43]. Der Fragment-Shader wird für jedes zu zeichnende Fragment des zu zeichnenden 3D-Modells aufgerufen. Ein Fragment beschreibt hierbei ein Bildelement (Pixel) bevor es auf dem Bildschirm dargestellt wird. Der Fragment-Shader bekommt interpolierte Informationen vom Vertex-Shader und verwendet diese, um die Farbe des Fragments zu bestimmen, für welches der Shader aufgerufen wurde. Zusätzlich können Variablen (sogenannte Uniform-Variablen) an die Shader übergeben werden, auf die alle Vertices oder Fragmente Zugriff haben, um zusätzliche Informationen zur Berechnung zu verwenden, wie beispielsweise Texturen, Vektoren, Matrizen und skalare Werte.

4.2 Umsetzung der Konzeptentwürfe

In diesem Abschnitt wird die Implementierung des Virtual Windows, der Farbdistanzkodierung, der Konturen und des Toon Shadings mithilfe der oben genannten Entwicklungsumgebung beschrieben. Grundlagen bilden hierbei die Konzepte und die Anforderungsanalyse aus Kapitel 3. Alle Konzepte basieren auf einer Kombination aus Knoten, die Coin3D bereitstellt und verschiedenen selbst entwickelten GLSL-Shadern. Die Shader werden hierbei vom Knoten *SoShaderProgram* gewrappt und in den Szenengraph direkt vor die entsprechenden Modelle gehangen. Hierbei sollten sie sich in einem gemeinsamen *SoSeparator* befinden, damit keine weitere ungewollte Geometrie von diesem Shader betroffen wird. Die Abbildung 4.2 zeigt den gegebenen Szenengraphen und die Abbildung 1.1 zeigt den Ausgangspunkt des visualisierten Szenengraphen ohne die Verwendung eines Konzeptes.

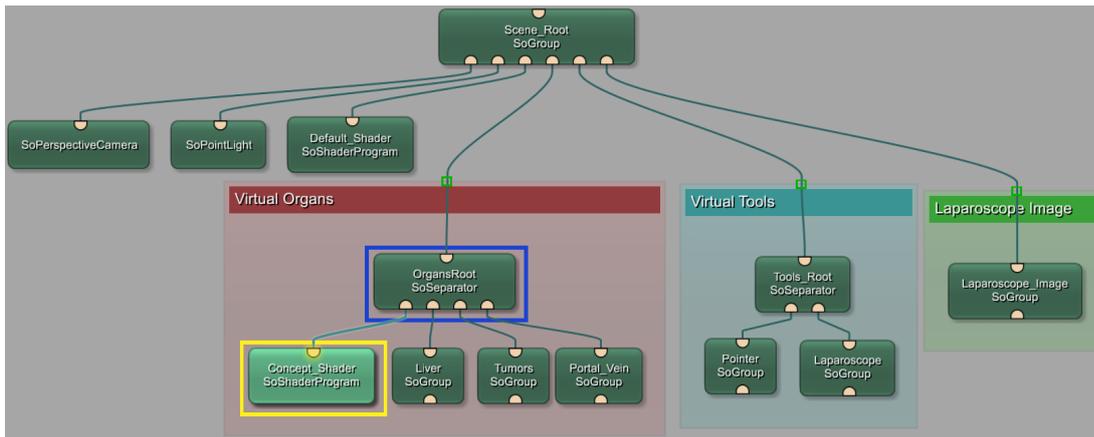


Abbildung 4.2: Der vorhandene Szenengraph mit einer Kamera, einer Punktlichtquelle und einem Standard-Shader. Die Knoten in der roten Box sind die zu augmentierenden Leberdaten. Die hellblaue Box beinhaltet die virtuellen, getrackten Werkzeuge und der Knoten in der grünen Box repräsentiert das Bild, welches vom Laparoskop aufgenommen wurde. Der gelb markierte Knoten ist der Shader, der die Konzepte umsetzt.

4.2.1 Virtual Window

Das Virtual Window soll Geometrie innerhalb und außerhalb anzeigen bzw. ausblenden können. Dazu wird die Sichtbarkeit der Geometrie über seinen Alphawert und die Blending-Funktion gesteuert. Die Blending-Funktion beschreibt, wie die Farbwerte von zwei überlagernden Fragmenten miteinander gemischt werden sollen. Im Vertex-Shader wird lediglich die zu augmentierende Geometrie und die übergebene 3D-Position des Virtual Windows auf Bildschirmkoordinaten transformiert. Die Hauptarbeit findet im Fragment-Shader statt. Als externe Uniform-Variablen werden der Radius des Virtual Windows, die Alphawerte innerhalb und außerhalb des Virtual Windows und ein Radius für den Übergangverlauf an den Fragment-Shader übergeben. Um das Virtual Window anzuwenden, wird die euklidische Distanz (siehe Formel 3.1) zwischen dem aktuellen Fragment zur 2D-Position des Fensters berechnet und mithilfe der folgenden Gleichung auf das Intervall $[0, 1]$ abgebildet.

$$t = clamp\left(\frac{d - r_0}{r_1}, 0, 1\right)$$

d ist der euklidische Abstand zwischen Fragment und Koordinate des Virtual Windows im 2D. r_0 ist der Radius des Virtual Windows und r_1 ist der Radius des

Übergangsbereiches in Bildschirmkoordinaten. Der Wert ist null für Fragmente, die sich innerhalb und eins für Werte, die sich außerhalb des Virtual Windows befinden und er liegt zwischen ihnen für Fragmente, die im Übergangsbereich liegen. Mit diesem normalisiertem Wert kann eine einfach lineare Interpolation zwischen den Alphawerten innerhalb und außerhalb des Virtual Windows durchgeführt und dem Fragment zugewiesen werden. Wenn ein Shading für die sichtbaren augmentierten Daten gewünscht ist, muss die Berechnung davon ebenfalls im Vertex- und Fragment-Shader stattfinden. Hierfür stehen die RGB-Kanäle zur Verfügung, die für das Virtual Window nicht benötigt werden. Das Listing 4.1 zeigt die Umsetzung des Virtual Windows als Pseudocode im Fragment-Shader.

```

1  //Transparenz außerhalb des Fensters:
2  uniform float alphaOut;
3  //Transparenz innerhalb des Fensters:
4  uniform float alphaIn;
5  //Radius des Fensters:
6  uniform float vwRadius;
7  //Radius des Übergangsbereiches:
8  uniform float falloffRadius;
9  //2D Position des Fensters aus dem Vertex-Shader:
10 in vec2 vwPos;
11
12 //Fragment-Shader
13 void main()
14 {
15     //Abstand zwischen Fenster und Pixelkoordinate:
16     float dist = distance(gl_FragCoord.xy, vwPos);
17     //Abstand auf das Intervall [0, 1] abbilden:
18     float t = normalizeDistance(dist, vwRadius, falloffRadius);
19
20     //Transparenz des Pixels durch lineare Interpolation setzen:
21     gl_FragColor.a = lerp(alphaIn, alphaOut, t);
22
23     //Farbe des Pixels durch Shading berechnen:
24     gl_FragColor.rgb = computeShading(gl_Color.rgb);
25 }

```

Listing 4.1: GLSL Pseudocode zur Berechnung des Virtual Windows im Fragment-Shader.

4.2.2 Farbdistanzkodierung

Bei der Farbdistanzkodierung sollen die zu augmentierenden Daten anhand eines Abstandes zu einem virtuellen Punkt im 3D-Raum eingefärbt werden, hierbei soll nur Geometrie innerhalb eines bestimmten Radius eingefärbt werden. Dieser Radius wird im Folgenden Risikobereich genannt und ist in Millimetern definiert. Als virtueller 3D-Punkt wird die Spitze eines Laparoskopiewerkzeugs gewählt. Der euklidische Abstand (siehe Formel 3.1) zu der Werkzeugspitze wird in Weltkoordinaten gemessen, sodass dieses Konzept im Vertex-Shader vor der Transformation auf Bildschirmkoordinaten umgesetzt wird. Innerhalb des Risikobereiches sollen die Daten anhand einer Farbskala eingefärbt werden. Um harte Übergänge zwischen dem Risikobereich und dem Bereich außerhalb zu vermeiden, wird als weiterer Parameter ein Übergangsradius zwischen beiden Bereichen festgelegt. Da der Farbverlauf aus einer Textur gelesen werden soll, muss der gemessene Abstand der Geometrie zur Werkzeugspitze normalisiert werden. Hierfür wird der Radius des Risikobereiches verwendet, sodass Werte innerhalb von diesem auf das Intervall $[0, 1]$ abgebildet werden. Dadurch ist es möglich den gewünschten Farbwert aus einer Textur zu lesen und dem aktuellen Vertex als Attribut zuzuweisen. Die Art des Shadings kann wieder frei gewählt werden, sofern dabei das eben berechnete Farbattribut des Vertex verwendet wird. Das Listing 4.2 zeigt die Umsetzung der Farbdistanzkodierung als Pseudocode im Vertex-Shader. Die Erstellung eines Farbverlaufes durch eine Textur kann über ein Bildbearbeitungsprogramm erfolgen und ermöglicht so einfache Anpassungen. Komplexe Farbverläufe können auf diese Art und Weise ebenfalls erstellt werden. Ebenso ist die Integration eines neuen Farbverlaufes durch eine Textur in das Programm simpel.

Ein großer Nachteil von dem Texturansatz ist, dass der Risikobereich unabhängig von der Textur definiert werden muss, sodass Textur und Risikobereich in sinnvollem Zusammenhang zu bringen sind, um das Ablesen realer Entfernungen zu ermöglichen. Aufgrund dieser Einschränkung wurde, alternativ zum Texturansatz, eine weitere, diskretere Art der Farbkodierung implementiert, die direkt die Definition verschiedener Farben auf Distanzen erlaubt. Dazu wird eine gewünschte Farbe mit einer Distanz versehen und als gebündelte Informationseinheit an den Vertex-Shader übergeben, dieser färbt dann anhand der Distanz zwischen Vertex und Werkzeugspitze den Vertex mit der korrespondierenden Farbe ein. Zum Beispiel das Einfärben von Bereichen zwischen 0 mm und 2 mm in

rot, zwischen 2 mm und 5 mm in gelb und zwischen 5 mm und 10 mm in grün. Abbildung 5.7 zeigt eine auf diese Art erstellte Ampelfarbskala. Dieser Ansatz erschwert die Definition von kontinuierlichen und komplexen Farbverläufen, erlaubt jedoch eine einfache Erstellung von diskreten, interpretierbaren Farbverläufen.

```

1  //Textur mit Farbverlauf:
2  uniform sampler2D colorScale;
3  //Position der Werkzeugspitze:
4  uniform vec3 toolPosition;
5  //Risikobereich:
6  uniform float riskRadius;
7  //Übergangsradius:
8  uniform float falloffRadius;
9
10 //Vertex-Shader
11 void main()
12 {
13     //Transformation vom Vertex auf Bildschirmkoordinaten:
14     gl_Position = ftransform();
15
16     //Abstand der Werkzeugspitze zum Vertex:
17     float dist = distance(toolPosition, gl_Vertex.xyz);
18     //Abstand auf das Intervall [0, 1] abbilden:
19     float t = normalizeDistance(dist, riskRadius, falloffRadius);
20
21     //Farbwert aus der Textur lesen:
22     vec3 scaleColor = texture2D(colorScale, vec2(t, 0)).rgb;
23     //Farbe des Vertex berechnen:
24     gl_Color.rgb = computeShading(scaleColor);
25
26     //Alphawert vom Vertex übernehmen:
27     gl_Color.a = gl_FrontColor.a;
28 }

```

Listing 4.2: GLSL Pseudocode zur Farbdistanzkodierung mit einer Textur im Vertex-Shader.

4.2.3 Illustrative Visualisierung

Wie in Abschnitt 3.4 beschrieben, kann eine abstraktere Art des Renderings helfen, eine Visualisierung verständlicher zu machen. Dazu werden in diesem Abschnitt Konturen und Toon-Shading implementiert.

Konturen

Um Renderings durchzuführen, für die Informationen in lokaler Umgebung benötigt werden, müssen diese manuell zugänglich gemacht werden. Sowohl der Vertex- als auch der Fragment-Shader haben nur Zugriff auf die eigenen Daten, für die sie aufgerufen werden. Jedoch können beide auf Texturen zugreifen, die als Uniform-Variablen zugänglich gemacht werden können [43]. Konturen benötigen solche lokalen Informationen über die Oberflächennormalen von umliegenden Fragmenten. Deshalb ist ein Multiple Pass Rendering für diese erforderlich. Im ersten Durchgang wird die Geometrie mit ihren Normalen als RGB-Werte in eine Textur gerendert (siehe Abbildung 4.3). Hierfür bietet Coin3D den Knoten *SoSceneTexture* an. Wenn dieser Knoten traversiert wird, wird eine ihm gegebene Szene in eine Textur gezeichnet und anschließend gebunden. Die Szene, die in die Textur gezeichnet wird, beinhaltet die Kamera und die zu augmentierenden Daten und den Shader, der die Normalen zeichnet. Anschließend wird ein bildschirmfüllendes Rechteck gezeichnet, welches die erzeugte Textur mit den Normalen enthält. Für jedes zu zeichnende Fragment werden alle Normalen der umliegenden Fragmente gelesen und das Skalarprodukt zwischen diesen Normalen und der Blickrichtung gebildet. Das Skalarprodukt lässt Rückschlüsse darauf zu, wie sehr die Normale in Richtung des Blickvektors zeigt. Ein Skalarprodukt von null bedeutet, dass die Vektoren senkrecht zueinander stehen, positive Werte bedeuten, dass die Vektoren zueinander zeigen und negative Werte bedeuten, dass die Vektoren in dieselbe Richtung zeigen. Es sollen nun nur Fragmente gezeichnet werden, deren Unterschiede der Skalarprodukte zwischen eigenem Fragment und umliegenden Fragmenten am größten ist, da solche Fragmente entweder Teil von markanten Features oder dem Rand der Geometrie sind. Dazu werden die Differenzen dieser Skalarprodukte gebildet und als Alphawert für das zu zeichnende Fragment verwendet. Die Farbe der Konturen kann über eine Uniform-Variable gesteuert werden.

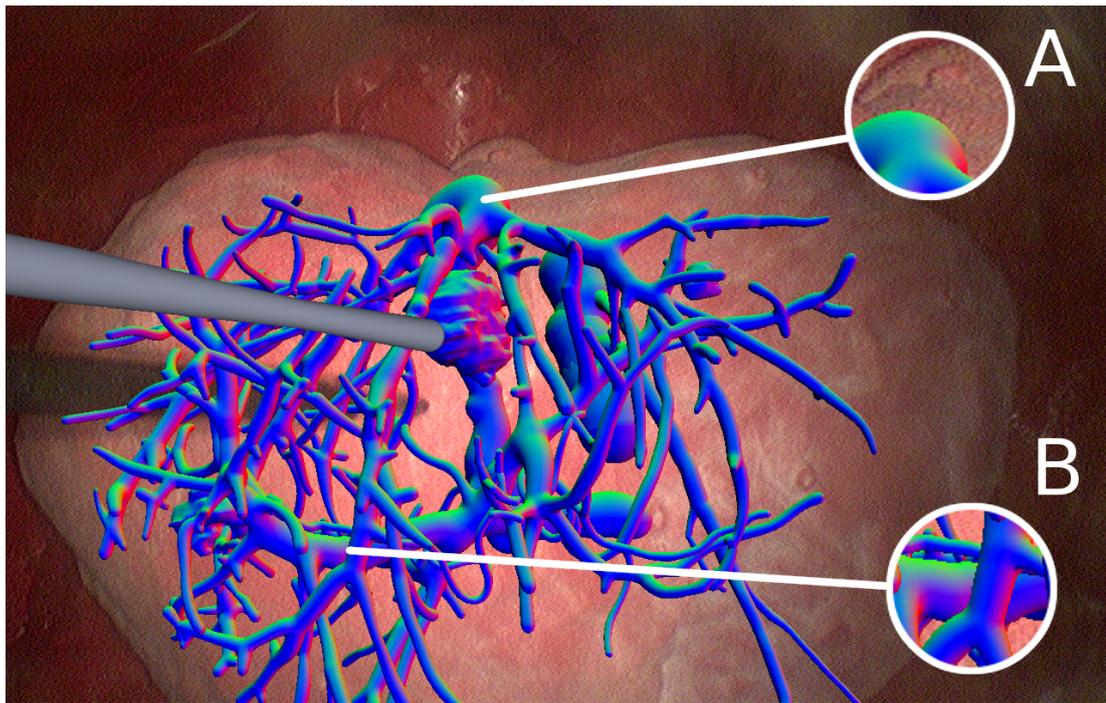


Abbildung 4.3: Die Visualisierung der Normalen der Lebergefäße und Tumore, welche für die Berechnung von Konturen benötigt werden. Orte mit hohen Differenzen der Normalen sind Teil der Konturen, wie beispielsweise die Ränder der Gefäße in Vergrößerung B. Vergrößerung A zeigt einen Ort, mit niedrigen Differenzen. Der Hintergrund besitzt keine Informationen über Normalen, sodass hier die Differenz zu den Gefäßen maximal ist.

Toon-Shading

Beim Toon-Shading wird ein beliebiges Shadingverfahren gewählt und diskretisiert. Hier wird das Toon-Shading auf das Phong-Shading angewendet. Beim Phong-Shading wird die Farbe des Fragments über drei Terme bestimmt: einen diffusen, einen ambienten und einen spekularen Anteil [44]. Der diffuse Anteil setzt sich aus der Farbe des Objektes, seinem Reflexionsfaktor gegenüber Licht, Lichtstärke und der Richtung des Lichts zusammen. Dabei werden Flächen stärker beleuchtet, die der Lichtquelle zugewandt sind. Der entscheidende Faktor hierbei ist die Zugewandtheit der Oberflächennormale und des Lichtvektors. Der ambiente Anteil ist eine Kombination aus globalem Umgebungslicht und der Farbe des Modells. Der spekulare Anteil beschreibt, wie das einfallende Licht vom Objekt reflektiert wird und dabei eventuell in das Auge des Beobachters fällt. Dies

zeichnet sich durch eine in der Lichtfarbe vorhandene Stellen auf der Oberfläche des Modells aus. Der entscheidende Faktor hierbei ist Zugewandtheit des Reflektionsvektor und des Blickrichtungsvektor. All diese Anteile sind kontinuierlich und machen einen physikalisch plausiblen Eindruck. Beim Toon-Shading werden diese Anteile diskretisiert, also in Stufen eingeteilt, indem die oben genannten Faktoren diskretisiert werden (siehe Formel 4.1). Die Anzahl der Stufen ist dabei beliebig. Das Toon-Shading wird hauptsächlich im Fragment-Shader umgesetzt und benötigt Materialeigenschaften des zu zeichnenden Objekts, sowie mindestens eine Lichtquelle. In Abbildung 5.10 wird das Toon-Shading dem Phong-Shading gegenüber gestellt.

$$t_d = \frac{\text{round}(t_c \cdot n)}{n} \quad (4.1)$$

t_d ist der diskretisierte Wert von der kontinuierlichen Zahl t_c und n ist die Anzahl der gewünschten Shadingstufen. Die Methode *round* rundet eine gegebene reale Zahl zur nächsten ganzen Zahl.

4.2.4 Kombination der Konzepte

Das Virtual Window arbeitet mit Alphawerten im Fragment-Shader, die Farbdistanzkodierung mit den RGB-Kanälen im Vertex-Shader. Beide Konzepte erlauben, wie oben bereits erwähnt, eine beliebige Shading-Technik. Hierfür ist bisher das Phong-Modell gewählt wurden. Eine Alternative dazu ist das in Abschnitt 4.2.3 vorgestellte Toon-Shading. Da die Konturen in einer separaten Textur zu finden sind, können diese nach dem Rendern der zu augmentierenden Daten einfach überlagert werden. Zusätzlich kann das Virtual Window auch auf die Konturen angewandt werden. Im nachfolgenden Kapitel werden einige dieser Kombinationen der Konzepte vorgestellt und diskutiert.

5 Ergebnisse

In diesem Kapitel werden die Ergebnisse der implementierten Konzepte und deren verschiedene Ausprägungen dargestellt. Hierbei werden die jeweiligen Parameter jedes Konzeptes gegenübergestellt und diskutiert. Die Konzepte funktionieren in stereoskopischer Laparoskopie und sind dort anwendbar, die nachfolgenden Ergebnisbilder sind jedoch zweidimensional dargestellt. Am Ende des Kapitels werden mögliche Kombinationen verschiedener Konzepte erprobt.

5.1 Virtual Window

Das Virtual Window lässt Veränderungen der Parameter Fenstergröße, Übergang zum Fenster und Transparenzen inner- und außerhalb des Fensters zu. Das Virtual Window bewegt sich mit dem laparoskopischen Werkzeug und ermöglicht so eine interaktive Visualisierung. Im Folgenden werden verschiedene Parameterkonfigurationen mit ihren jeweiligen Beispielbildern vorgestellt.

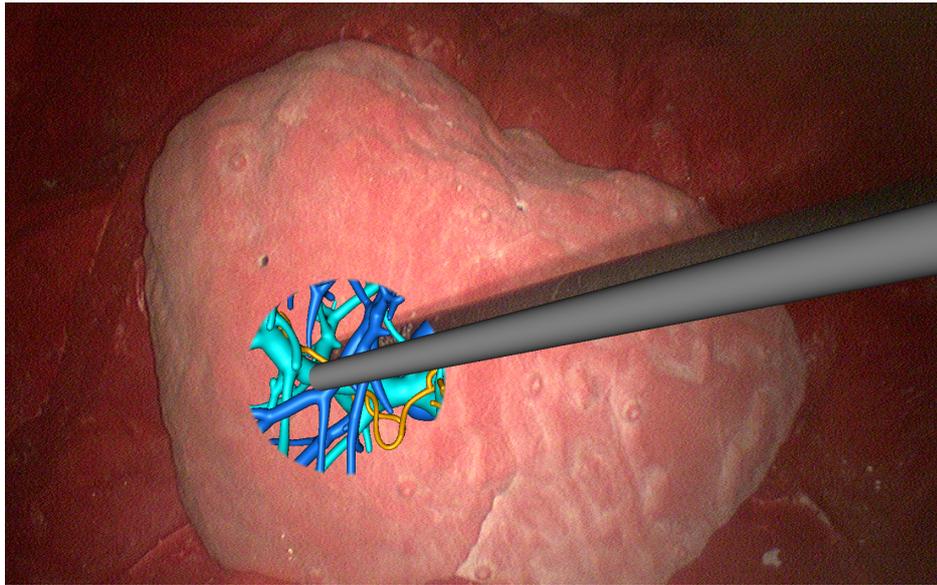


Abbildung 5.1: Ein Virtual Window mittlerer Größe und keinem Übergang. Innerhalb des Virtual Windows wird die Geometrie angezeigt, außerhalb wird sie ausgeblendet. Durch diese Darstellung wirkt das Fenster als Einblick in die Leber.

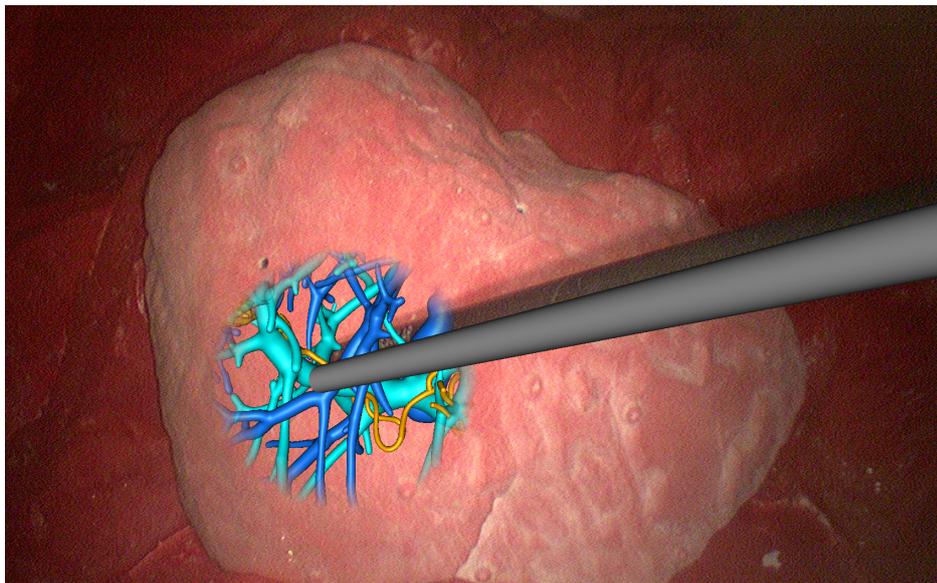


Abbildung 5.2: Durch Vergrößerung des Übergangsbereiches wird der nähere Kontext hervorgebracht. Dieser Effekt kann erwünscht sein, um eine bessere Orientierung zu ermöglichen, er kann jedoch auch die Wahrnehmung des Fensters als Einblick in die Daten reduzieren.

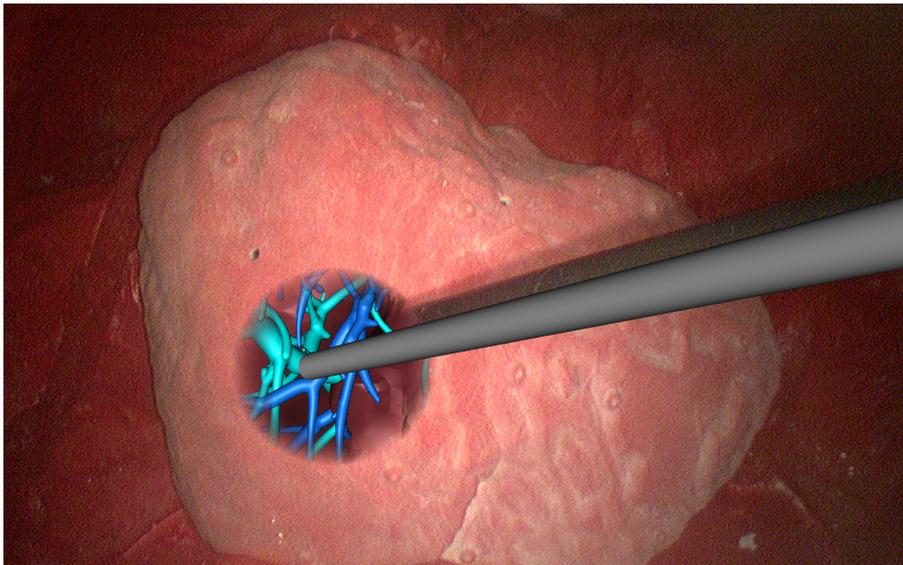


Abbildung 5.3: Ein interessanter Effekt tritt ein, wenn die Leberrückseite der augmentierten Daten innerhalb des Fensters in einer dunkleren Farbe dargestellt werden. Dadurch wirkt das Fenster verstärkt als Einblick in die Leber und erhöht somit weiterhin den Tiefeneindruck der augmentierten Daten.

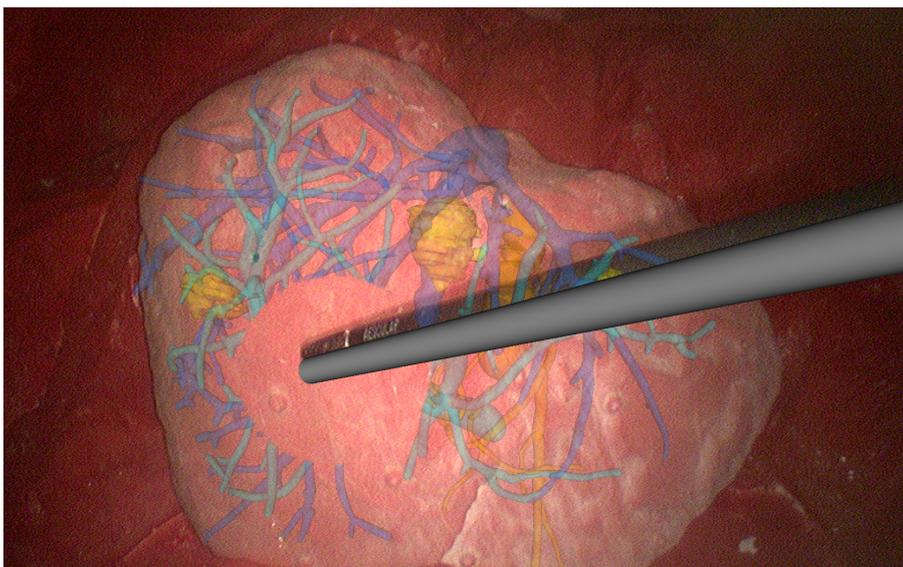


Abbildung 5.4: Das Virtual Window kann ebenfalls invers definiert werden. Die Abbildung zeigt ein Fenster mit voller Transparenz innerhalb und einer mittleren Transparenz außerhalb. Diese Konfiguration kann für die Resektion genutzt werden, um eine freie Sicht auf den Operationsort zu ermöglichen und um die Augmentierung der umliegenden Gefäße beizubehalten. Allerdings reduziert sich hierbei der wahrgenommene Tiefeneindruck drastisch.

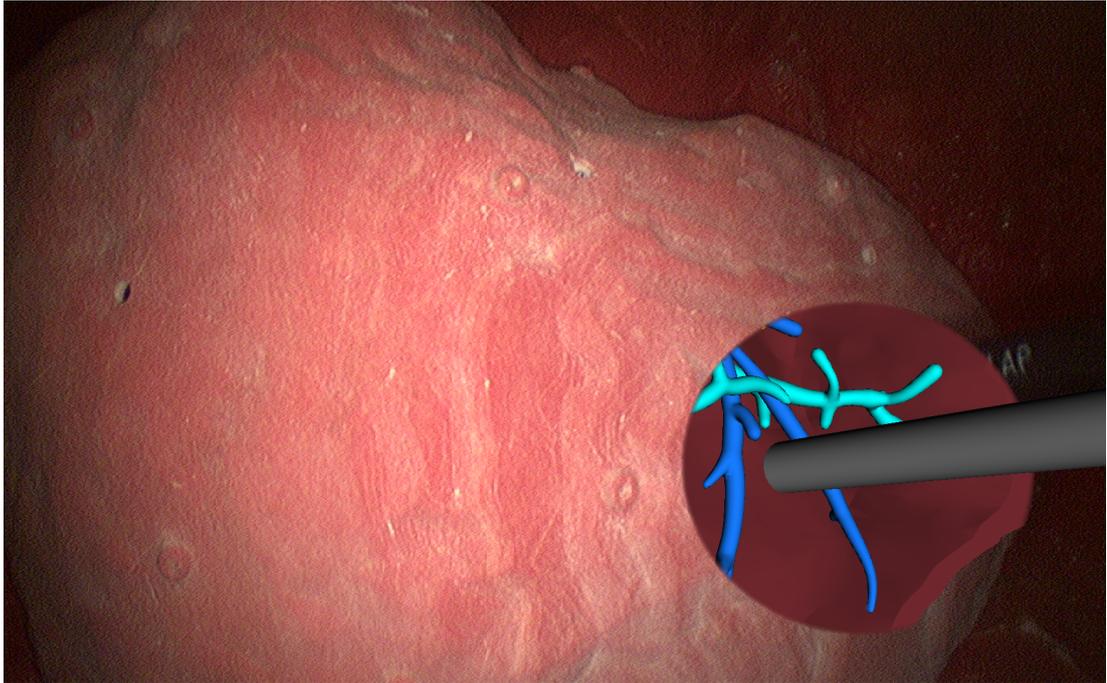


Abbildung 5.5: Bei einem zu großen Fenster oder bei der Positionierung des Fensters am Rand der augmentierten Geometrie, verliert die Visualisierung den Effekt einer korrekten Tiefe.

5.2 Konturen

Die Konturen können in den Parametern Farbe, Dicke und Transparenz gesteuert werden. Um die Effektivität der Konturen zu verbessern, sollten sie mit anderen Konzepten kombiniert werden (siehe Abschnitt 5.5).

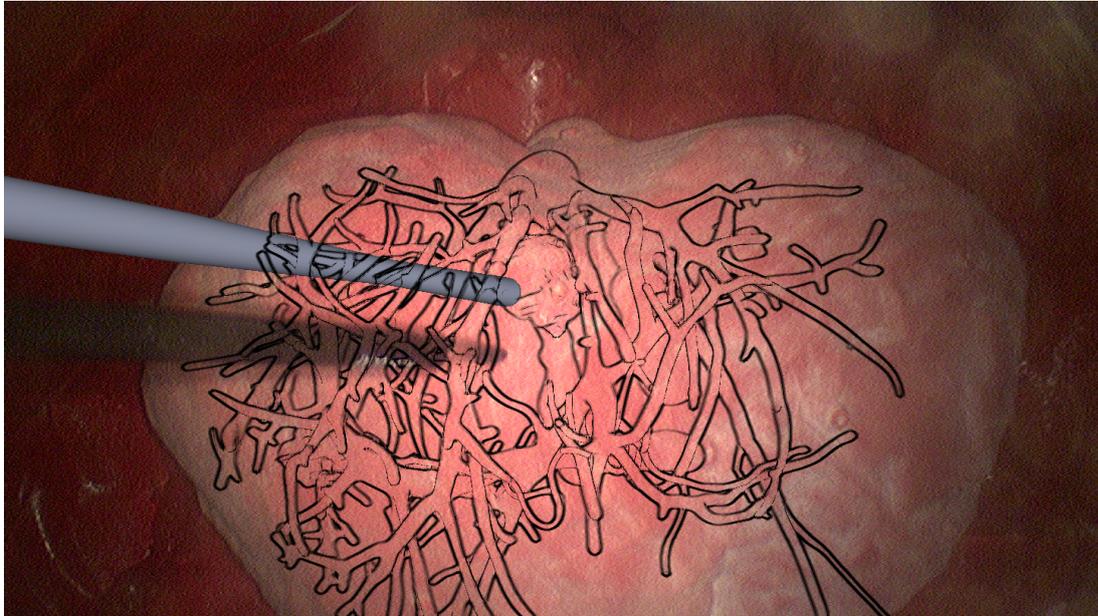


Abbildung 5.6: Durch die Überlagerung der Konturen auf das Bild wird das Werkzeug ebenfalls von den Konturen verdeckt. Dieser Effekt ist unerwünscht. In Kapitel 6 wird eine genauere Erklärung der Ursache und ein Lösungsansatz diskutiert.

5.3 Farbdistanz

Der Farbverlauf für die Farbdistanzkodierung kann mit Texturen oder mit einer direkten Definition von Distanzen auf Farben definiert werden. Der Farbverlauf kann hierbei diskret oder kontinuierlich sein. Im Folgenden werden einige mögliche Farbverläufe vorgestellt.

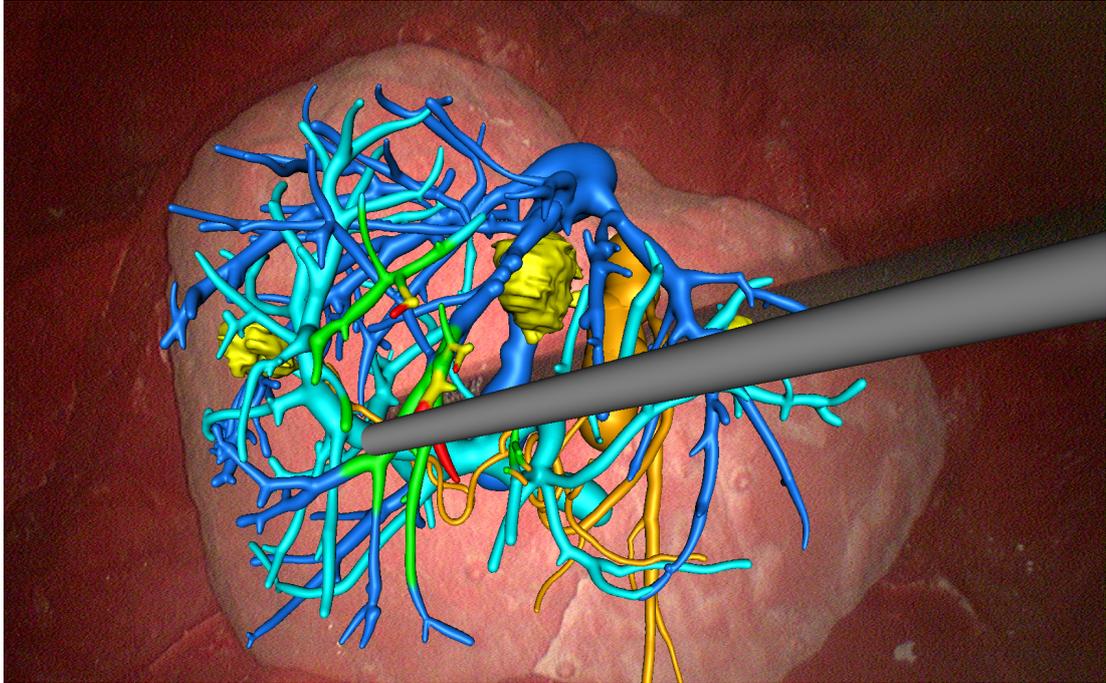


Abbildung 5.7: Die Abbildung zeigt eine diskrete farb-kodierte Distanzen zur Werkzeugspitze. Rot ist einem sehr hohem Risiko ausgesetzt, gelb einem erhöhten und grün nur einem geringeren. Für diesen Farbverlauf wurde die Möglichkeit zur direkten Definition der Farben auf Abstände genutzt.

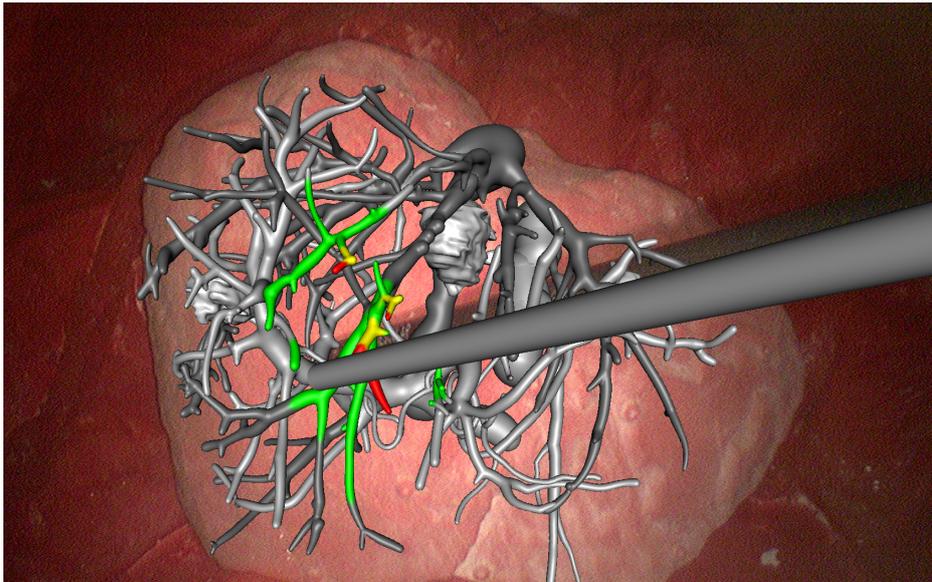


Abbildung 5.8: Die Verwendung von Farbskalen auf bereits farbige Geometrie hat gezeigt (siehe Abbildung 5.7), dass die Anzahl der Farben soweit ansteigt, dass die Unterscheidung der verschiedenen Farben erschwert wird. Deshalb wurde das Konzept um die Möglichkeit erweitert, die augmentierte Geometrie außerhalb des Risikobereiches auszugrauen.

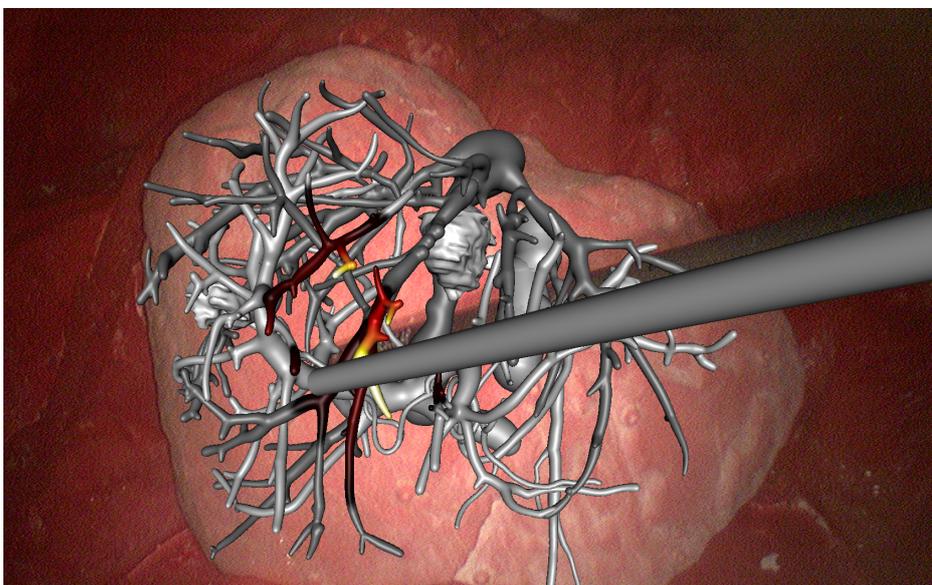
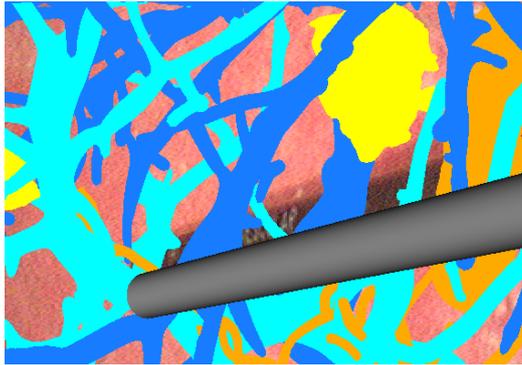


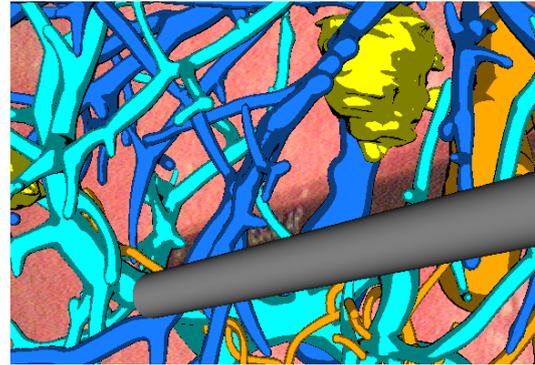
Abbildung 5.9: Die Abbildung zeigt die Verwendung einer Hitzefarbskala als Textur zur Visualisierung von kontinuierlichen Distanzen zur Werkzeugspitze. Außerhalb des Risikobereiches wurde die Geometrie ausgegraut.

5.4 Toon-Shading

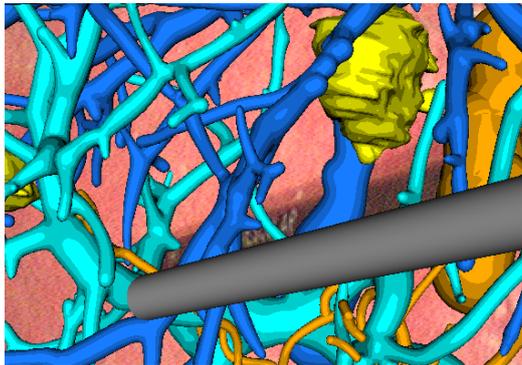
Die Anzahl der Diskretisierungsstufen ist beim Toon-Shading der einzige zu verstellende Parameter. Im Folgenden werden einige Stufen demonstriert und diskutiert.



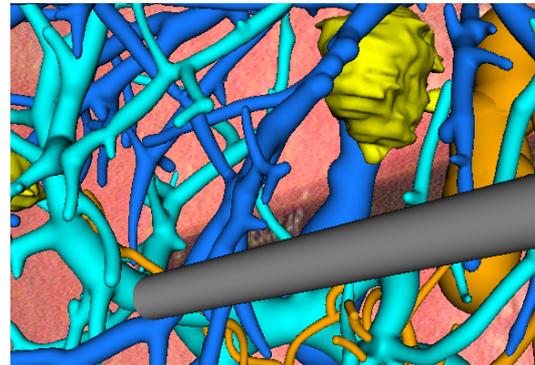
(a) Toon-Shading ohne Farbabstufung



(b) Toon-Shading mit zwei Farbabstufungen



(c) Toon-Shading mit fünf Farbabstufungen



(d) Phong-Shading

Abbildung 5.10: Die Abbildungen (a) bis (d) vergleichen verschiedene Stufen des Toon-Shadings mit dem Phong-Shading. Bei einer hohen Anzahl an Stufen nähert sich das Toon-Shading dem konventionellen Phong-Modell (d) an. Nach fünf Stufen lassen sich jedoch bereits kaum mehr Unterschiede wahrnehmen (c). Bei einer zu geringen Anzahl an Stufen degeneriert das gerenderte Modell zu einer flachen zweidimensional wahrgenommenen Darstellung (a). Zwei bis drei Stufen scheinen deshalb für das Toon-Shading sinnvoll zu sein (b).

5.5 Kombination der Konzepte

Ein großer Vorteil der ausgewählten Konzepte ist es, dass sie leicht miteinander kombinierbar sind. Nachfolgend werden einige dieser Kombinationen vorgestellt.

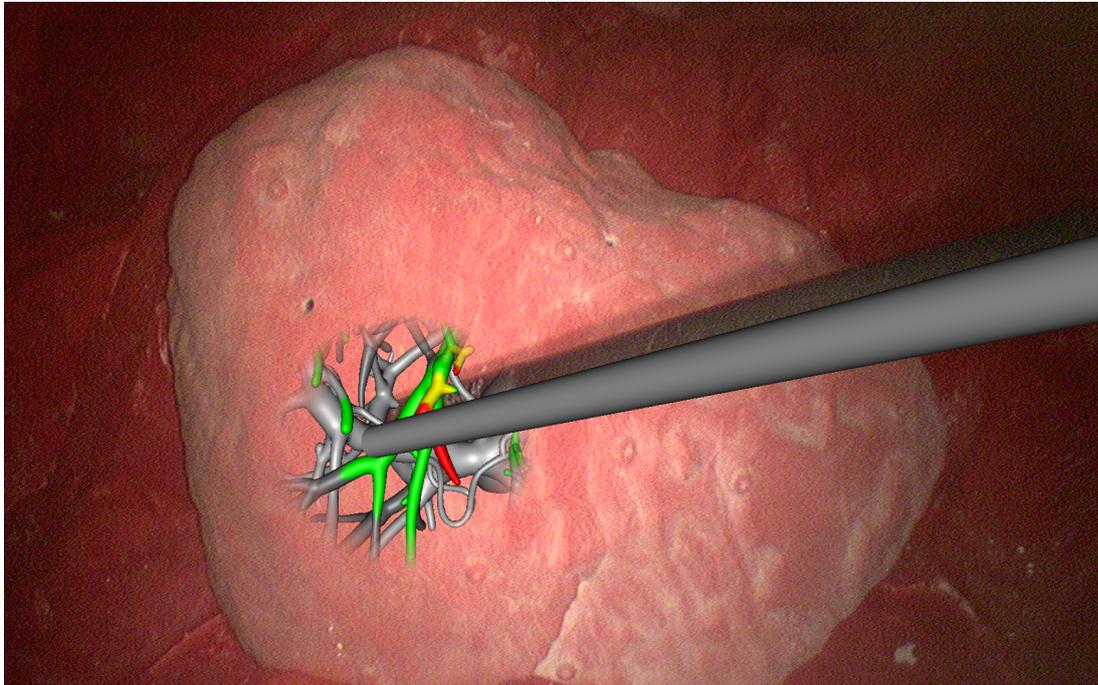


Abbildung 5.11: Die gleichzeitige Verwendung von farbkodierten Distanzen und dem Virtual Window mit Übergangsbereich. Durch diese Kombination wird der Fokus auf das Virtual Window gelegt und die Orientierung des laparoskopischen Werkzeugs innerhalb der Geometrie erhöht und der Risikobereich hervorgehoben.

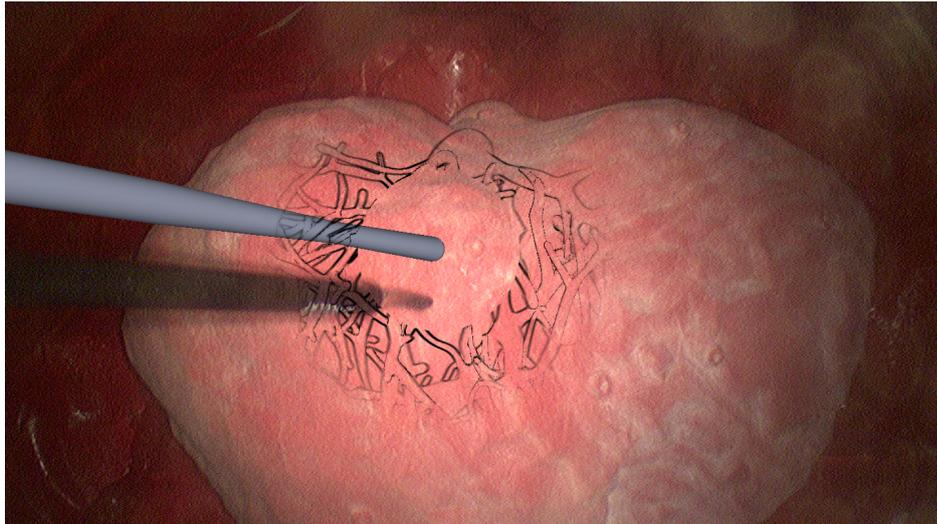


Abbildung 5.12: Alternativ zum inversen Fenster mit einer semitransparenten Darstellung der Patientendaten außerhalb (siehe Abbildung 5.4) kann das inverse Virtual Window auf die Konturen angewandt und die Patientendaten selbst ausgeblendet werden. Hierdurch wird die nähere Umgebung abgeschwächt dargestellt und eine freie Sicht auf den Operationsort ermöglicht.

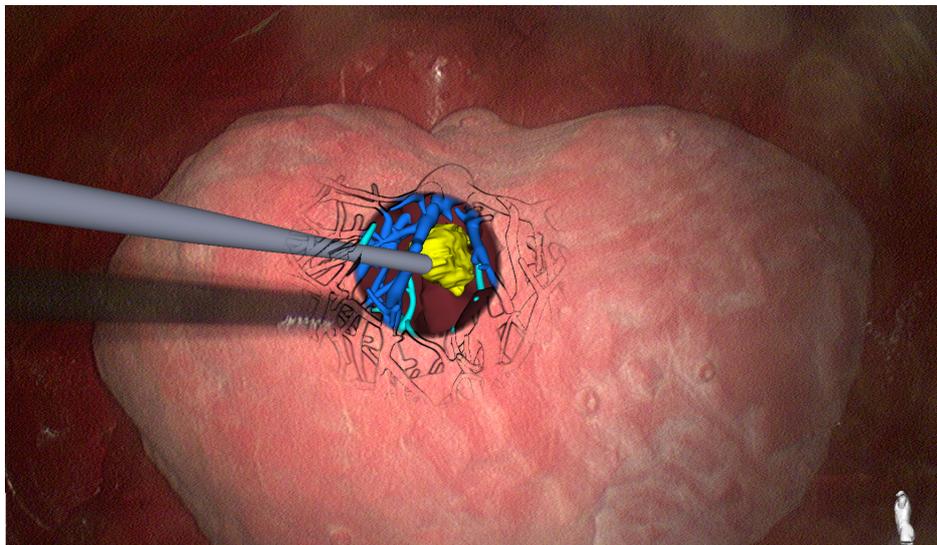


Abbildung 5.13: Die Verwendung eines normalen Virtual Windows in Kombination mit Konturen, auf die ebenfalls das Virtual Window angewandt wurde. Hierdurch bleibt die nähere Umgebung leicht sichtbar, um die Orientierung zu erhöhen.

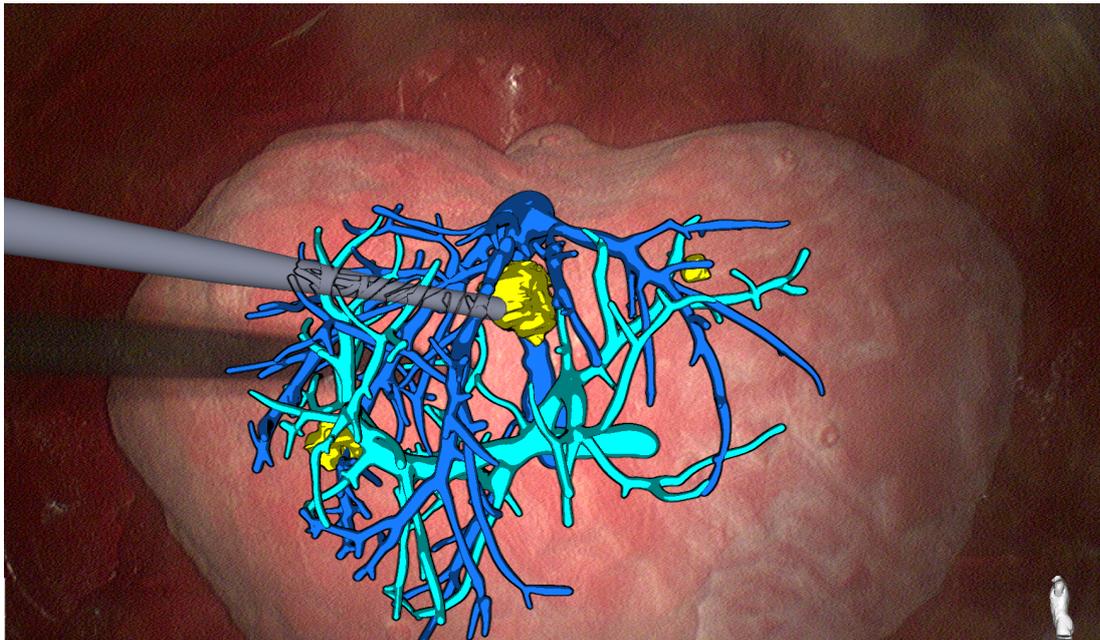


Abbildung 5.14: In Kombination mit dem Toon-Shading erhöhen die Konturen den Eindruck vom illustrativen Rendering.

6 Zusammenfassung

Diese Arbeit befasste sich mit der Konzeption von AR Visualisierungen von virtuellen Patientendaten in der stereoskopischen Laparoskopie. Das Ziel war es, die Tiefenwahrnehmung dieser Daten im Laparoskopiebild zu verbessern. Hierfür wurden verschiedene Konzepte implementiert, die anhand einer Anforderungsanalyse entworfen wurden. Das Virtual Window eignet sich gut, um durch Verdeckung einen starken Tiefeneindruck zu erzeugen. Durch Veränderungen verschiedener Parameter kann das Fenster angepasst werden, um weiteren Kontext anzuzeigen oder um den Fokus auf einen bestimmten Ort zu lenken. Das Konzept der Farbdistanzkodierung färbt die Geometrie anhand des Abstandes zu der Spitze eines Laparoskopiewerkzeugs ein, sodass die Orientierung im dreidimensionalen Raum erhöht und Risikobereiche deutlich hervorgehoben werden. Zur Reduzierung der Komplexität durch die Beleuchtung der augmentierten Daten wurde Toon-Shading eingeführt, welches die Anzahl der verschiedenen Farbabstufungen reduziert. Zuletzt wurden Konturen implementiert, um die illustrative Darstellung der augmentierten Daten zu erhöhen. Ebenso eignet sich dieses Konzept zur Darstellung von Verdeckung des Kontextes durch das Virtual Window. Die Ergebnisse lassen den Schluss zu, dass die Tiefenwahrnehmung durch die implementierten Konzepte verbessert werden kann. Das Virtual Window ist hierbei das vielversprechendste Konzept, da die Verdeckung einer der stärksten Tiefenhinweise ist [22] und hierdurch der Eindruck entsteht, dass sich die Patientendaten innerhalb der Leberaufnahmen befinden. Jedoch haben die Ergebnisse gezeigt, dass die Kombination mehrerer Konzepte nicht gleichbedeutend mit einer Verstärkung der Tiefenwahrnehmung einhergeht. Um einer Überladung des Bildes entgegenzuwirken, sollten nur bestimmte Konzepte mit geeigneten Parametern miteinander kombiniert werden. Eine weitere Möglichkeit einer Überladung entgegenzuwirken, ist die Reduzierung der Komplexität der Modelle selbst und die Verbesserung der vorgestellten Konzepte.

Wie bereits geschildert eignen sich Konturen allein nicht zur Erhöhung der Tiefenwahrnehmung. Eine mögliche Weiterentwicklung könnte die Anreicherung der Konturen mit Tiefenhinweisen wie beispielsweise Strichstärke, Farbe und Strichart sein.

Wie in Abbildung 5.6 sichtbar, werden die Konturen vor dem virtuellen Werkzeug gezeichnet. Dies liegt an dem Aufbau des Szenengraphen. Das bildschirmfüllende Rechteck wird am Ende des Szenengraphen gezeichnet und hat keine Tiefeninformationen mehr, mit denen es Werte im Tiefenbuffer vergleichen kann. Eine Möglichkeit dieses Problem in der Zukunft zu umgehen, wäre die Nutzung eines Geometry-Shaders zur Berechnung der Konturen. Der Geometry-Shader erfordert kein Multiple Pass Rendering, da er Informationen über lokale Geometrie besitzt. Jedoch unterstützt Coin3D in der verwendeten Version den Geometry-Shader nicht.

Der Radius des Virtual Windows ist in Bildschirmkoordinaten definiert (siehe Abschnitt 4.2.1). Dies hat den Nachteil, dass das Fenster seine relative Größe, abhängig von der Entfernung der Kamera zum Fenster, verändert. Die Definition des Fensters als Kreisform in Millimeter wäre besser. Jedoch hat dieser Ansatz ebenfalls das Problem der wechselnden Größe, da das Fenster eine Position in der virtuellen Welt benötigt. Bei einer Verankerung des Virtual Windows an die Werkzeugspitze, verändert das Fenster seine Größe mit der Tiefe des Werkzeugs. Bei einer konstanten Position verändert das Virtual Window seine Größe nicht, erlaubt jedoch auch keine sinnvolle Interpretation des Radius. Ein möglicher Ansatz, um diese Punkte zu verbessern wird im nächsten Absatz beschrieben.

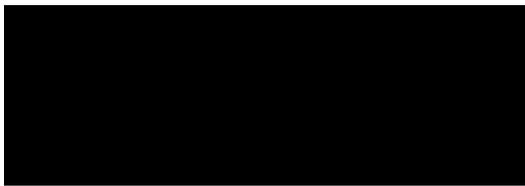
Das Virtual Window kann um den Tiefenhinweis der Bewegungsparallaxe (siehe Abschnitt 2.3.2) erweitert werden, indem das Fenster fest auf der virtuellen Leberoberfläche projiziert wird, um sich an die Form der Leber anzupassen. Durch Schwenken der Kamera würde sich das Fenster schneller bewegen, als die dahinterliegenden Strukturen und so die Tiefenwahrnehmung verbessern. Hierzu müsste der Tiefenbuffer als Textur vorhanden und auslesbar sein. Diese Erweiterung steht gegensätzlich zu der Anforderung, dass sich das Fenster mit einem laparoskopischen Werkzeug bewegen soll. Eine Möglichkeit zur Umschaltung beider Ansätze wäre nötig. Jedoch eliminiert diese Erweiterung das Problem der fehlenden Interpretierbarkeit der Fenstergröße (siehe vorherigen Abschnitt). Um das Verständnis des so projizierten Virtual Windows zu verbessern, sollte der Rand des Fensters durch Linien in einer anderen Farbe hervorgehoben werden.

Um die abschließende Anwendbarkeit der Konzepte zu testen, ist die Einbindung von Experten und die ausführliche Untersuchung der Visualisierungen auf ihre Effektivität durch Nutzerstudien erforderlich.

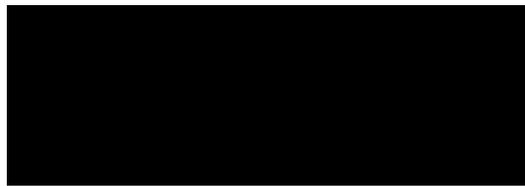
A Anhang: Laparoskopische Werkzeuge

Die Mehrzahl aller verwendeten Werkzeuge bei laparoskopische Eingriffe haben einen Durchmesser von 5 mm oder 10 mm, die Schaftlänge beträgt hierbei zwischen 30 cm und 35 cm. Im Allgemeinen unterscheidet man zwischen wiederverwendbaren und Einmalinstrumenten.

1 Veress-Nadel: Die Veress-Nadel (siehe Abbildung A.1) wird zur Schaffung des Pneumoperitoneum der Bauchhöhle verwendet. Sie besteht aus einer inneren Kanüle, die seitlich an ihrem Ende ein Loch hat, durch das das Gas insuffliert werden kann. Bei Druck zieht sich die Kanüle in die äußere, scharfe Nadel zurück, sodass die Nadel durch das Gewebe stechen kann. Beim Erreichen des Hohlraumes des Bauches springt die Kanüle wieder vor [12, 45].



(a)



(b)

Abbildung A.1: Abbildung (a) zeigt eine Veress-Nadel mit vorgeschobener Kanüle und Abbildung (b) zeigt eine Veress-Nadel mit zurückgeschobenen Kanüle (Quelle: [12]).

2 Trokar: Um die verschiedenen Werkzeuge sicher in die Bauchhöhle zu führen, werden sogenannte Trokare verwendet (siehe Abbildung: A.2). Ein Trokar ist ein spitzes Werkzeug mit einer Hülse (Tubus), welche zusammen nach der Schaffung des Pneumoperitoneum durch Punktion in den Bauchraum eingeführt werden. Nachdem die Hülse in den Bauchhöhle vorgeschoben wurde, kann der Trokar entfernt werden. Damit durch die Hülse das insufflierte Gas nicht entweichen kann, besitzen die Hülsen einen Ventilmechanismus, der je nach Ausführung manuell oder durch Druck geöffnet werden kann. Die Hülsen werden dann im weiteren Verlauf als Zugangspunkte für weitere Werkzeuge verwendet. Die Hülsen sind in der Regel 1 mm größer als die einzuführenden Werkzeuge [12, 45].

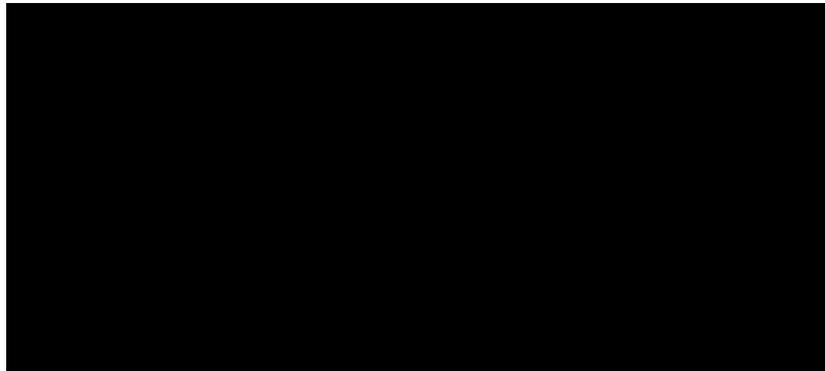
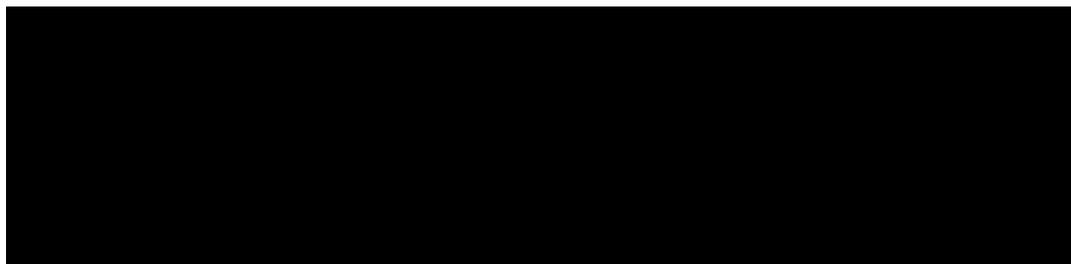
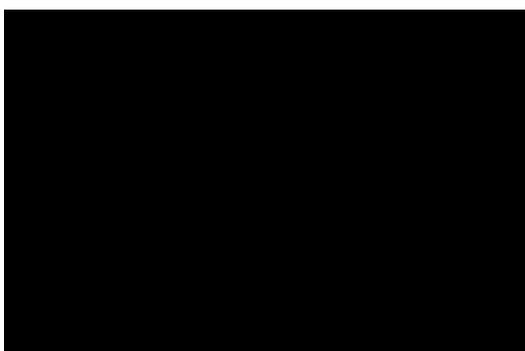


Abbildung A.2: Die Hülse des Trokars wird als Zugangspunkt für weitere Werkzeuge verwendet (Quelle: [12]).

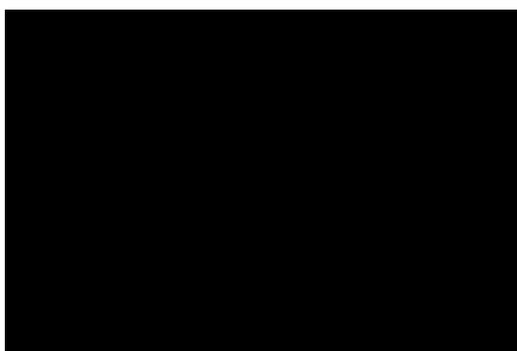
3 Optik: Am Schaftende der Optik befindet sich eine Kamera, welche dem Chirurgen den Blick auf das Operationsfeld ermöglicht. Über ein Lichtleitkabel wird das Innere der Bauchhöhle beleuchtet. Es gibt Optiken in verschiedenen Ausprägungen, die verschiedene Blickwinkel bieten. Am häufigsten sind 0° und 30° Optiken (siehe Abbildung A.3) verbreitet, es existieren jedoch auch 0° bis 120° winkelbare Optiken [12]. Eine Optik für laparoskopische Eingriffe wird häufig auch Laparoskop genannt [46]. Es gibt ebenfalls Optiken mit zwei Kameras, um eine stereoskopische Sicht zu ermöglichen. Die 3D-Bilder werden dann über ein Head-Mounted Display oder über einen speziellen 3D-Monitor ausgegeben. Votanopoulos et al. [47] und Kong et al. [48] verglichen 2D- und 3D-Laparoskopie und kamen zu dem Schluss, dass stereoskopische Laparoskopie die Performance der Chirurgen durch erhöhte Tiefenwahrnehmung verbesserte.



(a)



(b)



(c)

Abbildung A.3: Abbildung (a) zeigt eine 10 mm Optik mit Anschluss für das Lichtleitkabel. Auf Abbildung (b) ist der Kopf einer 0° Optik und auf Abbildung (c) der Kopf einer 30° Optik zu sehen (Quelle: [12]).

Abkürzungsverzeichnis

AR	Augmented Reality
AV	Augmented Virtuality
CAS	Computer-assistierte Chirurgie
GB	Gigabyte
GHz	Gigahertz
GLSL	OpenGL Shading Language
IDE	integrierte Entwicklungsumgebung
NPR	Non-photorealistic Rendering
VR	Virtual Reality

Abbildungsverzeichnis

1.1	Aktueller Stand der Visualisierung	2
2.1	Lebersegmente	7
2.2	Laparoskopie und Laparotomie im Vergleich	8
2.3	Reality Virtuality Continuum	10
2.4	Akkommodation	13
2.5	Verdeckung	14
2.6	Schattenillusion	14
2.7	Texturgradient	15
2.8	Virtual Window von Bichlmeier et al.	17
2.9	Konzept des minimalsten Abstandes	18
2.10	Methode zur virtuellen Leberdeformation	18
3.1	Das CAS-ONE Liver System	20
3.2	Konzeptentwurf: normale Sicht	22
3.3	Konzept Virtual Window	24
3.4	Diskrete und kontinuierliche Hitzefarbskala	25
3.5	Konzept der Farbdistanzkodierung	26
3.6	Verschiedene Feature Lines	27
3.7	Konzept der Konturen	27
3.8	Konventionelles Rendering und Toon-Shading	28
3.9	Konzept des Toon-Shadings	28
4.1	Szenengraph für eine beleuchtete Kugel	30
4.2	Szenengraph des Systems	33
4.3	Normalen der Lebergefäße	38
5.1	Virtual Window auf Lebergefäße	42
5.2	Virtual Window mit Übergangsbereich	42

5.3	Virtual Window und Leberrückwand	43
5.4	Inverses Virtual Window	43
5.5	Virtual Window Effektverlust	44
5.6	Ergebnis: Konturen	45
5.7	Diskrete farbkodierte Distanz der Lebergefäße	46
5.8	Ausgegraute Geometrie und diskrete Farbdistanzkodierung	47
5.9	Ausgegraute Geometrie und kontinuierliche Farbdistanzkodierung	47
5.10	Vergleich von Phong- und Toon-Shading	48
5.11	Kombination von Virtual Window mit farbkodierter Distanz	49
5.12	Konturen und inverses Virtual Window	50
5.13	Kombination von Virtual Window und Konturen	50
5.14	Kombination von Virtual Window und Toon-Shading	51
A.1	Veress-Nadel	57
A.2	Trokar mit Hülse	58
A.3	Laparoskop	59

Literaturverzeichnis

- [1] OLESCH, J.: *Bildregistrierung für die navigierte Chirurgie: Spezialisierte Ansätze zur Anwendung in der navigierten Leberchirurgie*. Aktuelle Forschung Medizintechnik – Latest Research in Medical Engineering. Springer Fachmedien Wiesbaden, 2014.
- [2] HUGHES-HALLETT, ARCHIE, ERIK K MAYER, HANI J MARCUS, THOMAS P CUNDY, PHILIP J PRATT, ARA W DARZI und JUSTIN A VALE: *Augmented reality partial nephrectomy: examining the current status and future perspectives*. 2014.
- [3] LANG, F. und P. LANG: *Basiswissen Physiologie*. Springer-Lehrbuch. Springer Berlin Heidelberg, 2007.
- [4] ZILLES, K. und B. TILLMANN: *Anatomie*. Springer-Lehrbuch. Springer Berlin Heidelberg, 2011.
- [5] PREIM, B. und C.P. BOTHA: *Visual computing for medicine : theory, algorithms, and applications*. The Morgan Kaufmann Series in Computer Graphics. Elsevier Science, 2nd ed. Auflage, 2014.
- [6] *Online Medizinlexikon: Tumore*. <http://flexikon.doccheck.com/de/Tumor>. Letzter Zugriff: 26.08.2016.
- [7] GNANT, M. und P.M. SCHLAG: *Chirurgische Onkologie: Strategien und Standards für die Praxis*. Springer, 2009.
- [8] GERBES, PROF. DR. ALEXANDER: *Lebererkrankungen: Superorgan unter Dauerstress*. <http://docmed.tv.vitanet.de/lebererkrankungen/video-krankheitsbilder>. Letzter Zugriff: 26.08.2016.

- [9] *Online Medizinlexikon: Laparotomie.* <http://flexikon.doccheck.com/de/Laparotomie>. Letzter Zugriff: 26.08.2016.
- [10] *Online Medizinlexikon: Laparoskopie.* <http://flexikon.doccheck.com/de/Laparoskopie>. Letzter Zugriff: 26.08.2016.
- [11] ALAMY: *Bildquelle: Vergleich von Laparoskopie und Laparotomie.* <http://c8.alamy.com/comp/ADW7WA/laparoscopic-cholecystectomy-gallbladder-removal-surgery-with-conversion-ADW7WA.jpg>. Letzter Zugriff: 26.08.2016.
- [12] CARUS, T.: *Operationsatlas Laparoskopische Chirurgie: Indikationen - Operationsablauf - Varianten - Komplikationen.* Springer Berlin Heidelberg, 2014.
- [13] *Online Medizinlexikon: Tumorresektion.* <http://flexikon.doccheck.com/de/Tumorresektion>. Letzter Zugriff: 26.08.2016.
- [14] VETTER, MARCUS, PETER HASSENPFUG, MATTHIAS THORN, GERALD-P GLOMBITZA, WOLFRAM LAMADÉ, GÖTZ MARTIN RICHTER, HANS-PETER MEINZER et al.: *Navigation in der Leberchirurgie-Anforderungen und Lösungsansatz.* In: *Rechner-und sensorgestützte Chirurgie, Proceedings zum Workshop*, Seiten 92–102. GI, 2001.
- [15] MILGRAM, PAUL, HARUO TAKEMURA, AKIRA UTSUMI und FUMIO KISHINO: *Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum.* Seiten 282–292, 1994.
- [16] DÖRNER, R., W. BROLL, P. GRIMM und B. JUNG: *Virtual und Augmented Reality (VR / AR) Grundlagen und Methoden der Virtuellen und Augmentierten Realität.* Springer Vieweg, 2013.
- [17] AZUMA, RONALD T.: *A survey of augmented reality.* Presence: Teleoperators and Virtual Environments, 1997.
- [18] FURHT, B.: *Handbook of Augmented Reality.* Springer New York, 2011.
- [19] WENDT, MIKE: *Allgemeine Psychologie - Wahrnehmung.* Hogrefe, 1. Aufl. Auflage, 2014.

- [20] GOLDSTEIN, E BRUCE: *Wahrnehmungspsychologie der Grundkurs*. Springer, 9. Aufl. Auflage, 2015.
- [21] WIKIPEDIA: *Bildquelle: Akkommodation*. https://upload.wikimedia.org/wikipedia/commons/thumb/8/81/Focus_in_an_eye.svg/800px-Focus_in_an_eye.svg.png. Letzter Zugriff: 26.08.2016.
- [22] CUTTING, JAMES E. und PETER M. VISHTON: *Chapter 3 - Perceiving Layout and Knowing Distances: The Integration, Relative Potency, and Contextual Use of Different Information about Depth*. In: EPSTEIN, WILLIAM und SHEENA ROGERS (Herausgeber): *Perception of Space and Motion*, Handbook of Perception and Cognition, Seiten 69 – 117. Academic Press, San Diego, 1995.
- [23] FLICKR: *Bildquelle: Texturgradient*. https://c2.staticflickr.com/6/5441/9363321139_fcdcc6f646_b.jpg. Letzter Zugriff: 26.08.2016.
- [24] KRUIJFF, ERNST, J. EDWARD SWAN II und STEVEN FEINER: *Perceptual issues in augmented reality revisited*. 2010.
- [25] DRASCIC, DAVID und PAUL MILGRAM: *Perceptual issues in augmented reality*. In: *Electronic Imaging: Science & Technology*, Seiten 123–134. International Society for Optics and Photonics, 1996.
- [26] NICOLAU, STÉPHANE, LUC SOLER, DIDIER MUTTER und JACQUES MARESCAUX: *Augmented reality in laparoscopic surgical oncology*. *Surgical oncology*, 20(3):189–201, 2011.
- [27] SIELHORST, TOBIAS, CHRISTOPH BICHLMEIER, SANDRO MICHAEL HEINING und NASSIR NAVAB: *Depth Perception – A Major Issue in Medical AR: Evaluation Study by Twenty Surgeons*. Springer Berlin Heidelberg, 2006.
- [28] BICHLMEIER, CHRISTOPH und NASSIR NAVAB: *Virtual Window for Improved Depth Perception in Medical AR*. MICCAI Society, 2006.
- [29] CHOI, HYUNSEOK, BYUNGHYUN CHO, KEN MASAMUNE, MAKOTO HASHIZUME und JAESUNG HONG: *An effective visualization technique for depth perception in augmented reality-based surgical navigation*. The International

- Journal of Medical Robotics and Computer Assisted Surgery, 12(1):62–72, 2016. RCS-14-0031.R5.
- [30] HAOUCHINE, NAZIM, JÉRÉMIE DEQUIDT, MARIE-ODILE BERGER und STÉPHANE COTIN: *Deformation-based augmented reality for hepatic surgery*. Studies in health technology and informatics, 184, 2013.
- [31] CASCINATION AG: *CAS-ONE Liver*. <http://www.cascination.com/products/casone-liver/>. Letzter Zugriff: 26.08.2016.
- [32] CASCINATION: *Bildquelle: CAS-ONE Liver System*. <http://www.cascination.com/wp-content/uploads/2013/05/System-Liver-e1369486091980.png>. Letzter Zugriff: 26.08.2016.
- [33] POLLMANN, S.: *Allgemeine Psychologie*. UTB / UTB. Reinhardt, 2008.
- [34] DIXON, BENJAMIN J, MICHAEL J DALY, HARLEY HL CHAN, ALLAN VESCAN, IAN J WITTERICK und JONATHAN C IRISH: *Inattentive blindness increased with augmented reality surgical navigation*. American journal of rhinology & allergy, 28(5):433–437, 2014.
- [35] HANSEN, CHRISTIAN, JAN WIEFERICH, FELIX RITTER, CHRISTIAN RIEDER und HEINZ-OTTO PEITGEN: *Illustrative visualization of 3D planning models for augmented reality in surgery*. International journal of computer assisted radiology and surgery, 5(2):133–141, 2010.
- [36] JUDD, TILKE, FRÉDO DURAND und EDWARD H. ADELSON: *Apparent ridges for line drawing*. ACM Trans. Graph., 26(3):19, 2007.
- [37] WIKIPEDIA: *Bildquelle: Toon Shading*. https://upload.wikimedia.org/wikipedia/commons/1/1a/Toon_Shader.jpg. Letzter Zugriff: 26.08.2016.
- [38] *Microsoft Visual Studio 2013*. <https://msdn.microsoft.com/de-de/library/dd831853%28v=vs.120%29.aspx>. Letzter Zugriff: 26.08.2016.
- [39] *Dokumentation von Coin3D*. <https://bitbucket.org/Coin3D/coin/wiki/Home>. Letzter Zugriff: 26.08.2016.

- [40] *OpenGL Dokumentation von GLSL*. <https://www.opengl.org/documentation/glsl/>. Letzter Zugriff: 26.08.2016.
- [41] *Qt Homepage*. <http://www.qt.io/>. Letzter Zugriff: 26.08.2016.
- [42] *Informationsseite MeVisLab*. <http://www.mevislab.de/mevislab/>. Letzter Zugriff: 26.08.2016.
- [43] ROST, RANDI J.: *OpenGL(R) Shading Language (2Nd Edition)*. Addison-Wesley Professional, 2005.
- [44] FOLEY, J.D.: *Grundlagen der Computergraphik: Einführung, Konzepte, Methoden*. Addison-Wesley, 1994.
- [45] SAXENA, A.K. und M.E. HÖLLWARTH: *Essentials of Pediatric Endoscopic Surgery*. Springer Berlin Heidelberg, 2008.
- [46] KIRSCHNIAK, A. und U. DREWS: *Endoskopisch-Chirurgische Anatomie: Topographie für die Laparoskopie, Gastroskopie und Coloskopie*. Springer Berlin Heidelberg, 2011.
- [47] VOTANOPOULOS, KONSTANTINOS, F. CHARLES BRUNICARDI, JOHN THORNBY und CHARLES F. BELLOWS: *Impact of Three-Dimensional Vision in Laparoscopic Training*. World Journal of Surgery, 32(1):110–118, 2008.
- [48] KONG, SEONG-HO, BYUNG-MO OH, HONGMAN YOON, HYE SEONG AHN, HYUK-JOON LEE, SUN GEUN CHUNG, NORIO SHIRAIISHI, SEIGO KITANO und HAN-KWANG YANG: *Comparison of two- and three-dimensional camera systems in laparoscopic performance: a novel 3D system with one camera*. Surgical Endoscopy, 24(5):1132–1143, 2010.