

gernot stangl

on no body

Die Gestalt des
Leeren Körpers

Deutsche Fassung:

Beschluss der Curricula-Kommission für Bachelor-,
Master- und Diplomstudien vom 10. 11. 2008
mit Genehmigung des Senates am 1. 12. 2008

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vor-
liegende Arbeit selbstständig verfasst, an-
dere als die angegebenen Quellen /
Hilfsmittel nicht benutzt, und die den be-
nutzten Quellen wörtlich und inhaltlich ent-
nommenen Stellen als solche kenntlich
gemacht habe.

Graz, am 11. Jänner 2011

(Unterschrift)

Englische Fassung:

Statutory Declaration

I declare that I have authored this thesis
independently, that I have not used other
than the declared sources / resources, and
that I have explicitly marked all material
which has been quoted either literally or by
content from the used resources.

Graz, 11 January 2011

(signature)

Dank

An Professor Urs Hirschberg

An Dr. Michael Stadler, der den Anstoß für diese Arbeit gegeben hat

An Prof. Klaus Bollinger und Univ. Lekt. Mag. Florian Medicus für ihre freundliche Unterstützung

An Prof. Peter Schreibmayer, der verhindert hat, dass dieses Studium "Endless" wird.

An die Architekturzeichensäle der TU Graz

An meine Eltern und meine Familie

An Julia, die mir diese Arbeit ermöglicht hat

Für Max und Antonia

on no body

Die Gestalt des
Leeren Körpers

Diplomarbeit

zur Erlangung des akademischen
Grades eines Diplom-Ingenieurs
Studienrichtung Architektur

Gernot Stangl

Technische Universität Graz
Erzherzog Johann Universität
Fakultät für Architektur

Betreuer:

Urs Hirschberg

Univ.-Prof. Dipl.-Arch. Dr.sc.ETH

Institut für Architektur und Medien

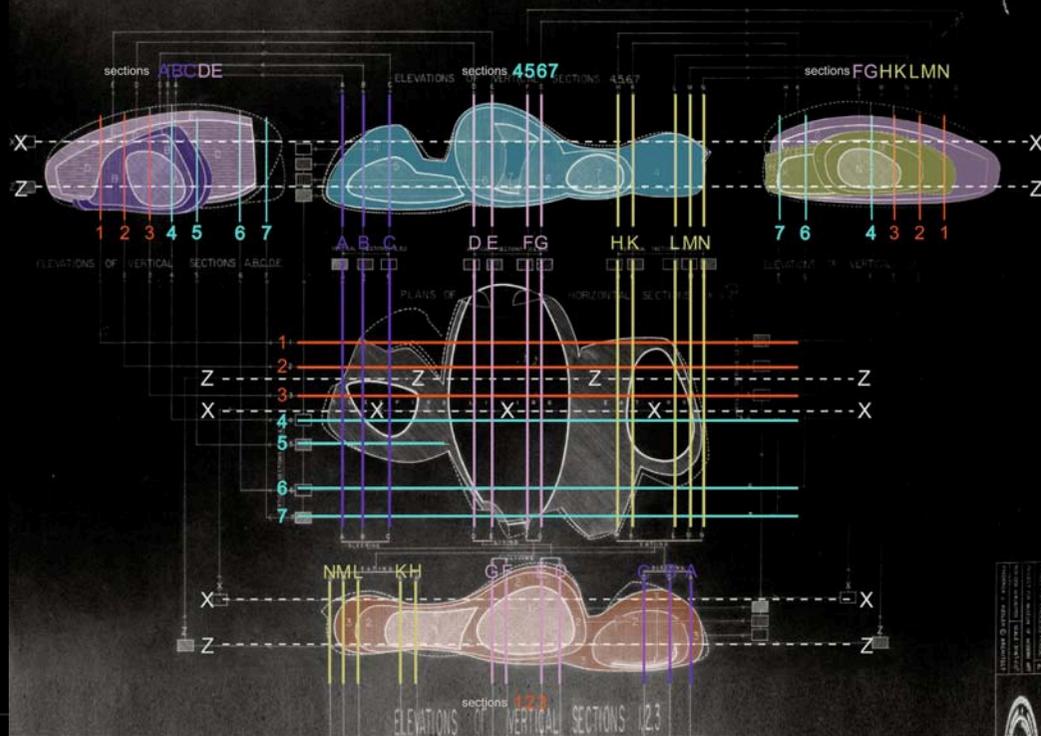
Im Jänner 2011

Die Struktur dieses Buches

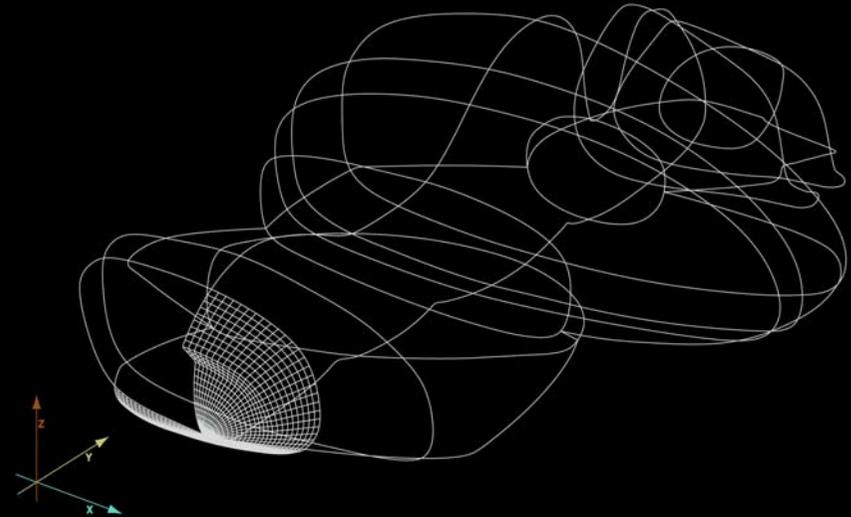
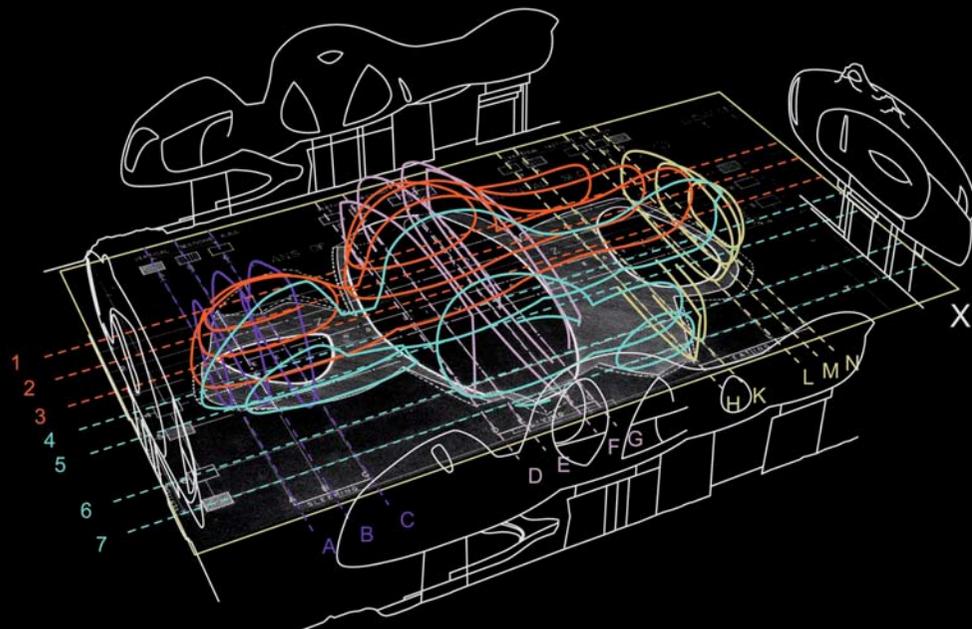
Spätestens jetzt ist dieses Buch aufgeklappt worden und hat damit bereits einen Eindruck hinterlassen. Eine für Bücher sehr wichtige Drehachse wurde von der Y- in die X-Richtung gelegt. Der Preis dafür ist vermutlich, dass seine "Haltbarkeit" etwas leidet - der Autor empfiehlt, es eher wie einen Laptop aufzustellen. In Analogie zum Laptop teilt sich das Buch in die eine Hälfte, die stärker grafisch aufbereiteten Inhalten vorbehalten ist und eine Basis, die stärker textlastig ist. Über weite Strecken ko-existieren beide, an einigen Stellen nehmen sie aufeinander Bezug - es wäre dann möglich, von einem zum anderen Pfad zu springen und dort weiterzugehen. Die Form des Buches hätte dann, falls das funktioniert, ein wenig von dem angenommen, was im Inhalt verhandelt wird.

Inhalt

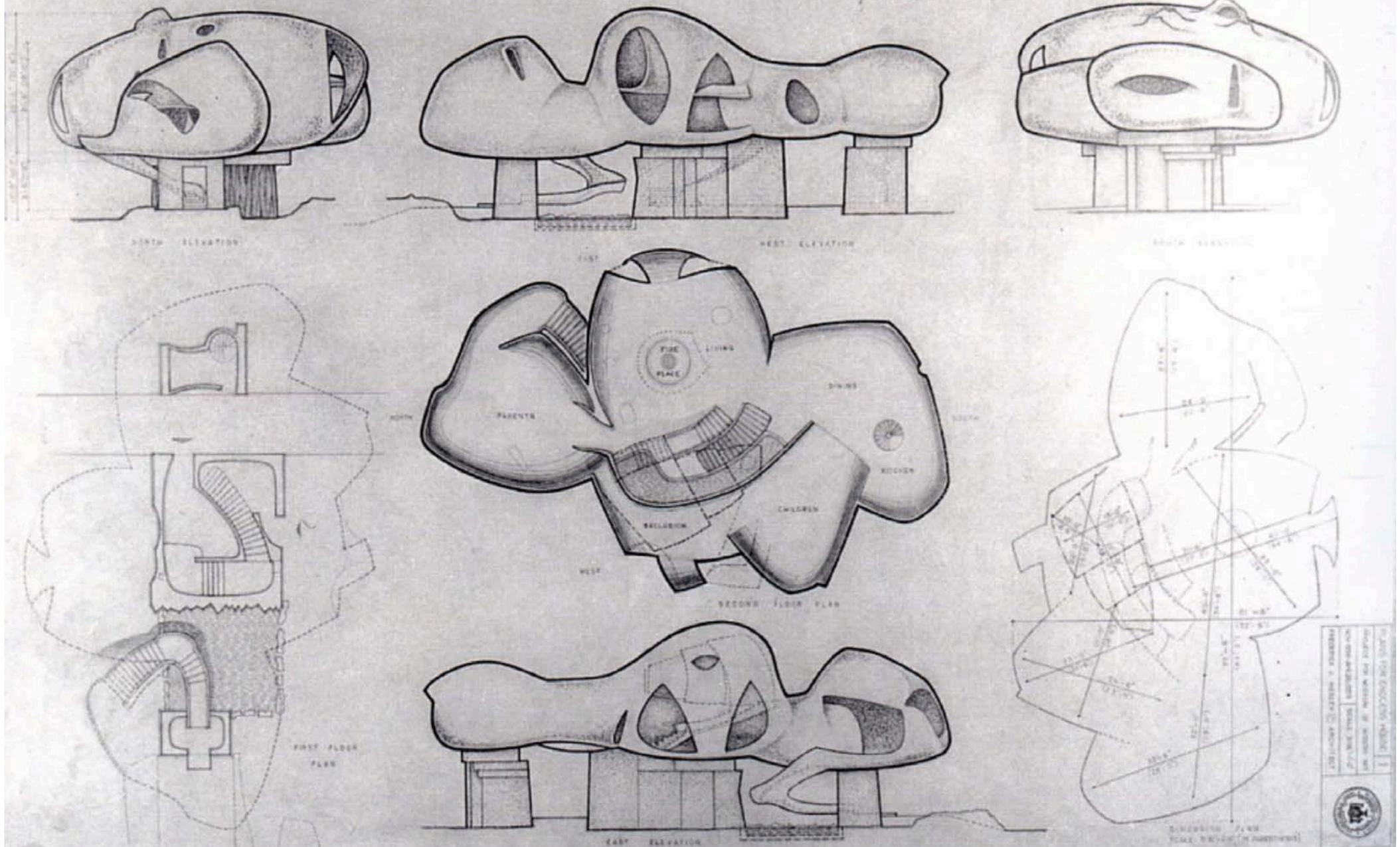
Die Struktur dieses Buches	5
Inhalt	6
Prolog: arbeiten mit dem "Endless House"	8
Rachel Whiteread: House	10
Die Gestalt des Leeren Körpers	16
Theory is analytic before it is normative.	18
über die Gleichförmigkeit des Raums	20
Das Leere ist nicht Nichts	22
Das endlose Haus	24
Die Sehräume erhellen: Raumsequenzen finden	26
Der Augenblick und die Wirklichkeit	28
Die Darstellung von Gleichzeitigem und Langzeitigem	30
Der Raum und seine Adressierung	32
Das manipulative Element in der Bildprojektion	34
Die Maschine-Die-Alles-Simultan-Sieht	36
Dreidimensionale quantitative Raumlandkarten	48
Die grafische Benutzeroberfläche	50
Die Bewertung des Raumes über eine Datenwolke	52
Numerische Schnittbilddarstellung	54
Transparenzen	56
Der Schwerpunkt des Sehkörpers	60
Die Kombination von Sehachse und Farbkarte	66
Ende und Anfang	68
Literaturverzeichnis	69
A Code Story	70

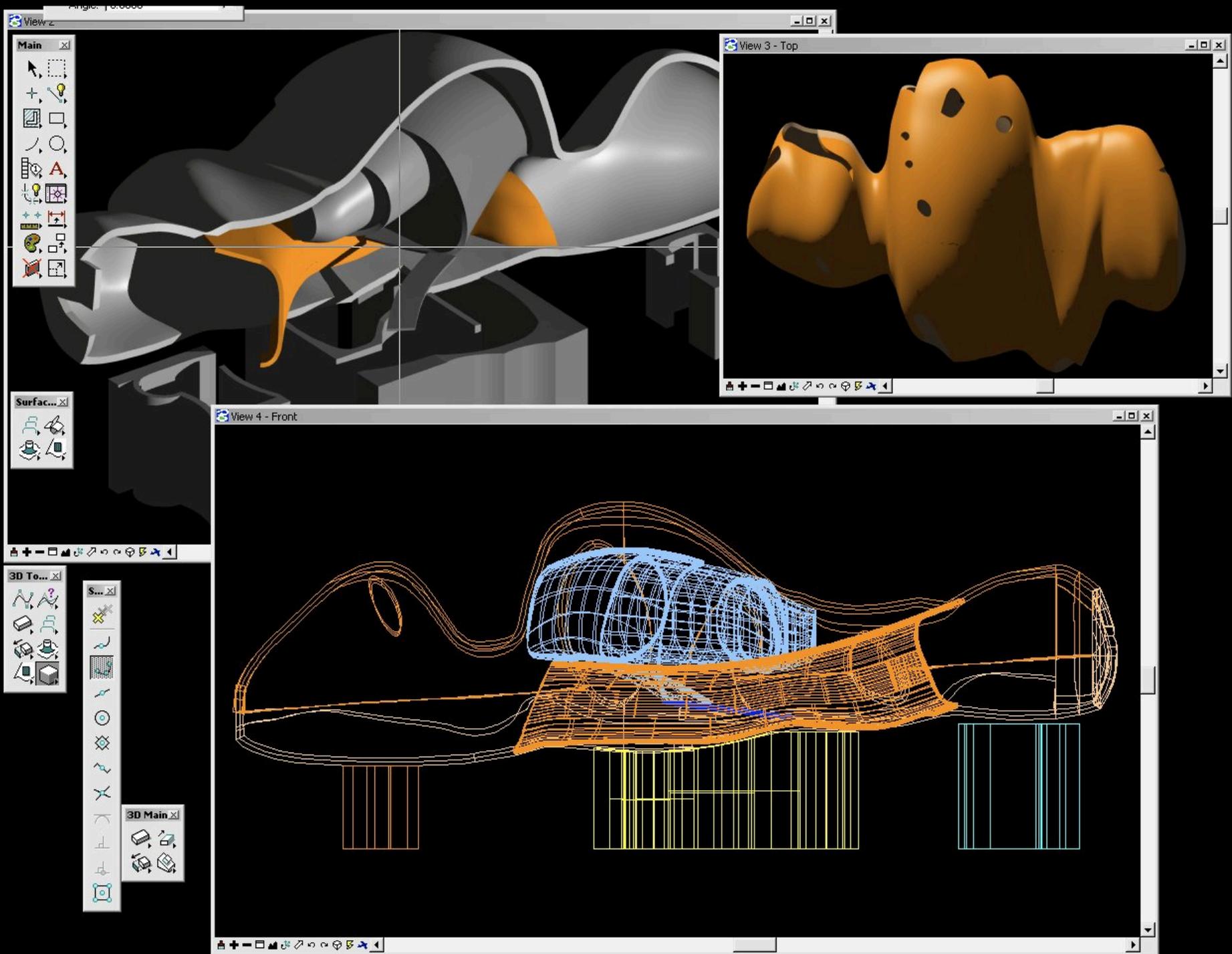


Friedrich Kiesler (1890 - 1965), österreichisch-amerikanischer Architekt, hat uns eine der bekanntesten ungebauten Architekturen hinterlassen. Das "Endless House", das als Betonmodell und in Plänen dokumentiert ist, wird in diesem Buch eine Rolle spielen. 2010 hat eine Lehrveranstaltung der Universität für Angewandte Kunst in Wien unter der Leitung von Prof. Klaus Bollinger die baukonstruktive Durchbildung des Hauses mit heutigen Mitteln zum Thema gehabt. Eines der Ergebnisse war ein Computermodell aus Freiformkurven, das in erweiterter Form auch Eingang in dieses Buch gefunden hat. Die hier gezeigten Illustrationen dokumentieren die Arbeit. Kieslers Übersichtsplan auf der Seite unten gibt uns etwas Aufschluß über eine mögliche Funktionszuweisung der Räume.



Prolog: arbeiten mit dem "Endless House"





Abbildungen oben: ein Computermodell des "Endless House" von Friedrich Kiesler. Es wurde mit gängiger Software (Bentley Microstation) gezeichnet. Ganz analog zu dem viktorianischen Reihenhaus, das

Rachel Whiteread abträgt, dient es als verlorene Form für die weitere Analyse. Die Sichtfenster zeigen eine Draufsicht (rechts oben) und einen räumlichen Längsschnitt durch die Schale (links

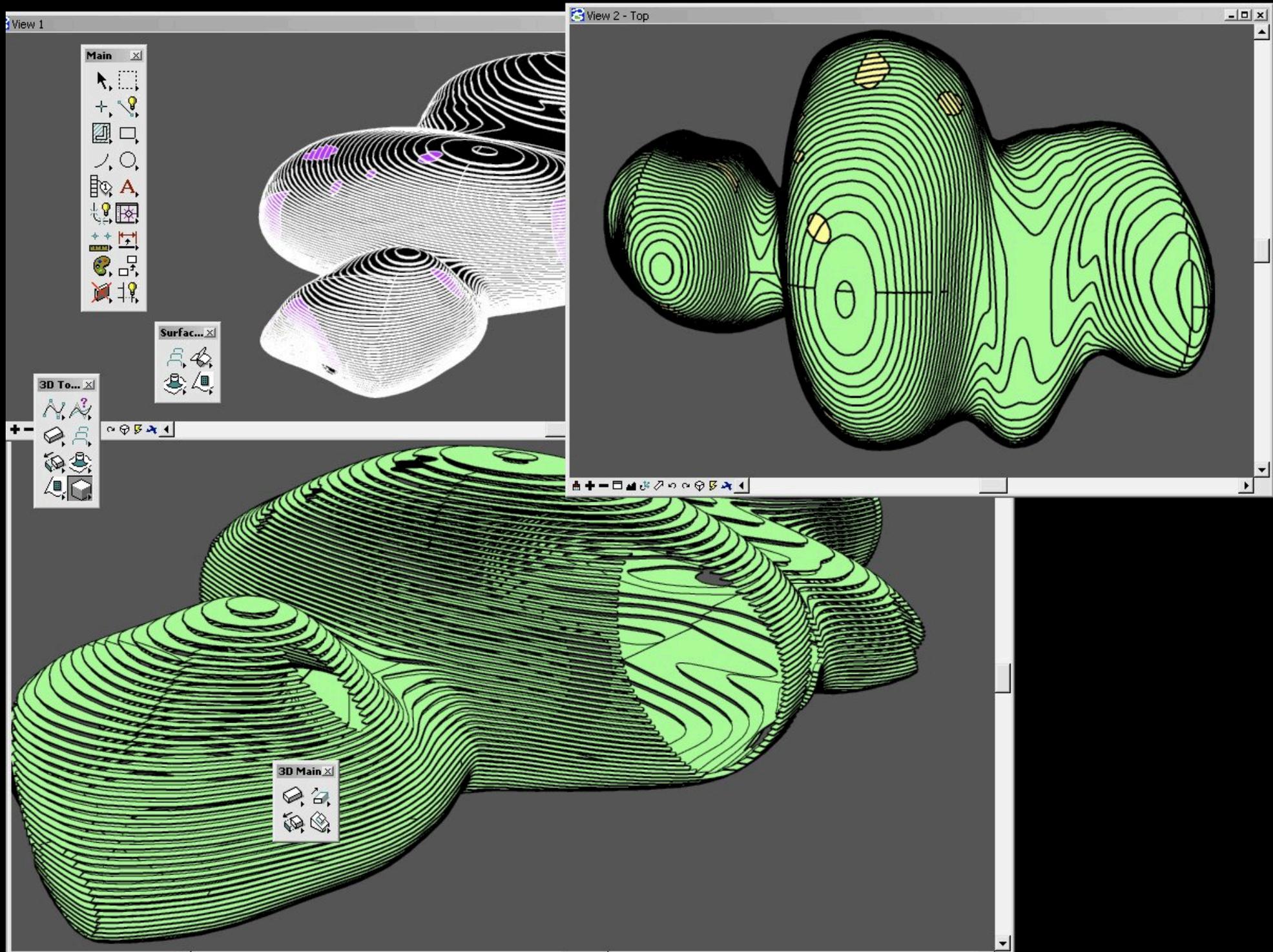
oben), in dem die Einbauten für das Kinderzimmer (orange) sichtbar sind. In dem Längsschnitt (unten) sind die Einbauten für die Räume im Obergeschoß hervorgehoben.

Rachel Whiteread: House

"House" ist eine Skulptur der britischen Künstlerin Rachel Whiteread aus dem Jahr 1993. Sie existierte als temporäres Objekt auf einem Grundstück in London. An diesem Ort befand sich ein kleines Reihenhaus aus der viktorianischen Zeit - das letzte, das als Teil einer schon zuvor demolierten größeren Anlage noch übrig geblieben war. Das Haus entsprach einem Archetypus, der auch anderswo häufig zu finden ist. Aber obwohl es sich um ein Reihenhaus handelte, stand es frei inmitten einer parkartigen Umgebung und war aus allen Richtungen gut zu sehen. Noch kurz zuvor dürfte es bewohnt gewesen sein: Kamine, Waschbecken, Türen und Fenster waren intakt und mit Gebrauchsspuren versehen.

Diese und folgende Seiten: Standbilder aus einem Dokumentarfilm zu "House"





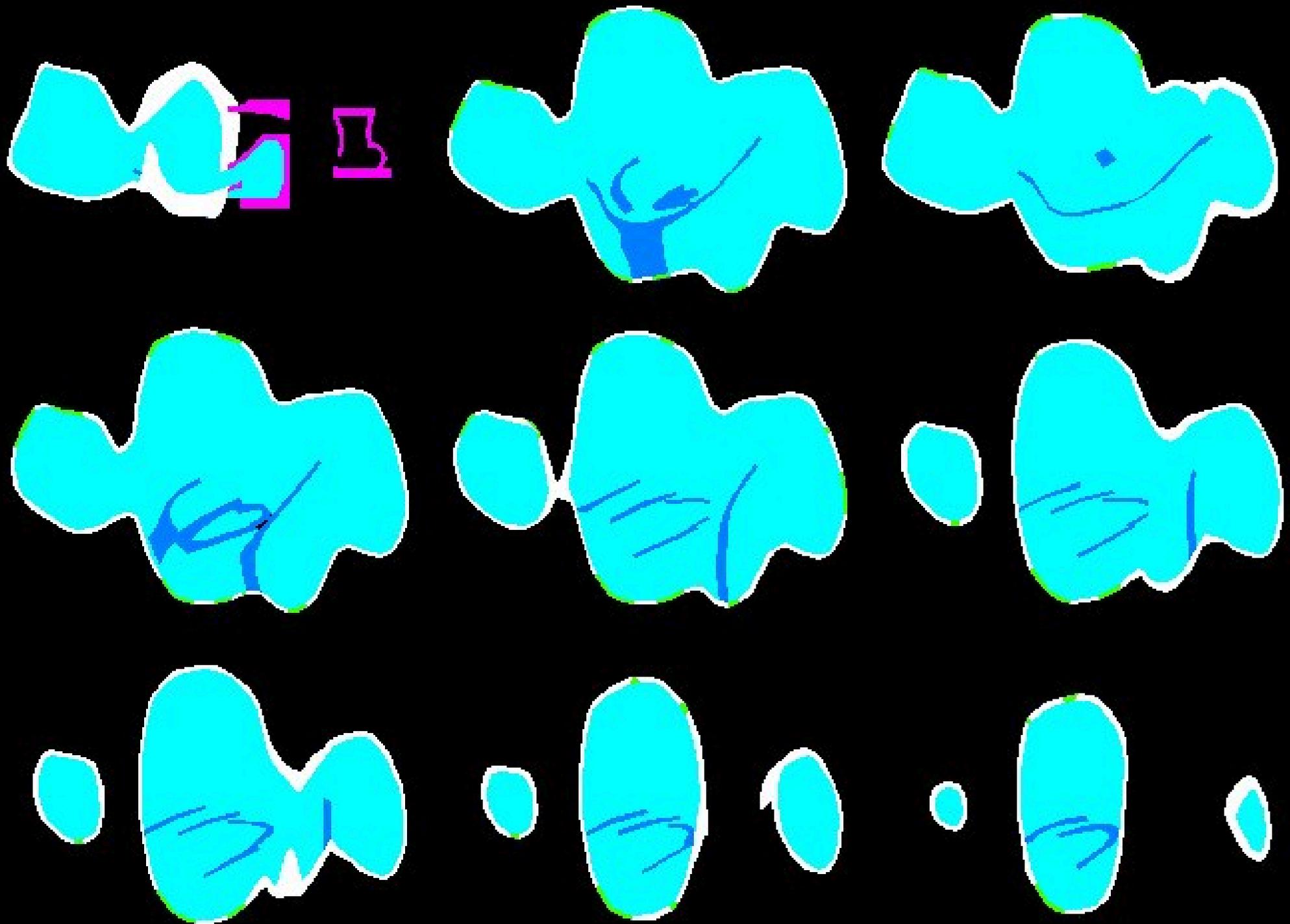
Abbildungen oben: das auf der vorangegangenen Doppelseite gezeigte Modell wurde in Höhenschichtlinien mit einem Abstand von 15 cm zerlegt. Das ist die Seitenlänge der Voxel

(Raumeinheiten), aus denen später das Analysemodell entstehen wird. Die Höhenschichtlinien sollen anschließend jeweils in ein Bild mit einer Auflösung von 15 cm umgewandelt werden.

Während der Entstehung der Skulptur wurde ein Dokumentarfilm gedreht, der die einzelnen Arbeitsschritte zeigt: eine Gruppe von Leuten, unter ihnen die Künstlerin, beginnt, im Inneren des Hauses Wände zu präparieren und in den Räumen Betonarmierungen zu verlegen. Einige Zeit später beginnt ein ungeheurer Vorgang: sieben Tage lang wird das Haus von Innen heraus mit Torkretbeton besprüht - vom Boden über die Wände bis zur Decke. Auf

eine erste Betonschicht werden nach innen hin schwere Armierungen aufgebracht, um eine tragfähige Innenschale zu erreichen. Der Beton härtet aus, damit der nächste Arbeitsschritt beginnen kann. Erst mit Stemmeisen, dann mit schwerem Gerät werden Stück für Stück die Außenwände und alle Einbauten, die die bis vor Kurzem die Substanz dieses Hauses darstellten, entfernt. Langsam schält sich das heraus, was zuvor von der Künstlerin und ihren Mitarbeitern in das Haus eingebracht wurde: ein gleichmäßig hellgraues Volumen aus Beton, das genau den Übergang markiert, an dem früher ein Innenraum durch seine undurchdringliche Umrandung begrenzt wurde.





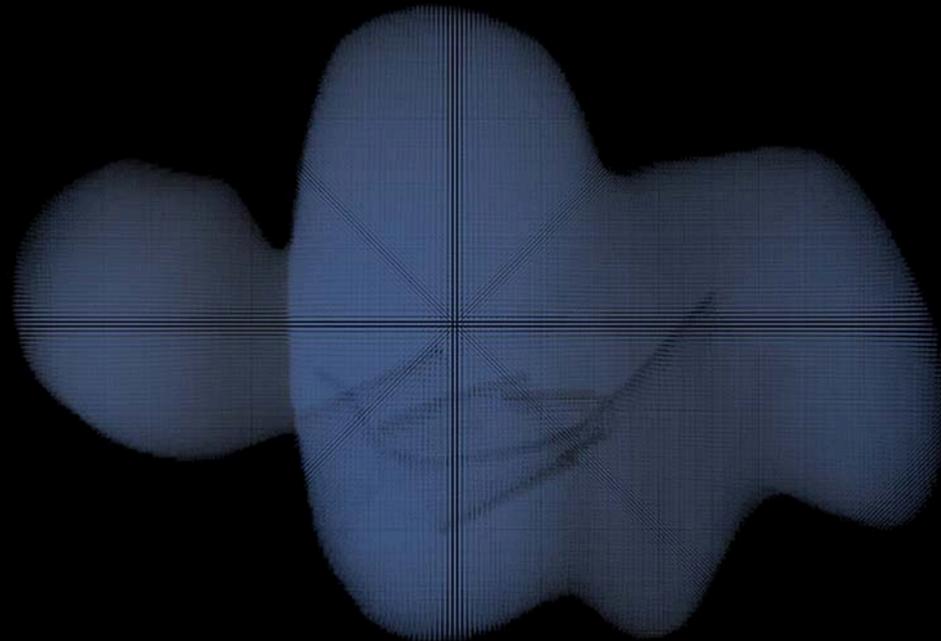
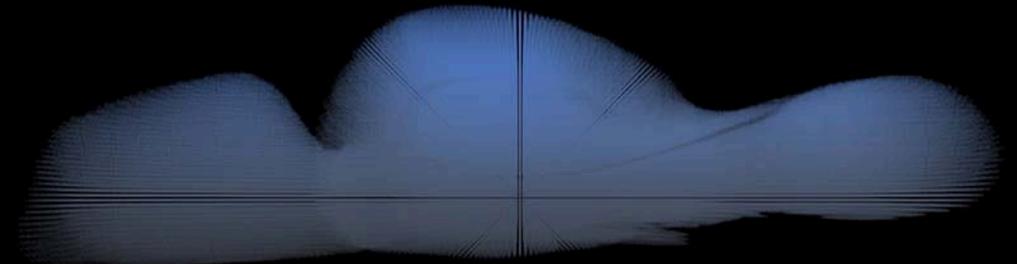
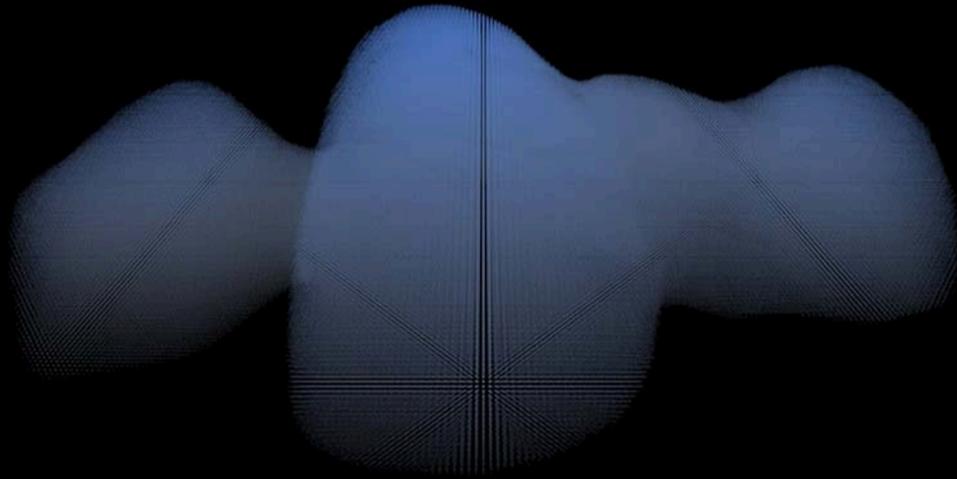
Abbildungen oben: die Form der Schale wird durch ca. 50 Höhenschichten erfaßt, die in vergleichsweise kleinen Bitmaps (Auflösung 220 x 160 Pixel) enthalten sind. Diese Bitmaps werden von der Software au-

tomatisch eingelesen und je nach Farbwert in verschiedene Datensammlungen aufgeteilt. Die hellen Begrenzungslinien und die blauen Einbauten werden als Begrenzungslinien interpretiert, Grün steht für

transparente Bauteile und schwarze Flächen sind Außenräume. Der Innenraum wurde mit hellblauer Farbe ausgegossen, um ihn zu kennzeichnen - ganz analog zu dem Torkretbeton in dem realen Beispiel.

Mit dem Verschwinden der Wände verschwindet nicht - wie sonst in einem derartigen Fall - auch das Haus. Ganz im Gegenteil, die zuvor durch Bauteile verdeckten Innenräume werden erstmals sichtbar. Die Skulptur erregt großes Aufsehen und wird von tausenden Leuten besichtigt. Nach einiger Zeit wird auch sie abgetragen. Erst jetzt verschwindet das alte Haus zur Gänze.





Der Negativabdruck der Innenräume des Hauses ist der Bereich, mit dem sich diese Arbeit befaßt. Die Beschaffenheit der Bauteile selbst ist nicht von Interesse und wir sparen sie folglich vollkommen aus. Das Modell wurde aus 100 einzelnen Schnittbildern erzeugt und umfaßt ca. 200.000 Raumpunkte, die im Folgenden untersucht werden sollen.

Die Gestalt des Leeren Körpers

Im Jahr 1993 hat eine britische Künstlerin ein altes Haus abgerissen, nachdem sie zuvor alle Inneräume mit Beton ausgegossen hat. Der Betonkörper, der viel stabiler war, blieb stehen. Die Räume, in denen zuvor Menschen gewohnt hatten, wurden frei sichtbar.

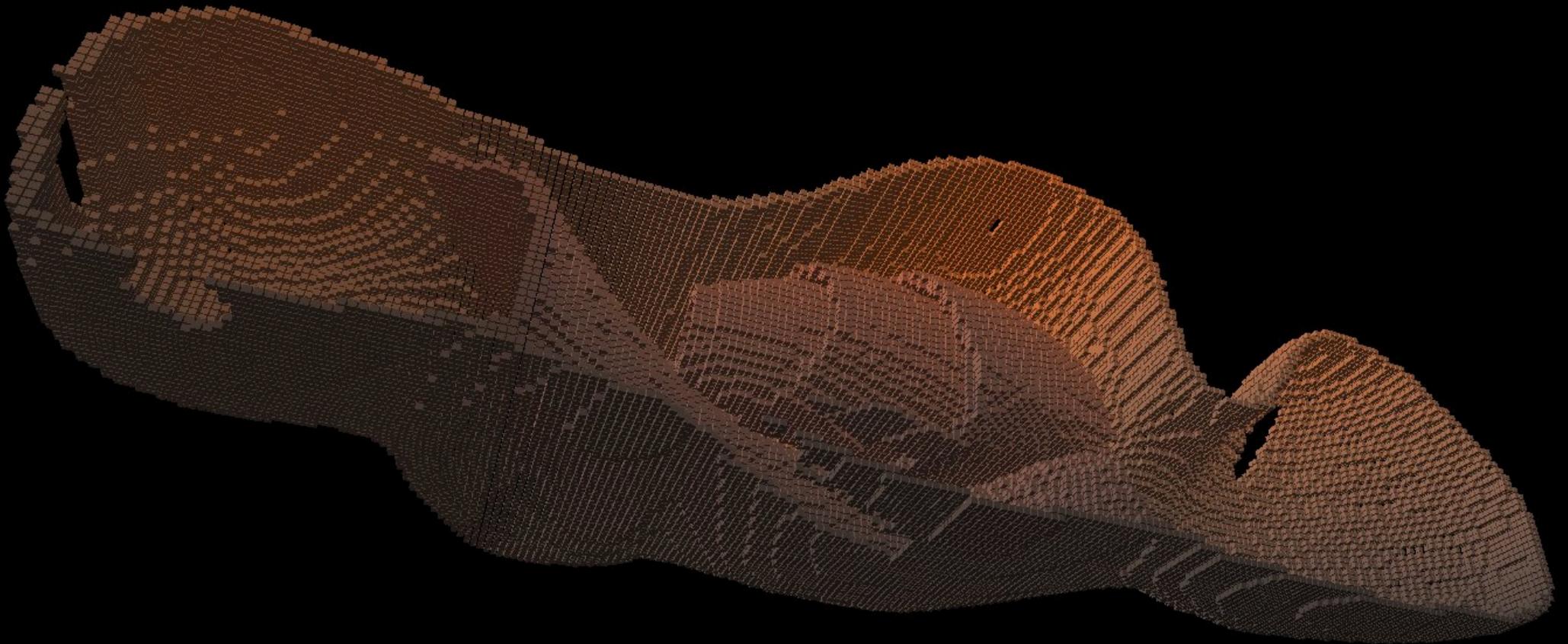
Rachel Whitereads Arbeit ist das wesentliche Leitmotiv für das hier vorliegende Buch.

Der Gegenstand dieses Buches ist eine Software, die der Verfasser entwickelt hat und ihre Anwendung an einem Beispiel aus der Architekturgeschichte. Es scheint auf den ersten Blick zwischen einer Betonskulptur und einer Software keine Gemeinsamkeiten zu geben.

Durch die Skulptur wird etwas sichtbar, von dem wir dauernd umgeben sind und das wir eigentlich nur anhand seiner Begrenzungsflächen wahrnehmen: der Leere Körper. Man kann anzweifeln, ob er überhaupt zur Dingwelt gehört und ihn wie eine Erscheinung aus der Metaphysik behandeln. Auf diese Weise entrückt, verliert der Leere Körper unser Interesse als ein Gegenstand, den man messen und erforschen kann.

In diesem Buch wird der andere Weg beschrieben: der Raum zwischen den Dingen kann gemessen und erforscht werden wie die Dinge selbst.

Seite für Seite sollen einfache Methoden gezeigt werden, nach denen man diesen Raum bewerten und mehr über ihn erfahren kann. Zu diesem Zweck habe ich eine Software geschrieben, die intuitiv und anschaulich Raumbeziehungen analysiert. Wie die Skulptur soll der leere Raum aus seiner Begrenzung herausgeschält und sichtbar gemacht werden.



Anhand dieser Grafik ist gut zu erkennen, wie aus den zweidimensionalen Pixel der Bitmaps ein Raum aus Voxeln entsteht. Die Software liest die Bilder Schicht für Schicht ein und setzt die Punkte an die entsprechende Position in einem räumlichen Raster.

Theory is analytic before it is normative.

- Bill Hillier¹

Zu den Wissenschaften, die die Beziehungen von Körpern untersucht, zählt die Mechanik. Eine verlässliche Vorhersage darüber zu treffen, wie sich Körper verhalten werden, war für lange Zeit nur dann möglich, wenn man empirisches Wissen darüber besaß. Der englische Universalgelehrte Robert Hooke (1635 - 1703) entdeckte die Proportionalität zwischen der Formänderung eines Gegenstandes und der Kraft, die auf ihn einwirkt.² Unser Verständnis von mechanischen Vorgängen wird von dieser Entdeckung abgeleitet. Der Anlaß für Hooke, sich mit diesem Thema zu befassen, war vermutlich ein praktisches Problem, das er lösen wollte: um auf offener See zuverlässig navigieren zu können, benötigte man genaue Uhren. So eine Uhr wollte er bauen. Das zentrale Element einer Uhr ist die Feder, die sie antreibt. Es war also wichtig, das Verhalten dieser Feder und ihre Elastizität zu kennen.

Hooke löste diese Frage und schaffte so eine der Grundlagen des Ingenieurwesens, denn Hookes Gesetz gilt überall. Tragwerkskonstruktionen in der Architektur sind offensichtlich komplexer als Uhrfedern. Die Uhrfeder ist sehr gleichförmig. Diese Einfachheit hat es Hooke ermöglicht, eine Theorie zu finden, die auch auf ungleichförmige Körper anwendbar ist.

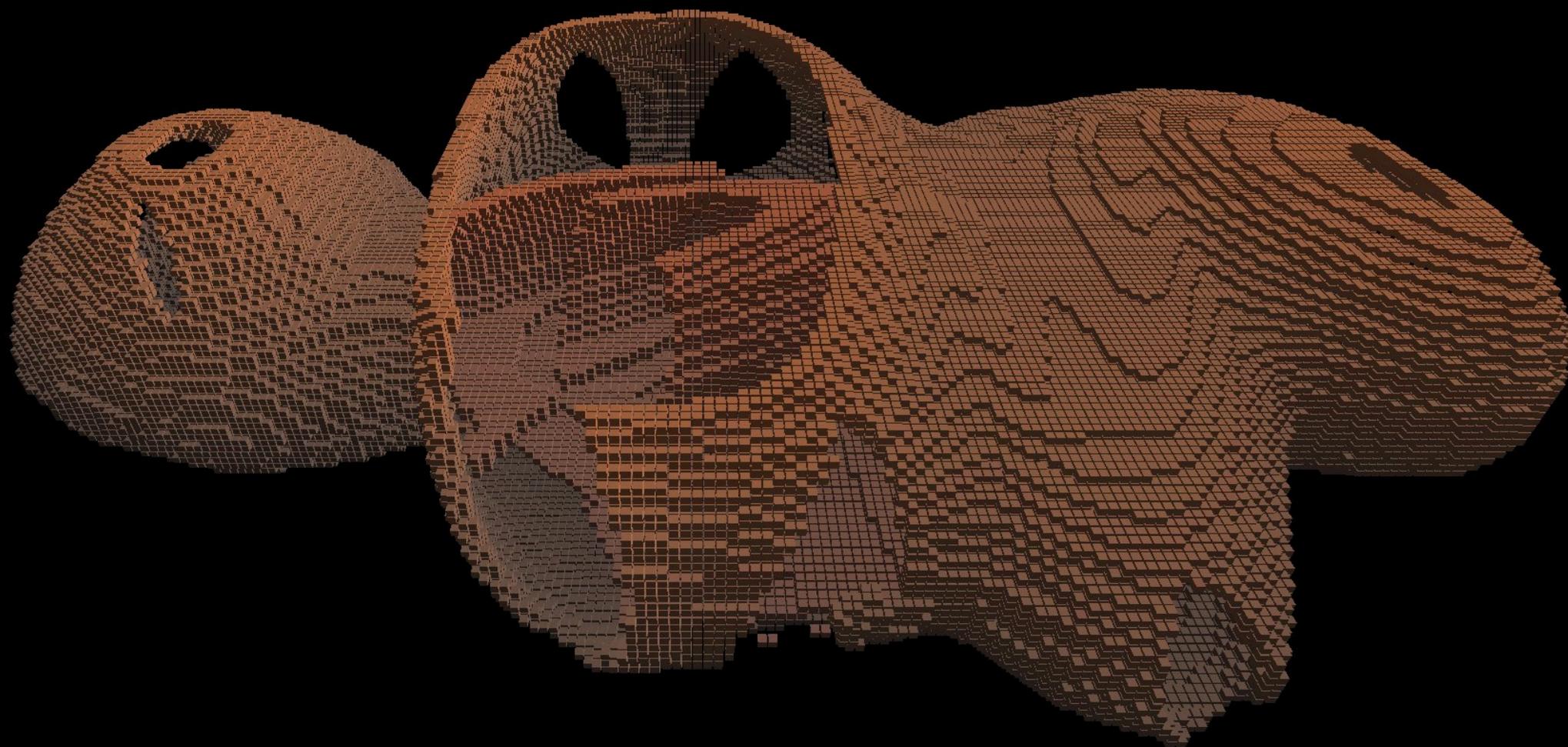
Die Idealisierte Sicht auf die Materie als etwas, von dem man Gleichförmigkeit in alle Richtungen voraussetzt, steht also am Beginn der Mechanik.

Hanson und Hillier beschäftigen sich mit den Bewegungsabläufen in städtischen Gebilden. Städte sind, ganz im Gegensatz zum regelmäßigen Aufbau einer Uhrfeder, komplexe und vielschichtige Strukturen. Die Wege ihrer Bewohner unterscheiden sich je nach Lebenssituation stark voneinander und so sind auch ihre Eindrücke stets unterschiedlich. Trotzdem gibt es Orte, die für die Mehrheit der Menschen, die die Stadt benutzen, eindeutige Attraktionspunkte darstellen.

Der Urbanismus erforscht das Zustandekommen dieser Attraktionspunkte. Geschichtliche und kulturelle Gegebenheiten einer Stadt scheinen eine wichtige Rolle in der Ausprägung dieser Orte zu spielen. Derartige Phänomene sind, anders als der Kräfteverlauf in einer Uhrfeder vielschichtig und bleiben rätselhaft. Hillier (S. 2) kritisiert, dass die Architekturtheorie keine analytische Herangehensweise sucht, sondern sich darauf beschränkt, normative Aussagen zu treffen. Dieser Umstand kann als Reaktion auf die vielschichtige Natur des Städtebaus gesehen werden, die sich einer quantitativen Analyse weitgehend entzieht, weil sie scheinbar der gegebenen Komplexität nicht gerecht wird. Aus der Kritik könnte man eine Aufforderung ableiten, die unerforschten Möglichkeiten zu suchen, die die quantitative Analyse für Architekturplanung bereithält. Das führt zu Robert Hooke zurück, der sich an einer bestimmten Stelle entschlossen hat, von einer sehr einfachen Annahme auszugehen und alles weitere davon abzuleiten.

1) Hillier 2007, S. 3.

2) Chapman 2005.



Das Volumensmodell des Hauses mit abgenommener Mittelkuppel. Die räumlich sehr komplexe Situation im Obergeschoß ist andeutungsweise zu erkennen.

Über die Gleichförmigkeit des Raums

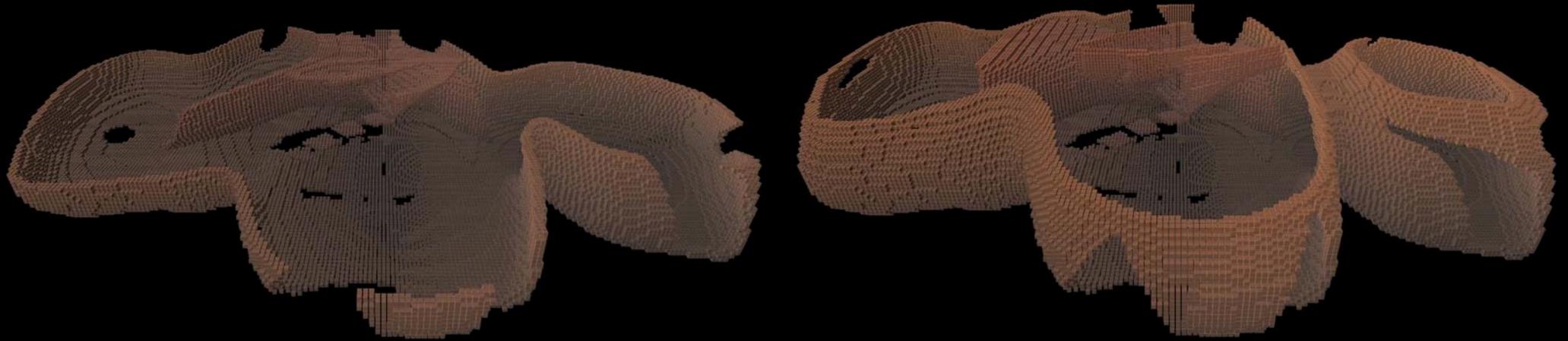


Das Diagramm aus dem Buch von Hillier³ zeigt, wie gut einzelne Straßenzüge in einem Stadtteil von London mit ihrer Umgebung in Beziehung stehen. Dunkle Bereiche sind besser integriert, helle weisen mangelhafte Verbindung auf.

3) Hillier 2007, S. 2-132.

In seinem Buch "Space is the machine" stellt Bill Hillier einige Strategien vor, zwei-dimensionale räumliche Layouts durch quantitative Methoden zu bewerten. Es werden zum Teil sehr großräumige Gebiete untersucht, etwa das gesamte Straßennetz von London. In die Untersuchung fließt im Wesentlichen nichts anderes ein als die Topografie der Verkehrswege. Es wird auch keine Unterscheidung zwischen Haupt- und Nebenwegen getroffen. Attraktoren wie wichtige Gebäude oder besondere Sehenswürdigkeiten fließen nicht in die Bewertung ein. Die Rechnergestützte Auswertung der Versuchsanordnung führt zu einer Karte, die die Wege des betrachteten Gebietes färbig darstellt und damit besser und schlechter vernetzte Straßenzüge ausweist. Durch diese Methode können Aussagen über die Attraktivität eines Gebietes getroffen werden, die mit anderen Kenngrößen (etwa Grundstückspreisen oder dem tatsächlichen Verlauf höherwertiger Wege) in der Wirklichkeit stark übereinstimmen.

Eine Annahme dabei lautet, dass es nicht die oben genannten Attraktoren wie wichtige Gebäude oder Sehenswürdigkeiten sind, die den Wegen ihre Bedeutung verleihen, sondern dass es das Wegenetz ist, das es manchen Punkten erlaubt, zu Attraktoren zu werden. Aus der eigenen Anschauung heraus fällt es vielleicht schwer, diese Annahme zu akzeptieren: wir sind ständig gezwungen, an Orte zu gelangen, die nicht gut erreichbar sind. Der Einwand von Hillier lautet: eine Vielzahl von Personen sucht eine Vielzahl von Orten auf, was zur gleichmäßigen Verteilung der Ziele führt. "Movement tends to be broadly from everywhere to everywhere else."



Die Schale des Gebäudes kann je nach Erfordernis als teilweise Darstellung gezeigt werden. Die Bilder zeigen verschiedene Schnitttiefen. Für Berechnungen wird das gesamte Datenmodell berücksichtigt, auch wenn die Darstellung nur Teile umfaßt.



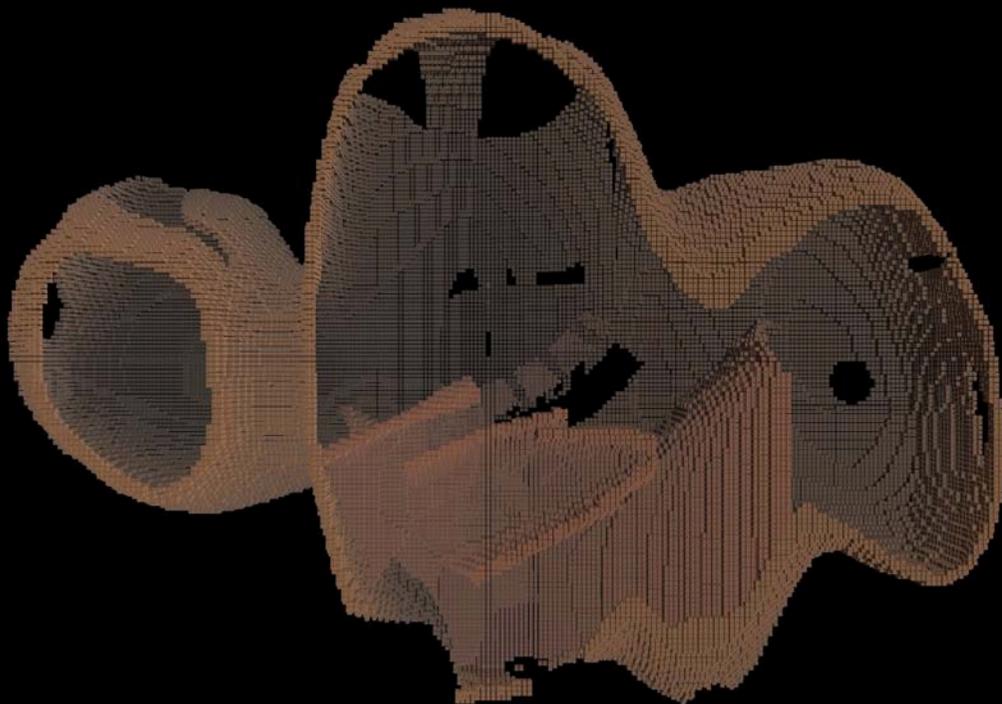
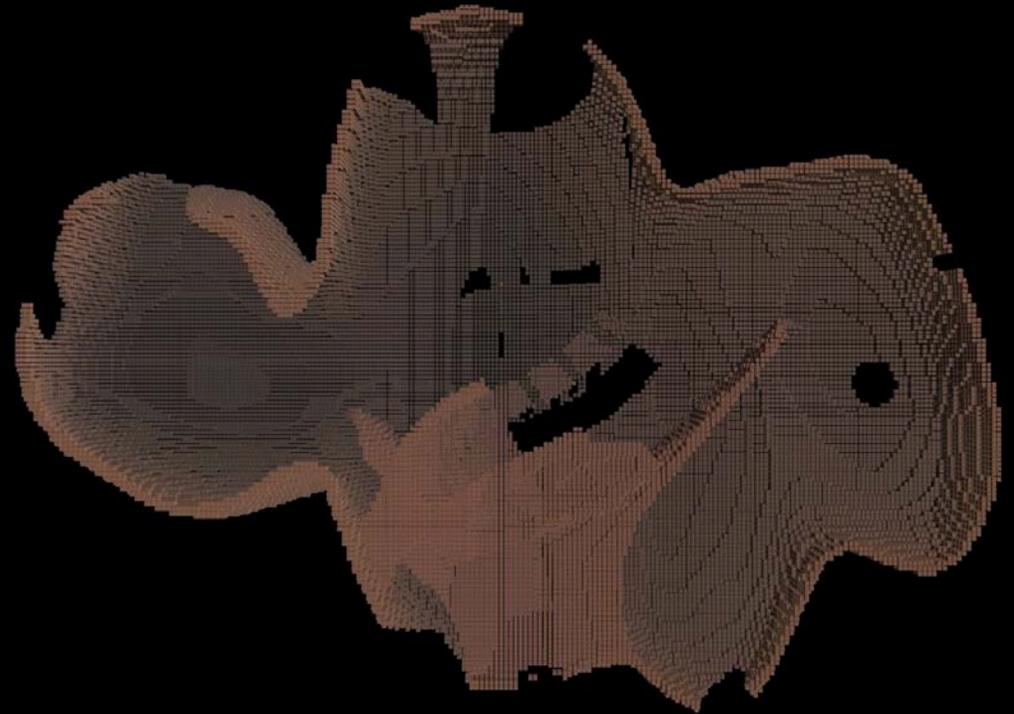
„Das Leere ist nicht Nichts“

- Martin Heidegger

Am Beginn dieses Buches wird ein Zerstörungsakt beschrieben: es handelt sich um die kunstvolle Schleifung eines Solitärbauwerkes. Später im Text wird eine wissenschaftliche Methode vorgestellt, die dem Weg alle Bedeutung zuweist und das einzelne Gebäude am Straßenrand absichtlich nicht in Betracht zieht. Vom Solitärbau wird abgesehen, aber der Bereich, der daran angrenzt, rückt ins Blickfeld. Diese Sehweise ist die Inversion eines Raumverständnisses, das sich mit dem Ende des 19. Jahrhunderts durchsetzte und das damals ebenfalls eine radikale Umkehrung der Blickrichtung darstellte.

„Beim modernen Städtebau kehrt sich das Verhältnis zwischen verbauter und leerer Grundfläche gerade um. Früher war der leere Raum (Strassen und Plätze) ein geschlossenes Ganzes von auf Wirkung berechneter Form; heute werden die Bauparzellen als regelmäßig geschlossene Figuren ausgeteilt, was dazwischen übrig bleibt, ist Strasse oder Platz.“ Diese Kritik stammt aus einem zeitgenössischen Text. Zwischen dem Text und der heutigen Welt liegt eine ganze Geschichte von Stadtentwürfen, die den Leerraum als eine Art Schaukasten ansah, der ausnahmsweise zur Präsentation von Solitärbauten freigehalten werden konnte, wenn er nicht verwertet wurde.

Martin Heidegger sagt 1929 über die moderne Wissenschaft: "Sie will vom Nichts nichts wissen." Diese "Seinsvergessenheit", so Heidegger, fördert paradoxerweise die Ausbreitung nihilistischer Orte. Zu dieser Zeit entstehen Hilberseimers und Le Corbusiers utopische Stadtmodelle, die ganz auf die Substanz der Gebäude konzentriert sind. Sie sind extreme Beispiele jener Blickumkehr, die den oben zitierten Kritiker aus dem 19. Jahrhundert so entsetzte: Camillo Sitte.



Draufsicht mit verschiedenen Schnitttiefen. Die Zerlegung der Geometrie in Raumquanten (Voxel, in diesem Fall mit 15 cm Seitenlänge) bietet Vorteile bei der Generierung von Teilmengendarstellungen, da nur eine sehr einfache Vergleichsoperation durchgeführt werden muß: wenn es wahr ist, dass das Voxel an derselben Adresse liegt wie ein Voxel einer Vergleichsmenge (das kann etwa eine Schnittebene sein), dann zeichne es. Im Gegensatz dazu sind Freiformgeometrien wie diese in Vektorbasierter CAD-Software schwierig zu beschreiben und schwierig zu manipulieren.

Das endlose Haus

Den Analysemethoden von Hillier liegt die Idee zugrunde, dass die Bewegung im Stadtraum zweidimensional ist und Aspekte der Höhenentwicklung vor der horizontalen Ausdehnung einer Stadt zurücktreten. Dementsprechend sind die Analysewerkzeuge (Computersoftware) auf zwei Dimensionen fixiert. Auch wenn dieser Ansatz in der Stadtplanung nachvollziehbar ist: sobald kleinere räumliche Einheiten betrachtet werden, fehlt ein wichtiges Element: Geländekanten bilden besondere räumliche Situationen, hohe Bauwerke werden zu Merkzeichen, stockwerksübergreifende Sichtbeziehungen in Gebäuden stellen die Orientierung her. dazu sollte eine Bewertung von Blickbeziehungen vorgenommen werden, die über die Möglichkeiten der Zentralperspektive hinausgeht. Darauf wird im Folgenden noch eingegangen.

Durch die Behandlung des Raums als eine uniforme Substanz, die ähnlich zu bewerten ist wie Festkörper in der Baumechanik verbietet es sich, im Vorhinein Gewichtungen und Bewertungen vorzunehmen, wie man sie etwa aus dem "Funktionsprogramm" eines Grundrisses kennt. Der Raum soll für sich selbst sprechen.

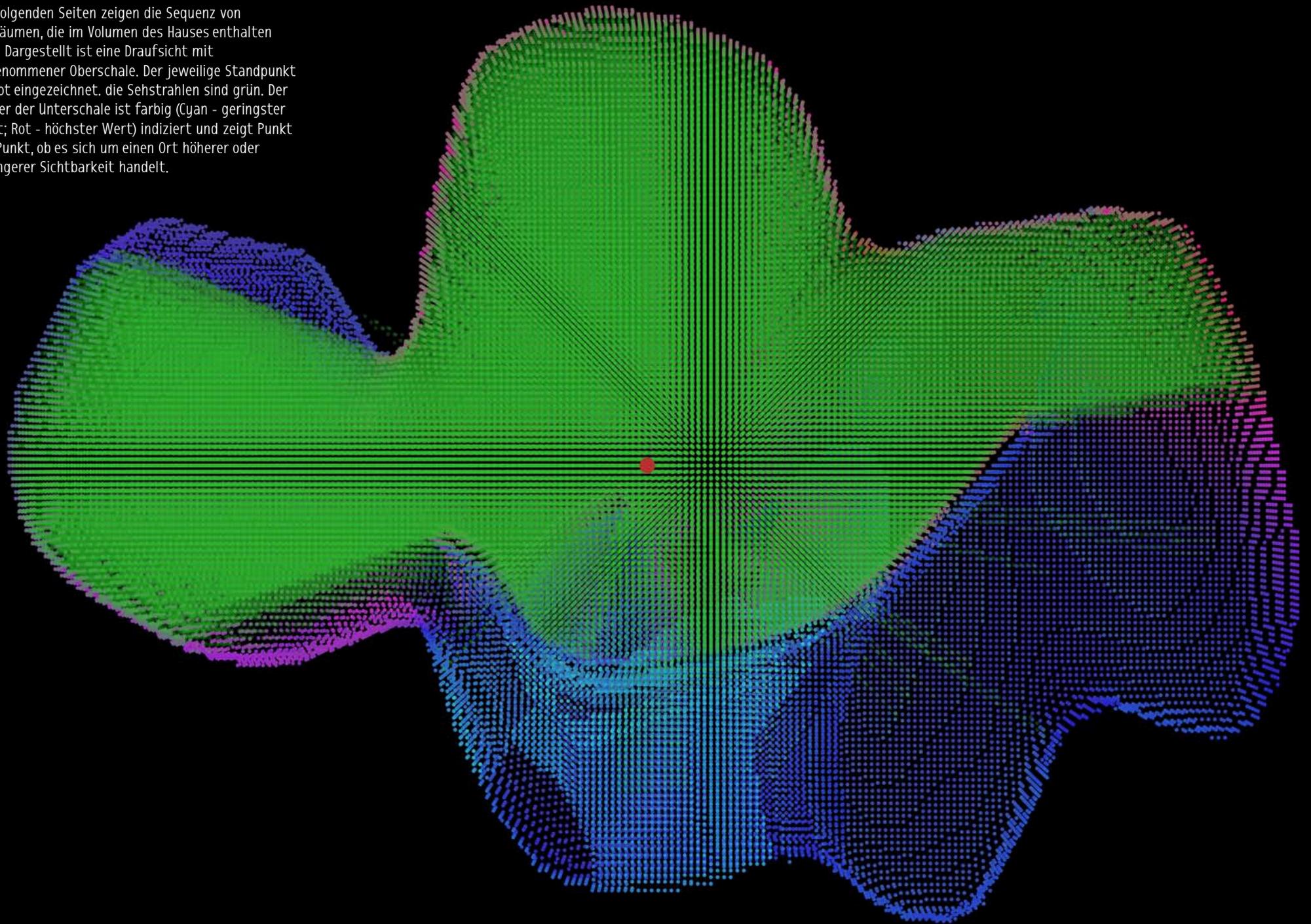
Von der Absicht, eine Software zu schreiben führt ein langer Weg bis zur Anwendung. Welches Ziel sollte man sich setzen, welchen Raum untersuchen? Die Wahl fiel auf ein singuläres Objekt in der Architekturgeschichte: Friedrich Kieslers Entwurf für ein "Endless House" aus dem Jahr 1959. Abseits jeder architekturhistorischen Einordnung kann das, was wir von dem Entwurf kennen als bizarr, skulptural, unzugänglich und räumlich anspruchsvoll angesehen werden. Durch das Grundthema des fließenden Raumkontinuums kann es in einer Reihe mit einigen diagrammatischen Entwürfen der klassischen Moderne angesehen werden, etwa Mies' Pavillion für die Weltausstellung in Barcelona. Man wird aber dem Haus nicht gerecht, wenn man es auf diesen Aspekt reduziert. Wo bei Mies Wände, Decken, Stützen und immer noch klare Raumgrenzen waren, entwirft Kiesler ein Schalentragwerk, das in dieser Form vielleicht erst seit wenigen Jahren bautechnisch durchführbar scheint.

Aus einigen "organischen" Entwürfen dieser Zeit spricht eine romantische Idee, ausgeglichene, harmonische Kurven einzusetzen, um damit Bewegung und Dynamik symbolhaft auszudrücken. Der Flughafenterminal in New York von Saarinen könnte als Beispiel dafür stehen. Kiesler hingegen entwirft eine Maschine für Raumsituationen, die in Beziehung zueinander stehen.

Diese Bewertung von Kieslers Entwurf wird untermauert von den Überlieferungen über die Entwurfsmethodik, die er in seinem gesamten Werk offenbar gesucht hat. Es ist eine anderes Verständnis von Funktionalismus, das aus Kieslers Arbeit spricht. Nicht eines folgt auf das andere, sondern Simultanität, Rückkopplung und Iteration formen uns und unsere Welt. Mit dieser Auffassung steht er in Beziehung zu dem etwas jüngeren Buckminster Fuller.

Es erscheint daher als reizvolle Idee, das Raumvolumen des Hauses näher zu untersuchen und hoffentlich einige der darin verborgenen Raumsituationen wiederzuentdecken.

Die folgenden Seiten zeigen die Sequenz von
Schräumen, die im Volumen des Hauses enthalten
sind. Dargestellt ist eine Draufsicht mit
abgenommener Oberschale. Der jeweilige Standpunkt
ist rot eingezeichnet, die Sehstrahlen sind grün. Der
Körper der Unterschale ist farbig (Cyan - geringster
Wert; Rot - höchster Wert) indiziert und zeigt Punkt
für Punkt, ob es sich um einen Ort höherer oder
geringerer Sichtbarkeit handelt.



Die Sehräume erhellen: Raumsequenzen finden

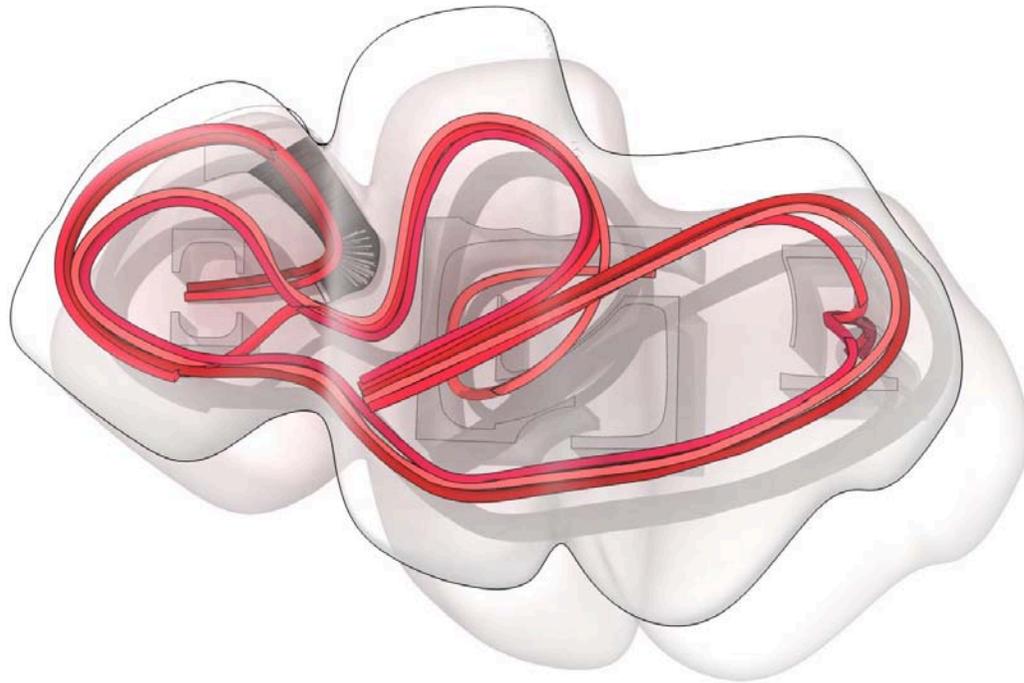


Illustration aus einer Studienarbeit der Meisterklasse für Architektur an der Universität für Angewandte Kunst, Wien: erst durch die Wegebeziehungen, die in das Haus eingearbeitet sind, entsteht eine Vielfalt von Raumzusammenhängen, die komplexer ist als die bloße Addition von Teilräumen.

'The "Endless House" is called the "Endless" because all the ends meet, and meet continuously.'

- Friedrich Kiesler ⁶

Vom Einzelbild zur Sequenz

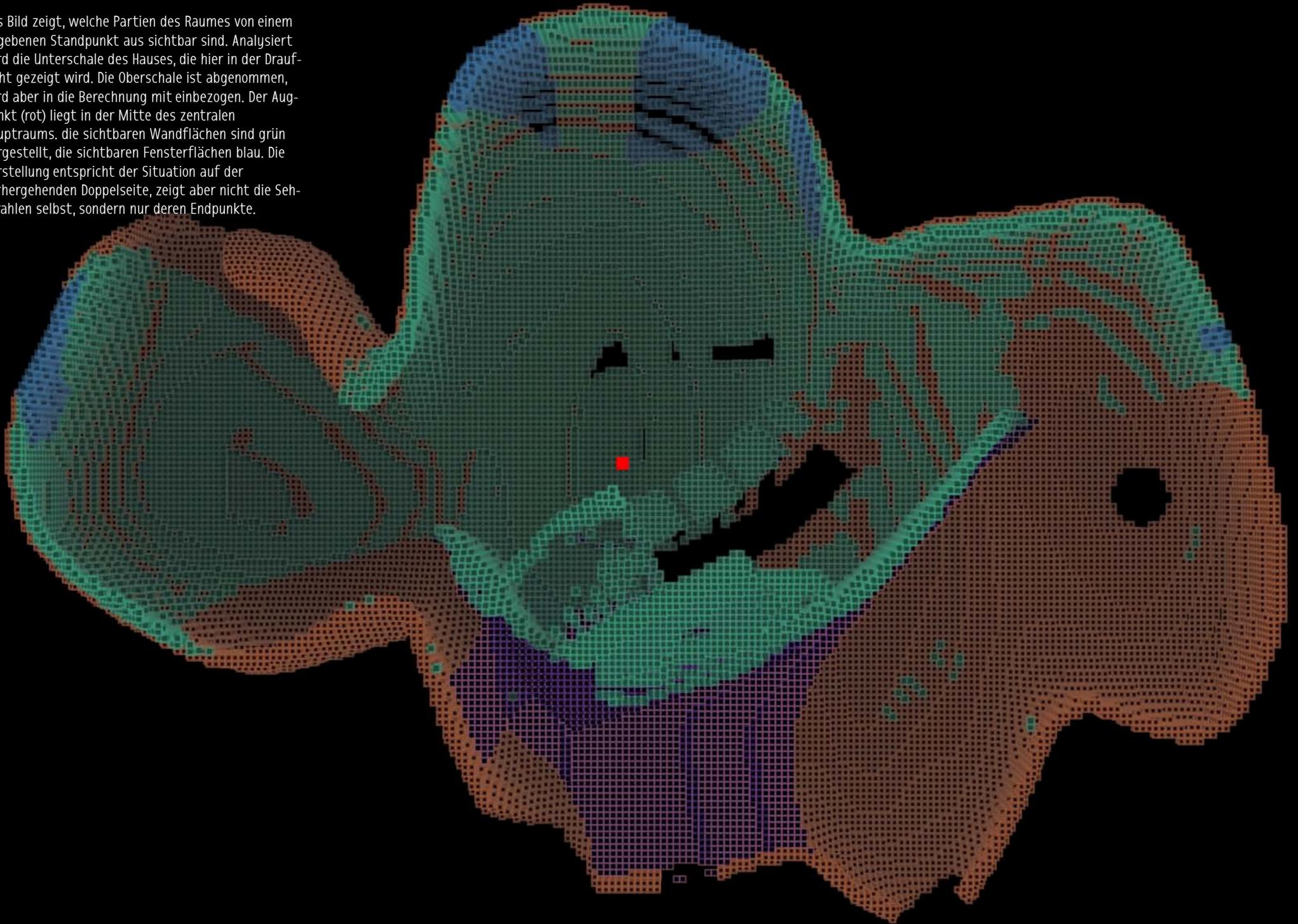
Ein leerer Körper, der unveränderlich ist, kann in einzelne Raumsequenzen zerlegt werden, wenn er durchquert wird. Der Fluß oder die Bewegung wird dem Raum zugeschrieben, obwohl es sich um die momentane Wahrnehmung des betrachtenden Subjekts handelt, die sich durch diesen Raum bewegt.

Nur die Verbindung mit Erinnerungen an kurz zuvor gesehene Raumsituationen verschafft uns überhaupt eine Vorstellung davon, wie unsere Umgebung beschaffen ist. Ein einzelnes Bild genügt dafür nicht, es muß schon eine ganze Sequenz sein.

Im Film gibt es den Begriff des "Establishing Shot" - eine Kameraposition am Beginn einer Szene, die dem Publikum die Zusammenhänge des Bühnenbildes zeigt. Das Fehlen einer solchen Einstellung ist ein Mittel des "Suspense" und ein Hinweis auf eine kommende Überraschung.

6) Kiesler 1996, S. 126.

Das Bild zeigt, welche Partien des Raumes von einem gegebenen Standpunkt aus sichtbar sind. Analysiert wird die Unterschale des Hauses, die hier in der Draufsicht gezeigt wird. Die Oberschale ist abgenommen, wird aber in die Berechnung mit einbezogen. Der Augpunkt (rot) liegt in der Mitte des zentralen Hauptraums. Die sichtbaren Wandflächen sind grün dargestellt, die sichtbaren Fensterflächen blau. Die Darstellung entspricht der Situation auf der vorhergehenden Doppelseite, zeigt aber nicht die Sehstrahlen selbst, sondern nur deren Endpunkte.



Der Augenblick und die Wirklichkeit

Wie kann eine visuelle Analyse eines Körpers aussehen? Welche Wirklichkeit bildet sie ab? Zeigen wir einen logischen oder einen geometrischen Ort? Es soll ein Weg gefunden werden, über die Perspektive hinauszugehen und mehr über Sichtbeziehungen zu erfahren, als aus dem Einzelbild hervorgeht.

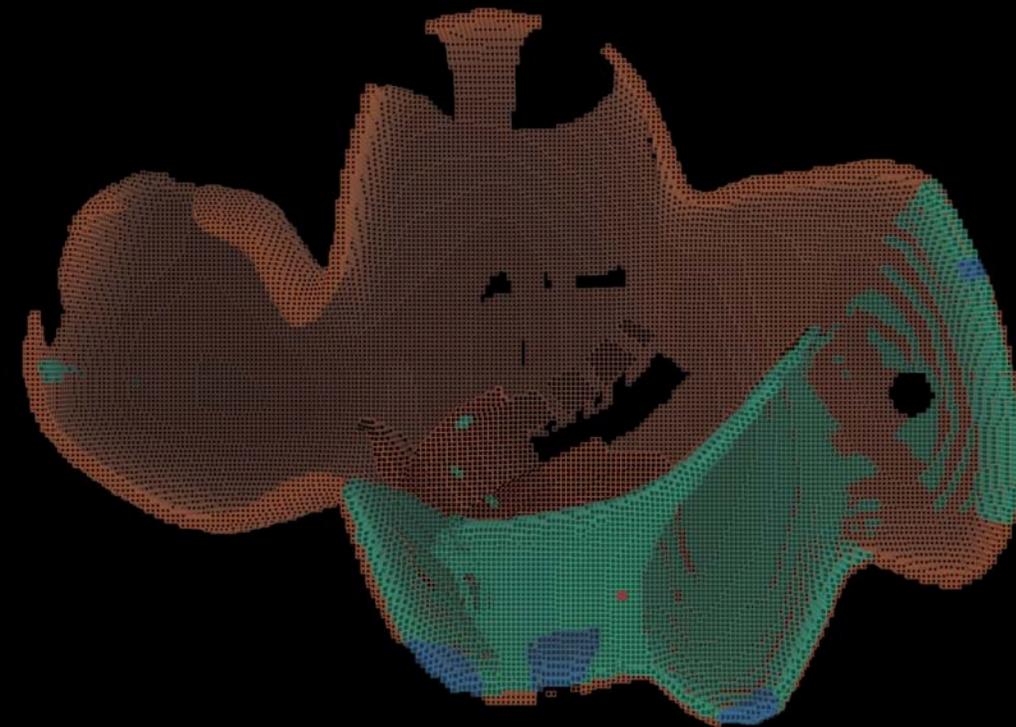
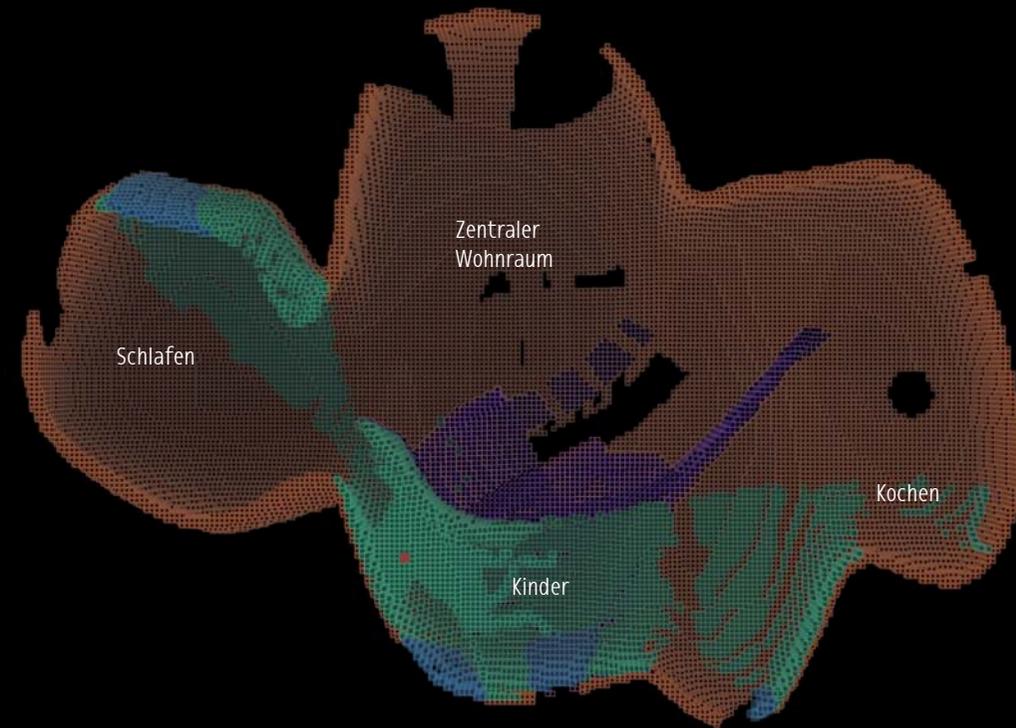
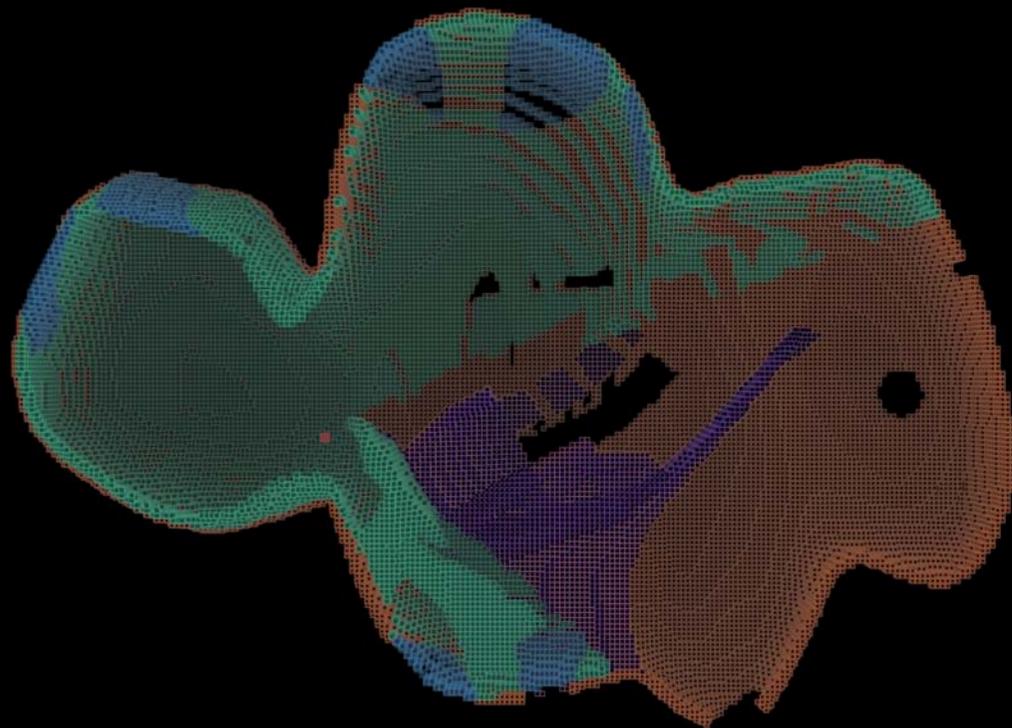
Der Sonderfall Fotografie

Das Fallen des Verschlusses in einer Kamera ist ein singulärer Moment, der an einem singulären Ort stattfindet. Der Fotografie fehlt die vorhin beschriebene sequentielle Komponente, weshalb es schwierig ist, mit Hilfe eines Einzelbilds bereits eine Vorstellung über die Beschaffenheit eines Raumes zu etablieren. In der Architekturfotografie werden bestimmte Konventionen eingehalten, die das Lesen der Bilder erleichtern sollen. Dazu gehören die möglichst planparallele Darstellung von Bauteilen oder das Entzerren senkrechter Perspektiven. Diese Techniken der Verzerrung (die als "Entzerrung" bezeichnet werden) erfordern besonders komplizierte und schwer zu bedienende Fotoapparate. Im Resultat steckt ein hohes Maß an Abstraktion, aber auch an Idealisierung.

Den einen Augenblick gibt es nicht: vier Schichten der visuellen Wahrnehmung

Die physikalisch treue Abbildung eines einzelnen Moments zeigt einen Sonderfall, den der Mensch mit freiem Auge gar nicht erfassen kann. Die Strategie des sequenziellen Abarbeitens verfolgt das Auge schon auf einer sehr unbewußten Ebene: der Sehkegel, der klar erfaßt werden kann hat einen Öffnungswinkel von nicht mehr als zwei Grad aus der Sichtachse. Es ist uns nicht möglich, auch nur an der eigenen ausgestreckten Hand gleichzeitig Daumen und kleinen Finger zu fokussieren. Wir benötigen Zeit für beides. Ein zweiter Sehkegel mit einem Öffnungswinkel von etwa 30 Grad zeigt das, was von uns scheinbar optisch wirklichkeitsgetreu wahrgenommen wird. Dieses Bild wird weniger deutlich erfaßt als der kleinere Sehkegel. Wir erliegen aber einer Illusion und bewerten beide Bereiche gleichwertig,

was sich auch in Fotografie und Malerei ausdrückt. Ein dritter Bereich, das laterale Sehen, das einen Winkel von beinahe 180 Grad umfaßt, tastet die Umgebung ständig nach Veränderungen ab und ist für unsere räumliche Orientierung von großer Wichtigkeit. Dieser Teil unseres Gesichtsfeldes wird in Fotografie, Film und Malerei nicht abgebildet. Es handelt sich dabei um den überwiegenden Teil dessen, was wir sehen und ein unverzichtbares Sensorium für die räumliche Orientierung. Die Halbkugel des Raumes der hinter uns liegt, kann nicht erfaßt werden. Wann immer es uns möglich ist, bemühen wir uns aber, die Erinnerung an diesen Raum präsent zu haben.



Abbildungen oben: Schon eine kurze Wegstrecke, die im Haus zurückgelegt wird kann zu einer starken Veränderung des Raumeindrucks führen. (Der momentane Standpunkt ist rot eingezeichnet, die jeweils sichtbaren Massivbauteile grün, Fenster blau.) In

der Abbildung links oben wird eine Position am Beginn des Korridors zum Kinderzimmer eingenommen. Das gesamte Schlafzimmer und Teile des zentralen Raums sind sichtbar und es gibt eine Sichtbeziehung über die gesamte Länge des Hauses sowie zu bei-

nahe allen Fensterflächen. In der Abbildung rechts oben sind die Haupträume nicht mehr sichtbar, es gibt aber zwei lange Sichtachsen, davon eine ganz neue vom Kinderzimmer bis zur Küche. Geht man weiter (rechte untere Abbildung), öffnet sich der

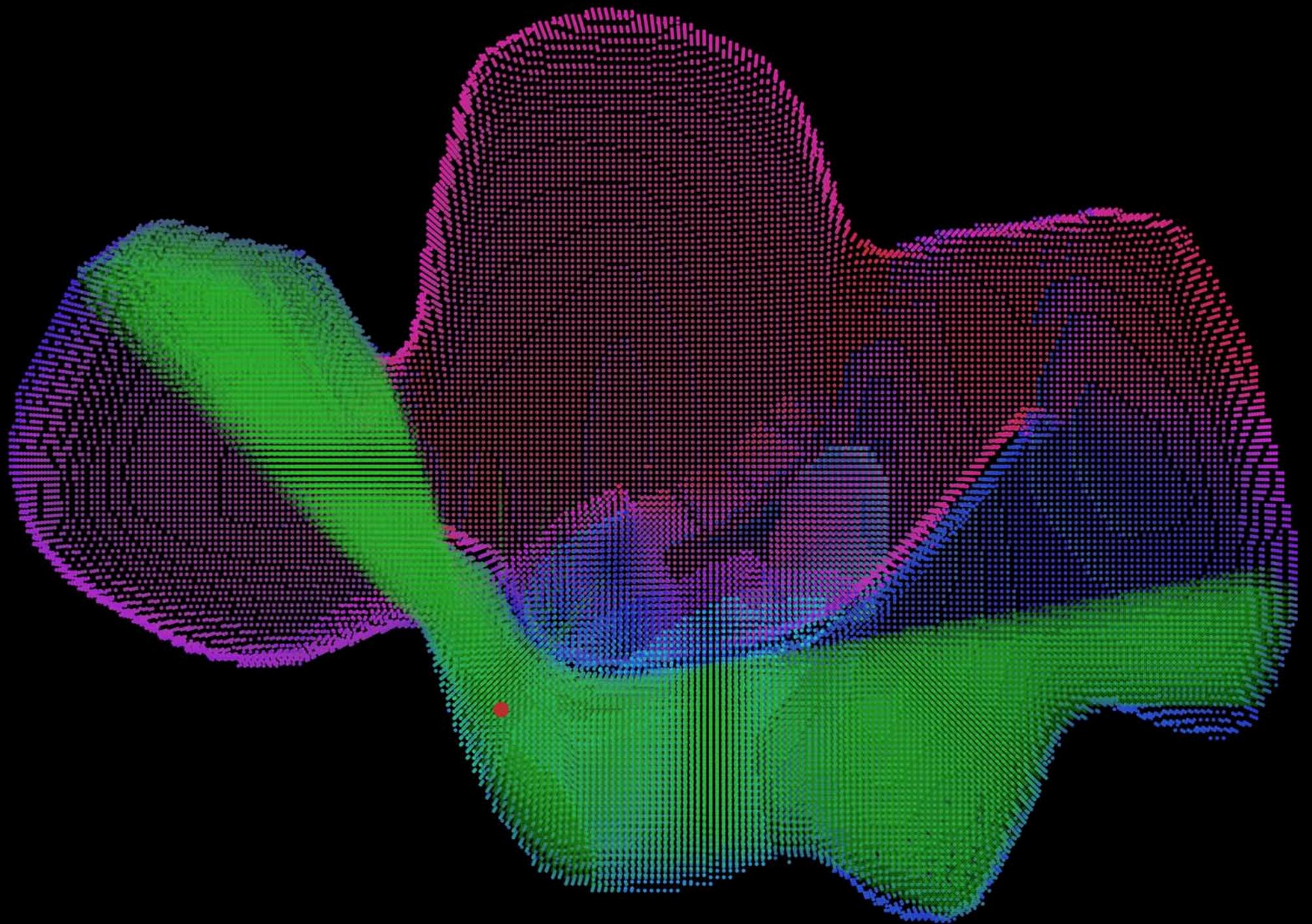
Raum hinter dem Kinderzimmer vollständig und es entsteht eine neue Sichtachse, die längs durch das Haus verläuft. Im Bild rechts unten schließlich ist der Sehraum beinahe vollständig komplementär zum Ausgangspunkt.

Die Darstellung von Gleichzeitigem und Langzeitigem

Eine fünfte Schicht: die Erinnerung

Als Edward Muybridge erstmals Momentaufnahmen eines galoppierenden Pferdes anfertigte, nützte er sich eine besondere Eigenschaft der Kamera: sie hat kein Erinnerungsvermögen und kann nur den jeweils letzten Augenblick sehen. Die Bilder waren eine Sensation, da sie das zeitgenössische Verständnis dessen, was sichtbar sein kann, stark veränderten. Der Eindruck des Sensationellen ist heute nicht mehr nachvollziehbar: Muybridges Zerschneidung

der bewegten Wirklichkeit in Einzelbildkader ist eine Art zu sehen, die uns überaus natürlich erscheint. Dass der Faktor Zeit aus dem Sehen nicht herauszunehmen ist, wurde vorhin schon angedeutet. Die Fotografie und die mit ihr verwandte Technik, naturgetreue Lichtsituationen im Computer zu simulieren, das Rendering, wollen den kleinstmöglichen Augenblick sichtbar machen. Um die Erinnerung miteinander zubeziehen, muß man versuchen, etwas anderes darzustellen: Simultanität.



Der Raum und seine Adressierung

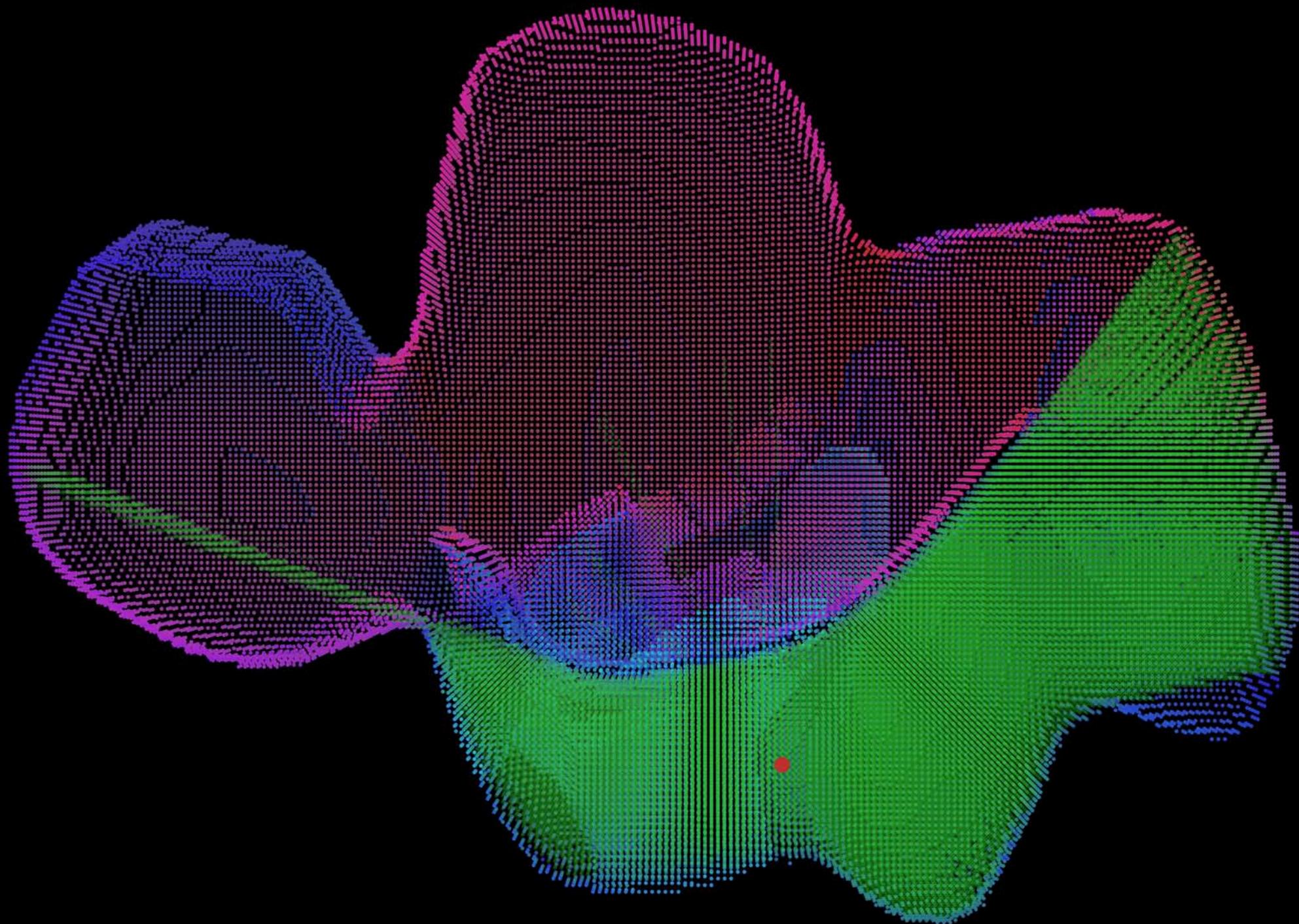
Der Raum ist überall wo wir sind und wir spüren ihn nicht. Warum auch sollte ein Lebewesen ein Sinnesorgan für einen konstanten Wert ausbilden?

Wir tasten die Welt nach den veränderlichen Werten ab, denen wir ausgesetzt sind. Druck, Temperatur oder Lichtstärke verwandeln sich und wandeln damit auch ihre Umgebung in etwas anderes um. Tritt das eine oder andere in unpassender Menge auf, dann verändert es unseren Körper auf sehr unmittelbare Weise und wir spüren Schmerz.

Das Bild vom Raum ist im Gegensatz dazu ein abstraktes Konzept. Der Sehsinn ist im Bereich der Netzhaut zweidimensional angelegt. Spätestens in den Nervenbahnen zum Gehirn geht auch diese zweite Dimension verloren. Doch schon eine kleine Winkelschiebung von Auge zu Auge hilft dem Denkapparat, uns räumliche Tiefe "sehen" zu lassen. Diese Sinnesreize könnten auch völlig anders aufbereitet werden, doch so wie sie sind, werden sie zu unserer Wirklichkeit.

Es gibt keinen "Raum" in unserem Gehirn, aber es muss Vorgänge darin geben, die zur Illusion des Raumes führen.

In der Computergrafik wird versucht, aus einer Maschine heraus ein Bild des Raumes auf eine "Netzhaut" zu bringen, die ein Bildschirm sein kann. Auch im Computer gibt es keinen "Raum" der manipuliert wird. Wir kennen den Prozeß der Bilderzeugung im Rechner sehr gut und setzen einige abstrakte Konzepte ein, um zu einem Ergebnis zu kommen. Es besteht die Möglichkeit, dass die Bildverarbeitung in Computer und Gehirn durch Phänomene ermöglicht wird, die von ihrem Wesen her nicht völlig verschieden sind.



Das manipulative Element in der Bildprojektion

Die Parallelprojektion wird zwar aus einem bestimmten Blickwinkel gesehen, sie trifft aber keine Aussage über die Distanz, aus der sie zu betrachten ist.

In der Perspektive wird der Standpunkt des Individuums zum unveränderlichen Fixpunkt, der innerhalb der dargestellten Sache gefangen ist und um den herum alle Dinge ihre Gewichtung nach Größe erhalten. Die Perspektivdarstellung stellt alle, die sie betrachten, auf diesen Fixpunkt. Dieser Effekt ist ein Instrument der Manipulation, das nur jemand in Händen haben kann, dem die Freiheit gewährt wurde, einen Standpunkt einzunehmen. Mit dem Aufkommen des Humanismus in der Renaissancezeit wurde die subjektive Sichtweise zur bevorzugten Bildprojektion.

Es scheint paradox, dass die Technik der Perspektive als Ausdruck subjektiver Wahrnehmung gleichzeitig dem Anspruch folgen kann, ein äußerst objektives Abbild der Welt zu erzeugen. Denn trotz der Unendlichkeit möglicher Standpunkte folgt sie als geometrisches Axiom einem Satz von Prinzipien, die nicht verändert werden dürfen. Folglich führen gleichartige Voraussetzungen zum immer gleichen Ergebnis. Die Parallelprojektion und vor allem das Diagramm erfüllen diese Anforderung nicht.

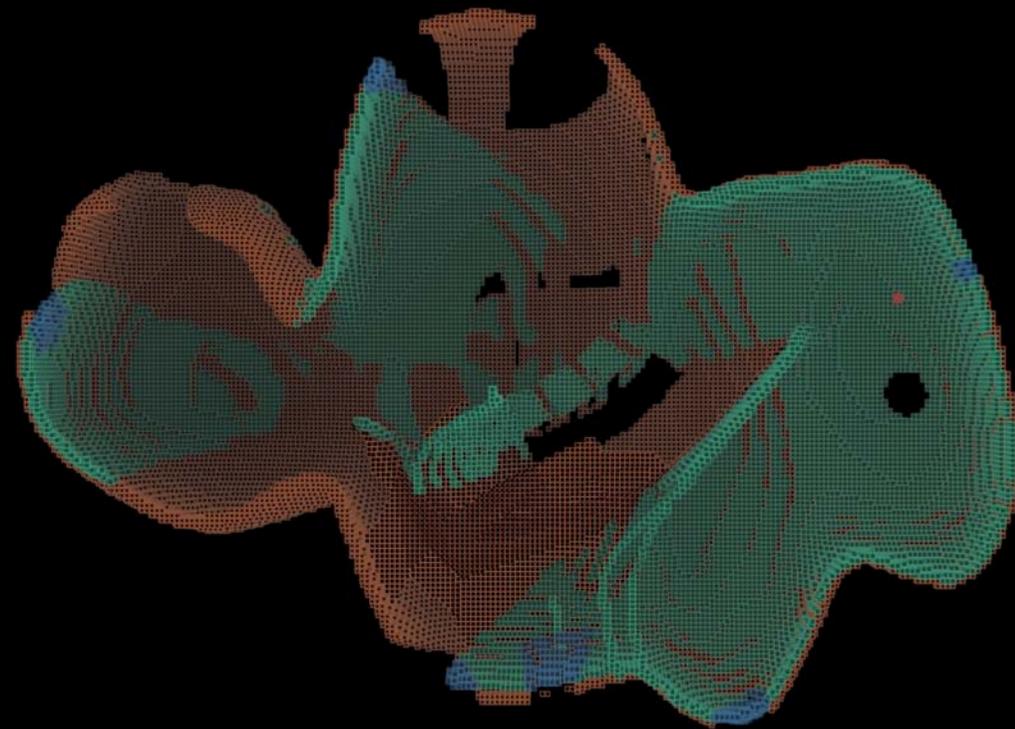
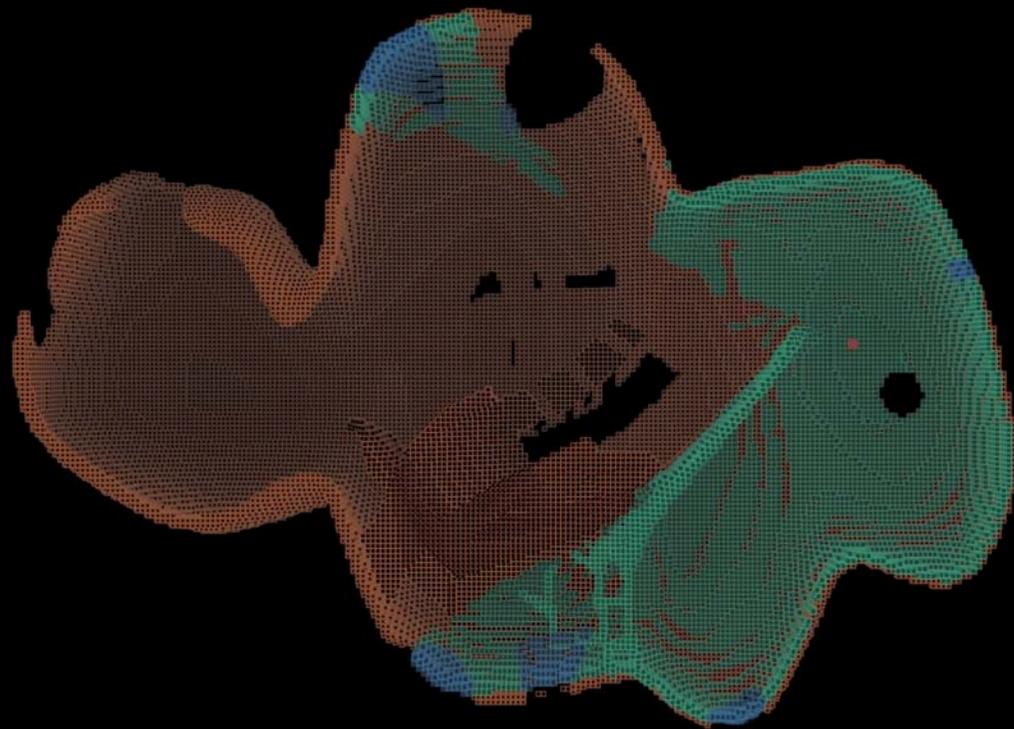
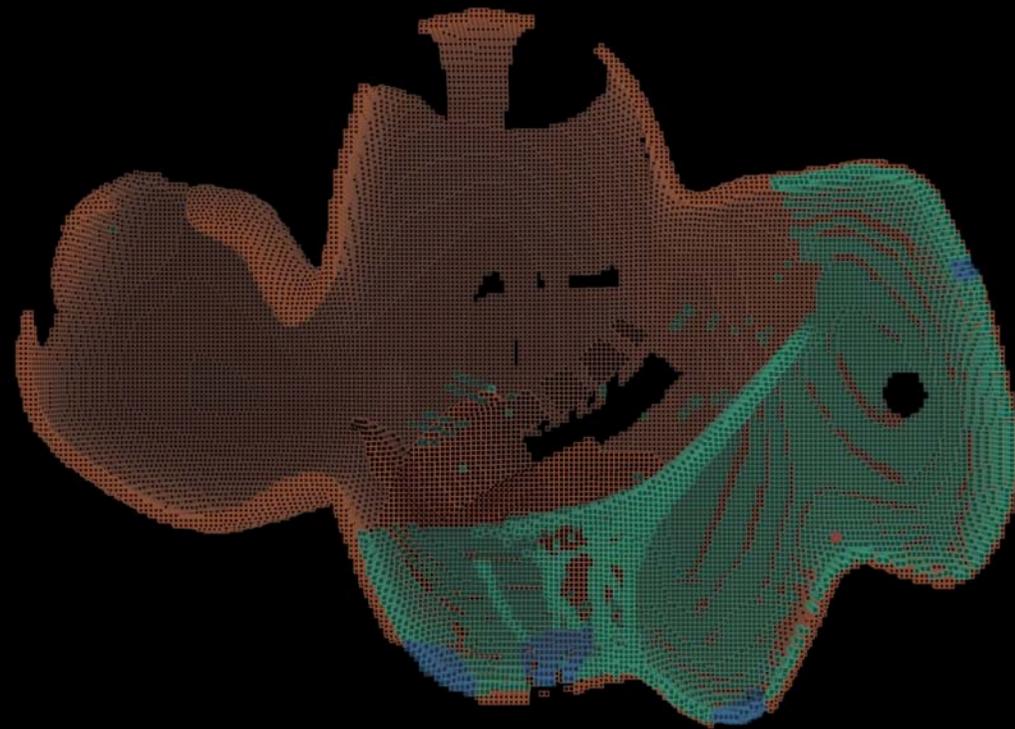
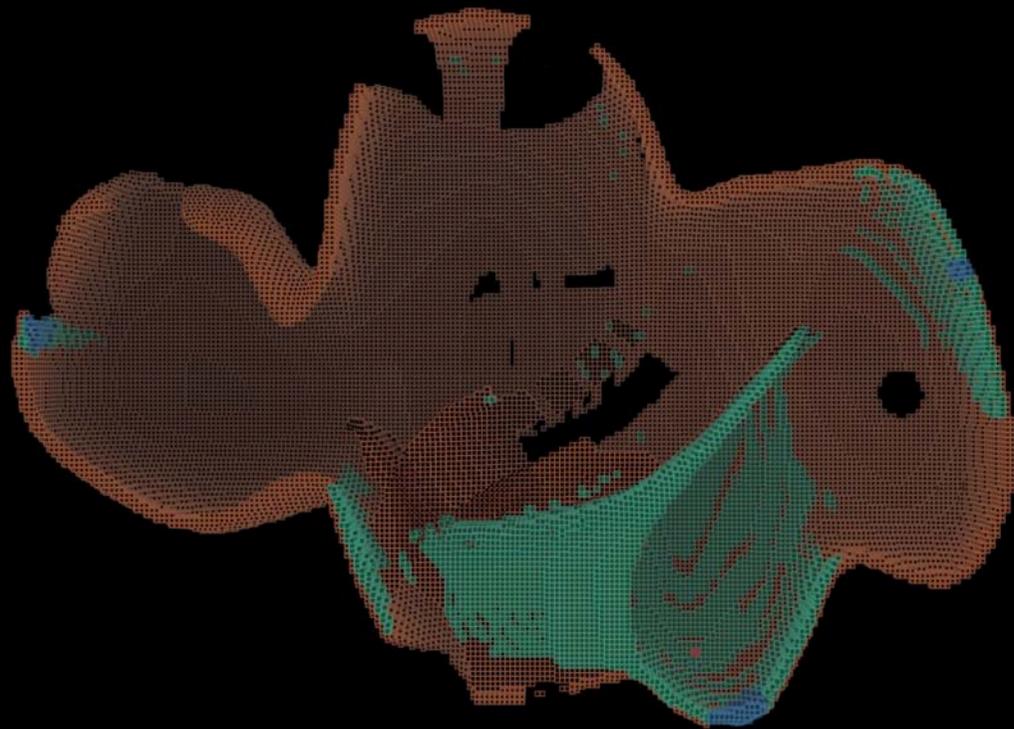
Wir können auf einfache Weise räumliche Informationen aus einem Koordinatensystem in eine Perspektivdarstellung umcodieren. Umso schwieriger ist es, den umgekehrten Weg zu gehen und aus der Perspektive verlässlich räumliche Beziehungen abzuleiten.

Worauf zielen diese langwierigen Überlegungen ab? Die Perspektive ist ein anerkanntes Mittel, eine Raumsituation darzustellen. Sie schafft aber auch Unklarheiten, verzerrt und bietet nur ein eingeschränktes Gesichtsfeld.

Gesucht wird eine Art Meta-Perspektive, die das Gesehene zeigt, aber eben nicht aus der Sicht des Betrachters.

Und ausgehend von dieser Meta-Perspektive sollen alle denkbaren Perspektiven zusammenfassend gezeigt werden können. Diese Anforderung ist das grundlegende Prinzip der Software, die hier gezeigt wird.

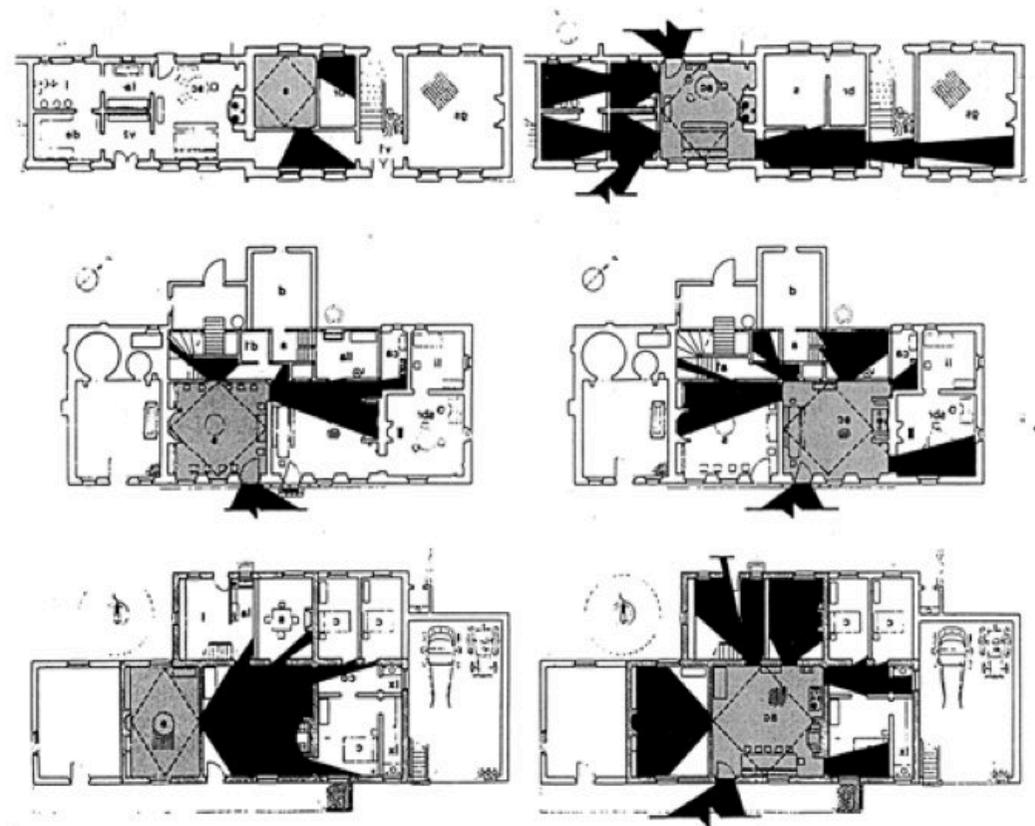
Das Bild oben zeigt die Entstehung einer Meta-Perspektive. Vom Augpunkt (rot) ausgehend werden Lichtstrahlen (grün) allseitig in den Raum gelegt.



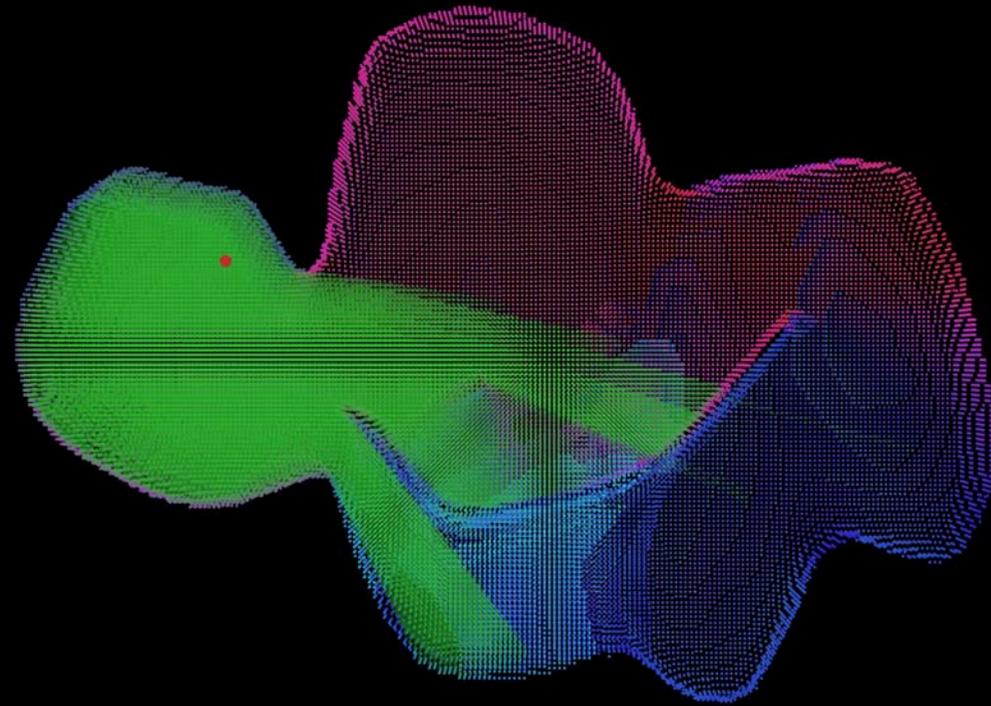
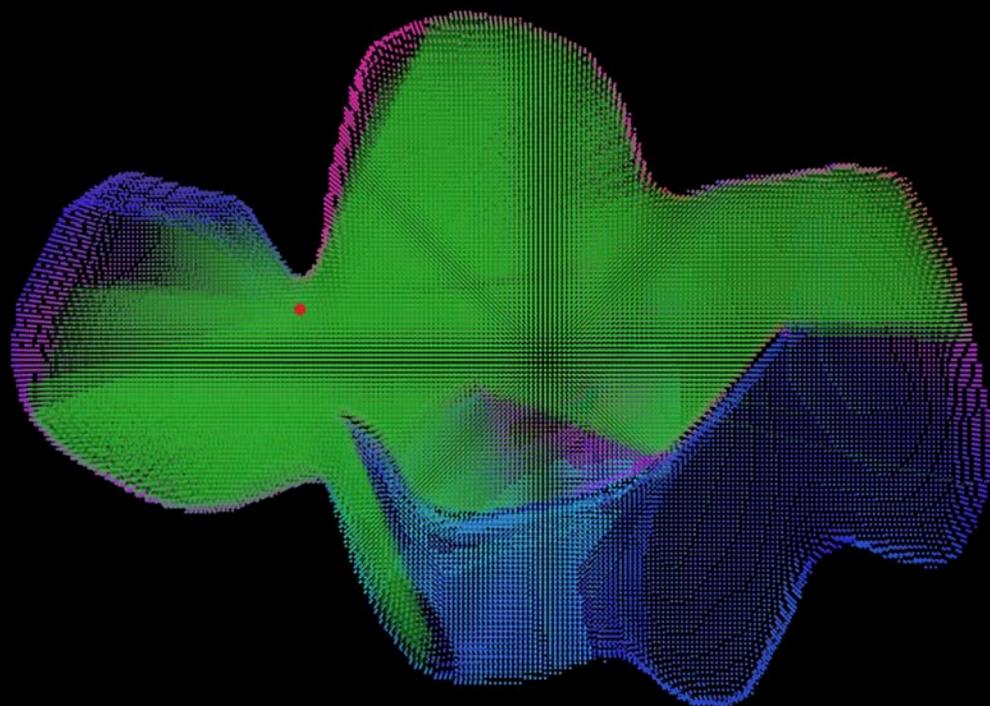
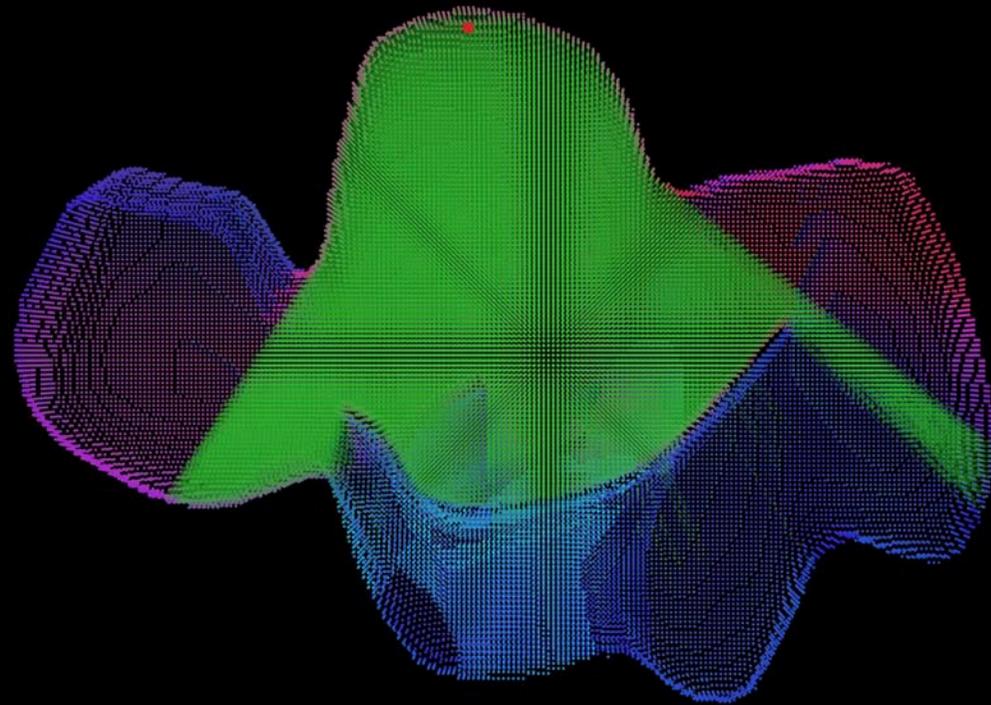
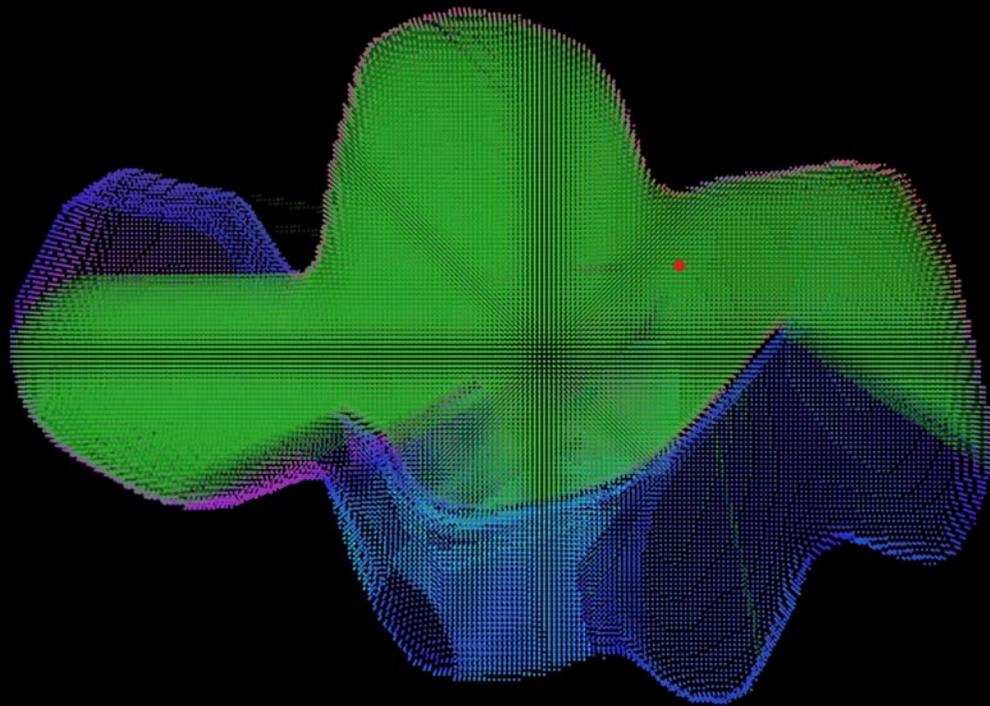
Die Maschine-Die-Alles-Simultan-Sieht

Die nebenstehende Illustration von Hillier stellt bereits diagrammatisch dar, was in späterer Folge als Simulation im Computer entstehen soll: die Projektion von Lichtstrahlen ist das zentrale Element, um das herum die gesamte Software aufgebaut ist. Ein sogenannter Raytracer verfolgt üblicherweise den Weg vom Auge zum Objekt und erstellt jeweils einen einzelnen Bildpunkt auf der Bildebene - so lange, bis das gesamte Bild sichtbar wird. Aus dem Diagramm und obenstehendem Farbbild geht hervor, dass man auch einen anderen Weg verfolgen kann: hier steht die Darstellung des Sehstrahls im Mittelpunkt. Der Eindruck ähnelt dem einer Taschenlampe, die durch Nebel leuchtet, was den Weg des Lichts durch die Luft sichtbar werden lässt. Im Unterschied zu der physikalischen Wirklichkeit, können im Computer bestimmte Kenngrößen abgefragt werden, etwa die Anzahl der ausgeleuchteten Raumeinheiten.

Die Illustration zeigt die Lage sichtbarer Bereiche im Inneren eines Wohnhauses. Abbildung aus "Space is the machine" von Bill Hillier.⁷

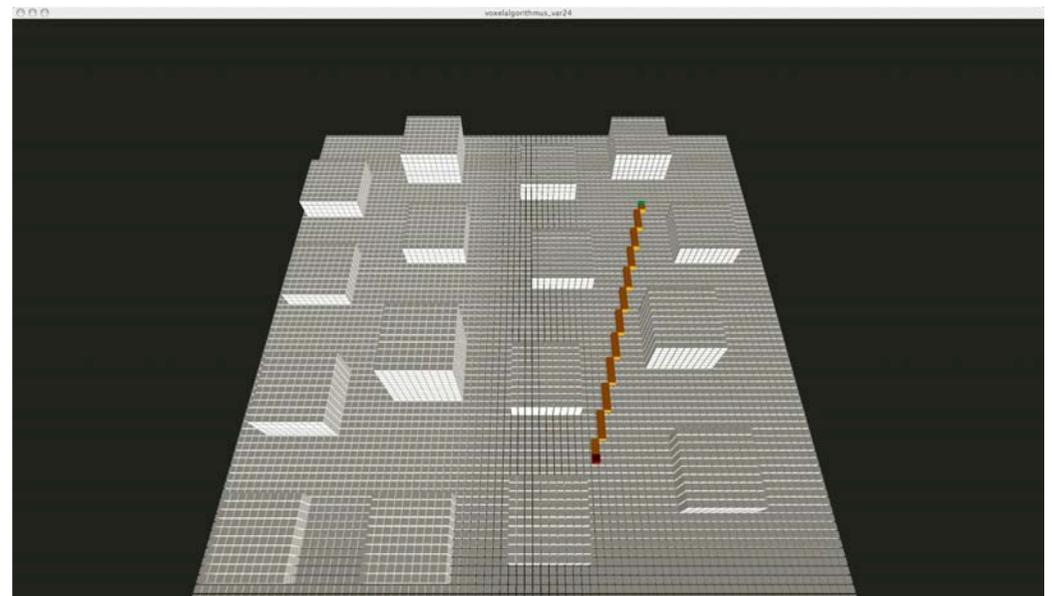


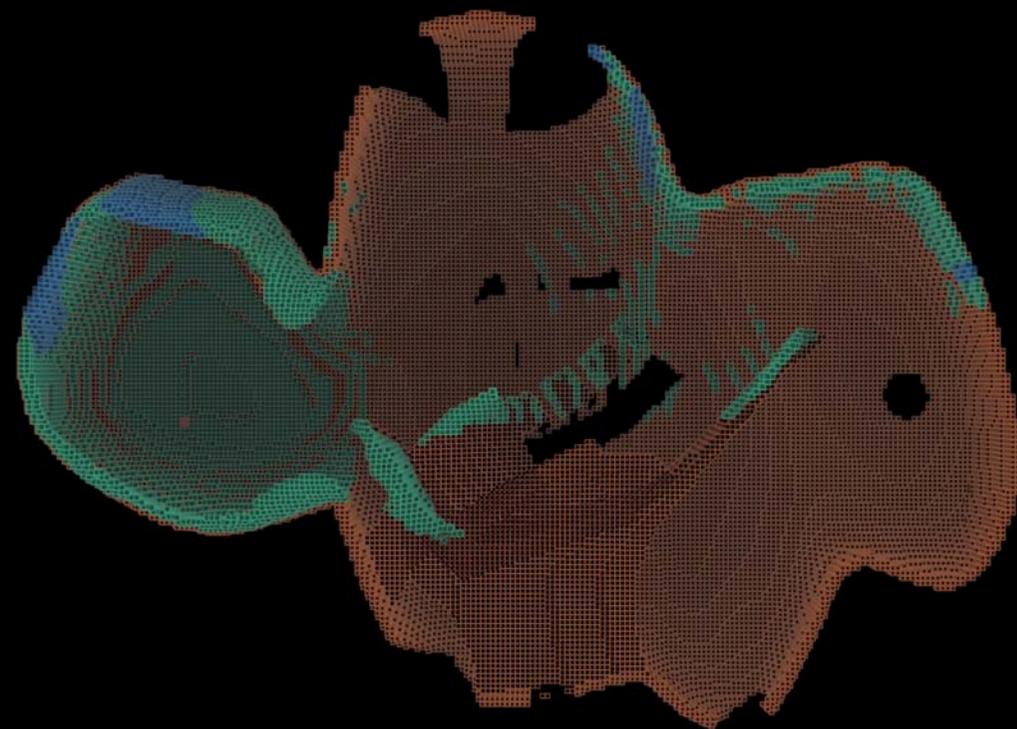
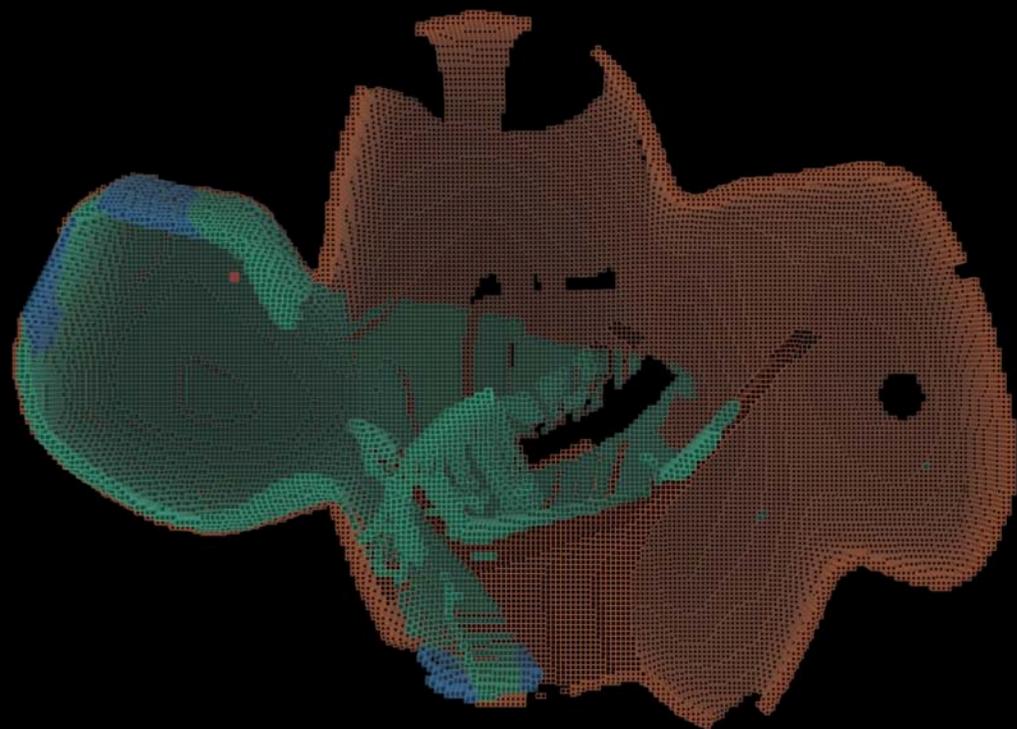
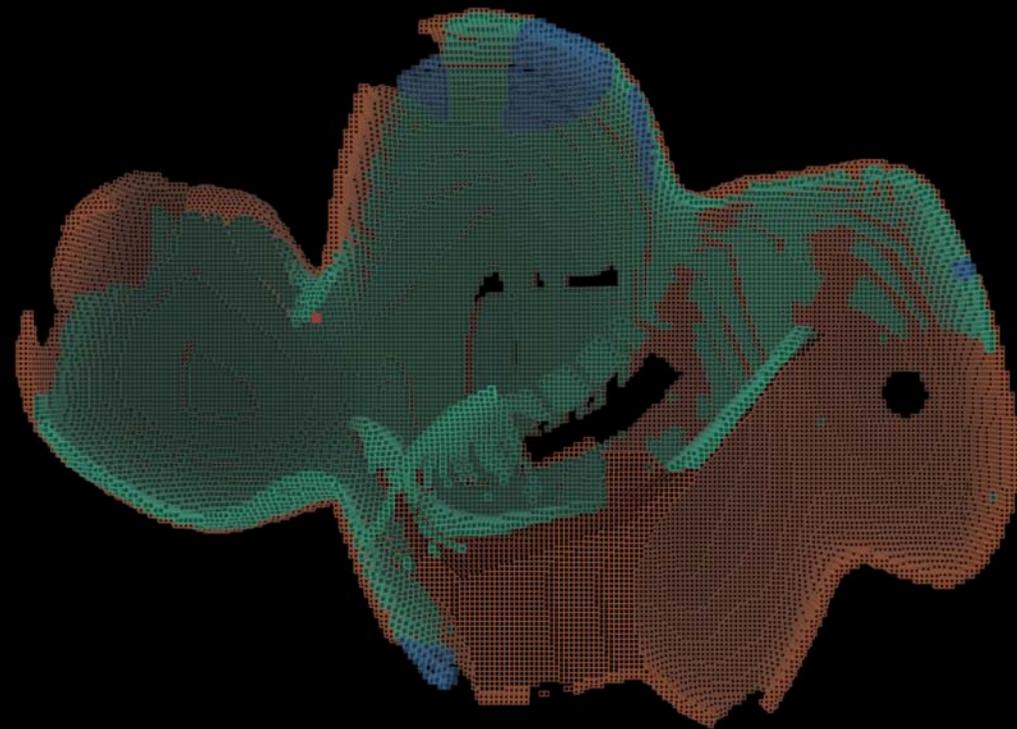
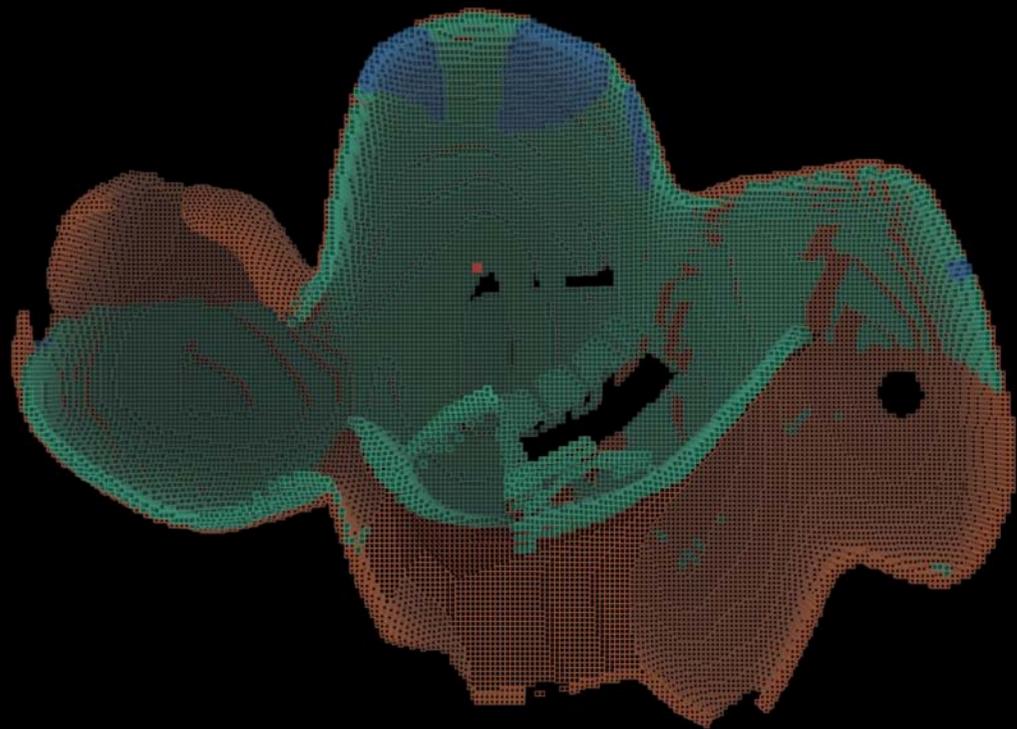
7) Hillier 2007, 2-26.



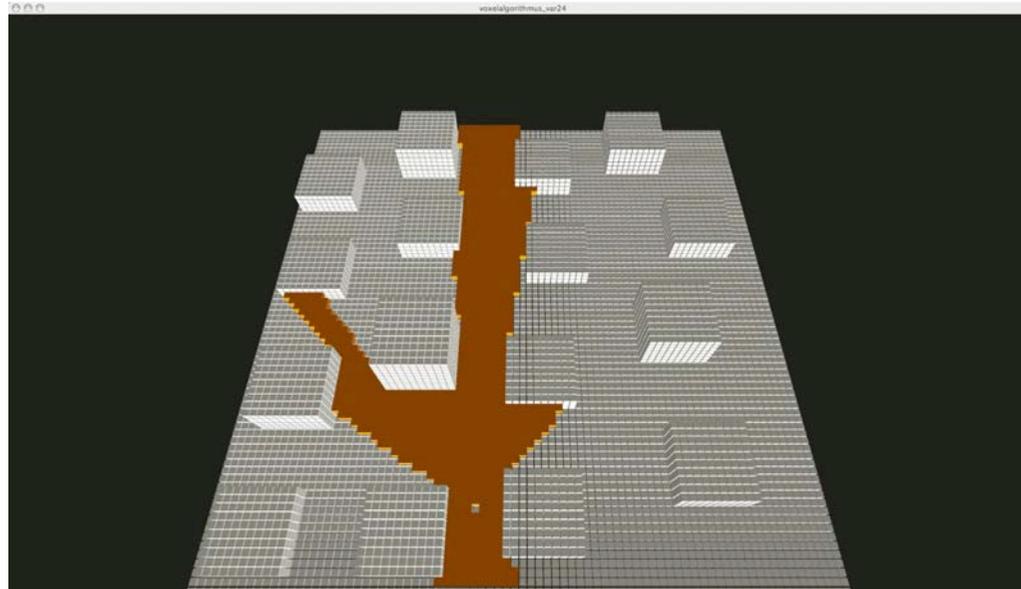
Am Beginn steht der einzelne Sehstrahl oder seine abstrakte Repräsentation. Aus den Illustrationen im Buch wird augenfällig, dass die Repräsentation des Raumes kein Kontinuum darstellt, sondern in würfelförmige Einheiten gebrochen ist. Das liegt an der Art der Adressierung des Raumes, in dem ein an sich eindimensionaler Punkt im Raum jeweils für ein kleines Raumvolumen steht. Linien, die den Raum durchqueren, können in diesem System nicht gerade verlaufen, wie das in jedem trigonometrischen Verfahren grundsätzlich der Fall ist, sondern müssen einen möglichst kurzen Weg durch diese Raumpunkte gehen. Man berechnet also für jede Koordinate, welcher Raumpunkt einer geraden Linie am nächsten kommt und setzt diesen Punkt in den Sehstrahl ein. Dieses Verfahren muß bei jeder Bildschirmdarstellung angewendet werden, wenn der Bildschirm aus orthogonalen Rasterpunkten besteht. Der Vorteil gegenüber der reinen trigonometrischen Berechnung ist die schnelle und eindeutige Adressierung von Raumpunkten durch ganze Zahlen. Da wir ja herausfinden wollen, auf welchen Raumwürfel ein Sehstrahl auftrifft, steht uns damit eine schnelle Methode zur Verfügung, die ausreichend genau ist.

Ein einzelner Sehstrahl in einem frühen Prototypen der Software. Die Versprünge zwischen den Voxeln sind gut erkennbar.

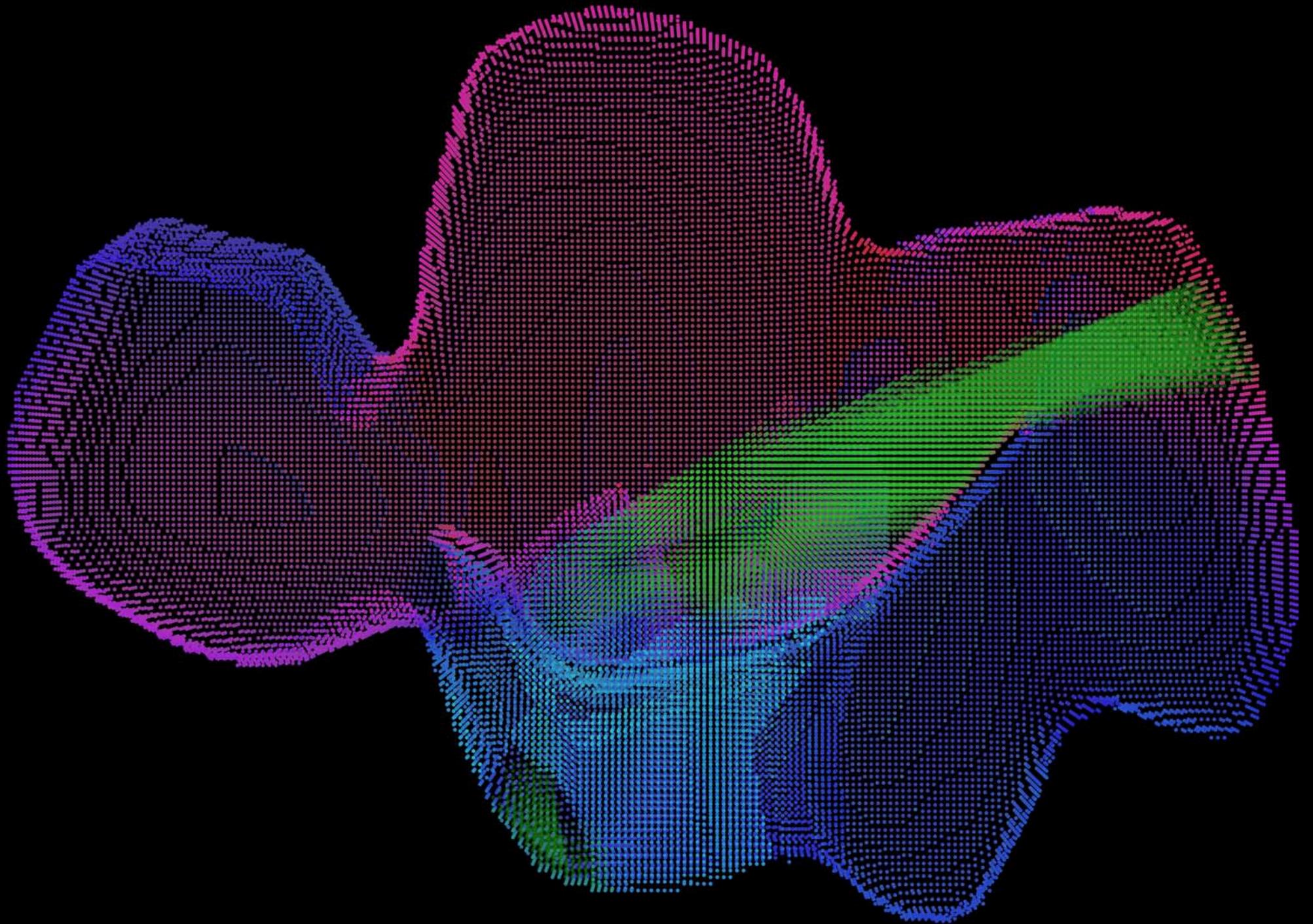




Die Bilder zeigen einen Prototypen, der in der Lage ist, in einer Ebene Sehstrahlbündel zu zeichnen. Diese Funktion ist analog zu der in der Illustration von Hillier gezeigten.



Dieser Sehstrahl wird von einer Funktion erzeugt, die die Software ständig aufruft, wenn sie ein Bündel von Sehstrahlen zeigen soll. Wie die Taschenlampe, die Photonen aussendet, werden auch die einzelnen Punkte der Sehstrahlen vom Augpunkt weg nach aussen geschickt. Feste Körper verschlucken die Photonen in der Realität - im Computerprogramm muß eine Randbedingung dafür sorgen, dass das Erzeugen von Punkten endet, sobald der Sehstrahl im selben Raum steht wie ein fester Körper. Dazu ist nur eine einfache Vergleichsoperation nötig. Die Standpunkte aller festen Körper sind in einer Liste gespeichert, die die Raumkoordinaten beinhaltet. Die oben erwähnte einfache Adressierungsmethode, erlaubt es uns, den Strahlengang abzubrechen, sobald ein Lichtpunkt an der selben Stelle steht wie ein fester Körper. Ist das nicht der Fall, wird der Lichtpunkt in einer Lichtpunktliste gespeichert und ein nächster Lichtpunkt gesucht. Wenn am Schluß eines Durchlaufs alle Strahlen gelegt worden sind, enthält die Lichtpunktliste alle beleuchteten Volumina des Leeren Raums. Eine Funktion kann ihre Anzahl messen und eine weitere kann die Lichtpunkte darstellen.

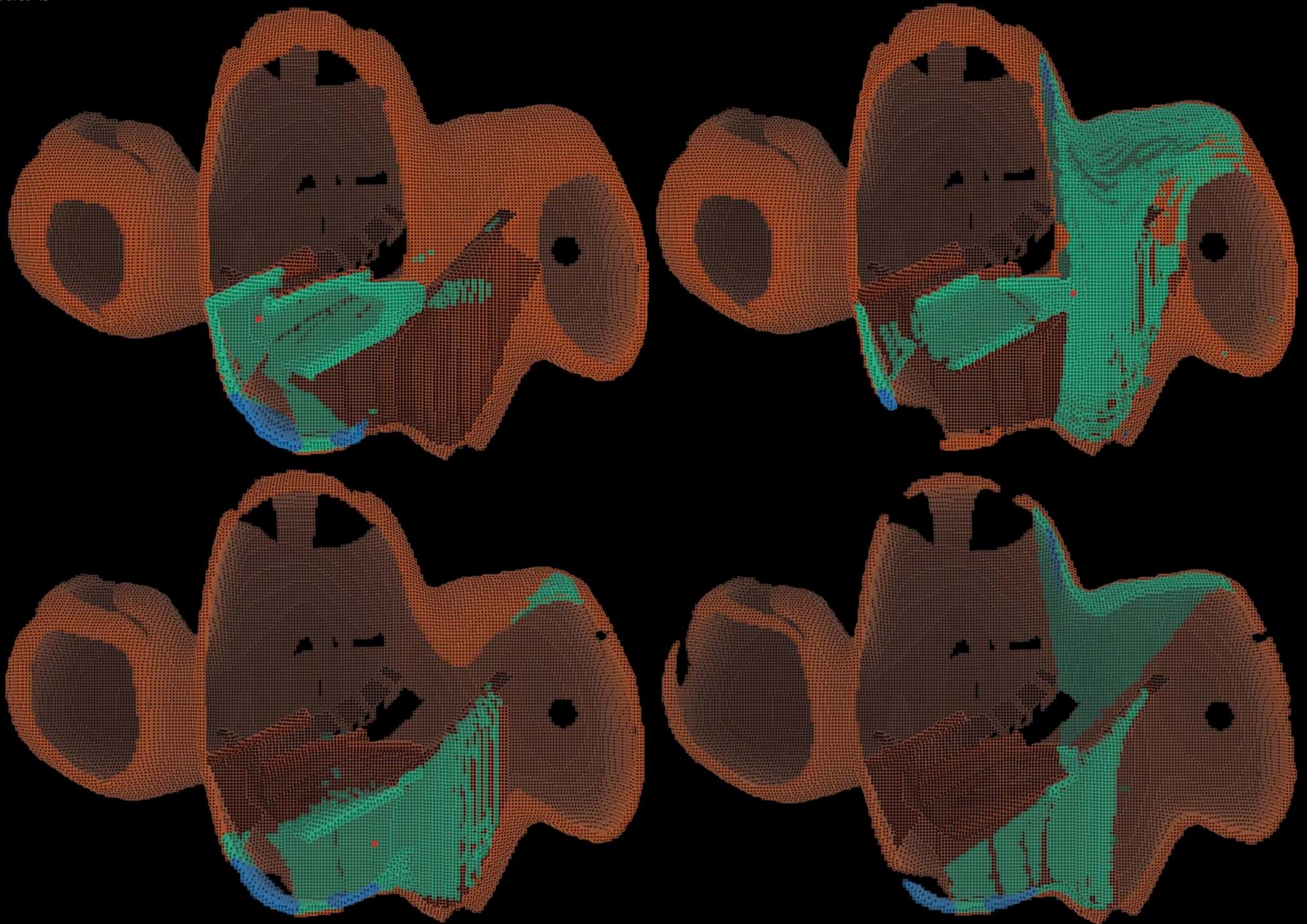


'Each and every one of the space-nuclei can be separated from the totality of the dwelling, secluded. At will you can reunify to meet various needs: the congregation of the family, of visitors from the outer world, neighbors, friends, strollers. Or again, you'll womb yourself into happy solitude.'

- Friedrich Kiesler

Die so gewonnene Information über den Verlauf des Sehstrahls kann uns jetzt von praktischem Nutzen sein. Denn wir kennen nicht nur den Lichtkörper, den die einzelnen Sehstrahlen bilden, sondern auch die Punkte, an denen sie von Festkörpern unterbrochen werden. In einigen der oben gezeigten Abbildungen sind die derart ermittelten sichtbaren Flächen gut ablesbar (blaugrüne Hervorhebung).

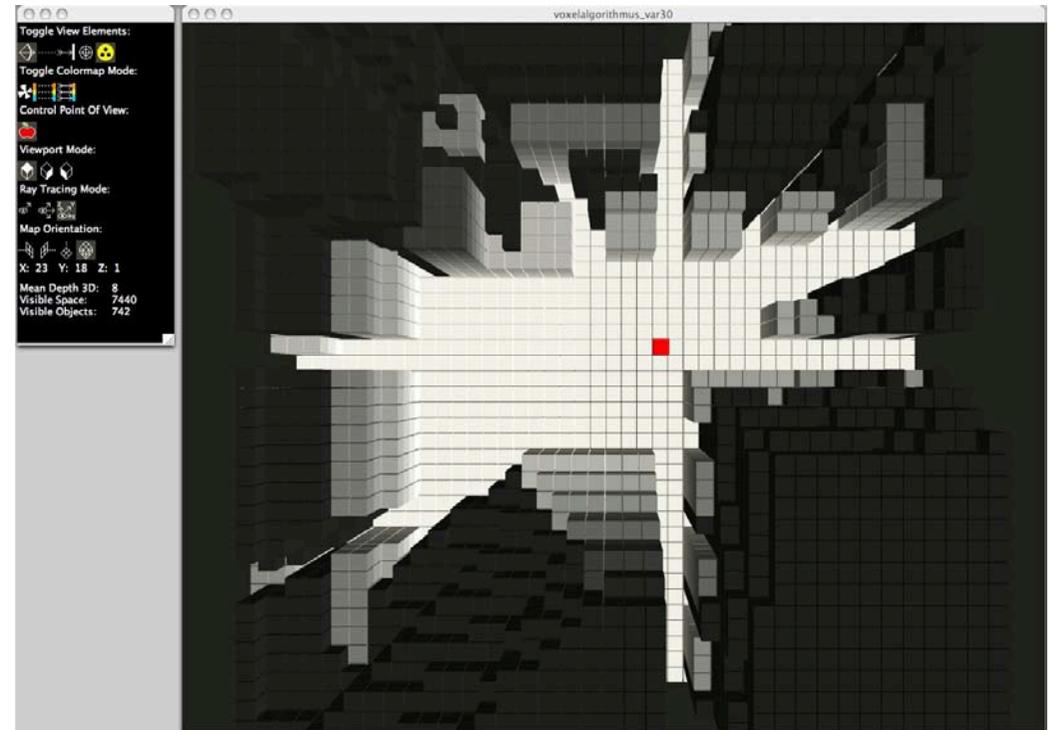
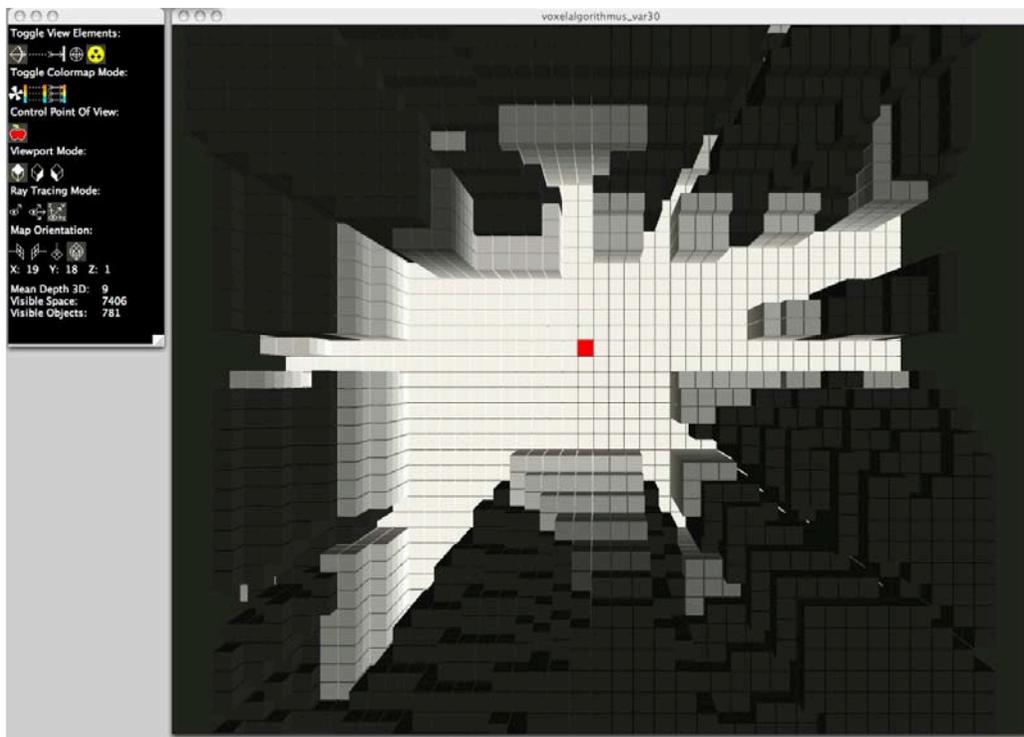
Es zeigt sich deutlich, welche vielfältigen Raumsequenzen entstehen, wenn man das Haus durchquert. Die Anzahl der Raumkonfigurationen ist wesentlich höher als die, die sich aus Grundriß oder Schnittdarstellung ablesen lässt. Die diagrammatische Art der Darstellung ist wesentlich klarer, als es eine Zentralperspektive jemals sein kann: es sind auch die nicht sichtbaren Bereiche ausgewiesen. Das Zitat von Kiesler kann als eine Bezugnahme auf diese Eigenschaft des Raumes verstanden werden: in das eine unifizierte Raumvolumen ist eine Vielzahl von Subvolumina eingearbeitet, die sich überschneiden oder voneinander unterscheiden.

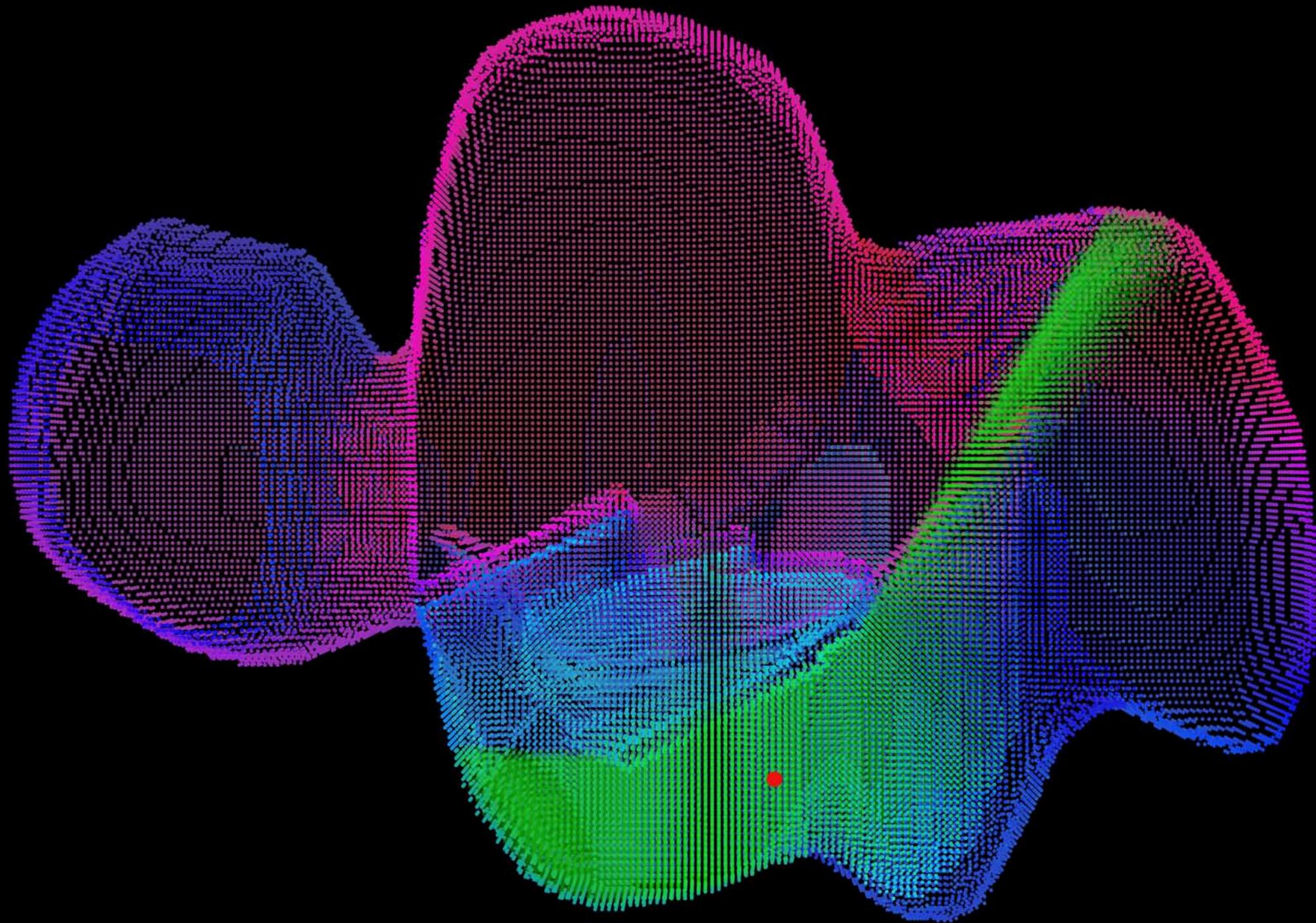


Da das Programm in der Lage ist, die sichtbaren Bereiche eines Körpers in sehr kurzer Zeit zu finden, kann die Simulation in Echtzeit stattfinden. Es entsteht dadurch ein sehr klares Bild darüber, wie sich Räume öffnen und verbergen

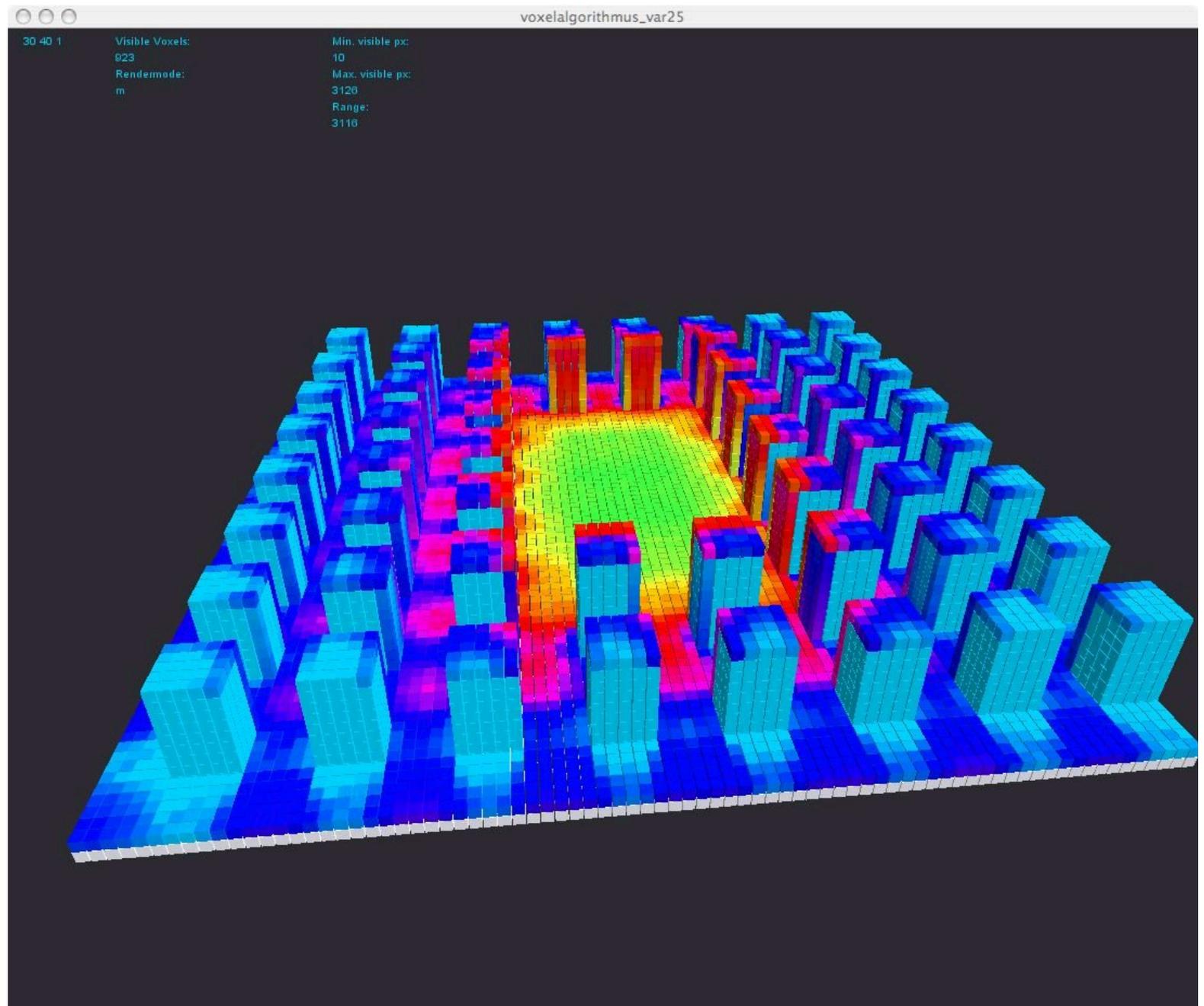
Wir werden in der Folge sehen, welche Informationen man gewinnen kann, wenn man die Daten aus einzelnen Augpunkten miteinander vergleicht

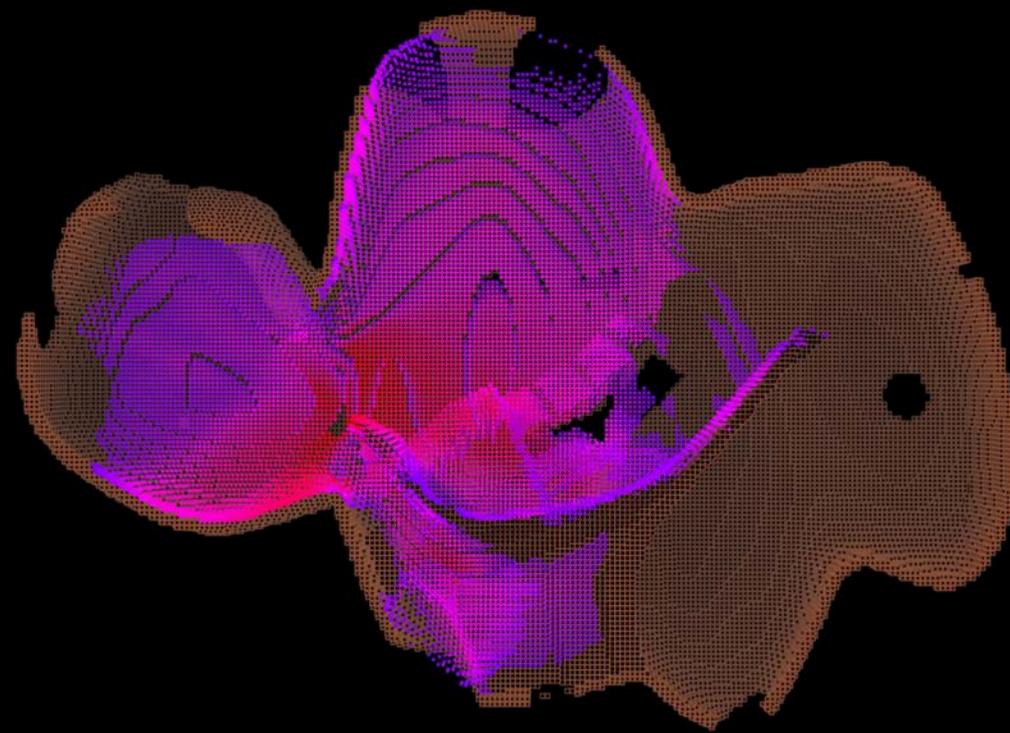
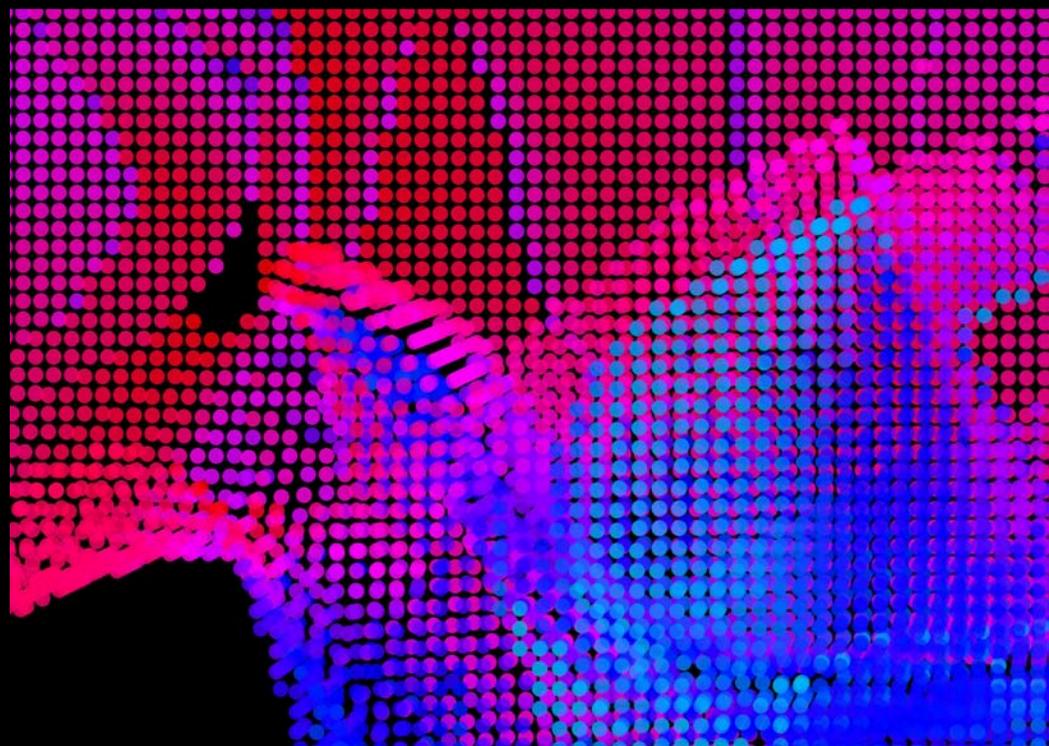
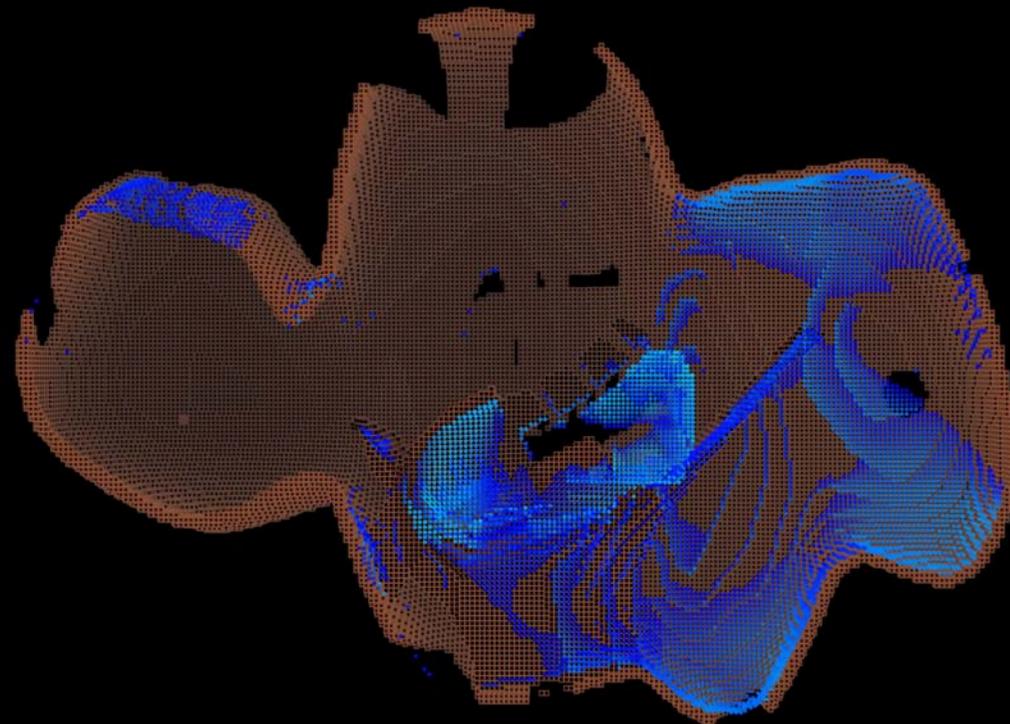
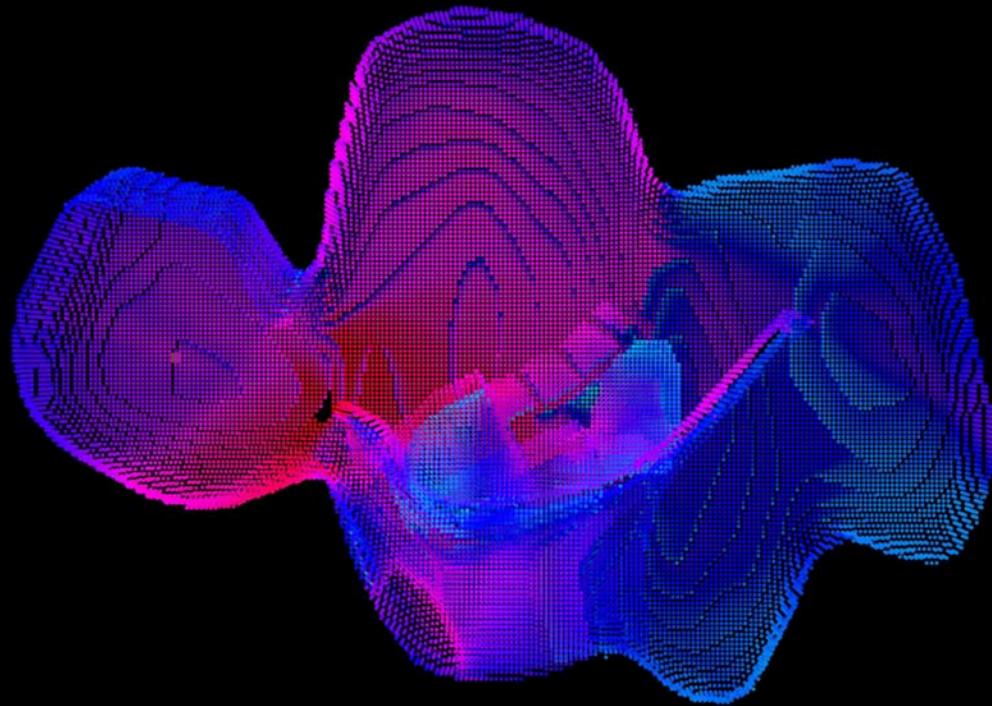
Die beiden Illustrationen auf dieser Seite geben einen Eindruck davon - nicht sichtbare Bereiche wurden durch schwarze Pixel verdeckt. Es entsteht ein Negativraum, der dem Sehkegel entspricht.





Die Daten, die über einen einzelnen Augpunkt gewonnen wurden, können auch mit denen aller anderen Augpunkte verglichen werden. aus dem Vergleich von Werten entstehen dreidimensionale farbige Landkarten. Das Bild zeigt ein abstrahiertes Stadtmodell. Gemessen wurde, wieviele Festkörper von dem jeweiligen Augpunkt aus sichtbar sind. Hellblau markiert den niedrigsten Wert, Rot den höchsten.





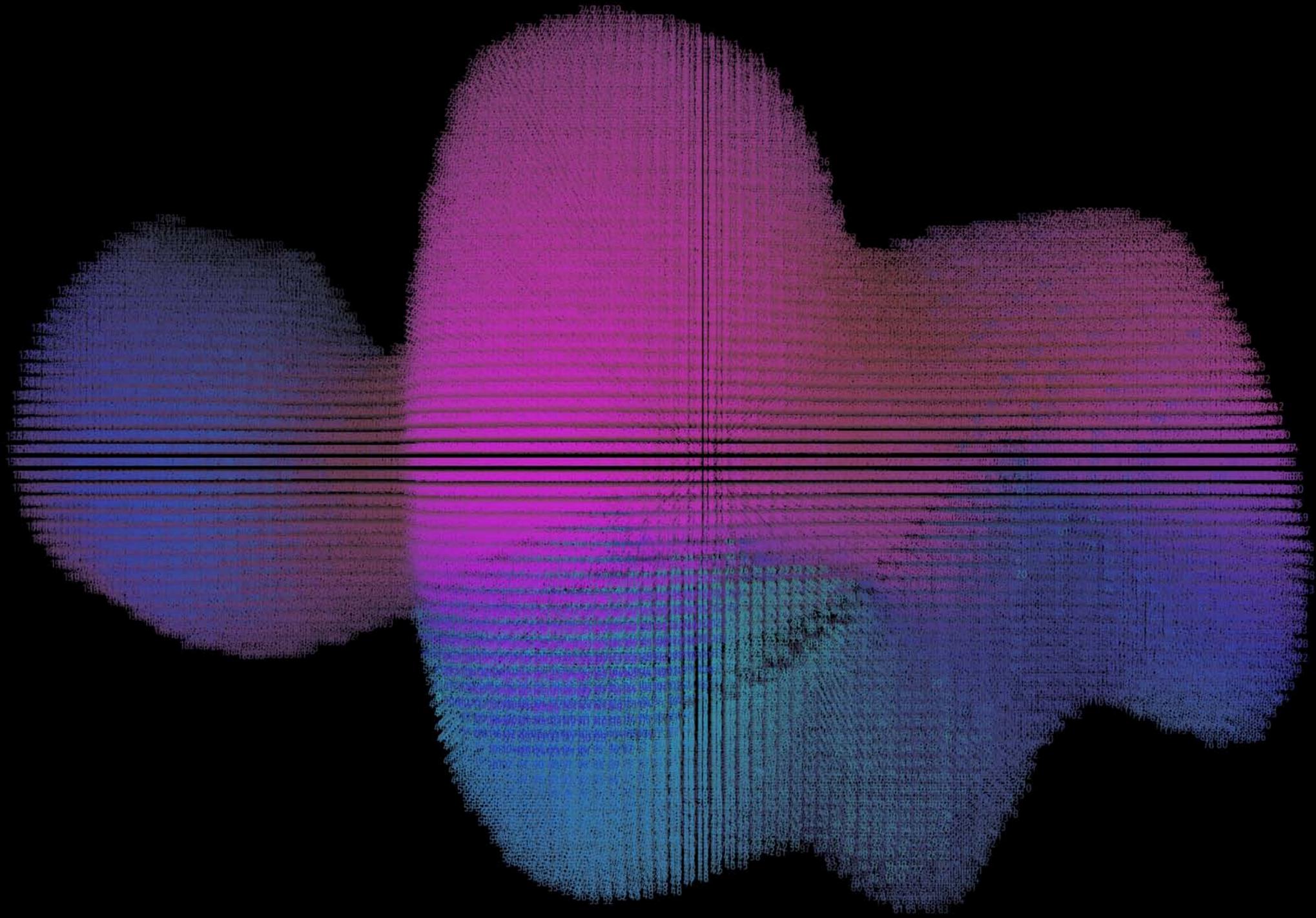
Dreidimensionale quantitative Raumlandkarten

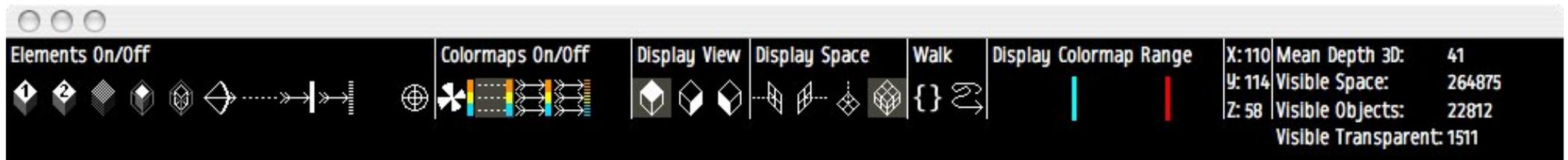
Sobald wir einen Raum betreten, beginnen wir, ihn in Gedanken in verschiedene Zonen zu gliedern. Wir sehen Gebiete, in denen die Übersicht möglicherweise größer ist und andere Gebiete, die einen Rückzugsraum darstellen. Die Anzahl der Sichtbeziehungen zu anderen Raumteilen spielt dabei eine wesentliche Rolle. Mit Hilfe einer quantitativen Analyse kann die Software dieses Verhalten nachvollziehen. Sie greift dabei zu einem etwas langwierigeren Verfahren: es wird einfach jeder Punkt an dem interessierenden Ort besucht und bewertet.

Diese Wertung wird in Listen eingetragen - sie umfaßt für jeden Augpunkt die Anzahl der sichtbaren festen Körper, die Anzahl der sichtbaren transparenten Körper, das Volumen des Luftraums, der sichtbar ist und die mittlere Entfernung aller sichtbaren Objekte vom Augpunkt. Diese vier Listen können jeweils grafisch aufgearbeitet werden. Eine Zeichnungsfunktion bestimmt erst den Tiefst- und den Höchstwert in einer Liste und bemißt das Farbspektrum, mit dem alle Werte erfaßt werden können.

Es ist weiters möglich, diese Werte für alle Punkte im Innenraum zu bestimmen oder sie nur für die Randzonen abzufragen. Die oben gezeigten Bilder weisen die Anzahl sichtbarer Objekte entlang der unteren Schalenhälfte des Hauses aus. Blau steht für geringe Sichtverbindung, Rot für hohe Sichtbarkeit. Das Bild am oberen linken Rand zeigt eine Gesamtzusammenstellung dieser Punkte, das Bild darunter ein Detail derselben Zeichnung. Alternativ dazu können auch Wertebereiche separat dargestellt werden. Das Bild am rechten oberen Rand zeigt niedrige, das Bild darunter hohe Werte.

Es wird schnell deutlich, wo im Haus die Rückzugszonen liegen und wo man eher exponiert ist. Bei den Zonen mit geringer Exposition handelt es sich um an den Rand angelagerte nischenartige Räume, während der große Zentralraum natürlich gute Sichtbarkeit aufweist - das gilt auch für Teile des Schlafzimmers (die äußerste linke Raumzelle), was eine gewisse Überraschung darstellt.



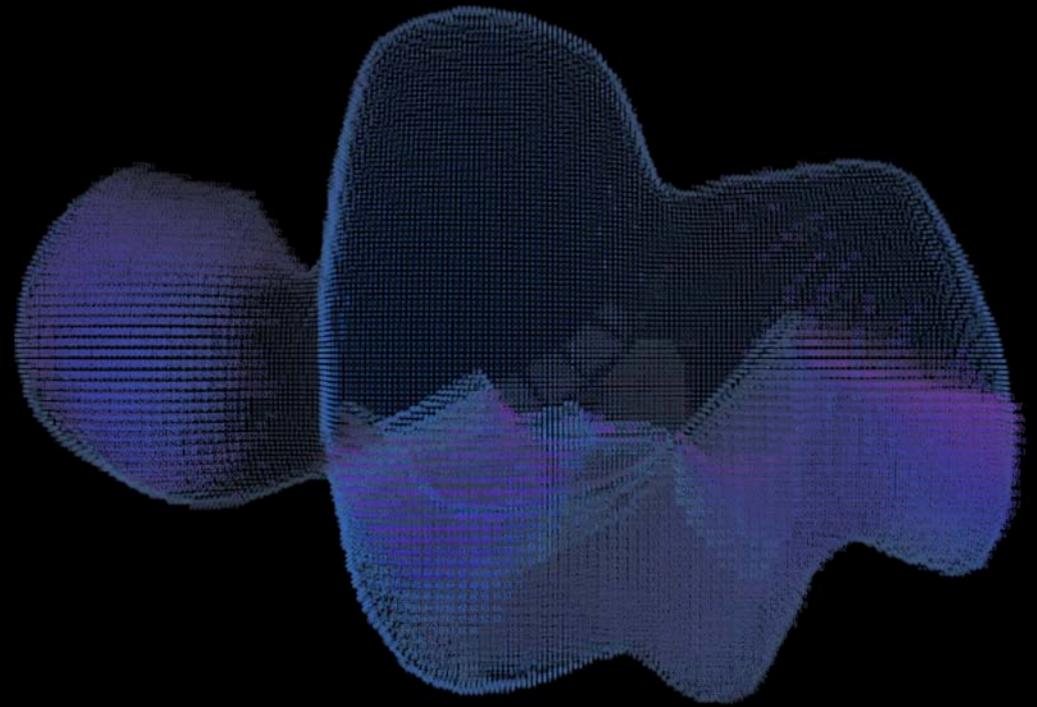
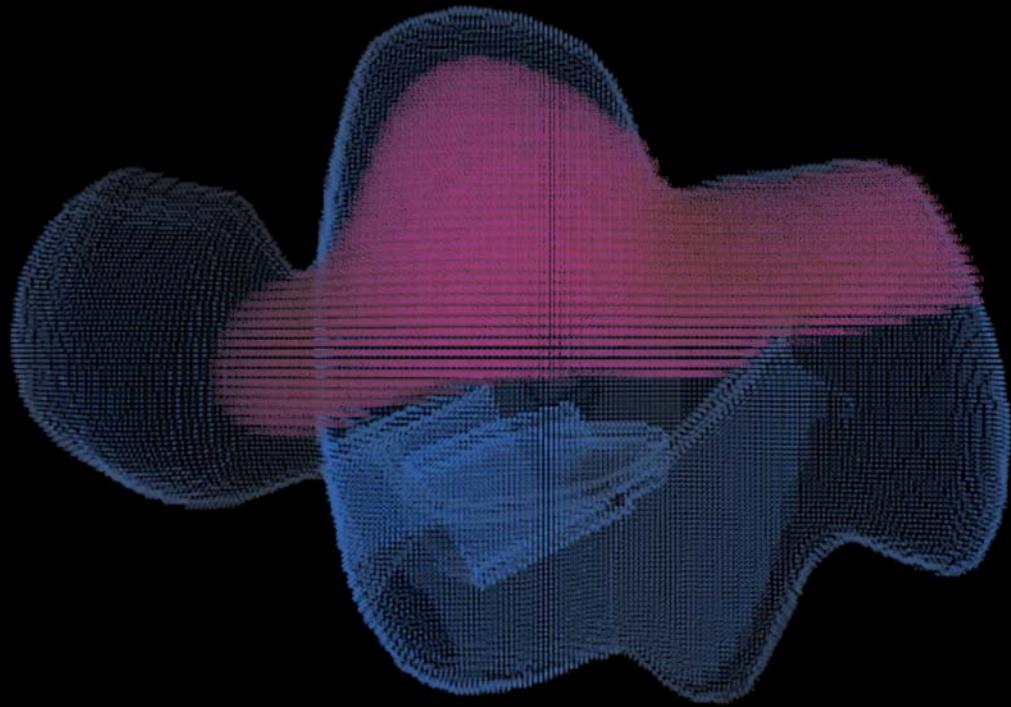
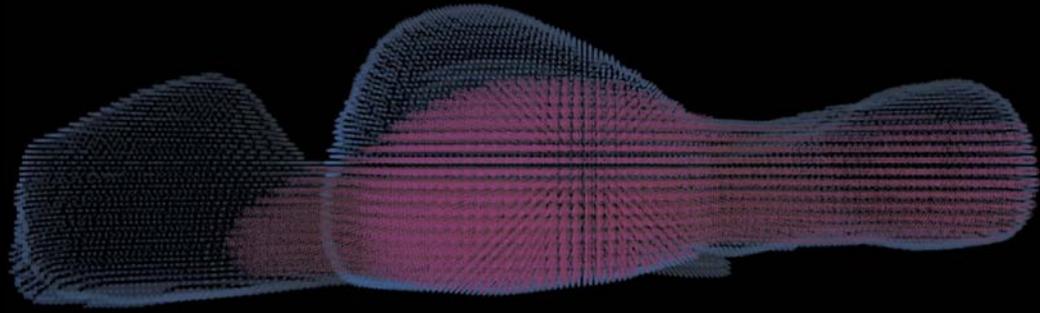


Die grafische Benutzeroberfläche

Die Steuerung der Software erfolgt durch verschiedene, miteinander kombinierbare Eingaben, die hauptsächlich Filteroperationen darstellen. Die Darstellung von Bauteilen, Sehstrahlen oder sichtbaren Elementen kann getrennt aktiviert und aus verschiedenen Ansichten betrachtet werden. Es gibt einen echten 3D-Modus, der eine interaktive Kamerasteuerung aufweist und alternativ dazu einen Pseudo-3D-Modus, der in Wahrheit aus zweidimensionalen Geometrieobjekten zusammengesetzt ist, die aber perspektivisch erscheinen.

Letzterer Modus weist nur wenige programmierte Ansichtsrichtungen auf, erlaubt aber eine vektorgrafische Aufbereitung der Daten mit hoher Qualität. Verschiedene quantifizierbare Werte werden jeweils für den aktuellen Augpunkt ausgegeben.

über Switches, die direkt im Programmiercode verborgen sind, können weitere Darstellungsmodi aktiviert werden. So ist es etwa möglich, gezählte Werte nicht nur farblich zu indizieren, sondern direkt numerisch auszugeben. (siehe Abbildung oben: für den gesamten Innenraum wird angezeigt, wieviel Luftraum jeweils von einem Punkt aus sichtbar ist.

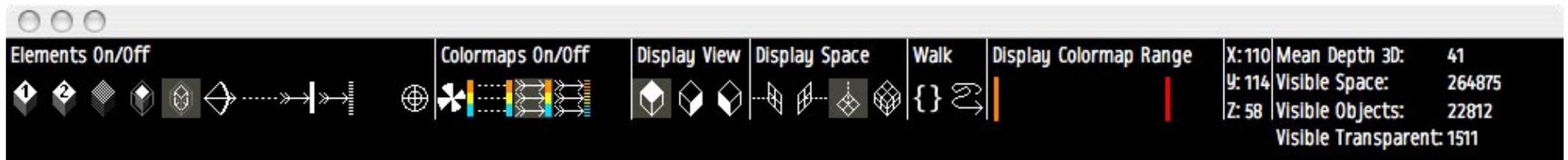


Die Bewertung des Raumes über eine Datenwolke

Die Betrachtung der Ränder gibt uns Aufschluß über die Eigenschaften der Wandzonen, aber wir können den Raum auch in der Tiefe analysieren. Eine Schwierigkeit in der Darstellung besteht darin, dass tiefer im Inneren liegende Raumschichten meist von der äußeren Datenschicht verdeckt werden. Darum besteht die Möglichkeit, nur einen Ausschnitt aus dem gesamten Wertebereich anzuzeigen. Auf diese Weise werden wolkenartige Gebilde im Rauminnen sichtbar, die in etwa Zonen ähnlicher Werte liegen. Im Fall der oben gezeigten drei Illustrationen bedeutet das, dass die Bereiche im Inneren sichtbar werden, die einen großen Luftraum überblicken. Höhere Werte sind rot ausgewiesen, niedrigere blau. Während aus der linken Bildhälfte deutlich hervorgeht, dass sich eine Zone guter Sichtbarkeit durch nahezu die gesamte Längsachse des Hauses zieht, wird am rechten Bild deutlich, wo die Rückzugszonen liegen.

Diese Art der Darstellung kann bestimmte Werte exakt ausfiltern, da sie auf einer einfachen Wahrheitsbedingung beruht: wenn der ausgelesene Wert in einem bestimmten numerischen Bereich liegt, dann zeichne ihn - andernfalls tu gar nichts.

Das Bild auf der folgenden Seite zeigt einen anderen Weg, diese Art der Filterung zu verwenden: wir haben die Möglichkeit, die Gebäudedaten schnittweise auszugeben.



Gezeigt wird die Transition zwischen Innen und Außen

Gezeigt wird die Farbkarte sichtbarer Objekte

Draufsicht ist an

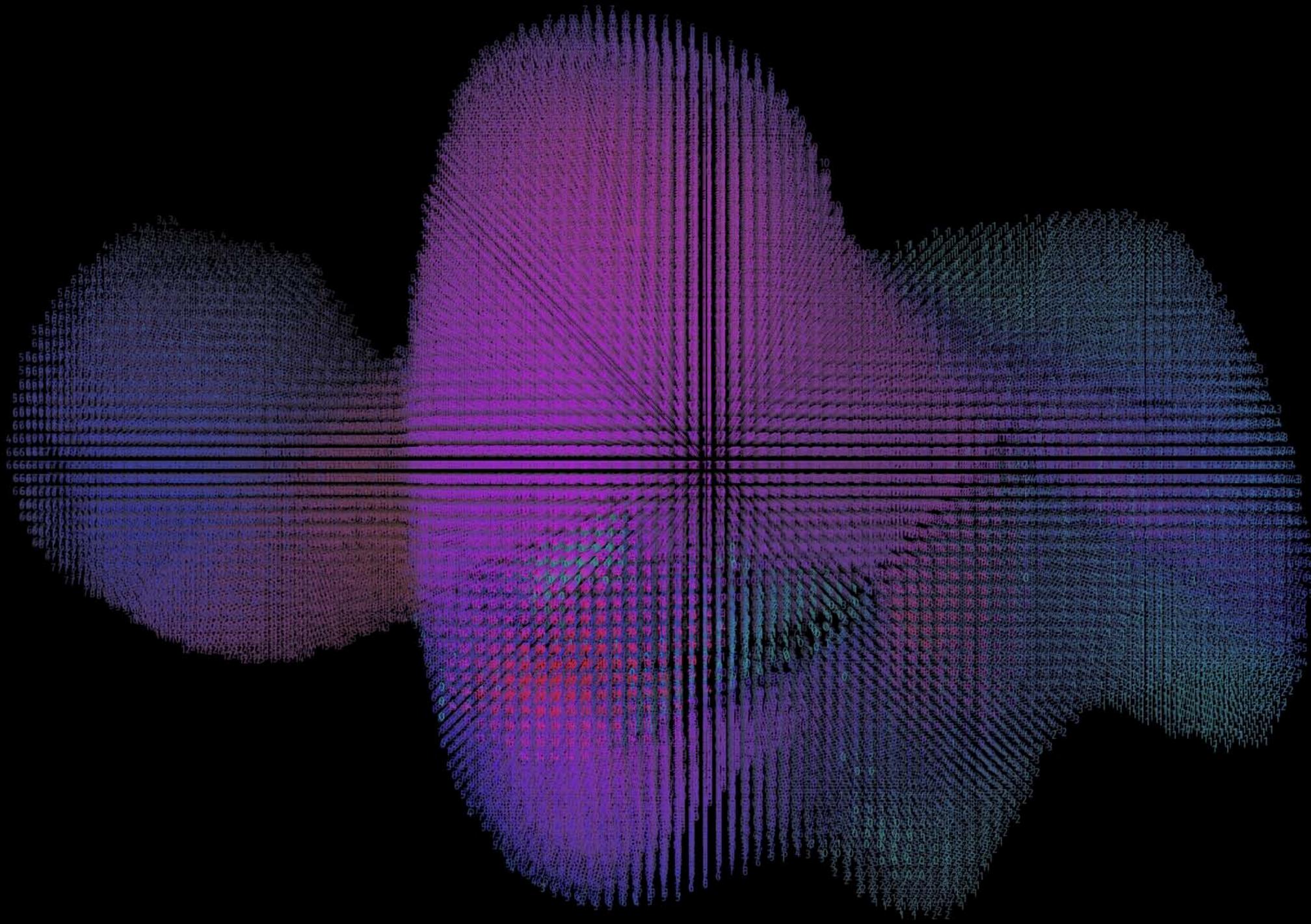
Der Ansichtsraum ist "Grundriß"

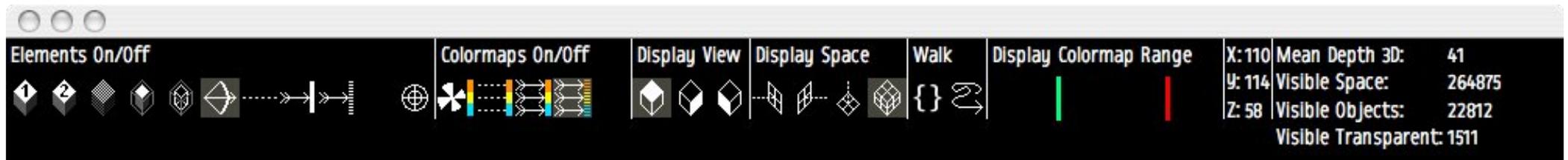
Das gesamte Spektrum der Farbkarte wird gezeigt

Numerische Schnittbilddarstellung

Durch Vergleichsoperationen werden Ansichtsräume darstellbar und die Erstellung von Schnittbildern möglich. Es ist ein besonderer Vorteil, der voxelbasierenden Adressierung, dass die geometrische Form des Baukörpers für die Zuverlässigkeit dieser Operation keine Rolle spielt. Damit steht sie im Gegensatz zum vektorgrafischen Datenmodell, das derartige Schnitte nur mit hohem Aufwand berechnen kann. Wir können jede beliebige dreidimensionale Wahrfalsch-Liste zu unserem Ansichtsraum machen und daraus völlig neuartige Schnittmengen finden, die dargestellt werden sollen.

In der oben gezeigten Grafik sind die Werte numerisch ausgewiesen - es handelt sich dabei um die Anzahl der vom jeweiligen Punkt aus sichtbaren Festkörper, zur besseren Lesbarkeit geteilt durch 1000.





Gezeigt wird der Augpunkt

Gezeigt wird die Farbkarte transparenter Objekte

Draufsicht ist an

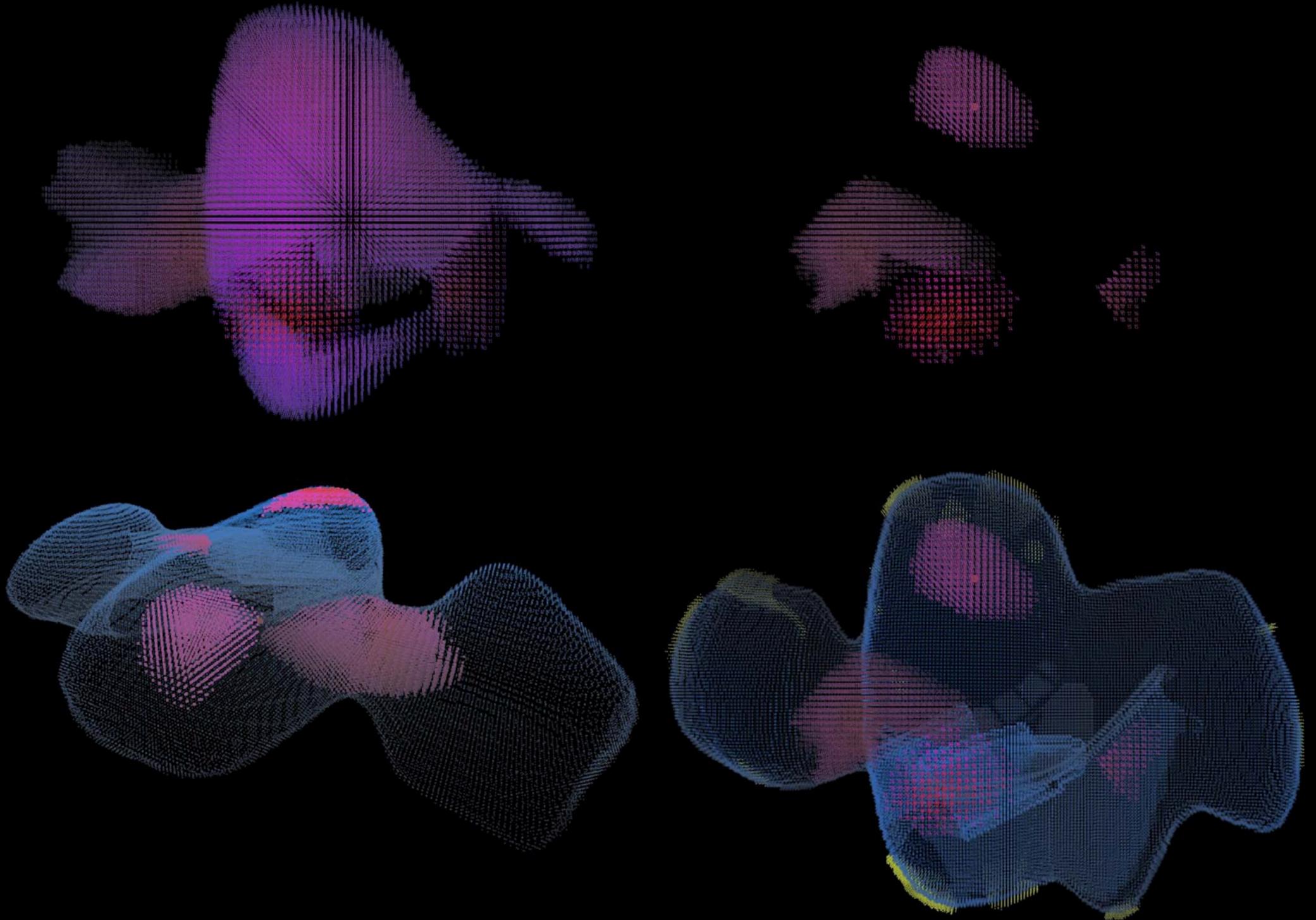
Der Ansichtsraum ist "3D"

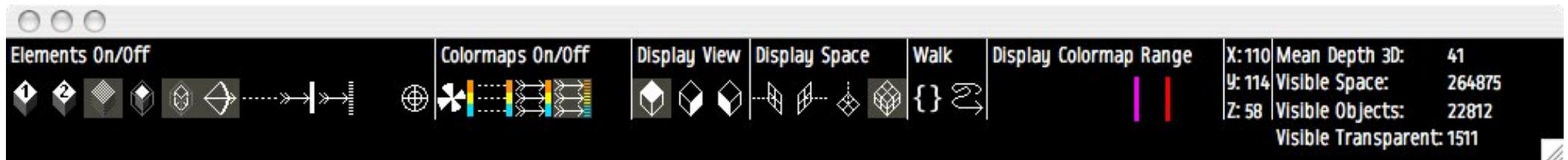
Ein Teilspektrum der Farbkarte wird gezeigt

Transparenzen

Wenn wir auch die gebaute Substanz, die den Raum umgibt, für das Rechenmodell grundsätzlich mit uniformen Eigenschaften versehen, so gibt es einen Sonderfall, der differenziert behandelt werden sollte. Transparente Bauteile, die den Inneraum umschließen. Man könnte sie als Verbindungselemente zu weiteren Räumen ansehen, die ebenfalls wieder analysiert werden können. Für den weiteren Fortgang der Lichtstrahlen, die das Haus verlassen, können wir keine Annahme treffen.

Als sichere Kenngröße gilt aber, dass ein Lichtstrahl von innen auf eine Fensterfläche fällt. Insofern werden Fenster gleich behandelt wie undurchsichtige Körper, sie werden aber separat gezählt. Mit Hilfe einer räumlichen Quantifizierungsmethode können wir etwa die Punkte finden, die Ausblick auf möglichst viele Fensterflächen bieten. Dabei ist zu beachten, dass die Entfernung vom Fenster keine Rolle spielt, da wir nicht perspektivisch gewichten.





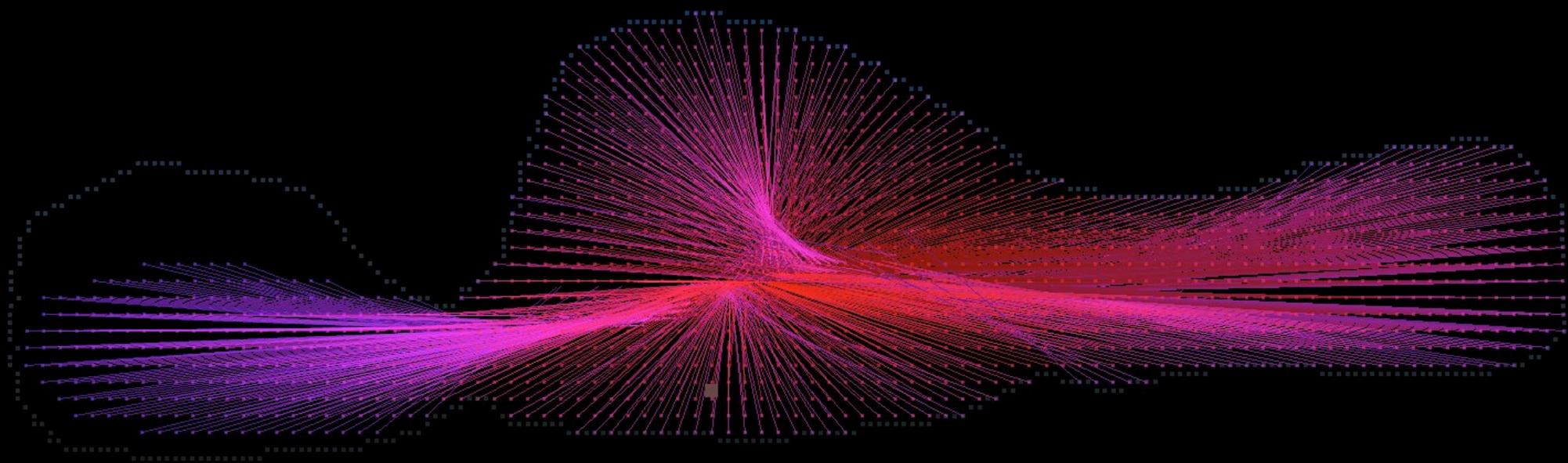
Gezeigt werden transparente Bauteile
 Gezeigt wird die Transition Innen/Außen
 Gezeigt wird der Augpunkt

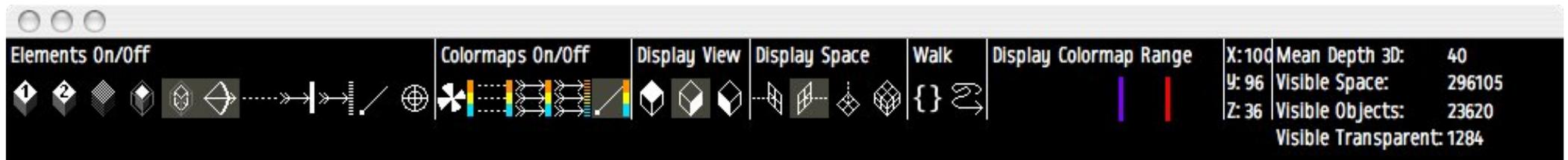
Draufsicht ist an
 Gezeigt wird die Farbkarte
 transparenter Objekte

Der Ansichtsraum
 ist "3D"

Ein Teilspektrum
 der Farbkarte wird gezeigt

Die Abbildungen zeigen die Orte mit der stärksten Anbindung an transparente Bereiche (diese sind im Bild rechts unten gelb eingetragen). Anders als die zuvor gezeigten Farbkarten konzentrieren sie sich nicht auf einen scharf umrissenen Bereich, sondern zerfallen in vier Felder unterschiedlicher Position. Das Feld mit der stärksten Anbindung ist paradoxerweise ein nicht zugänglicher Raum unmittelbar unter der höchsten Kuppelschale. Die Karte links oben beschränkt sich auf die Felder mit guter Anbindung an Fensterflächen - aus den Leerstellen wird aber deutlich, dass Teile jener Bereiche des Hauses, die auch sonst eher als Rückzugsräume dienen, weniger Fensteranteil aufweisen.





Gezeigt wird die Transition Innen/Außen
Gezeigt wird der Augpunkt

Seitenansicht ist an
Gezeigt werden die
Hauptachsen

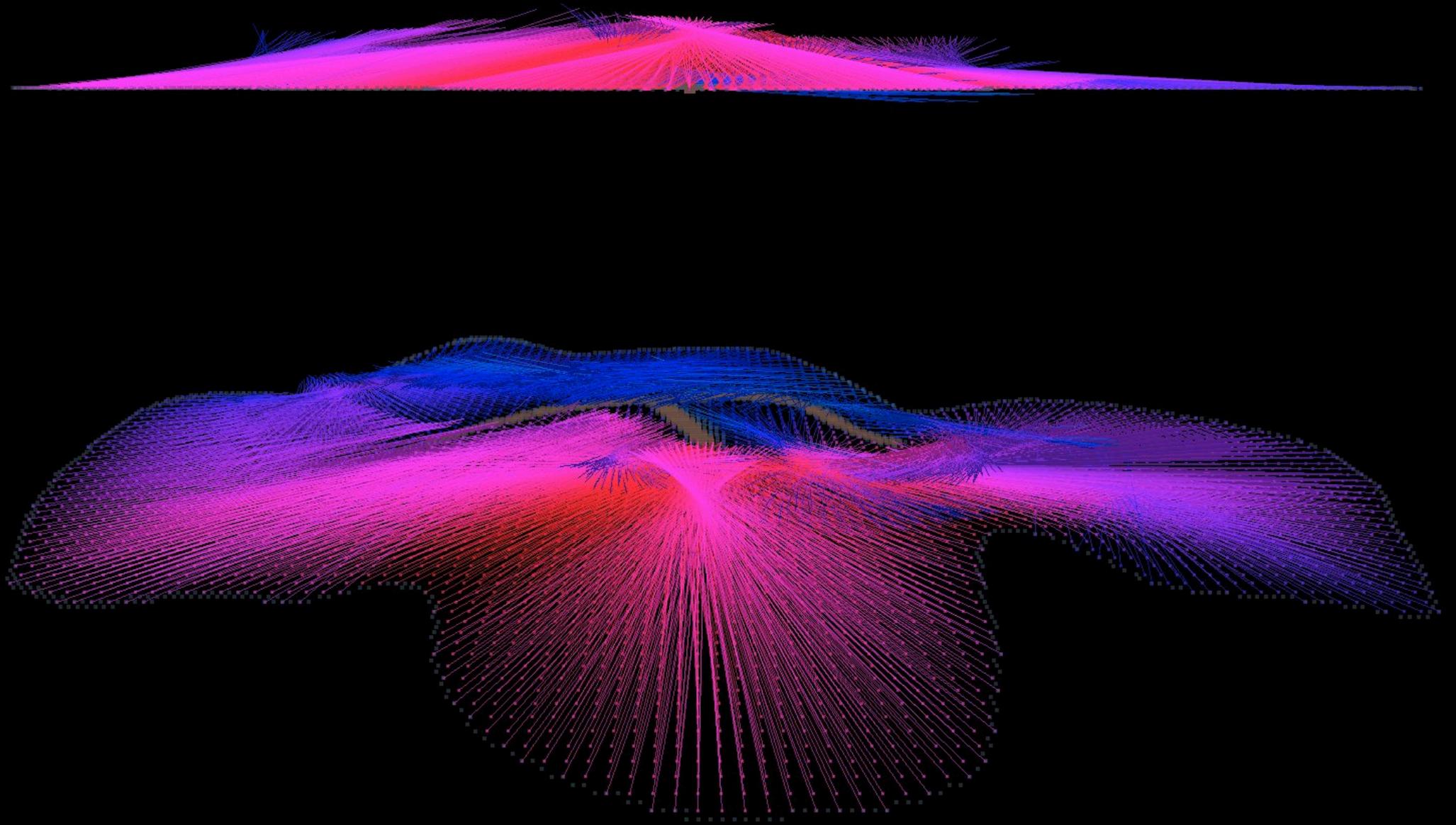
Der Ansichtsraum
ist "Schnitt"

Ein Teilspektrum
der Farbkarte wird gezeigt

Der Schwerpunkt des Sehkörpers

Aus der Möglichkeit, Sehstrahlbündel allseitig von einem Augpunkt aus in den Raum zu senden und anschließend den sichtbaren Raum zu Vermessen, leitet sich ein zusätzlicher Weg ab, die Beschaffenheit des Raums zu Charakterisieren. Starre Körper besitzen zwar eine Ausdehnung, sie verhalten sich aber in vieler Hinsicht wie ein punktförmiges Objekt, das Masse besitzt. Dieser besondere Punkt ist der Schwerpunkt, der in der Mechanik eine große Rolle spielt. Er bezeichnet den Ort, der die durchschnittliche Lage aller Punkte des starren Körpers im Raum darstellt. Der Schwerpunkt und die durch ihn gezogenen Achsen bieten in der Mechanik erst die Möglichkeit, ein reales Objekt als eine abstrakte, berechenbare Entität zu betrachten. Am Anfang aller Überlegungen in diesem Buch stand die Idee, den leeren Körper als etwas anzusehen, das ungerichtet und homogen ist. aus der Mechanik stammt der Begriff der Spannungstrajektorien.

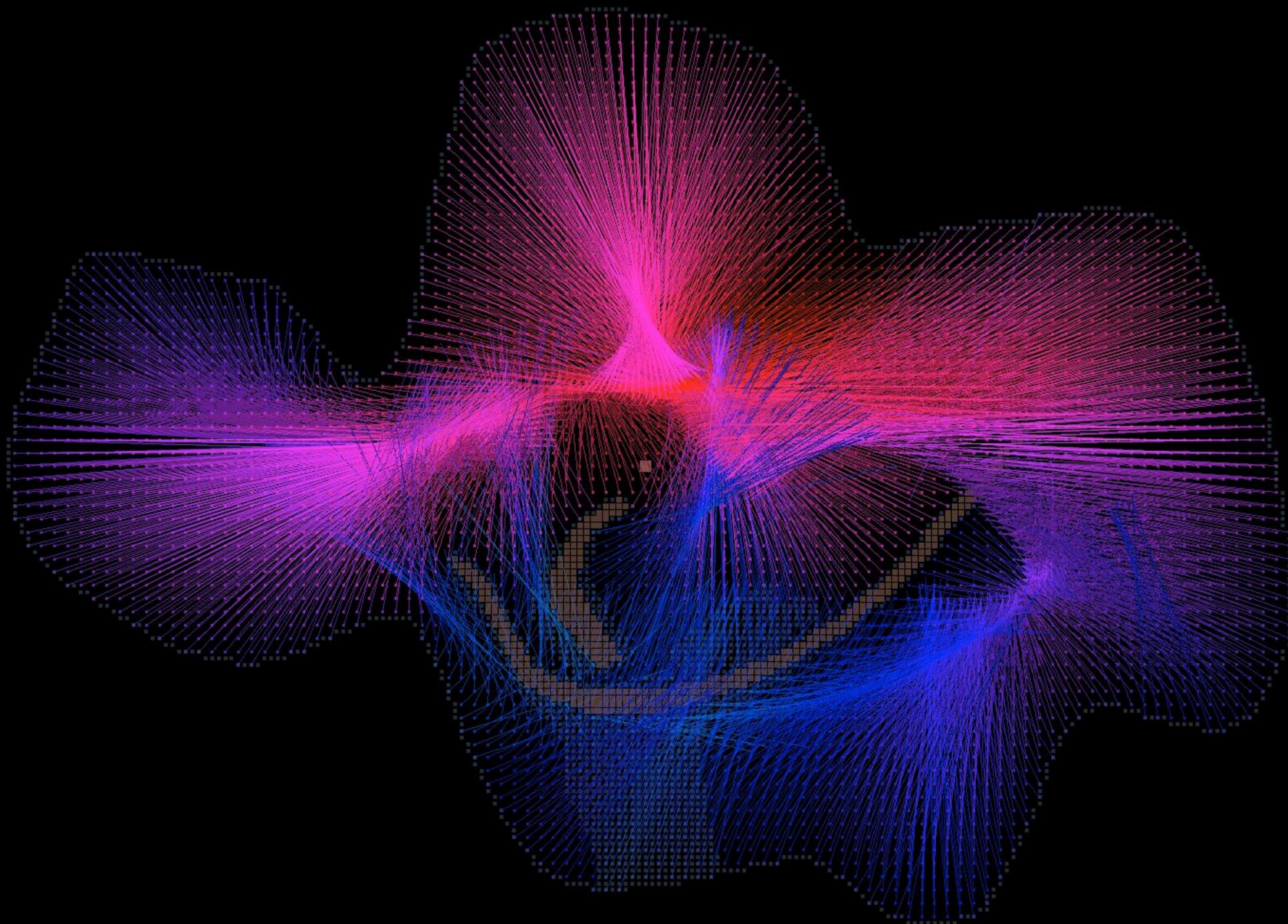
Die folgende Überlegung führt zu einem Konzept, das dem leeren Raum Hauptachsen zuweist und uns helfen kann, mehr über ihn zu erfahren: Das zuvor beschriebene Sehstrahlbündel besitzt eine meßbare Ausdehnung und durchläuft ein Raumvolumen, das eine klar definierte Form hat. Dieses Raumvolumen kennen wir, da es die Software in allen Punkten berechnet hat. Weist man jedem Punkt dasselbe Gewicht zu und mittelt die Gewichte in X-, Y- und Z-Koordinate, so erhält man einen Punkt, der in einem vergleichbaren festen Körper genau dem Schwerpunkt entspricht. Es entsteht also ein Punktepaar aus dem Augpunkt und dem Schwerpunkt des Sehkörpers.

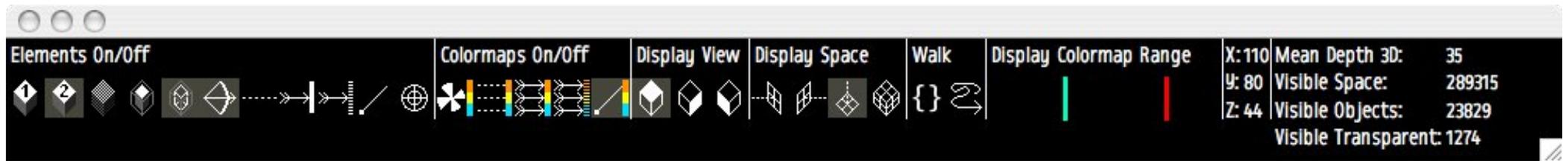


Warum ist der Schwerpunkt des Sehkörpers wichtig? Anders als das menschliche Auge, das nur einen begrenzten Sehkegel aufweist, hat unser künstliches Auge stets die volle Information über die allseitigen visuellen Eigenschaften eines Punktes. Es kann also die Hauptrichtung, die ein unregelmäßiges Sehvolumen aufweist berechnen. Das kann der Mensch nicht, aber er kann aus der Bewegung und der Erinnerung heraus ähnliche Betrachtungen anstellen und zu dem Schluß kommen, dass eine bestimmte Richtung im Raum die größte visuelle Ausbeute liefert. Das tun wir beispielsweise immer dann, wenn wir uns entscheiden, die Mitte eines Raumes stärker zu beachten und den Blick aus einer Ecke herauszulenken.

Unter der Annahme, dass die bevorzugte Blickachse zur bevorzugten Bewegungsachse wird, könnte man sich jetzt entscheiden, dieser Hauptachse ein wenig zu folgen. Weil sich damit aber der Augpunkt verändert, kann die Hauptachse neu berechnet werden und möglicherweise eine neue Richtung erhalten. Wir folgen dieser neuen Richtung und erhalten eine neue Achse. Die Summe dieser neu gefundenen Punkte ergibt ein längliches Gebilde im Raum - von ihnen können wir unendlich viele finden und wir können sie als Sehtrajektorien bezeichnen.

Von der bloßen visuellen Eigenschaft abgesehen, haben sie auch Einfluß auf unsere Bewegungsmuster im Raum. Die Illustrationen in diesem Kapitel zeigen diese Muster für das räumlich sehr differenzierte Gebilde, das Gegenstand der Untersuchung war. Sie weisen aber auch in räumlich weniger differenzierten Situationen, und damit sind insbesondere orthogonale Räume gemeint, starke Dreidimensionalität auf.





Gezeigt wird ein Teil des Baukörpers

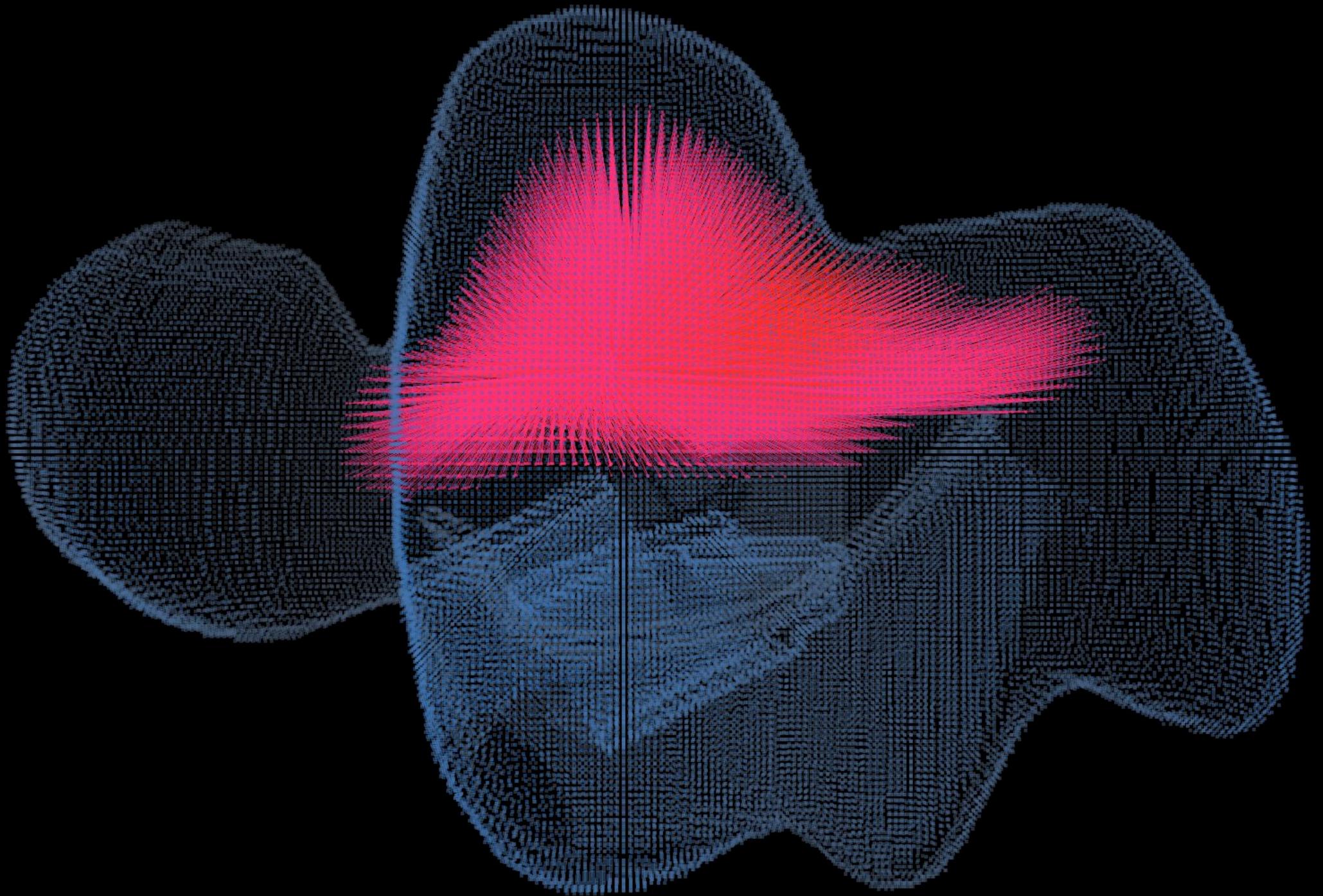
Gezeigt wird die Transition Innen/Außen
Gezeigt wird der Augpunkt

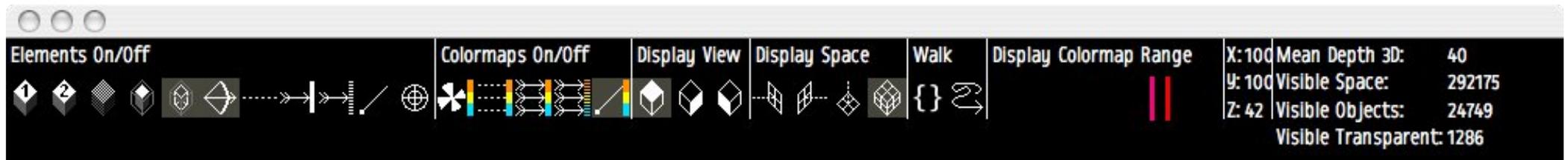
Draufsicht ist an
Gezeigt werden die Hauptsichtachsen

Der Ansichtsraum ist "Grundriss"

Die gesamte Farbkarte wird gezeigt

Für das Endless House wurde der gesamte Inneraum als Punktraster nach seinen Hauptsehrichtungen vermessen. Tatsächlich entsteht ein Bild, das ein wenig an magnetische Feldlinien erinnert und das uns Aufschluß darüber gibt, wohin an einem bestimmten Ort der Blick gezogen wird. Sehr auffällig ist der Fokuspunkt im mittleren Zentralraum, der in etwa dort liegt, wo Kiesler in seinem Entwurf den offenen Kamin situiert. Im Bild oben entsteht auch ein Eindruck von einer umlaufenden Gehlinie, die mögliche Hauptverkehrswege andeutet.





Gezeigt wird die Transition Innen/Außen
Gezeigt wird der Augpunkt

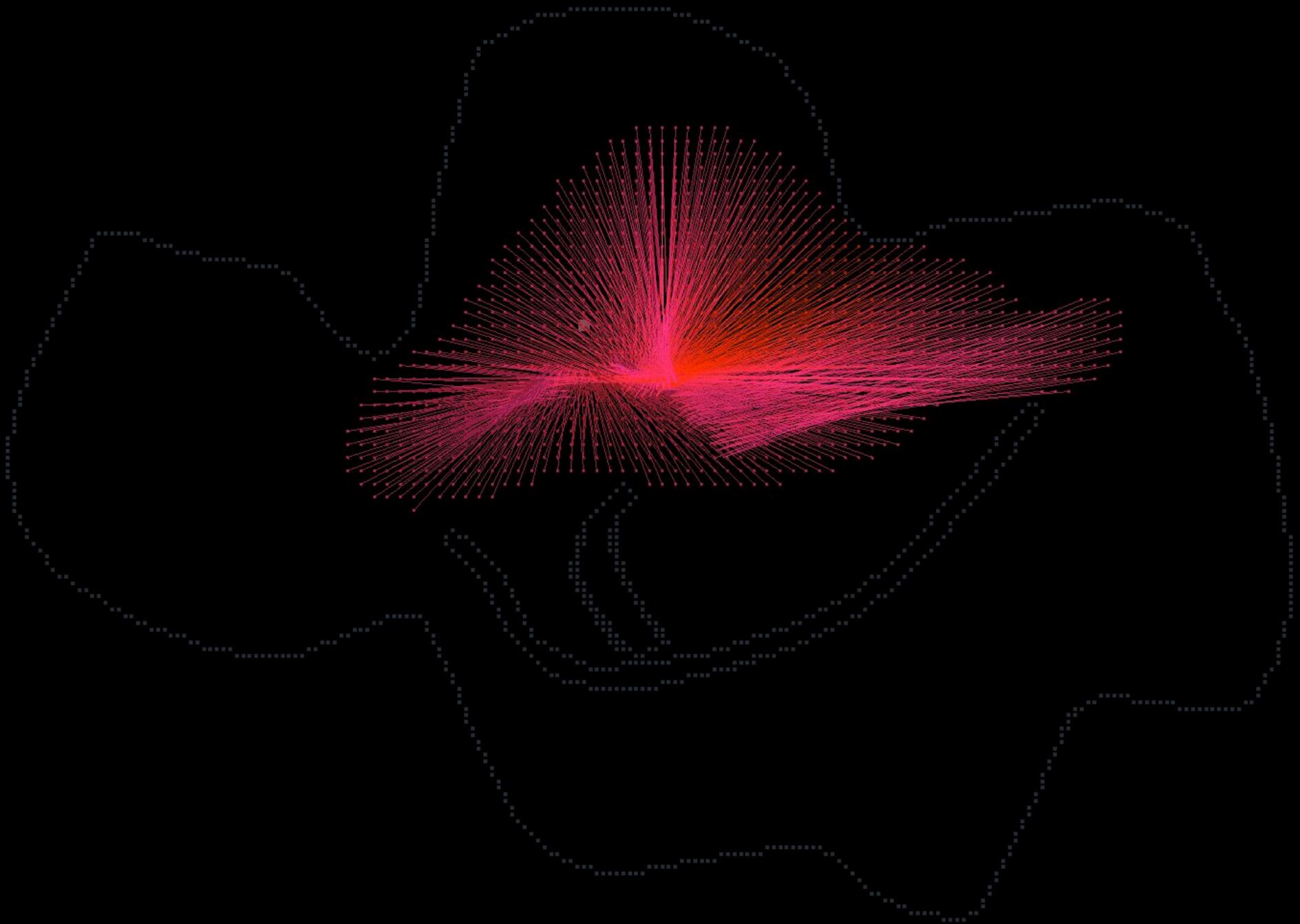
Draufsicht ist an
Gezeigt werden die
Hauptsichtachsen

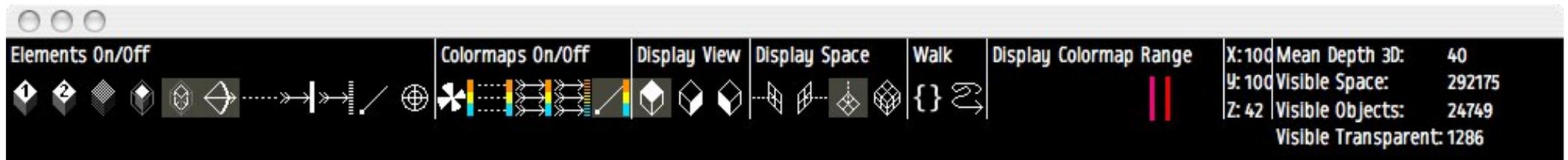
Der Ansichtsraum
ist "3D"

Ein Teil der Farbkarte wird
gezeigt

Die Kombination von Sehachse und Farbkarte

Weil wir auch andere Eigenschaften der Untersuchten Raumpunkte bereits kennen, besteht die Möglichkeit, kombinierte Abfragen durchzuführen und die Sehachse mit anderen Kenngrößen zu überlagern. Eine Frage könnte lauten: wo sind die Achsen, die besonders große Sehräume durchqueren? Wie liegen die Achsen, die stark zu transparenten Bauteilen hin orientiert sind? Wohin blickt man in Rückzugszonen? Als Beispiel für derartige Fragestellungen wird eine Kombination aus starker Sichtbarkeit von Objekten und Sehachsen gewählt. Die Zentralität des Raumes wird dadurch sehr anschaulich.





Gezeigt wird die Transition Innen/Außen
Gezeigt wird der Augpunkt

Draufsicht ist an
Gezeigt werden die
Hauptsichtachsen

Der Ansichtsraum
ist "Grundriss"

Ein Teil der Farbkarte wird
gezeigt

Ende und Anfang

Die in diesem Buch gezeigten Beispiele sind das Ergebnis einer praktischen Auseinandersetzung mit den Möglichkeiten computergestützter Planung. Die theoretischen Anforderungen an die Software müssen immer mit den praktischen Möglichkeiten des Programmierers abgeglichen werden. Insofern handelt es sich um eine endlose Aufgabe, da die Arbeit mit dem Programm zu ständig neuen Anforderungen führt, die dann in die Programmierung einfließen. Die hier gezeigten Methoden stellen also einen Anfang dar.

Nach einigen Entwicklungsschritten liegt nun ein Programm vor, das auch ohne Programmierkenntnisse bedienbar ist und schnell Resultate liefert.

Einige Schnittzeichnungen, die als Bitmaps eingelesen werden, reichen schon aus, um zu einem Ergebnis zu kommen. Der Autor ist zuversichtlich, dass es ein großes Einsatzgebiet für die hier gezeigten Lösungen gibt. Das gilt nicht nur für die Architektur, sondern auch für andere Aufgaben in Raumplanung, Bühnenbild oder der Konzeption von Veranstaltungen. Das Vergnügen, dass die Arbeit mit dem Programm bereitet hat, gibt Grund zur Hoffnung, dass es auch anderen Menschen Spaß machen wird, es einzusetzen.

Literaturverzeichnis

Alexander, Christopher,/Czech, Herrmann, Eine Muster-Sprache, Wien 1995

Bogner, Dieter, Friedrich Kiesler: inside the endless house, Wien 1998

Bollinger, Klaus, Lehrveranstaltung am Institut für Tragwerksentwurf, Universität für Angewandte Kunst, Wien 2010

Chapman, Allan, England's Leonardo: Robert Hooke and the seventeenth-century scientific revolution, London 2005

Fink, Stefan, Im Endlosen Raum des Wissens. Friedrich Kieslers Correalismus als Wissensarchitektur. Diplomarbeit, TU Graz 2010

Fry, Stephen/Reas, Casey, Processing: A Programming Handbook for Visual Designers and Artists, Cambridge 2009

Hillier, Bill, Space is the machine, London 2007

Kiesler, Frederick J., Selected Writings, Frankfurt 1996

Krause, Joachim, Your private sky: R. Buckminster Fuller, the art of design science, Zürich 1999

Lynch, Kevin, Das Bild der Stadt, Basel 2001

Ratti, Carlo, Urban texture and space syntax: some inconsistencies, Cambridge 2003

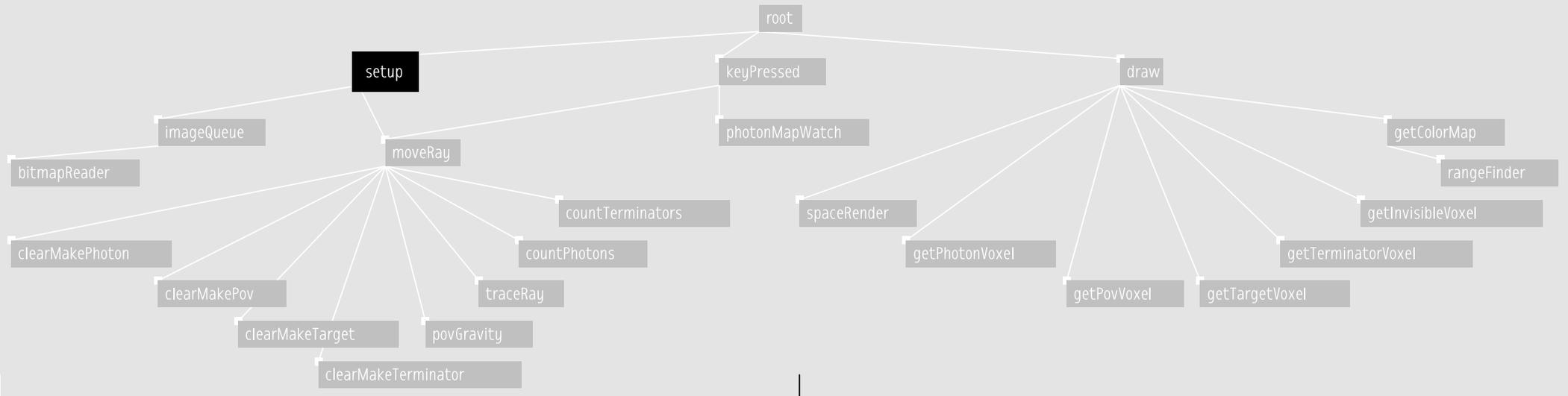
Watt, Alan, 3-D Computergrafik, München 2002

Whitread, Rachel/Lingwood, James/Bird, Jon, House, London 1995

Dokumentarfilm auf

http://www.artangel.org.uk/projects/1993/house/video/video_rachel_whitread_house

A Code Story



Spiel das Spiel **Verarbeite das Bild** (dazu bekommst du den

Wert der Bildbreite, den
Wert der Bildhöhe, ein
Bild mit Bildnamen und den
Wert der Höhenlage aller Bildpunkte).

Das Bild mit dem entsprechenden Bildnamen

liest du Bildpunkt für Bildpunkt ein.

Beende diesen Arbeitsschritt,

aber nicht bevor die Bedingung erfüllt wurde

(am **Anfang** stell dich ganz nach links;

achte darauf, dass du noch nicht ganz rechts bist;

geh jeweils einen Schritt nach rechts)

aber nicht bevor die andere Bedingung erfüllt wurde

(am **Anfang** stell dich ganz nach vorne;

achte darauf, dass du noch nicht ganz hinten bist;

geh jeweils einen Schritt nach hinten).

Es gibt einen Wert für den Bildpunkts-Ort,

er entspricht der Anzahl der Schritte,

die du schon nach links gegangen bist und

der Bildbreite mal der Anzahl der Schritte,

die du schon nach hinten gegangen bist.

Die Bildpunktfarbe entspricht dem Farbwert

des Bildes mit dem passenden Bildnamen

am Bildpunkts-Ort, den du jetzt gerade ansiehst.

Wenn die Bildpunktfarbe hell ist,

merke dir, dass an dieser Stelle ein farbiger Bildpunkt existiert.

Wenn die Bildpunktfarbe dunkel ist,

merke dir, dass an dieser Stelle kein farbiger Bildpunkt existiert.

In die dreidimensionale Liste der Raumpunkte trägst

du ein, ob an der Stelle, die du untersucht hast,

ein farbiger Bildpunkt existiert oder nicht.

Vergiß alle Bildpunkte.

```
void bitmapReader (
```

```
int imageXlen,
```

```
int imageYlen,
```

```
PImage imageID,
```

```
int zNumber) {
```

```
    imageID.loadPixels();
```

```
    for (int x = 0; x < imageXlen; x++) {
```

```
        for (int y = 0; y < imageYlen; y++) {
```

```
            int pixelLocation = x + (y * imageXlen);
```

```
            pixelColor = imageID.pixels[pixelLocation];
```

```
            if (pixelColor == -1) {
```

```
                pixelColor = 1;
```

```
            }
```

```
            if (pixelColor == -16777216) {
```

```
                pixelColor = 0;
```

```
            }
```

```
            voxelSpace[x][y][zNumber] = pixelColor;
```

```
        }
```

```
    }
```

```
    updatePixels();
```

```
}
```

Der Name des Spiels ist Lösche die Zielpunkte der Sehstrahlen aus der Liste.

Es gibt die Werte der Länge, Breite und Tiefe der Liste.

Geh an die erste, vorderste Stelle der obersten Zeile und hör sofort auf zu lesen,

aber nicht ohne vorher je eine Stelle nach rechts gegangen zu sein, solange, bis du ganz rechts ankommst,

aber nicht ohne vorher je eine Stelle nach hinten gegangen zu sein, solange, bis du ganz hinten ankommst,

aber nicht ohne vorher je eine Stelle nach unten gegangen zu sein, solange, bis du ganz unten ankommst,

und immer eine "0" an die Stelle der Liste setzt, an der du gerade bist.

```
void clearMakeTarget() {
```

```
int x, y, z;
```

```
for ( x = 0; x < imgXlength; x++) {
```

```
for ( y = 0; y < imgYlength; y++) {
```

```
for ( z = 0; z < imgZrange; z++) {
```

```
makeTarget[x][y][z] = 0;
```

```
}
```

```
}
```

```
}
```

```
}
```

Der Name des Spiels ist Lösche die letzten sichtbaren Punkte
der Sehstrahlen aus der Liste.

Es gibt die Werte der Länge, Breite und Tiefe der Liste.

Geh an die erste, vorderste Stelle der obersten Zeile
und hör sofort auf zu lesen,

aber nicht ohne vorher je eine Stelle
nach rechts gegangen zu sein, solange,
bis du ganz rechts ankommst,

aber nicht ohne vorher je eine Stelle nach hinten gegangen zu sein,
solange, bis du ganz hinten ankommst,

aber nicht ohne vorher je eine Stelle nach unten gegangen zu sein,
solange, bis du ganz unten ankommst,
und immer eine "0" an die Stelle der Liste setzt,
an der du gerade bist.

```
void clearMakeTerminator() {
```

```
int x, y, z;
```

```
for ( x = 0; x < imgXlength; x++) {
```

```
for ( y = 0; y < imgYlength; y++) {
```

```
for ( z = 0; z < imgZrange; z++) {
```

```
makeTerminator[x][y][z] = 0;
```

```
}
```

```
}
```

```
}
```

```
}
```

Finde die Zahl, die die Summe aller sichtbaren Punkte ist.

Es gibt die Werte der Länge, Breite und Tiefe der Liste.

Du weißt bereits, dass es eine Summe aller sichtbaren Punkte geben muss, kennst aber keinen der Punkte. Also beginne bei Null.

Geh an die erste, vorderste Stelle der obersten Zeile und hör sofort auf zu lesen,

aber nicht ohne vorher je eine Stelle nach rechts gegangen zu sein, solange, bis du ganz rechts ankommst,

aber nicht ohne vorher je eine Stelle nach hinten gegangen zu sein, solange, bis du ganz hinten ankommst,

aber nicht ohne vorher je eine Stelle nach unten gegangen zu sein, solange, bis du ganz unten ankommst,

Wenn es zutrifft, dass

in der Liste der sichtbaren Punkte auf der jeweiligen Koordinate ein sichtbarer Punkt eingetragen ist, dann

steigere den Wert der Summe aller sichtbaren Punkte um 1.

Beende alle Schritte.

Gib bekannt, wie groß die Summe der sichtbaren Punkte ist.

```
int countPhotons() {
```

```
    int x, y, z;
```

```
    numberOfPhotons = 0;
```

```
    for ( x = 0; x < imgXlength; x++) {
```

```
        for ( y = 0; y < imgYlength; y++) {
```

```
            for ( z = 0; z < imgZrange; z++) {
```

```
                if (makePhoton[x][y][z] == 1) {
```

```
                    numberOfPhotons = numberOfPhotons + 1;
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
    return numberOfPhotons;
```

```
}
```

Spiel das Spiel zeichne alles.

Mal den Hintergrund (in dunklem Grau),

mach ein wenig Licht (von oben) und weiters

schalte bitte einen Scheinwerfer ein

(mit weißem Licht, stell ihn seitlich auf).

Wenn der Schalter (er regelt die Darstellungsart)

auf 'Ansicht von Oben' eingestellt ist, dann

ist der Abstand deines Standpunkts die Hälfte

der Gesamtanzahl der Schritte mal der Schrittweite.

Du blickst auf den dir gegenüberliegenden

Punkt nach unten und

richtest die Hochachse deines Sichtfeldes

parallel zur Hochachse der Grundfläche.

Zeig, was du siehst (wie es vorhin beschrieben wurde) und

beende diese Aufgabe.

Zeichne alle sichtbaren Objekte.

Wenn es stimmt, dass sichtbarer Raum gezeigt

werden soll, dann

zeige sichtbaren Raum.

Wenn es stimmt, dass der Augpunkt gezeigt

werden soll, dann

zeige den Augpunkt.

Wenn es stimmt, dass der Zielpunkt jedes

Sehstrahls gezeigt werden soll, dann

zeige den Zielpunkt.

Wenn es stimmt, dass der letzte sichtbare

Punkt jedes Sehstrahls gezeigt werden soll, dann

zeige den letzten sichtbaren Punkt.

Wenn es stimmt, dass unsichtbare Punkte des Raums

gezeigt werden sollen, dann

zeige unsichtbare Punkte.

Wenn es stimmt, dass eine farbige Übersichtskarte

gezeigt werden soll, dann

zeige die Übersichtskarte (übergib dazu die Werteliste).

```
void draw() {
    background(35,35,30);
    ambientLight(120, 120, 110);
    directionalLight(255, 255, 255, -0.66, -0.33, -0.5);

    switch(viewmode) {
        case 't':
            eyeX = gridStepX * imgXlength/2;
            eyeY = gridStepY * imgYlength/2;
            eyeZ = gridStepY * imgYlength;
            centerX = gridStepX * imgXlength/2;
            centerY = gridStepY * imgYlength/2;
            centerZ = gridStepY * imgYlength * (-1);
            upX = 0;
            upY = 1;
            upZ = 0;
            camera(eyeX, eyeY, eyeZ, centerX,
                centerY, centerZ, upX, upY, upZ);
            break;
    }
    spaceRender();
    if (showPhotonVoxel == true) {
        getPhotonVoxel();
    }
    if (showPovVoxel == true) {
        getPovVoxel();
    }
    if (showTargetVoxel == true) {
        getTargetVoxel();
    }
    if (showTerminatorVoxel == true) {
        getTerminatorVoxel();
    }
    if (showInvisibleVoxel == true) {
        getInvisibleVoxel();
    }
    if (showColorMap == true) {
        getColorMap(valueList);
    }
}
```

Der Name des Spiels ist **zeichne die farbige Landkarte**
(Dazu holst du dir die Liste von Zahlen, die dir gerade
übergeben wurde und die dreidimensional ist
und nennst sie Werteliste).

Du weißt ja schon, dass unsere Zeichnung einen Farbumfang
braucht. Rufe das Programm auf, das den Farbumfang
berechnet (und übergib ihm die Werteliste) Es wird dir
den Farbumfang bekanntgeben.

Es gibt die Werte der Länge, Breite und Tiefe der Liste.

Für alles, was du in Zukunft zeichnen wirst, verwendest
du deine Farbpalette (vom Typ Wert-Sättigung-Helligkeit,
Farbumfang wie soeben berechnet);

Wenn die Randbedingung stimmt (sie lautet: der
momentane Ansichtsmodus ist 'zeige alles')

dann geh an die erste, vorderste Stelle der obersten Zeile
und hör sofort auf zu lesen,

aber nicht ohne vorher je eine Stelle

nach rechts gegangen zu sein, solange,

bis du ganz rechts ankommst,

aber nicht ohne vorher je eine Stelle nach hinten

gegangen zu sein, solange, bis du ganz hinten ankommst,

aber nicht ohne vorher je eine Stelle nach unten

gegangen zu sein, solange, bis du ganz unten ankommst,

aber nur, wenn du vorher nachsiehst,

ob in der Werteliste an der Koordinate, an der du

jetzt stehst, kein Wert eingetragen ist, der Null ist.

In diesem Fall tust du Folgendes:

Fülle alles, das du in Zukunft zeichnest nach meiner
Formel (die erste Formel berechnet den Farbton je nach
Farbumfang und gegebenem Farbwert an der Stelle, die
Sättigung entspricht dem Maximalwert des
Farbumfangs, die Helligkeit auch)

In Zukunft zeichne bitte keine Umrißlinien mehr.

Nimm dein Zeichenwerkzeug,

geh an die angegebene Stelle und

zeichne einen Würfel (Die Seitenlängen entsprechen
dem Standardraster minus einem Spaltmaß).

Jetzt leg das Zeichenwerkzeug wieder weg.

Für alles, was du in Zukunft zeichnen wirst, verwendest du
meine Farbpalette (rot-grün-blau, Farbumfang 255);

```
void getColorMap(int[][][] valueList) {  
  
    mapRange = rangeFinder(valueList);  
  
    int x, y, z;  
    colorMode(HSB, mapRange);  
  
    if (perimeterViewSurfaceNormal == 'd') {  
        for ( x = 0; x < imgXlength; x++) {  
  
            for ( y = 0; y < imgYlength; y++) {  
  
                for ( z = 0; z < imgZrange; z++) {  
  
                    if (valueList[x][y][z] != 0) {  
  
                        fill(mapRange/(valueList[x][y][z] + 1) +  
                            valueList[x][y][z] - (minRangeVal + minRangeVal/6),  
                            mapRange, mapRange);  
  
                        noStroke();  
                        pushMatrix();  
                        translate(gridStepX * x, gridStepY * y, gridStepZ * z);  
                        box(gridStepX - voxelGapX * 2, gridStepY -  
                            voxelGapY * 2, gridStepZ - voxelGapZ * 2);  
                        popMatrix();  
  
                    }  
  
                }  
  
            }  
  
        }  
  
        colorMode(RGB, 255);  
    }  
}
```

Der Name des Spiels ist **finde sichtbaren Raum und stelle ihn dar.**

Du **füllst** alles, was du darstellst mit Farbe und zeichnest

keine Umrißlinien.

Es gibt die Werte der Länge, Breite und Höhe des Raumes.

Stell dich ganz nach links und ganz nach vorne und ganz nach unten

und beende sogleich deine Wanderung,

aber nicht ohne vorher je einen Schritt

nach rechts gegangen zu sein, den du machst,

bis du ganz rechts ankommst,

aber nicht ohne vorher je einen Schritt nach hinten gegangen zu sein,

den du machst, bis du ganz hinten ankommst,

aber nicht ohne vorher einen Schritt nach oben gegangen zu sein,

den du machst, bis du ganz oben ankommst,

aber nur wenn du vorher nachsiehst,

ob in der Liste der vorhandenen sichtbaren Elemente *kein* Element

an der Koordinate eingetragen ist, an der du gerade stehst -

in welchem Fall du *nichts* tust.

Wenn das nicht zutrifft, aber es zutrifft, dass

in der Liste der vorhandenen sichtbaren Elemente

an der Koordinate etwas eingetragen ist, dann

nimm dein Zeichenwerkzeug,

geh an die angegebene Stelle und

zeichne einen Würfel.

Jetzt leg das Zeichenwerkzeug wieder weg.

```
void getPhotonVoxel () {
```

```
fill(255, 127, 0);
```

```
noStroke();
```

```
int x, y, z;
```

```
for ( x = 0; x < imgXlength; x++) {
```

```
for ( y = 0; y < imgYlength; y++) {
```

```
for ( z = 0; z < imgZrange; z++) {
```

```
if (makePhoton[x][y][z] == 0) {  
}
```

```
else if (makePhoton[x][y][z] == 1) {
```

```
pushMatrix();
```

```
translate(gridStepX * x, gridStepY * y, gridStepZ * z);
```

```
box(gridStepX/2, gridStepY/2, gridStepZ/2);
```

```
popMatrix();
```

```
    }  
  }  
}
```

Spiel das Spiel Bilder laden.

Es gibt ein Bild, dessen aktueller Name ist Schicht-Null.

Es gibt einen Zahlenwert für die Höhenlage, nimm ihn mit 'null' an.

Verarbeite das Bild(ich sage dir die Länge, die Breite, den Namen und seine Höhenlage)

Ab nun ist der aktuelle Name des Bildes Schicht-Eins.

Die gültige Höhenlage ist eins.

Verarbeite das Bild(ich sage dir die Länge, die Breite, den Namen und seine Höhenlage)

Ab nun ist der aktuelle Name des Bildes Schicht-Zwei.

Die gültige Höhenlage ist zwei.

Verarbeite das Bild(ich sage dir die Länge, die Breite, den Namen und seine Höhenlage)

Ab nun ist der aktuelle Name des Bildes Schicht-Drei.

Die gültige Höhenlage ist drei.

Verarbeite das Bild(ich sage dir die Länge, die Breite, den Namen und seine Höhenlage)

Ab nun ist der aktuelle Name des Bildes Schicht-Vier.

Die gültige Höhenlage ist vier.

Verarbeite das Bild(ich sage dir die Länge, die Breite, den Namen und seine Höhenlage)

Ab nun ist der aktuelle Name des Bildes Schicht-Fünf.

Die gültige Höhenlage ist fünf.

Verarbeite das Bild(ich sage dir die Länge, die Breite, den Namen und seine Höhenlage)

Ab nun ist der aktuelle Name des Bildes Schicht-Sechs.

Die gültige Höhenlage ist sechs.

Verarbeite das Bild(ich sage dir die Länge, die Breite, den Namen und seine Höhenlage)

Ab nun ist der aktuelle Name des Bildes Schicht-Sieben.

Die gültige Höhenlage ist sieben.

Verarbeite das Bild(ich sage dir die Länge, die Breite, den Namen und seine Höhenlage)

Ab nun ist der aktuelle Name des Bildes Schicht-Acht.

Die gültige Höhenlage ist acht.

Verarbeite das Bild(ich sage dir die Länge, die Breite, den Namen und seine Höhenlage)

Ab nun ist der aktuelle Name des Bildes Schicht-Neun.

Die gültige Höhenlage ist neun.

Verarbeite das Bild(ich sage dir die Länge, die Breite, den Namen und seine Höhenlage)

Ab nun ist der aktuelle Name des Bildes Schicht-Zehn.

Die gültige Höhenlage ist zehn.

Verarbeite das Bild(ich sage dir die Länge, die Breite, den Namen und seine Höhenlage)

Ab nun ist der aktuelle Name des Bildes Schicht-Elf.

Die gültige Höhenlage ist elf.

Verarbeite das Bild(ich sage dir die Länge, die Breite, den Namen und seine Höhenlage)

```
void imageQueue() {
```

```
    PImage imageIdentifizier = bitmapLayer00;
```

```
    int z = 0;
```

```
    bitmapReader(imgXlength, imgYlength, imageIdentifizier, z);
```

```
    imageIdentifizier = bitmapLayer01;
```

```
    z = 1;
```

```
    bitmapReader(imgXlength, imgYlength, imageIdentifizier, z);
```

```
    imageIdentifizier = bitmapLayer02;
```

```
    z = 2;
```

```
    bitmapReader(imgXlength, imgYlength, imageIdentifizier, z);
```

```
    imageIdentifizier = bitmapLayer03;
```

```
    z = 3;
```

```
    bitmapReader(imgXlength, imgYlength, imageIdentifizier, z);
```

```
    imageIdentifizier = bitmapLayer04;
```

```
    z = 4;
```

```
    bitmapReader(imgXlength, imgYlength, imageIdentifizier, z);
```

```
    imageIdentifizier = bitmapLayer05;
```

```
    z = 5;
```

```
    bitmapReader(imgXlength, imgYlength, imageIdentifizier, z);
```

```
    imageIdentifizier = bitmapLayer06;
```

```
    z = 6;
```

```
    bitmapReader(imgXlength, imgYlength, imageIdentifizier, z);
```

```
    imageIdentifizier = bitmapLayer07;
```

```
    z = 7;
```

```
    bitmapReader(imgXlength, imgYlength, imageIdentifizier, z);
```

```
    imageIdentifizier = bitmapLayer08;
```

```
    z = 8;
```

```
    bitmapReader(imgXlength, imgYlength, imageIdentifizier, z);
```

```
    imageIdentifizier = bitmapLayer09;
```

```
    z = 9;
```

```
    bitmapReader(imgXlength, imgYlength, imageIdentifizier, z);
```

```
    imageIdentifizier = bitmapLayer10;
```

```
    z = 10;
```

```
    bitmapReader(imgXlength, imgYlength, imageIdentifizier, z);
```

```
    imageIdentifizier = bitmapLayer11;
```

```
    z = 11;
```

```
    bitmapReader(imgXlength, imgYlength, imageIdentifizier, z);
```

Spiel das Spiel Tastatureingabe.

Wenn eine Pfeiltaste gedrückt wird und

wenn es die RECHTE ist und

der Augpunkt noch nicht ganz rechts ist

rück mit dem Augpunkt nach rechts.

Zeichne einen Sehstrahl (dazu übergibst du den Entfernungswert des Augpunkts in horizontaler Richtung, den Entfernungswert des Augpunkts in der Tiefe, den Entfernungswert des Augpunkts in der Höhe, den Entfernungswert des Blickpunkts in horizontaler Richtung, den Entfernungswert des Blickpunkts in der Tiefe und den Entfernungswert des Blickpunkts in der Höhe).

Wenn die Taste 'Sehstrahl' gedrückt wurde

dann berechne in Zukunft einen 'einzelnen' Sehstrahl.

Der Benutzeroberflächen-Sehstrahl-Reglerzustand ist auf 'an' gestellt.

Der Benutzeroberflächen-Sichtflächen-Reglerzustand ist auf 'aus' gestellt.

Der Benutzeroberflächen-Sichtkörper-Reglerzustand ist auf 'aus' gestellt.

Der Benutzeroberflächen-Kartenmodus-Reglerzustand ist auf 'aus' gestellt.

Zeichne einen Sehstrahl (dazu übergibst du den Entfernungswert des Augpunkts in horizontaler Richtung, den Entfernungswert des Augpunkts in der Tiefe, den Entfernungswert des Augpunkts in der Höhe, den Entfernungswert des Blickpunkts in horizontaler Richtung, den Entfernungswert des Blickpunkts in der Tiefe und den Entfernungswert des Blickpunkts in der Höhe).

Wenn die Taste 'Draufsicht' gedrückt wurde, dann

ist der Benutzeroberflächen-Draufsichts-Reglerzustand auf 'an' gestellt und

der Benutzeroberflächen-Vorderansichts-Reglerzustand auf 'aus' gestellt.

Der Benutzeroberflächen-Seitenansichts-Reglerzustand ist auf 'aus' gestellt.

Der Ansichtsmodus ist 'Draufsicht'.

Wenn die Taste 'Raumtiefenkarte' gedrückt wurde, dann

werden 'alle' raumbegrenzenden Bereiche untersucht.

Der Benutzeroberflächen-Querschnitts-Reglerzustand ist aus.

Der Benutzeroberflächen-Längsschnitts-Reglerzustand ist aus.

Der Benutzeroberflächen-Höhenschichten-Reglerzustand ist aus.

Der Benutzeroberflächen-Modellvolumen-Reglerzustand ist an.

Die gültige Werteliste ist 'sichtbare mittlere Tiefe räumlich.'

Zeichne einen Sehstrahl (dazu übergibst du den Entfernungswert des Augpunkts in horizontaler Richtung, den Entfernungswert des Augpunkts in der Tiefe, den Entfernungswert des Augpunkts in der Höhe, den Entfernungswert des Blickpunkts in horizontaler Richtung, den Entfernungswert des Blickpunkts in der Tiefe und den Entfernungswert des Blickpunkts in der Höhe).

```
void keyPressed() {
  if (key == CODED) {
    if ((keyCode == RIGHT) &&
        (povXKoord < imgXlength - 1)) {
      povXKoord = povXKoord + 1;
      moveRay(povXKoord, povYKoord, povZKoord,
              targetXKoord, targetYKoord, targetZKoord);
    }
  }
  if (key == 'r') {
    rendermode = 'r';
    gui.visualControl.switchState = true;
    gui.visualControl2d.switchState = false;
    gui.visualControl3d.switchState = false;
    gui.showPhotonsMap.switchState = false;
    moveRay(povXKoord, povYKoord, povZKoord,
            targetXKoord, targetYKoord, targetZKoord);
  }
  if (key == 't') {
    gui.viewTop.switchState = true;
    gui.viewFront.switchState = false;
    gui.viewSide.switchState = false;
    viewmode = 't';
  }
  if (key == '3') {
    perimeterViewSurfaceNormal = 'd';
    gui.viewSurfNormX.switchState = false;
    gui.viewSurfNormY.switchState = false;
    gui.viewSurfNormZ.switchState = false;
    gui.viewSurfNorm3d.switchState = true;
    valueList = visibleMeanDepthList3D;
    moveRay(povXKoord, povYKoord, povZKoord,
            targetXKoord, targetYKoord, targetZKoord);
  }
}
```

Der Name des Spiels ist **bewege den Sehstrahl** (dazu bekommst du den Entfernungswert des Augpunkts in horizontaler Richtung, den Entfernungswert des Augpunkts in der Tiefe, den Entfernungswert des Augpunkts in der Höhe, den Entfernungswert des Blickpunkts in horizontaler Richtung, den Entfernungswert des Blickpunkts in der Tiefe und den Entfernungswert des Blickpunkts in der Höhe).

Lösche die Liste der sichtbaren Punkte.

Lösche den Augpunkt aus seiner Liste.

Lösche die Zielpunkte der Sehstrahlen aus der Liste.

Lösche die letzten sichtbaren Punkte der Sehstrahlen aus der Liste.

Es gibt einen Wert, der die Summe der Längen der sichtbaren Sehstrahlen darstellt. Stell ihn auf null zurück.

Du weißt, es gibt den Längenwert des sichtbaren Sehstrahls. Nimm jetzt an, er sei null.

Es gibt auch die Anzahl der Arbeitsschritte, die du durchläufst. Weil du noch gar keinen Schritt gegangen bist, ist sie null.

Betrachte den Schalter (er heißt Schwerkraft):

Steht der Schalter auf der Stellung 'Schwerkraft an', dann erfährst du die Höhe des Augpunkts bei der Augpunktberechnungsinstanz (sag ihr, wie weit der Augpunkt horizontal und in der Tiefe vom Ursprung entfernt ist).

Ab nun ist die Stellung des Schalters nicht mehr wichtig.

Es gibt noch einen Schalter, betrachte ihn. (er regelt die Art der Darstellung).

Steht der Schalter auf der Stellung 'zeige *einen* Sehstrahl' dann

Zeichne einen Sehstrahl (dazu übergibst du den Entfernungswert des Augpunkts in horizontaler Richtung, den Entfernungswert des Augpunkts in der Tiefe, den Entfernungswert des Augpunkts in der Höhe, den Entfernungswert des Blickpunkts in horizontaler Richtung, den Entfernungswert des Blickpunkts in der Tiefe und den Entfernungswert des Blickpunkts in der Höhe).

Zähle die sichtbaren Punkte, die erzeugt wurden.

Zähle die sichtbaren Endpunkte der Sehstrahlen, die erzeugt wurden.

Die Stellung dieses Schalters ist ab nun nicht mehr wichtig.;

```
void moveRay(  
int povXLocal,  
int povYLocal,  
int povZLocal,  
int targetXLocal,  
int targetYLocal,  
int targetZLocal) {  
clearMakePhoton();  
clearMakePov();  
clearMakeTarget();  
clearMakeTerminator();  
int visiblerayLengthSum = 0;  
visiblerayLength = 0;  
int stepCounter = 0;  
switch(povGravity) {  
case 'g':  
povZLocal = povGravity(povXLocal, povYLocal);  
break;  
}  
switch(rendermode) {  
case 'r':  
traceRay(povXLocal, povYLocal, povZLocal,  
targetXLocal, targetYLocal, targetZLocal);  
countPhotons();  
countTerminators();  
break;  
}  
}
```

Es gibt den Wert des Farbumfangs (dazu brauchst du die Werte der aus drei Dimensionen bestehenden Werteliste) {

Es gibt eine Länge

Es gibt eine Breite

Es gibt eine Höhe

Es gibt einen maximalen Wert in der Liste, den wir vorerst ganz niedrig mit 1 ansetzen.

Du weißt, es gibt einen minimalen Wert in der Liste, setz ihn zuerst möglichst hoch an, mit Länge mal Breite mal Höhe.

Geh an die erste, vorderste Stelle der obersten Zeile und hör sofort auf zu lesen,

aber nicht ohne vorher je eine Stelle nach rechts gegangen zu sein, solange, bis du ganz rechts ankommst,

aber nicht ohne vorher je eine Stelle nach hinten gegangen zu sein, solange, bis du ganz hinten ankommst,

aber nicht ohne vorher je eine Stelle nach unten gegangen zu sein, solange, bis du ganz unten ankommst,{

wenn dann der Wert in der Liste an der Stelle, an der du gerade stehst nicht Null ist und

wenn dieser Wert größer ist als der bisherige Maximalwert, dann ist der neue Maximalwert dieser Wert aus der Liste.

wenn aber der Wert in der Liste an der Stelle, an der du gerade stehst, kleiner ist als der bisherige Minimalwert

dann ist der neue Minimalwert dieser Wert aus der Liste.

Der Farbumfang ist der Maximalwert weniger dem Minimalwert.

Übergib den Wert des Farbumfangs an die Instanz, die dich aufgerufen hat.

```
int rangeFinder(int[][][] valueList) {
    int x;
    int y;
    int z;
    int maxRangeVal = 1;
    minRangeVal = imgXlength * imgYlength * imgZrange;
    for ( x = 0; x < imgXlength; x++) {
        for ( y = 0; y < imgYlength; y++) {
            for ( z = 0; z < imgZrange; z++) {
                if (valueList[x][y][z] != 0) {
                    if (valueList[x][y][z] > maxRangeVal) {
                        maxRangeVal = valueList[x][y][z];
                    }
                    if (valueList[x][y][z] < minRangeVal) {
                        minRangeVal = valueList[x][y][z];
                    }
                }
            }
        }
    }
    mapRange = maxRangeVal - minRangeVal;
    return mapRange;
}
```

Spiele das Spiel **bereite alles vor.**

Von nun an stelle alles auf einer Fläche von 1100 mal 900 Punkten dar.

Nach der dir bekannten Art, Bilder anzufertigen, erzeuge f, das ein **neues Bild** ist.

Schicht-Elf nennen wir **deine Erinnerung an jenes Bild (Schicht-Elf)**.

Schicht-Zehn nennen wir **deine Erinnerung an jenes Bild (Schicht-Zehn)**.

Schicht-Neun nennen wir **deine Erinnerung an jenes Bild (Schicht-Neun)**.

Schicht-Acht nennen wir **deine Erinnerung an jenes Bild (Schicht-Acht)**.

Schicht-Sieben nennen wir **deine Erinnerung an jenes Bild (Schicht-Sieben)**.

Schicht-Sechs nennen wir **deine Erinnerung an jenes Bild (Schicht-Sechs)**.

Schicht-Fünf nennen wir **deine Erinnerung an jenes Bild (Schicht-Fünf)**.

Schicht-Vier nennen wir **deine Erinnerung an jenes Bild (Schicht-Vier)**.

Schicht-Drei nennen wir **deine Erinnerung an jenes Bild (Schicht-Drei)**.

Schicht-Zwei nennen wir **deine Erinnerung an jenes Bild (Schicht-Zwei)**.

Schicht-Eins nennen wir **deine Erinnerung an jenes Bild (Schicht-Eins)**.

Schicht-Null nennen wir **deine Erinnerung an jenes Bild (Schicht-Null)**.

Arbeite weiter mit den Bildern, die du dir gemerkt hast.

Zeichne einen Sehstrahl im Raum (nenne dazu den momentanen

Entfernungswert des Augpunkts in horizontaler Richtung, den

Entfernungswert des Augpunkts in der Tiefe, den

Entfernungswert des Augpunkts in der Höhe, den

Entfernungswert des Blickpunkts in horizontaler Richtung, den

Entfernungswert des Blickpunkts in der Tiefe und den

Entfernungswert des Blickpunkts in der Höhe).

void setup () {

size(1100, 900, OPENGL);

PFrame f = new **PFrame**();

bitmapLayer11 = loadImage("Bitmap_gradient_80x80_1-bit_Layer11.gif");

bitmapLayer10 = loadImage("Bitmap_gradient_80x80_1-bit_Layer10.gif");

bitmapLayer09 = loadImage("Bitmap_gradient_80x80_1-bit_Layer09.gif");

bitmapLayer08 = loadImage("Bitmap_gradient_80x80_1-bit_Layer08.gif");

bitmapLayer07 = loadImage("Bitmap_gradient_80x80_1-bit_Layer07.gif");

bitmapLayer06 = loadImage("Bitmap_gradient_80x80_1-bit_Layer06.gif");

bitmapLayer05 = loadImage("Bitmap_gradient_80x80_1-bit_Layer05.gif");

bitmapLayer04 = loadImage("Bitmap_gradient_80x80_1-bit_Layer04.gif");

bitmapLayer03 = loadImage("Bitmap_gradient_80x80_1-bit_Layer03.gif");

bitmapLayer02 = loadImage("Bitmap_gradient_80x80_1-bit_Layer02.gif");

bitmapLayer01 = loadImage("Bitmap_gradient_80x80_1-bit_Layer01.gif");

bitmapLayer00 = loadImage("Bitmap_gradient_80x80_1-bit_Layer00.gif");

imageQueue();

moveRay(povXKoord,

povYKoord,

povZKoord,

targetXKoord,

targetYKoord,

targetZKoord);

}

Spiele das Spiel **zeichne alle festen Körper()** {

Es gibt einen Zahlenwert, den wir Dunkelheit nennen,
und der ganz gering angenommen wird.

Sieh dir die Liste der festen Körper an und hör auf damit,
aber nicht bevor du die Bedingung erfüllt hast (geh ganz nach links;
dann geh bis zum rechten Ende;
mach immer nur einen Schritt)

aber nicht bevor du die andere Bedingung erfüllt hast (geh ganz
nach vorne; dann geh bis zum hinteren Ende;
mach immer nur einen Schritt)

und setze jetzt den Wert der Dunkelheit auf null,

aber nicht bevor du die letzte Bedingung erfüllt hast
(geh ganz nach unten; dann geh bis zum obersten Ende;
mach immer nur einen Schritt)

und erhöhe jeweils den Wert der Dunkelheit um 20.

Wenn die Randbedingung stimmt (Die Liste der festen Körper weist
an der Stelle, an der du bist, einen Wert aus, der null ist)

dann fülle alles, das du in Zukunft zeichnest mit dem hellsten
Farbton, von dem du den Dunkelheitswert abziehst.

Zeichne von jetzt an keine Umrißlinien mehr.

nimm dein Zeichenwerkzeug,

geh an die angegebene Stelle und

zeichne einen Würfel.

Jetzt leg das Zeichenwerkzeug wieder weg.

void **spaceRender()** {

int darkness = 0;

for (voxelXKoord = 0; voxelXKoord < imgXlength ;
voxelXKoord++) {

for (voxelYKoord = 0; voxelYKoord < imgYlength ;
voxelYKoord++) {

darkness = 0;

for (voxelZKoord = 0; voxelZKoord < imgZrange ;
voxelZKoord++) {

darkness = darkness + 20;

if (voxelSpace[voxelXKoord][voxelYKoord][voxelZKoord] == 1) {

fill(255 - darkness, 255 - darkness, 255 - darkness);

noStroke();

pushMatrix();

translate(gridStepX * voxelXKoord, gridStepY * voxelYKoord,
gridStepZ * voxelZKoord);

box(gridStepX - voxelGapX, gridStepY - voxelGapY,
gridStepZ - voxelGapZ);

popMatrix();

}

}

}

}