



Herausgeber:
Arndt von Koenigsmarck

Femme digitale

3D-Charaktere modellieren
und in Szene setzen

Das menschliche Auge

Im ersten Workshop erstellten wir bereits Skizzen und Bilder, die für die folgende Modellierung als Referenzmaterial dienen. In diesem Kapitel können wir daher direkt mit der Umsetzung beginnen.

2.1 Allgemeine Vorgehensweise

Wie bei allen folgenden Arbeitsschritten wurden die vorgestellten Techniken und Werkzeuge so ausgewählt, dass sie in allen gängigen 3D-Programmen leicht reproduziert werden können. Aus diesem Grund benutzen wir auch Polygon-Modelling, wir konstruieren also alle Objekte aus Dreiecken und Vierecken. Dies ist praktisch der kleinste Nenner aller 3D-Programme und steht somit überall zur Verfügung.

Was die nachfolgende Texturierung, also die Definition der Oberflächen, betrifft, so lässt sich dabei leider nicht so einfach ein allgemein gültiges Schema entwickeln. Jedes Programm verfolgt andere Ansätze, die sich zudem optisch stark voneinander unterscheiden können. So basieren fast alle Systeme auf so genannten Nodes, die wie kleine Recheneinheiten funktionieren.

Es gibt Nodes, die z.B. einfach ein geladenes Bild darstellen, Nodes, die wie Photoshop-Filter funktionieren, und Einheiten, die direkt auf die Berechnungsstrahlen des Raytracers zugreifen, um z.B. ein Ergebnis in Abhängigkeit vom Betrachtungswinkel der Oberfläche abhängig zu machen.

Raytracer: Dies ist die wohl am häufigsten benutzte Berechnungsmethode, bei der ausgehend von der Position des virtuellen Betrachters Strahlen ausgesendet werden. Trifft einer dieser Strahlen auf eines der Objekte in der Szene, kann der Raytracer an diesem Punkt die Helligkeit berechnen und mit den vergebenen Oberflächeneigenschaften verrechnen. Bei transparenten oder spiegelnden Objekten werden von diesem Auftreffpunkt aus weitere Strahlen losgeschickt, um z.B. hinter dem transparenten Objekt liegende Objekte auf die gleiche Weise abzutasten. Im Vergleich zu physikalisch noch exakteren Methoden, die z.B. diffuse Lichtstreuung und Reflexion mit einbeziehen können, bietet der Raytracer eine gute Alternative, da damit recht schnell hochwertige Bilder berechnet werden können. Die Nachteile der nicht ganz exakten Lichtsimulation lassen sich oft mit etwas Erfahrung durch Setzen zusätzlicher Lichtquellen ausgleichen.

Dies stellt nur einen Bruchteil der Möglichkeiten dar und soll Ihnen verdeutlichen, wie komplex allein diese Thematik sein kann.

Wir werden uns daher bei der Besprechung der Oberflächen auf allgemeingültige Bestandteile beschränken. Diese übertragen Sie dann in das von Ihrer Software benutzte Materialsystem.

Bevor es so weit ist, beschäftigen wir uns nun aber mit dem menschlichen Auge und seiner Umsetzung in 3D. Das Auge ist der Spiegel der Seele, wie schon der Dichter weiß, und daher sollte diesem Objekt besonders viel Aufmerksamkeit zuteil werden.

Glücklicherweise ist zumindest die äußere Form des Auges recht leicht umzusetzen. Zudem kann der Durchmesser eines menschlichen Auges mit ungefähr 2,5 cm angegeben werden.

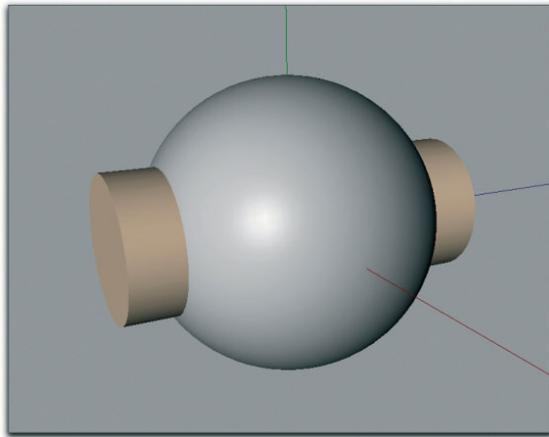


Abbildung 2.1: Kugel und Zylinder als Basis für die Modellierung des Auges

2.2 Grundkörper als Basis

Aufgrund dieser Eckdaten erzielen wir durch eine Kugel in der entsprechenden Größe bereits eine gute Annäherung an die tatsächliche Form. Dabei bietet es sich an, in Ihrer 3D-Software auf die so genannten Grundobjekte oder Primitives zurückzugreifen.

Dies sind geometrische Grundformen wie Kegel, Zylinder oder eben Kugeln. Sie brauchen diese Objekte daher nicht selbst zu konstruieren, sondern können diese in beliebiger Größe sofort abrufen.

Erzeugen Sie also eine Kugel mit einem Radius von 1,25 cm und zusätzlich einen Zylinder mit einem Radius von 0,6 cm. Legen Sie die Höhe des Zylinders so an, dass er die Kugel durchstößt, wie es in **Abbildung 2.1** zu sehen ist.

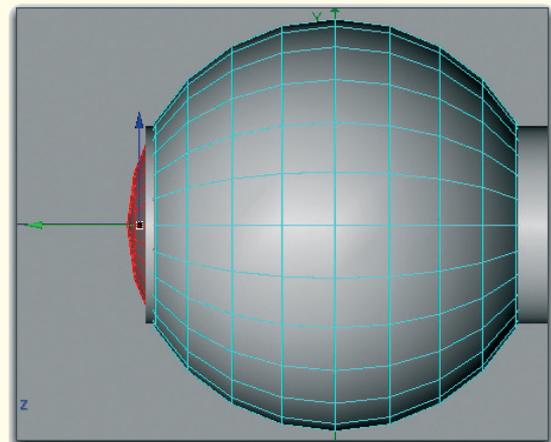
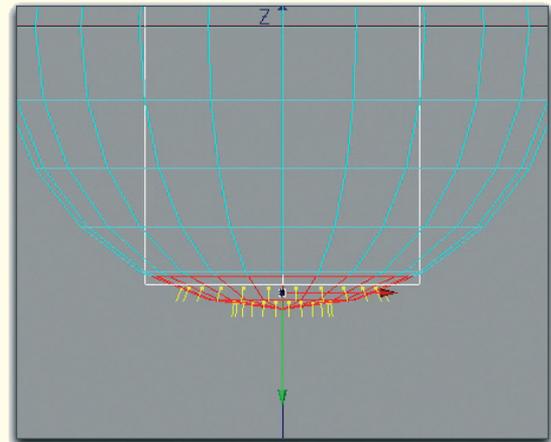
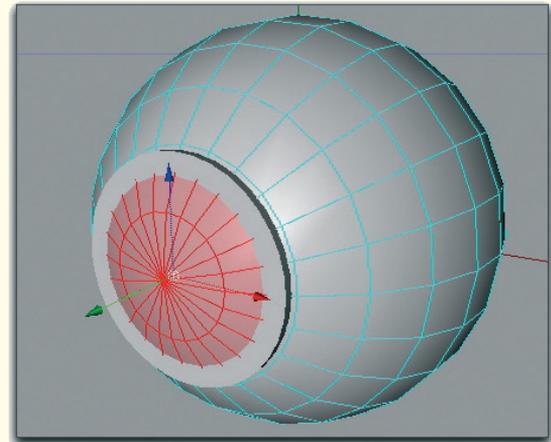


Abbildung 2.2: Modifizierung des Kugel-Pols

Die Vorgabe des Zylinderdurchmessers kommt zustande, weil die Pupille bei frontaler Betrachtung des Auges ungefähr das halbe Auge ausfüllt.

Die Umsetzung der Kugel aus Polygonen sieht in allen 3D-Programmen gleich aus. Es gibt zwei Polpunkte, von denen ausgehend sich die Flächen ringförmig über die Kugel ausbreiten. Diese ringförmige Anordnung der Flächen kommt uns hier sehr entgegen, da die Pupille schließlich auch kreisrund ist.

Da wir uns hier mit der möglichst anatomisch korrekten Modellierung des Auges beschäftigen wollen, kommt jedoch noch hinzu, dass sich die klare Hornhaut über der Pupille nach außen wölbt.

Sorgen Sie also durch Rotation der Kugel dafür, dass die beiden Kugelpole auf der Längsachse des Zylinders liegen, der Zylinder die Kugel also an den Polen durchstößt.

Erzeugen Sie dann einen zusätzlichen Kantenring dort, wo der Zylinder die Kugel durchstößt. Dies ist nur dort nötig, wo später die Pupille liegen soll, also nicht zusätzlich auf der hinteren Halbkugel des Auges.

Selektieren Sie die zwischen dem vorderen Polpunkt und dem neu hinzugekommenen Kantenring liegenden Polygone, so wie es [Abbildung 2.2](#) zeigt.

Um die angesprochene Auswölbung zu konstruieren, dürfen diese Flächen nicht einfach nach vorne verschoben werden. Es sollte schon eine Art Linse geformt werden. Verschieben Sie daher den selektierten Teil etwas nach vorne und ergänzen Sie darin zusätzliche Kantenringe. Diese erlauben es uns, die Form in diesem Bereich weiter zu steuern.

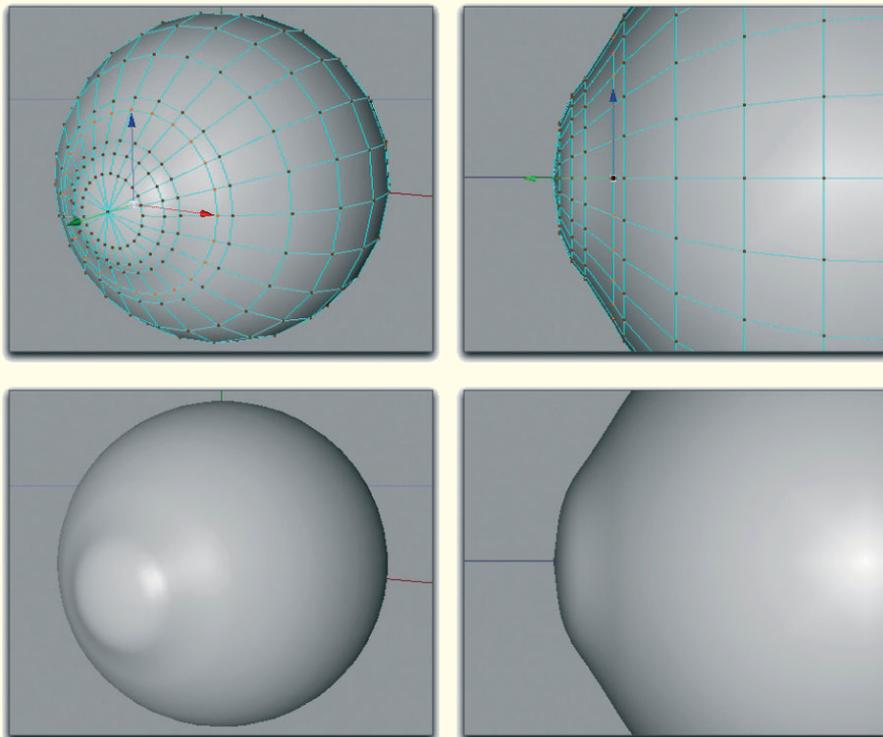


Abbildung 2.3: Ausbeulung über der Pupille

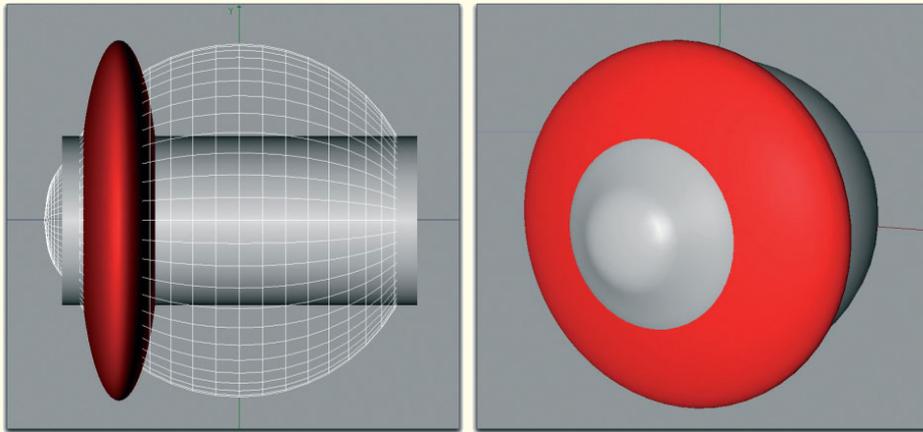


Abbildung 2.4: Gequetschte Kugel als Platzhalter für die Pupille

Abbildung 2.3 verdeutlicht, wie dies aussehen könnte. Wie Sie vor allem in den seitlichen Ansichten sehen können, bleibt der Bereich um den vorgezogenen Pol relativ flach. Ein sich anschließender Polygonring erzeugt den eigentlichen Übergang von der Linse zum Augenkörper. Der Effekt ist besonders in [Abbildung 2.3](#) am schattierten Objekt gut zu erkennen.

Letztlich sorgt diese Ausbeulung später für realistischere Glanzlichter und Reflexionen auf dem Auge. Übertreiben Sie es jedoch nicht zu sehr, damit die Augenlider später noch eng genug am Auge modelliert werden können.

Damit ist die äußere Hülle des Auges bereits komplett und wir können uns mit der Pupille einschließlich Iris beschäftigen.

2.3 Die Pupille mit Iris

Die Pupille ist ein ringförmiger Muskel, in dessen Mitte eine kreisförmige Öffnung verbleibt. Die Größe dieser Iris kann durch die Pupille gesteuert werden, um die Menge an Licht zu kontrollieren, die in das Auge fällt.

Hinter der Iris liegt mit Bändern fixiert eine Linse. Diese drückt die Pupille im Bereich der Iris leicht nach außen.

Wir werden diese Form wie beschrieben modellieren. Es gibt jedoch noch zahlreiche andere Möglichkeiten, die z.B. eine effektivere Beleuchtung des Auges erleichtern. Wie andere Künstler damit umgehen, erfahren Sie in den Zwischenkapiteln. Wie immer gibt es mehrere Wege nach Rom.

Um uns eine Grundlage für die Modellierung der Pupille zu schaffen, greifen wir erneut auf ein Kugelgrundobjekt zurück.

Weisen Sie diesem die gleiche Größe wie der Augenkugel zu und sorgen Sie auch hierbei dafür, dass ein Polypunkt in Blickrichtung des Auges liegt.

Skalieren Sie dann die neue Kugel so entlang der Blickrichtung des Auges, dass diese zu einer Linse gequetscht wird. [Abbildung 2.4](#) zeigen diese Linse rot eingefärbt aus zwei verschiedenen Perspektiven.

Wie ebenfalls aus diesen Bildern ersichtlicht wird, verschieben Sie diese Linse so in Richtung Pupille, dass der in die Augenkugel ragende Teil der Linse ungefähr mit der Zylinderfläche übereinstimmt. Der Zylinder hat dann seine Schuldigkeit als Hilfsobjekt getan und kann aus der Szene entfernt werden.

Löschen Sie die hinteren und die außerhalb des Auges liegenden Polygone der Linse. Es verbleibt eine flache Schale in Form einer Kontaktlinse. Durch Extrudieren des äußeren Rands und der Flächen nahe am Pol entsteht eine Form, wie sie in [Abbildung 2.5](#) zu sehen ist.

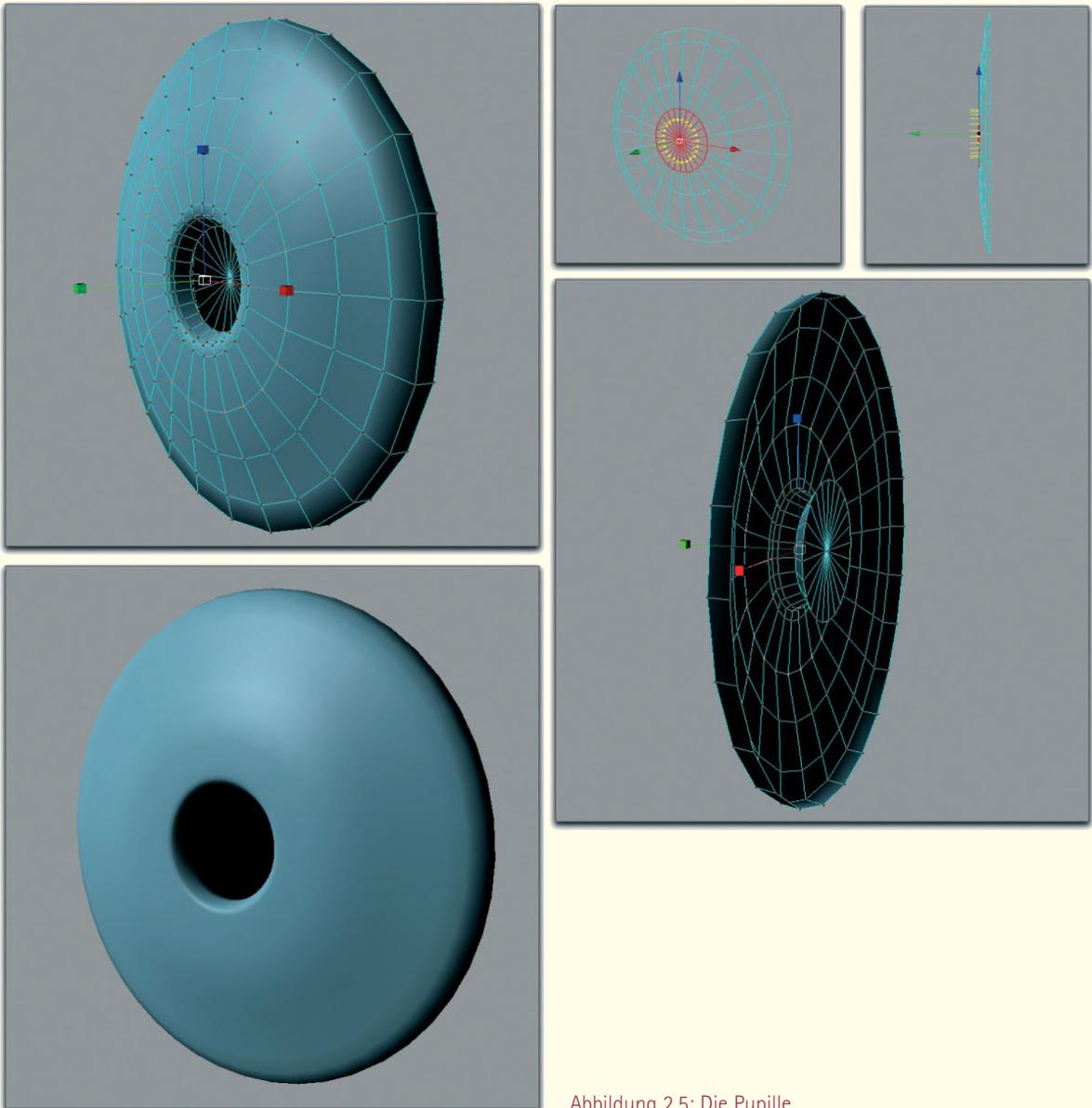


Abbildung 2.5: Die Pupille

Extrudieren: Bei dieser Aktion werden selektierte Kanten oder Flächen verschoben. Gleichzeitig entstehen an den Rändern der verschobenen Elemente neue Flächen. Auf diese Weise lassen sich dünne Objekte einfach verdicken oder z.B. Ausbuchtungen erzeugen, wie wir sie an der Iris benötigen.

Im Normalfall kommt das Schwarz der Iris nur zustande, weil man durch diese Öffnung in das Innere des Auges schaut. Von dort wird jedoch nur sehr wenig Licht wieder aus dem Auge heraus reflektiert. Dafür sorgt nicht zuletzt der Muskel der Pupille, der die Iris steuert. Nur in seltenen Fällen, wie dem direkten und unerwarteten Einleuchten z.B. bei einer Blitzlichtaufnahme, wird der Augenhintergrund rot sichtbar.

Wir sparen uns daher das Konstruieren des inneren Auges und erzeugen stattdessen eine sackförmige Ausbeulung hinter der Iris. Dieser Bereich wird später schwarz eingefärbt und simuliert damit ausreichend genau das Innere des Auges.

Der äußere Rand der Pupillenscheibe wird leicht nach hinten extrudiert. Dies erzeugt eine höherwertige Schattierung der Oberfläche an den Rändern, die für uns wichtig ist, da die Transparenz der Auswölbung des Auges weich zum Augenkörper übergehen soll.

Ansonsten sorgen Sie bei der Pupille dafür, dass der Bereich um die Iris abgerundet wird und etwas nach vorne rückt. Dies simuliert den bereits angesprochenen Druck der dahinter liegenden Linse. Fällt Ihnen dies aufgrund einer zu geringen Anzahl an Polygonen schwer, fügen Sie zusätzliche Polygonreihen hinzu.

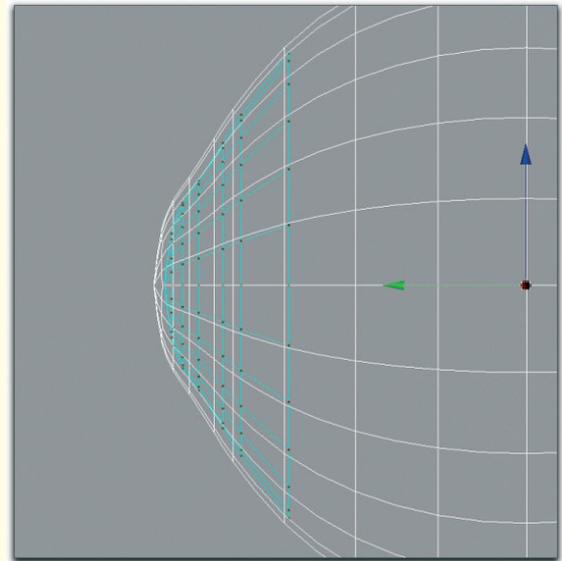


Abbildung 2.6: Verdoppelte Front des Auges

2.4 Doppelwandig arbeiten

Damit ist die Modellierung des Auges praktisch abgeschlossen. Ich möchte jedoch noch einen Schritt weiter gehen und den vorderen Teil des Auges doppelwandig ausführen.

Dies hat aus meiner Sicht den Vorteil, dass sich die Refraktion in diesem Bereich später natürlicher verhält.

Refraktion: Abweichung der ursprünglichen Lichtrichtung in transparenten Materialien. Man spricht auch von der Brechung des Lichts. Die Stärke dieses Effekts kann numerisch über den Brechungsindex angegeben werden. So hat z.B. Luft bei Raumtemperatur einen Wert von 1.0, flüssiges Wasser ca. 1.33 und Glas ungefähr einen Brechungsindex von 1.6.

Gehen Sie einfach so vor, dass Sie die Flächen im vorderen Teil des Auges in ein neues Objekt kopieren und dieses etwas kleiner skalieren. Das Ergebnis ist in [Abbildung 2.6](#) zu sehen.

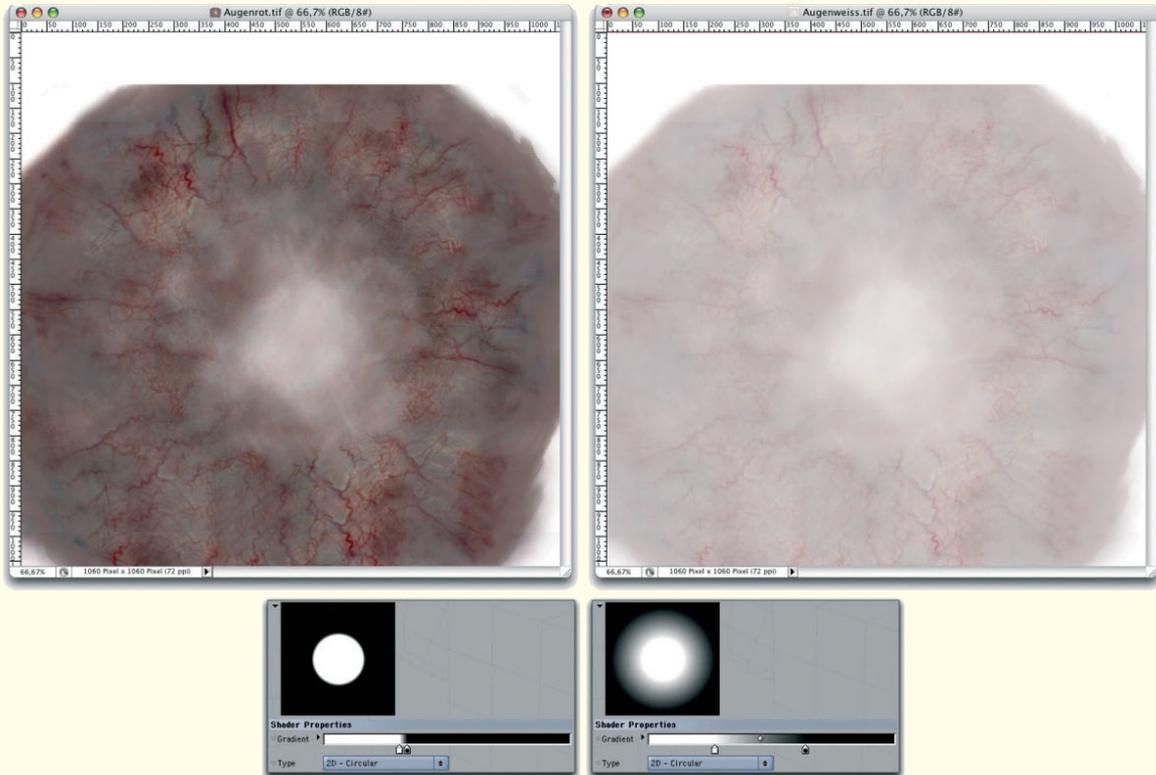


Abbildung 2.7: Bildtexturen für das Auge

2.5 Texturierung des Auges

Für die Texturierung von Objekten, also das Zuweisen von Oberflächeneigenschaften, kommen diverse Techniken in Frage. Man kann alle Eigenschaften parametrisch zuweisen und Shader benutzen. Darunter versteht man im 3D-Programm implementierte Algorithmen, die aufgrund von mathematischen Berechnungen Muster und physikalische Eigenschaften erzeugen können.

Vorhersehbarer ist die Benutzung von Bildern, wobei diese dann jedoch oft vorher aufwändig erstellt werden müssen.

Zudem unterliegen Bilder einer festen Pixelgröße, was beim Heranzoomen zum Aufpixeln oder Weichzeichnen führen kann. Shader sind dagegen weitestgehend auflösungsunabhängig. Beide Varianten haben also ihre Vor- und Nachteile. Wir werden sowohl mit Shadern als auch mit Bildern bei der Texturierung arbeiten.

Wir beginnen dabei mit der Oberfläche des Auges. Ich habe mir dazu eine Collage verschiedener Aufnahmen menschlicher Augen erstellt. **Abbildung 2.7** zeigt zwei Varianten dieser Collage, einmal mit Schwerpunkt auf den Blutgefäßen und einmal leicht entsättigt für die eher helle Oberfläche des Auges.

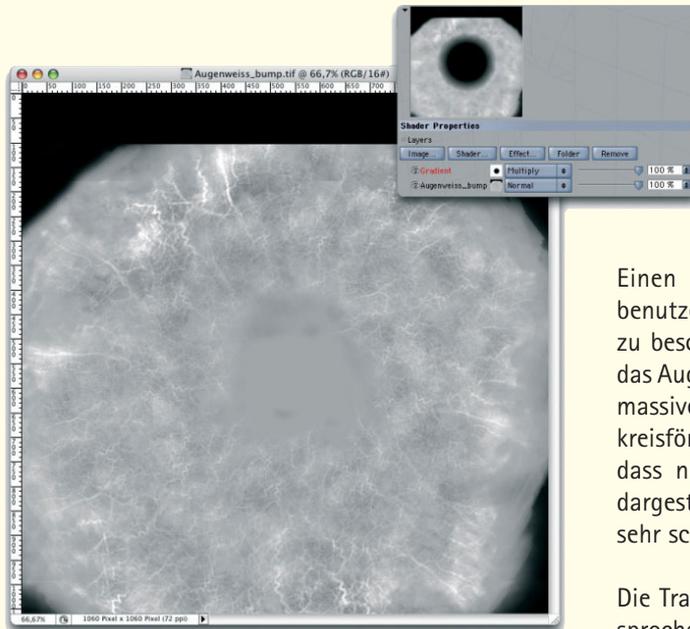


Abbildung 2.8: Bump-Map des Auges

Wichtig ist bei allen Arbeiten, die Fotos einbeziehen, dass Glanzlichter und Schatten möglichst aus den Bildern zu entfernen sind. Ansonsten können sich später Unstimmigkeiten bei der Benutzung ergeben, wenn z.B. die virtuelle 3D-Beleuchtung von links kommt und das dargestellte Objekt eine Bildtextur benutzt, auf der das Licht von rechts kommt.

Die Bildinformation sollte sich daher möglichst nur z.B. auf die Farbwerte des dargestellten Objekts beschränken.

Mithilfe eines kreisförmigen Helligkeitsverlaufs mische ich die beiden Bildinformationen so, dass die farbintensiven Adern und Rötungen nur am Rand des Auges sichtbar werden. Solch einen Farbverlauf könnten Sie auch als Bild vorbereiten, dies hätte jedoch den Nachteil, dass Sie nie genau wissen, ob z.B. der Übergang weich genug ist oder an der richtigen Stelle beginnt.

Ich verwendete daher hier einen Verlauf-Shader, der direkt in der 3D-Software angepasst werden kann.

Einen ähnlich dimensionierten Verlauf-Shader benutze ich, um die Transparenz des äußeren Auges zu beschreiben. Dort, wo der Shader weiß ist, bleibt das Auge durchsichtig. Je dunkler der Shader ist, desto massiver bleibt das Objekt sichtbar. Die Größe dieses kreisförmigen Helligkeitsverlaufs wird so eingestellt, dass nur der Bereich über der Pupille transparent dargestellt wird. Der Übergang zum Augenweiß wird sehr schmal gehalten.

Die Transparenz sollte zudem mit dem bereits angesprochenen Brechungsindex verknüpft werden, damit das Licht, das durch diese Hornhaut auf die Pupille fällt, auch natürlich gebrochen wird.

Da das Auge zu einem großen Teil mit Wasser gefüllt ist, können wir hier einen Brechungsindex von 1.33 annehmen.

Dem Auge wird zudem das in [Abbildung 2.8](#) dargestellte Bild als Bump-Map zugewiesen. Eine Bump-Map ist ein Graustufenbild oder ein Shader, dessen Helligkeitsinformationen benutzt werden, um die Schattierung auf der Oberfläche des Objekts zu beeinflussen. Helle Bereiche erscheinen dann leicht erhaben und dunkle wie Absenkungen oder Riefen. Der Effekt ist rein optischer Natur und verändert daher die Geometrie des Objekts nicht. Man setzt Bump-Maps hauptsächlich für feine Details ein, deren Modellierung zu aufwändig sein würde. Denken Sie z.B. an Hautporen, Pickel oder kleine Fältchen.

[Abbildung 2.9](#) demonstriert diesen Effekt beispielhaft an einer Kugel. Wie im unteren Teil der Abbildung zu sehen, ist der Effekt eine rein optische Täuschung. Die Umrisse der Kugel bleiben auch mit Bump-Map identisch.

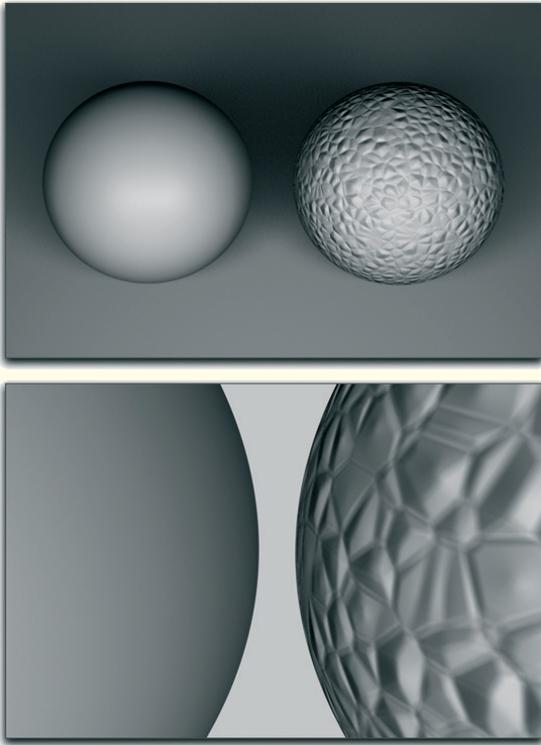


Abbildung 2.9: Der Bump-Effekt

Die Bump-Map für die Adern ist aus einer Kontrastverstärkung des farbigen Adernbilds mit nachfolgender Entsättigung entstanden.

Auch hier wird wieder ein runder Farbverlauf eingesetzt, der durch Multiplikation mit der Bump-Map deren Intensität steuert. Auf diese Art und Weise wird die transparente Hornhaut über der Pupille von dem Bump-Effekt frei gehalten. Dies ist besonders wichtig, weil Bump-Maps auch das Aussehen der Glanzpunkte und Spiegelungen verändern. Die Front des Auges soll schließlich schöne runde Glanzlichter zeigen.

Neben Farbe und Bump geben wir dem Auge noch leicht spiegelnde Eigenschaften, um den Eindruck einer feuchten Oberfläche zu verstärken.

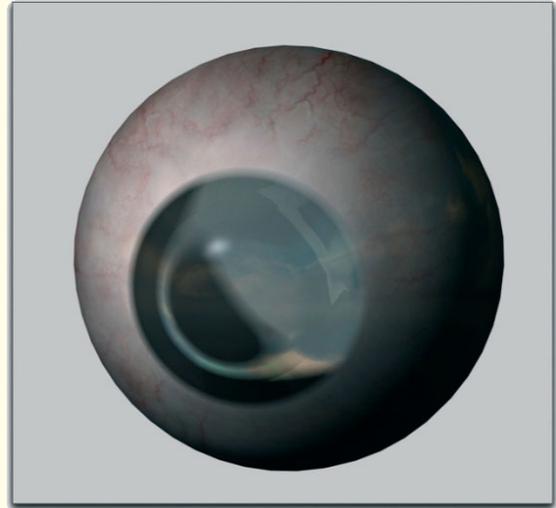


Abbildung 2.10: Das äußere Auge samt Textur

Schließlich wird ein schwaches Glanzverhalten aktiviert und das komplette Material dem äußeren Auge zugewiesen. In diesem Fall projiziere ich das Material einfach frontal auf das Auge. Dies führt zwar an den Rändern der Kugel zu Texturverzerrungen, aber dieser Bereich wird später sowieso nie sichtbar werden.

Passen Sie gegebenenfalls die kreisförmigen Helligkeitsverläufe an, damit die Transparenz wirklich nur über der Pupille sichtbar wird und sich der Bump-Effekt nur auf dem Augenweiß zeigt.

Wie Sie in [Abbildung 2.10](#) erkennen können, habe ich kurzfristig noch eine Landschaftsaufnahme hinzugefügt, um die Intensität der Spiegelung und die Refraktion der Linse überprüfen zu können.

Ein entsprechendes Bild können Sie z.B. auf eine sehr große Kugel legen, um Spiegelungen auf Objekten zu simulieren.

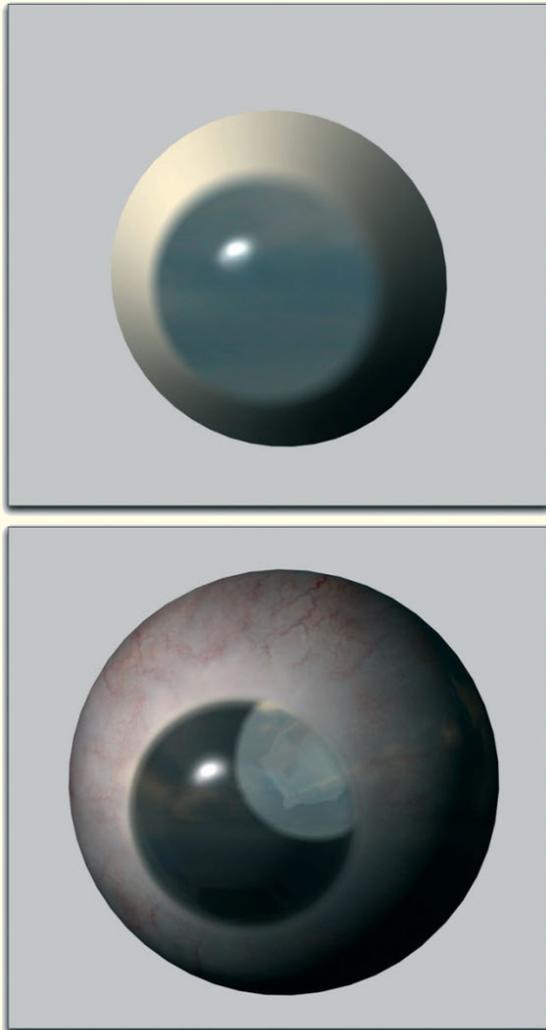


Abbildung 2.11: Zusätzlich texturierte Linse

2.5.1 Die Linse

Duplizieren Sie das Material des äußeren Auges und tauschen Sie in der Kopie den Farbanteil durch ein helles Gelb aus.

Auf die Spiegelung und den Bump-Effekt können wir hier ganz verzichten. Verstärken Sie jedoch die Darstellung von Glanzlichtern etwas. Die Transparenz und deren Brechungsindex können so übernommen werden.

Weisen Sie dieses neue Material dann dem Objekt zu, das die Doppelwandigkeit der Hornhaut darstellen soll. Das obere Bild vom [Abbildung 2.11](#) zeigt, wie das Material dort wirkt.

In der unteren Hälfte der Abbildung sehen Sie das zusätzlich berechnete äußere Auge. Nur auf den ersten Blick hat sich an dessen Darstellung nichts verändert. Vergleichen Sie besonders den transparenten Bereich der Linse mit der vorherigen Abbildung des Auges ohne den doppelwandigen Effekt. Die Lichtbrechung wirkt nun dort ganz anders.

Zudem wurde der Glanzpunkt verstärkt und er erhält durch die Zweiteilung der Oberflächen mehr Tiefe und Variation.

Kommen wir also zum letzten noch fehlenden Teil, der Pupille. Deren Oberfläche ist aus der Nähe betrachtet sehr komplex. Unterschiedlichste Farbvariationen sind möglich und die faserige Struktur sorgt für unterschiedliche Mischungsverhältnisse dieser Farben bei Veränderungen der Iris.

Der einfachste Weg zur Texturierung ist sicherlich die Verwendung von geeigneten Fotos. Diese sind jedoch nicht leicht zu beschaffen, da das Auge im entspannten Zustand teilweise von den Lidern verdeckt wird. Hinzu kommen die bereits angesprochenen störenden Einflüsse wie z.B. Spiegelungen, Schatten oder Glanzpunkte, die sich oft nicht vollständig aus den Bildern entfernen lassen.

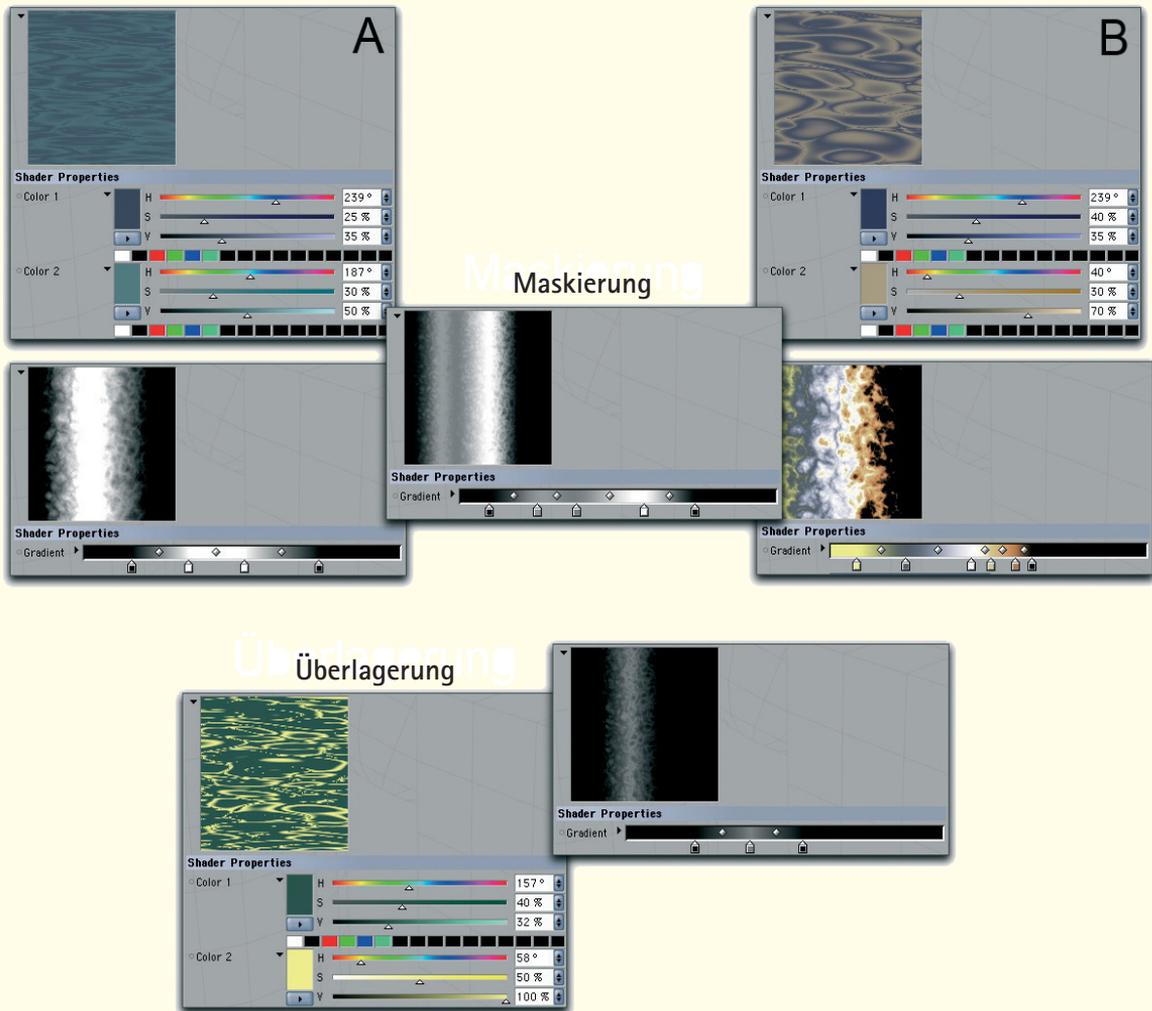


Abbildung 2.12: Shader-Verknüpfungen für die Oberfläche der Pupille

2.5.2 Die Pupille

Ich möchte daher die Oberfläche der Pupille mit Shadern aufbauen. **Abbildung 2.12** bietet eine entsprechende Anleitung.

Ich beginne mit dem Erzeugen einer leicht streifigen Struktur mit Grau- und Blautönen.

Dafür bieten sich so genannte Noise-Shader an, die verschiedene Muster mit beliebigen Farbwerten erzeugen können. Das Ergebnis dieses Shaders wird mit einem senkrechten Helligkeitsverlauf multipliziert. Diese beiden Shader sind oben unter dem Buchstaben A zu finden.

Ein ähnlich strukturierter Noise-Shader mit bläulichen und orangefarbenen Strukturen wird ebenfalls mit einem Verlauf multipliziert, der diesmal jedoch ein chaotisches Farbgemisch abbildet.

Nur der äußere rechte Rand, der später den äußeren Rand der Pupille belegen wird, bleibt schwarz. Diese Shader sind unter dem Buchstaben B zu finden.

Der mittig abgebildete Helligkeitsverlauf funktioniert nun wie eine Überblendung zwischen den Ergebnissen von A und B.

Im Prinzip entspricht dies einer Alpha-Maske in Photoshop. Hierbei wird das Ergebnis der B-Shader über dem Ergebnis der A-Shader angeordnet und entsprechend dem Maskierungsverlauf dargestellt.

Dort, wo die Maskierung schwarz ist, bleibt also das B-Ergebnis sichtbar. Helle Bereiche lassen dagegen das A-Ergebnis durchschimmern. Auf diese Weise können Sie schnell mehrere Farbkombinationen und Mischungsverhältnisse durchspielen, ohne mit Bildern arbeiten zu müssen.

Generell ist darauf zu achten, dass die Pupille zum äußeren Rand hin dunkler wird. Bei vielen Menschen ist sogar eine deutliche schwarze Umrandung zu erkennen.

Um dem Ganzen noch mehr Variationen zu verleihen, multipliziere ich einen Noise-Shader mit gelben und grünen Farbflächen mit einem Helligkeitsverlauf. Dieser bildet nur einen schmalen senkrechten Streifen geringer Helligkeit und mit ausgefransten Rändern ab.

Das Resultat dieser im unteren Teil der Shader-Abbildung dargestellten Berechnung wird dem bislang kombinierten Material der Pupille überlagernd beige-mischt. Das Ergebnis ist in [Abbildung 2.14](#) zu sehen.

Neben diesen rein farblichen Zuweisungen sollte auch die Oberflächenstruktur aufgegriffen werden. Ich habe dazu einen Noise-Shader mit einer blasigen

Struktur benutzt und diesem horizontale Linien überlagert.

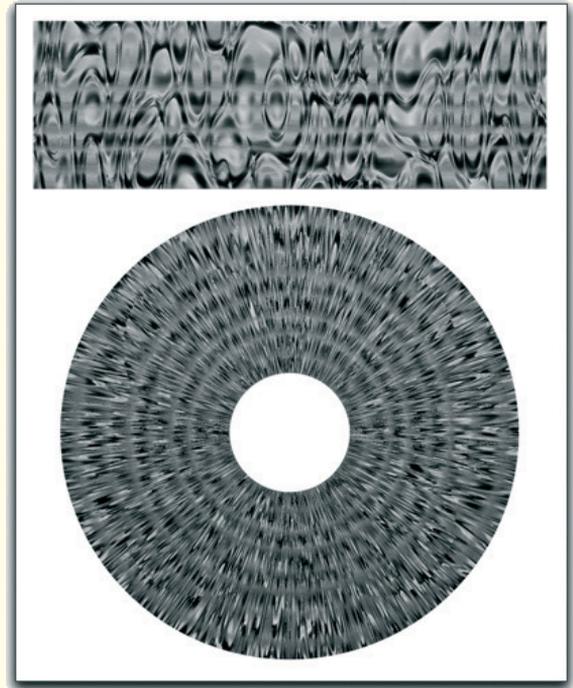


Abbildung 2.13: Bump-Struktur der Pupille

[Abbildung 2.13](#) zeigt diese Struktur auf einer Ebene. Im unteren Teil ist die gleiche Shader-Kombination einer Scheibe zugewiesen worden. Durch die Krümmung wird der gewünschte Effekt sichtbar, der die radialen Fasern der Pupille darstellen soll.

Dieser Effekt kommt im Bump-Kanal des Materials zum Einsatz. Zusätzlich wird ein schwaches und breites Glanzlicht aktiviert, damit die Oberfläche nicht zu trocken und rau wirkt.

Schließlich können alle Objekte wieder sichtbar gemacht werden, um das Gesamtergebnis zu begutachten. [Abbildung 2.14](#) gibt das Erscheinungsbild der Pupille und des kompletten Auges nach Abschluss der Texturierung wieder.

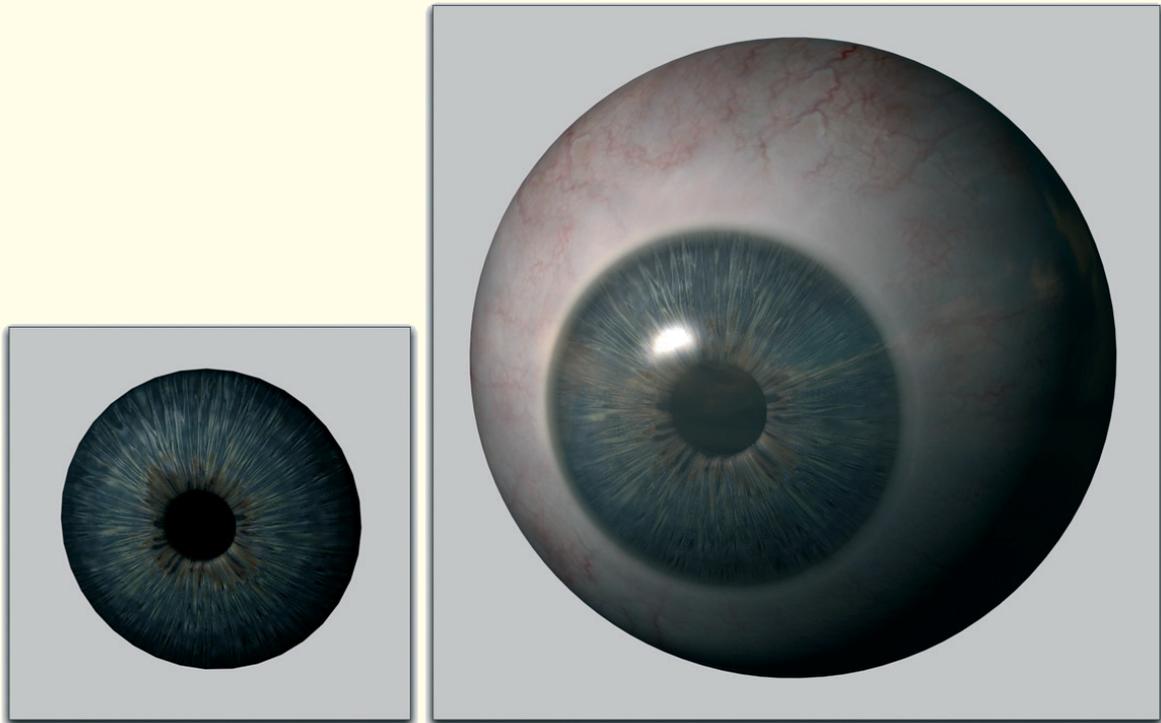


Abbildung 2.14: Das fertig texturierte Auge

2.6 Vorbereitungen für die Modellierung des Gesichts

Der Grund, weshalb wir das Auge zuerst modelliert und texturiert haben, ist, dass es uns als eine Art Anker oder Richtschnur für die folgende Modellierung des Gesichts dienen kann.

Da wir auch dort mit dem Bereich um die Augen beginnen, können wir gleich alle Polygone an die Augen anpassen und für einen optimalen Sitz der Lider sorgen.

Zudem bestimmen der Abstand der Augen voneinander und deren Größe die Proportionen des gesamten Kopfs, wie Sie bereits im ersten Workshop gesehen haben.

Öffnen Sie also Ihre 3D-Applikation und laden Sie dort das im ersten Kapitel vorbereitete Bild mit den zwei Ansichten des Kopfs.

Je nachdem, wie Sie normalerweise Bildvorlagen innerhalb Ihres Programms verwenden, können Sie die Bilder als Hintergrundbild einblenden lassen oder als Textur auf zwei Flächen verwenden.

Wichtig ist in jedem Fall, dass beide Bildvorlagen im 3D-Raum einen Winkel von 90° zwischen sich einschließen.

Es ist zudem sinnvoll, das Bild mit der frontalen Kopfansicht in der XY-Ebene bzw. der frontalen Editoransicht und die seitliche Kopfansicht in der ZY-Ebene, also der seitlichen Editoransicht, zu platzieren.

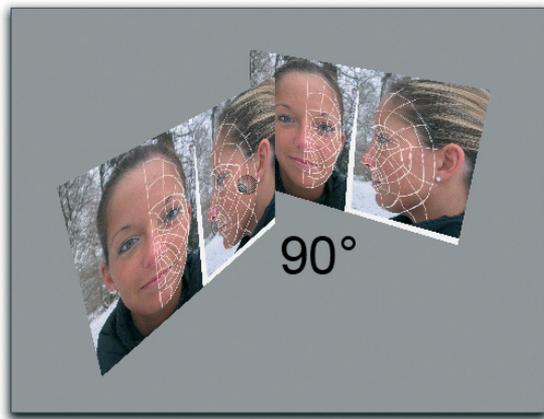


Abbildung 2.15: Einbringen der Bildvorlagen in den 3D-Raum

Dies kann z.B. wie in [Abbildung 2.15](#) aussehen. Wenn Sie wie hier beide Kopfansichten in einem Bild gespeichert haben, kann das Abgleichen der Ansicht entfallen. Dies haben wir zuvor bereits in Photoshop erledigt.

Sie brauchen also das Bild nur auf eine entsprechend skalierte 3D-Ebene zu projizieren und davon eine Kopie anzufertigen. Diese Kopie drehen Sie 90° um die Senkrechte. Verschieben Sie dann beide Stellwände so, dass Sie in der frontalen und der seitlichen Editoransicht freies Sichtfeld auf die jeweilige Kopfansicht haben.

Laden Sie zu dieser Szene das zuvor erzeugte 3D-Auge hinzu und skalieren Sie dieses gegebenenfalls auf eine zu den Bildern passende Größe.

Platzieren Sie das Auge so, dass die Lagen der Iris und der Pupille ungefähr mit der auf der Bildvorlage übereinstimmen.

Duplizieren Sie dann das Auge und verschieben Sie diese Kopie in der frontalen Ansicht horizontal auf die andere Seite des Gesichts. Für diesen und folgende Arbeitsschritte ist es sinnvoll, die Symmetrieachse des Gesichts auf die Y-Achse Ihres 3D-Bezugssystems zu verschieben.

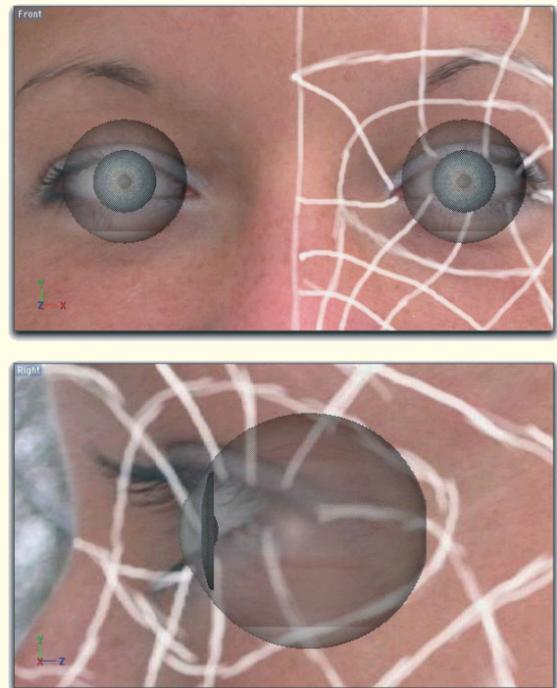


Abbildung 2.16: Einpassen des Auges

Auf diese Weise lassen sich Objekte von einer Gesichtshälfte einfacher auf die dazu passende Position der anderen Hälfte verschieben, da Sie numerische Eingaben benutzen können. So kann das linke Auge z.B. auf einer X-Position von 150.0 liegen. Ein einfacher Vorzeichenwechsel dieser Koordinate verschiebt das Auge dann auf die andere Seite des Gesichts.

Dies ist auch deshalb sinnvoll, weil wir bei der Modellierung des Gesichts auf dessen symmetrische Eigenschaften zurückgreifen. Wir brauchen also nur eine Gesichtshälfte zu modellieren und lassen dann die fehlende Hälfte durch Spiegelung automatisch ergänzen.

[Abbildung 2.16](#) gibt eine gute Position für die Augenobjekte wieder. Im nächsten Teil dieses Arbeitsbeispiels beginnen wir mit der Modellierung des Gesichts.