

# DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

JANUAR | FEBRUAR 01:2021



## Hardware

Loupedeck, Sensel, TourBox,  
AMD vs. Intel und mehr

## Praxis

Flame, Blender, Fusion,  
Resolve 17, Mocha...

## Theorie

EU-Drohnenverordnung,  
Rigging, Remote Rendering

## ... und Projekte

L'Artista, Jamiri, Aixterior  
und Carmodeling

# Jamiri Titeldesign

„Gödel, Escher, Gott“ klingt episch und sieht auch so aus: Das Konterfei von niemand Geringerem als dem Allmächtigen selbst zielt das neue Album des deutschen Comiczeichners Jan Michael Richter aka Jamiri (Bild 01). Selbstredend, dass der Schriftzug für die Titelseite entsprechend erzählerisch daherkommen muss. Mit diesem Artikel möchte ich einen Einblick gewähren in den Entstehungsprozess dahinter.

von Marc Potocnik

Jamiris Comics lese ich im Grunde schon seit der Studienzeit. Seine Comics erschienen regelmäßig in Medien wie Unicum und Spiegel Online und seine eigenen Comicbände sind nach wie vor begehrte Artikel in der Buchhandlung. Ich selbst zeichne quasi, seit ich denken kann; ein leeres Blatt Papier übte schon in Kindertagen eine ungeheure Faszination auf mich aus. Zwar habe ich selbst irgendwann den Tuschestift gegen Maus und Tastatur eingetauscht, dennoch liegen die Wurzeln meines heutigen Tuns im Zeichnen. Umso mehr be-

wunderte ich die Zeichenkunst hinter Jamiris Arbeiten, die er früher mit Filzmarkern zu Papier brachte (siehe Bild 02) und heute mit einem Wacom Cintiq Pen Display anfertigt.

Über Pointen und Gags in Jamiris satirischen Beobachtungen habe ich mich sowie so schon oft schlapp gelacht. Daher freute ich mich besonders auf diese Zusammenarbeit und deren Umsetzung in Cinema 4D und Photoshop. Die fertige Arbeit kann unter [www.renderbaron.de/stills](http://www.renderbaron.de/stills) begutachtet werden.



**Bild 02:** Wirklich echte Zeichenkunst: Einst mit Filzmarkern (Originalblatt aus „Homepages“, 1997), heute mit einem Wacom Cintiq Pen Display (s. Bild 01).

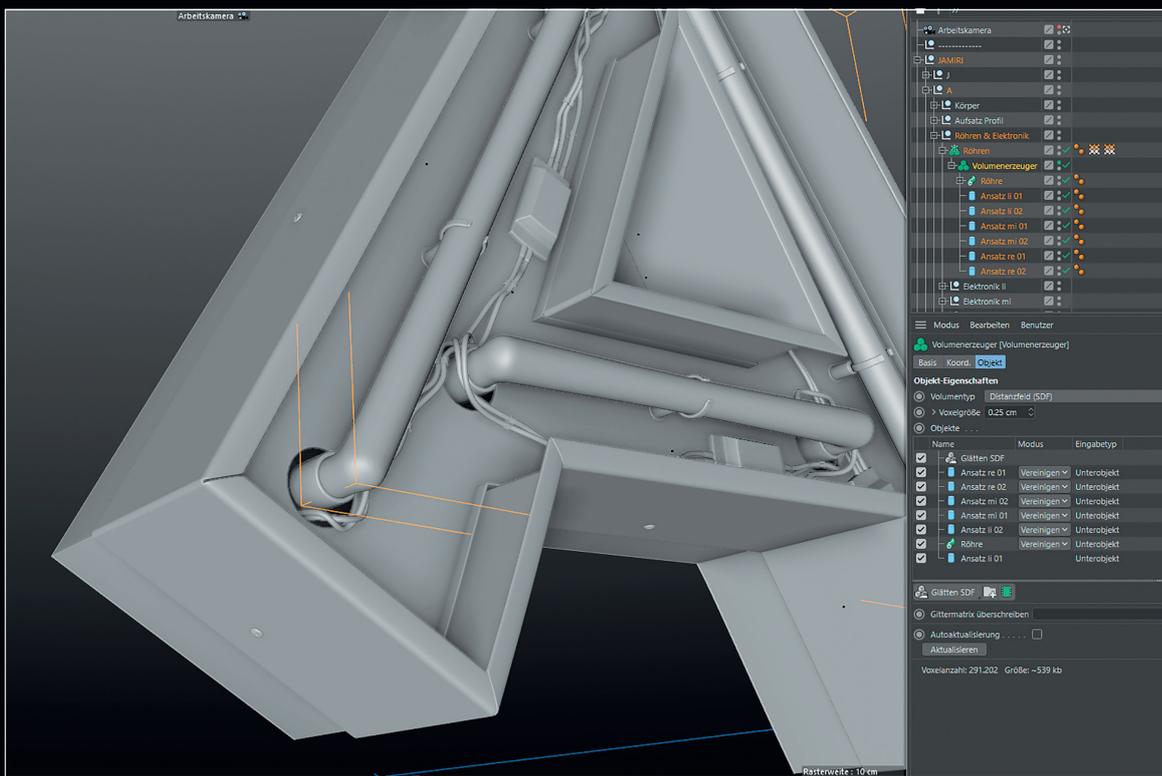




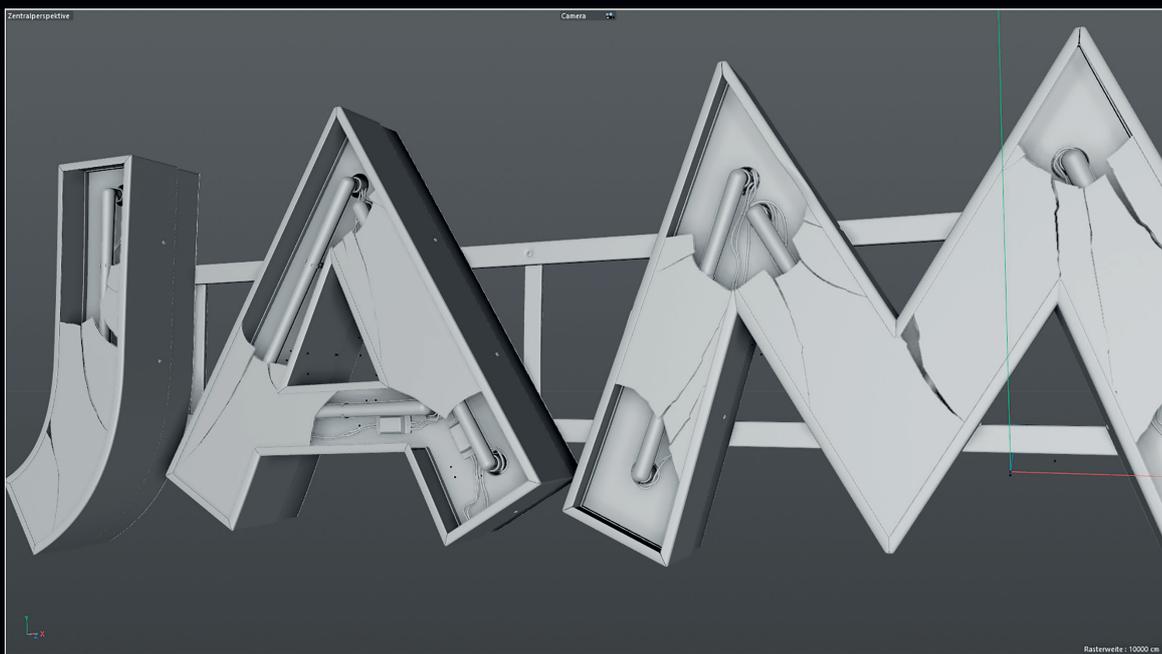
**Bild 01:**  
Titelseite des neuen  
Jamiri-Bandes „Gödel,  
Escher, Gott“. Der All-  
mächtige war schon  
mal sympathischer.

Der Schriftzug für das  
neue Jamiri-Album  
„Gödel, Escher, Gott“,  
made by renderbaron.

[www.renderbaron.de](http://www.renderbaron.de)



**Bild 03:** Gestopfte Kabel, Blechprofile und Leuchtstoffröhren. Letztere wurden aus einfachen Sweep-Objekten und Zylindern zusammengestellt und mittels eines Cinema-4D-Volumenobjekts mit einer organischen Oberfläche versehen.



**Bild 04:** Verdecken und dennoch andeuten: Die Linienführung der Acrylglasrisse setzt die der Leuchtstoffröhren kompositorisch fort.

## Modeling & Formensprache

Der Gedanke hinter dem Titelschriftzug war, ihm die Anmutung einer ramponierten Leuchtreklame zu verleihen. Einst in voller Pracht strahlend, sollten nun Kratzer, Abnutzungen und Verfall die beschädigten Lettern dominieren. Parallelen zum Autor sind dabei vollkommen unbeabsichtigt. Der Schriftzug besteht aus den folgenden Komponenten (siehe Bild 03):

- ▷ Die Grundform der Buchstaben beruht auf dem Jamiri-Originalfont, den ich als Spline extrudierte und das Ergebnis in Polygone umwandelte. Ein umlaufendes Gehäuse und eine nach vorne abschließende Blende als Aufhängung für Leuchtstoffröhren ergänzten das Ganze.
- ▷ Elektronische Komponenten wie Kabel und Trafokästchen sorgen für einen erhöhten Detailgrad. Die Formen folgen dabei einer gewissen Gesetzmäßigkeit:

Sie hängen durch, sind schlampig in Öffnungen reingestopft, haben Steifheit und Knicke. Das Modeling der Kabel erfolgte dabei mit frei gezeichneten Splines und Sweep-Objekten.

- ▷ Die Leuchtstoffröhren sind ebenfalls nicht ganz perfekt. Sie verjüngen sich in der Mitte ein wenig und münden krumm und schief in ihre Fassungen. Für eine solche organische Form ist Cinema 4Ds Volumenobjekt ideal, welches das Verbinden

mehrerer einfacher Objekte mit einer gemeinsamen geschwungenen Oberfläche ermöglicht.

- ▷ Aufsatzprofile schließen jeden Buchstaben nach vorne ab und sorgen visuell für eine stärkere Linienführung. Vor allem der vordere Falz wirkt visuell betonend und hält außerdem das Acrylglas an Ort und Stelle. Auf Basis der Grundform der Buchstaben (s.o.) war für die Aufsatzprofile klassisches Polygon-Modeling gefragt.
- ▷ Das Acrylglas ist in so einer Szene natürlich das Material, das zuerst zu Bruch geht. Acrylglas zersplittert durch seine Flexibilität jedoch großteiliger als normales Glas. Deswegen sind Trümmer und Risse recht großzügig und geschwungen dimensioniert. Basis waren auch hier gezeichnete Splines (siehe Bild 04).

Die Komponenten des Schriftzugs muten nach einem Baujahr in den 60er- oder 70er-Jahre an. Das passt somit zum Konzept des mittlerweile desolaten Zustands. Die Szenerie folgt dabei mehr einer visuellen statt einer mechanischen Logik. Heißt: Was plausibel aussieht und die Wirkung unterstreicht, wird einer physikalischen Korrektheit (Das würde doch niemand so montieren!) vorgezogen. So liegt z.B. die gesamte Elektronik auf der Vorderseite der Blende, was eigentlich wenig Sinn macht. Es ist aber visuell viel interessanter, als wenn man das Ganze schön langweilig hinter die Blende packt.

Diese visuelle Logik wird auch in anderen Bereichen durchdekliniert: Z.B. folgen die Bruchkanten des Glases ebenfalls dem Prinzip der visuellen Logik. Das Problem war nämlich, dass das Acrylglas die Leuchtstoffröhren und die Elektronik dahinter zu guten Teilen verbirgt. Die Risse zeichnete ich jedoch so, dass sie im Grunde die Linienführung der Leuchtstoffröhren fortführen. Das Acrylglas würde so also niemals zerbrechen, die Linienführung der Brüche lässt die Leuchtstoffröhren dahinter aber besser erahnen und stärkt so die Komposition (siehe Bild 04).

## Bildbasiert oder prozedural?

Grundsätzlich kann man Shading – also das Gestalten von Oberflächen und deren Beschaffenheit – in zweierlei Herangehensweisen anpacken: auf Basis von Fotos oder prozedural.

Für den **bildbasierten Ansatz** nimmt man eine Fototextur und klebt sie quasi wie eine Tapete auf das Objekt. Ad hoc bieten Fototexturen – wenn sie zuvor anständig aufbereitet und bearbeitet

wurden – einen hohen Realismus. Ein zerkratztes Metall fotografieren, in Photoshop aufbereiten, fertig. Das Problem ist aber die Auflösung: Je nach gewünschter Kameraannäherung und Auflösung des Renderings muss man unter Umständen mit sehr hoch aufgelösten Texturen von z.B. 8K oder 16K arbeiten. Und dennoch sind auch dieser Auflösung Grenzen gesetzt. Dann sieht man relativ schnell Treppchen, Kompressionsartefakte, Unschärfen usw. Schwierig wird es erst recht, wenn der Kunde wünscht, Charakteristika von Beschädigungen der Oberflächen oder grundsätzliche Strukturen des Materials zu ändern – das kann in langwierige Retusche-Orgien in Photoshop ausarten oder eine völlig neue Textur erfordern.

In diesem Fall ist der zweite Ansatz zum Kreieren einer Oberflächentextur besser: der **prozedurale Ansatz**. Er basiert eben nicht auf Pixelbildern als Fototapete, sondern auf regelbasierten Eigenschaften in Form von Shadern. Shader sind im Grunde kleine Programme, die es erlauben, bestimmte Attribute des Aussehens mit Parametern zu bestimmen. Nach einem initial höheren Arbeitsaufwand sind Änderungen an der Textur dann meist eine Sache von wenigen Mausklicks.

Zudem sind eigens entwickelte Shader-Setups sehr recyclingfreundlich. Der Erstaufwand ist natürlich höher, aber meist verfeinert man solche Setups von Projekt zu Projekt wie kleine Software-Versionen. So ist das zerkratzte Metall im Jamiri-Schriftzug mit Bestandteilen aus Setups aufgebaut worden, die ich bereits in meinen vorherigen Projekten „Apollinarisstr.“ und „Oberbilk“ er-sonnen habe.

All diese Überlegungen führten schlussendlich zu der – dann recht leichten – Entscheidung, das Shading des Jamiri-Schriftzuges prozedural zu gestalten.

## Shading – Materialsystem

Okay, prozeduraler Ansatz, das wäre geklärt. Nun stellte sich noch die Frage, welches der beiden verfügbaren Materialsysteme ich benutzen würde: a) das Standardmaterialsystem, in dem sämtliche Materialattribute als sogenannte Kanäle gegliedert sind, oder b) das neuere Node-Materialsystem, bei dem Materialattribute als miteinander zu verknüpfende logische Einheiten (Nodes) vorliegen? Die Wahl fiel auf (Trommelwirbel!!) das Standardmaterialsystem. Warum?

- ▷ **Erstens:** Das Standardmaterialsystem bietet zwar nicht die logische Verknüpfung der Node-Materialien, hat allerdings den Vorteil, dass es in der Bearbeitung sehr viel performanter ist als das

Node-Materialsystem. Das macht sich vor allem beim Arbeiten mit dem interaktiven Renderbereich bemerkbar. Für schnelle Vorschauen benutze ich häufig den Progressive Mode des physikalischen Renderers, da dieser ziemlich schnell ein grobes Bild der Szene erzeugt. Dies fügt sich gut zusammen mit dem interaktiven Renderbereich von Cinema 4D. Wenn dieses Zusammenspiel jedoch durch langsame Node-Materialien gebremst wird, macht das Arbeiten keinen Spaß. Deswegen: Standardmaterialsystem. Übrigens: Im neuen Release 23 von Cinema 4D sollte sich die Performance der Node-Materialien spürbar verbessert haben.

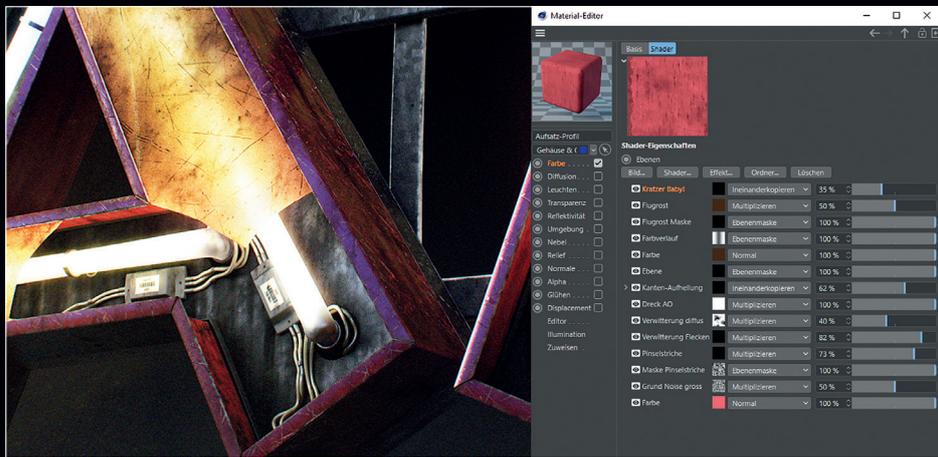
- ▷ **Zweitens:** Ich brauchte für das Acrylglas unbedingt Subsurface Scattering (SSS), also die korrekte Darstellung von Transluzenz bzw. einer Lichtstreuung im Volumen des Objekts. Und das ist momentan im Node-Materialsystem nicht für Standard-Renderer und physikalischen Renderer verfügbar. Da das Acrylglas ein zentrales Element der Szene ist, habe ich mich dann gleich komplett für das Standardmaterialien entschieden.

## Ein Haufen Kratzer

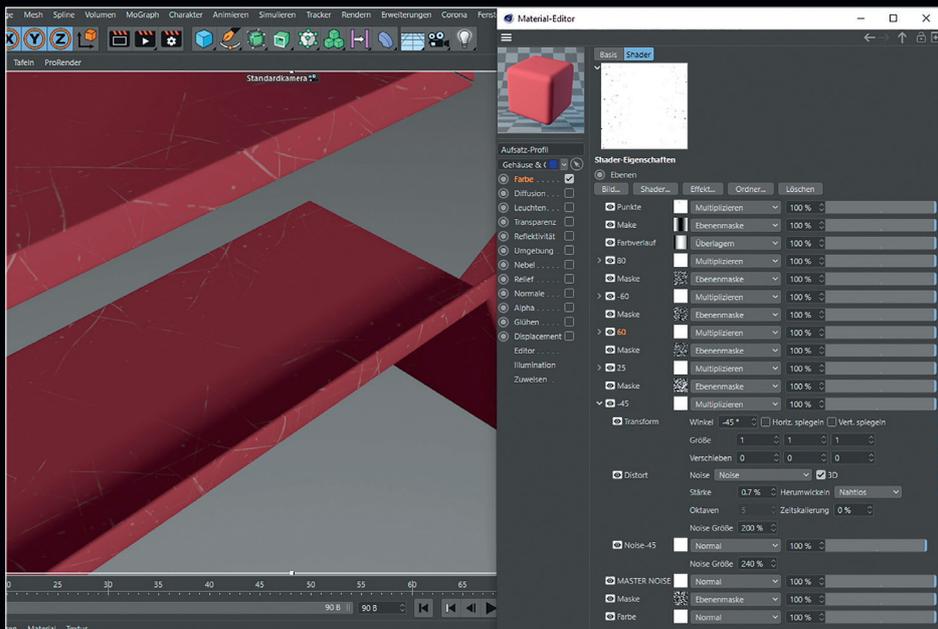
Das zerkratzte Metall der Buchstabengehäuse und der Aufsatzprofile besteht aus einem Ebenen-Shader als Container, in dem mehrere prozedurale Shader nach Photoshop-Manier übereinandergestapelt und miteinander verrechnet sind. Die Basis ist eine rötliche Farbe, über der in mehreren Ebenen Noises für großflächige Strukturvariationen und Verschmutzung liegen. Hinzu kommen Verwitterungseffekte, Kantenkennungseffekte für Beschädigungen auf Basis invertierter Ambient Occlusion (s.u.) und natürlich Kratzer.

Die Kratzer liegen an oberster Stelle in einem wiederum eigenen Ebenen-Shader, den ich „Kratzer, Baby!“ nannte (siehe Bild 05). Mit Blick in diesen Ebenen-Shader zeigt sich: Die Kratzer sind nichts anderes als an sich sehr ähnliche Noises, die lang und schmal skaliert wurden und in Ordnern liegen. Dort werden sie jeweils durch einen Transform-Effekt gedreht und einen Distort-Effekt verzerrt (siehe Bild 06).

Die Noises sind jeweils eine Kopie eines von mir als „Master Noise“ benannten Originals, das ich in diesem Ebenen-Shader zuunterst platziert habe. Zwischen den Ordnern liegen außerdem andere, größere Noises, die die Ordner jeweils maskieren und so für Unterbrechungen in den Noises darin sorgen.



**Bild 05: Der Farbkanal des rötlichen Metalls: ein Ebenen-Shader als Container für weitere prozedurale Shader. Ganz oben die Ebene „Kratzer, Baby!“ – ein weiterer Ebenen-Shader nur für Kratzer.**

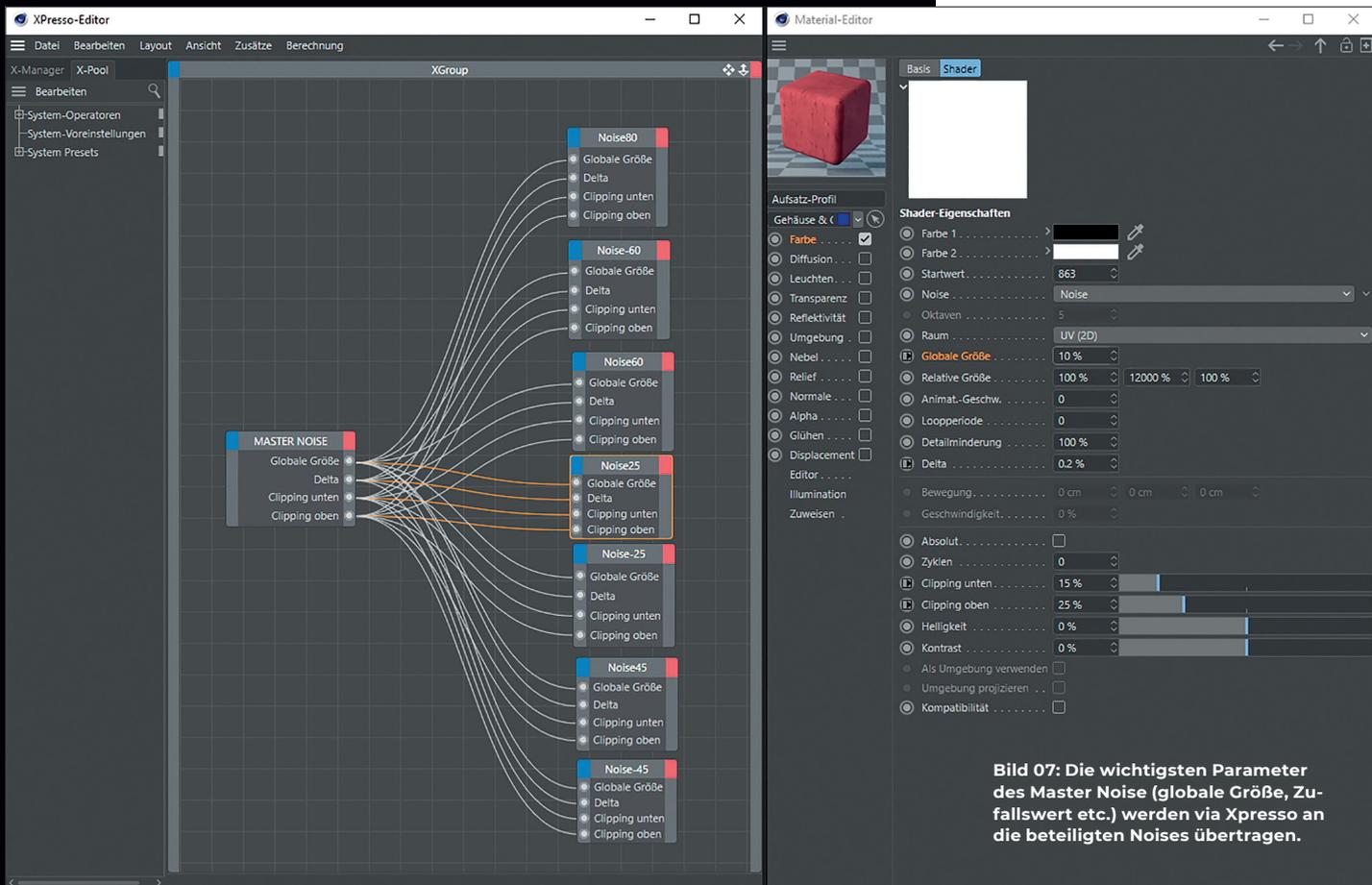


**Bild 06: „Kratzer, Baby!“ – ein langes, schmales Master-Noise wird in mehrere Ordner kopiert und unterschiedlich gedreht und verzerrt. Angewendet werden die Noises je im Raum „Textur“, denn das fertige Material wird am Ende in Quaderprojektion aufgetragen. Sorgen um UVs entfallen so.**

## Xpresso-gesteuerte Shader

Jetzt stellt sich die Frage: Wie kann ich, ausgehend von einem Shader-Parameter im Master Noise die gleichlautenden Parameter in den anderen Noises steuern? Die Antwort lautet: Xpresso. Der Xpresso-Editor ist im Grunde Node-basiertes Programmieren. Dazu wird ein Xpresso-Tag auf dem betreffenden Objekt erzeugt, per Doppelklick darauf öffnet sich der Xpresso-Editor. Die einzelnen Noise Shader werden hier hineingezogen und so als kleine logische Stückchen, also Nodes, abgelegt. Auf der rechten Seite der Nodes (z.B. des Master Noise) lassen sich bestimmte Eigenschaften durch Ports ausgeben und mit Verbindungen in entsprechende Ports auf der linken Seite anderer Nodes einklinken. Damit werden Parameter, wie z.B. globale Größe, Zufallswert (Seed) etc. übertragen.

**Beispiel:** Im Master Noise habe ich den Ausgangs-Port für die globale Größe erzeugt. Diese globale Größe habe ich weitergegeben an den Eingangs-Port aller anderen Noises (siehe Bild 07). Im Master Noise zeigt sich neben dem Parameter „Globale Größe“ daraufhin ein kleines Symbol aus einem Kreis und einem Dreieck: Das bedeutet, dass der Wert des Parameters via Xpresso weitergegeben wird. Wenn ich jetzt nun einen bestimmten Wert eingabe, wird genau dieser Wert an alle anderen Noises weitergegeben.



**Bild 07: Die wichtigsten Parameter des Master Noise (globale Größe, Zufallswert etc.) werden via Xpresso an die beteiligten Noises übertragen.**

Dasselbe Vorgehen habe ich ebenfalls für weitere Parameter angewendet wie z.B. Delta (Prägebrite für die Verwendung des Noise im Relief-Kanal) oder Clipping unten/Clipping oben (Tonwertverengung des Noise).

So eine Schaltung ist im Grunde ein Ausflug in die Denke der Elektrotechnik – für Designer (wie mich) erst einmal gewöhnungsbedürftig. Allerdings hat man auf diese Weise einen Ansatz, um abseits von Node-Materialien ein ewiges Copy-and-paste von Shadern zu vermeiden und mehr globale Kontrolle ins Spiel zu bringen.

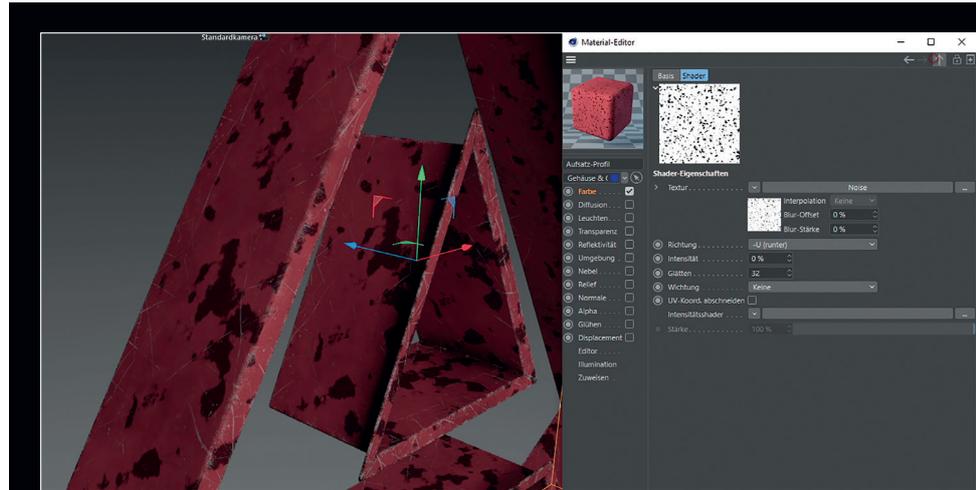
### Verwitterung und Taubendreck

Zwei Aspekte am Material der Aufsatzprofile fallen auf, nämlich Verwitterung und beschädigte Kanten. Der Verwitterungs-Shader ist eigentlich relativ simpel: Er nutzt Input-Material, in diesem Fall ein Noise (siehe Bild 08), und verwäscht es mit einem frei definierbaren Prozentwert (siehe Bild 09). Dies erzeugt einen Effekt, als ob lange Zeit Regenwasser über die Oberfläche mit ihren Unregelmäßigkeiten gelaufen wäre. Im konkreten Fall entschloss ich mich für einen Wert von 300 Prozent, da dies einen sehr weichen, lang gezogenen Effekt erzielt. Bei solchen Werten wird natürlich der Parameter „Glätten“ umso wichtiger: Damit ist im Grunde die Unterteilung, das Sampling, des Effekts gemeint. In unserem Fall wird eine Unterteilung von mindestens 32 Samples benötigt, um ein homogenes Ergebnis zu liefern.

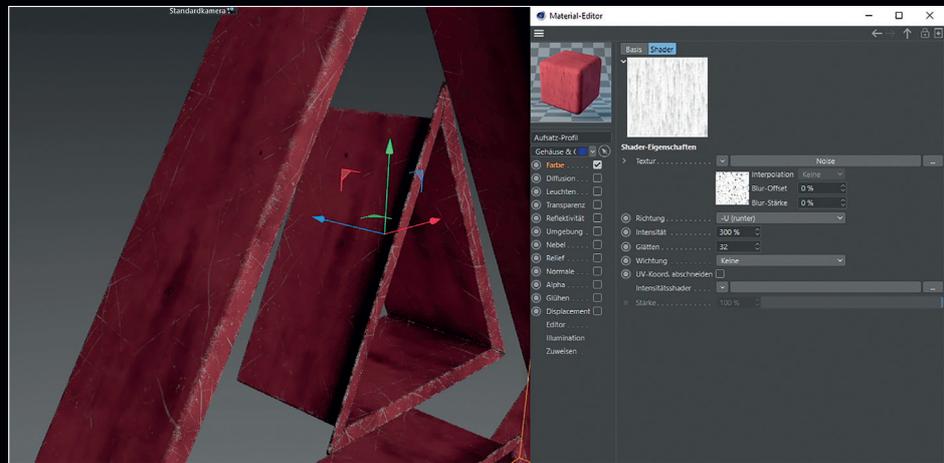
Auf dem rechten I des Jamiri-Schriftzugs findet sich übrigens eine recht großzügig dimensionierte Hinterlassenschaft einer Taube (siehe Bild 10). Dabei handelt es sich um eine einfache Ebene mit einem eigenen Material. Dieses basiert auf einem turbulent verzerrten kreisförmigen Farbverlauf von weiß nach graugrün und einem entsprechenden Pendant zum Freistellen im Alpha-Kanal. Die Startwerte beider Farbverläufe werden wie o.a. via Xpresso gesammelt gesteuert. Schlussendlich wird das Ganze noch durch einen Verwitterungs-Shader verwaschen, so wie es sich gehört. Zerkratztes Metall, vermodertes Acrylglas und sogar ein prozeduraler Shader für Taubendreck – was will man mehr?

### Kantenbeschädigungen

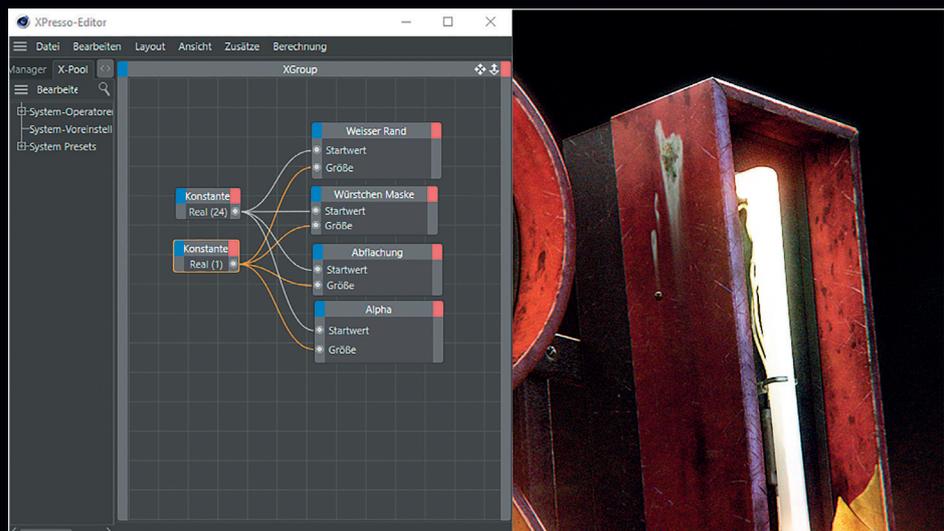
Kommen wir nun zu den Kantenbeschädigungen. Diese machen sich in Form von helleren, scheinbar abgeriebenen Stellen an den Kanten der Aufsatz-Profile bemerkbar. Erzeugt wurde dieser Effekt mit einem Stan-



**Bild 08:** Ein grobes, kontrastreiches Turbulenz-Noise in einem Verwitterungs-Shader, noch ohne Effekt.



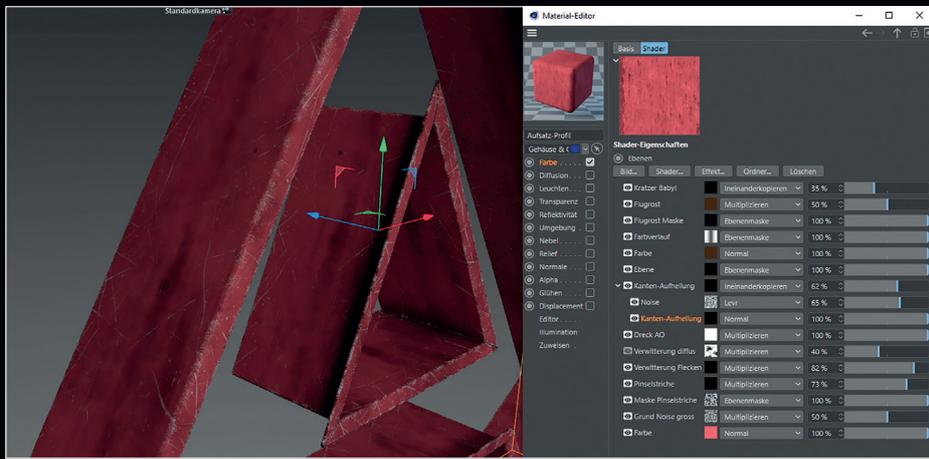
**Bild 09:** Dasselbe Noise mit 300% Verwaschen-Effekt durch den Verwitterungs-Shader.



**Bild 10:** Prozeduraler Taubendreck: zwei Zahlenwerte steuern via Xpresso Zufallswert und Größe der beteiligten Farbverlaufsturbulenzen.

dardverfahren für Kantenbeschädigungen: inverser Ambient Occlusion mit Noise-basierter Unregelmäßigkeit. In einem Ordner des Ebenen-Shaders liegt ein „Kanten-Aufhellung“ genannter Ambient Occlusion (AO)

Shader (siehe Bild 11). Normalerweise tastet AO seine Umgebung ja halbkugelförmig in Richtung der Flächennormalen nach Abdeckungen ab. Mit der Checkbox „Gegenrichtung“ (siehe Bild 12) kehrt man dieses

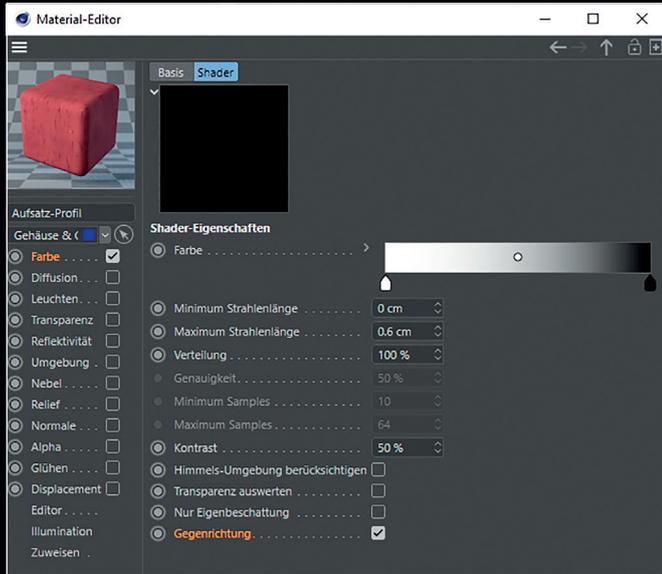


und berechnet aus beidem kombiniert eine Art Mittelwert-Scherschnitt. Das bedeutet: Durch Levr wird die langweilige, gerade Kantenaufhellung mit den Strukturen des darüberliegenden Noise aufgebrochen. Je transparenter ich dieses Noise im Ebenen-Shader beimische (hier 65 %), desto weicher wird der Levr-Effekt angewendet.

## Erweitertes Subsurface Scattering

Ein zentrales Element dieses Schriftzugs ist ja das hinterleuchtete Acrylglas. Als Cinema-4D-Artist denkt man da wahrscheinlich direkt an ein Stichwort: Subsurface Scattering (SSS), also die Streuung von Licht im Volumen eines Objekts. Und genau das ist auch hier im Spiel. Werfen wir einmal einen Blick in den Leuchten-Kanal des Acrylglas-Materials. Dort befindet sich – wie sollte es anders sein? – ein Ebenen-Shader (siehe Bild 13). An unterster Position, quasi als Basis, haben wir einen Subsurface Scattering Shader. Darüber gestapelt sind einige übliche Verdächtige: leicht farbige Noises, um das Ganze ein bisschen farblich zu variieren, und Ambient Occlusion für Kanten und Dreck.

Bleiben wir aber einmal beim Subsurface Scattering. Dieses wird mit einer recht brachialen Stärke von 4.000 % angewendet (siehe Bild 14). Das hängt damit zusammen, dass wir den Shader später durch weitere Komponenten abschwächen. Doch dazu später mehr. Die Farbe des SSS ist recht kräftig gesättigt, zudem läuft der Effekt nur über eine sehr kurze Pfadlänge von 0,01 Zentimetern. Das heißt, die Streuung des Lichts im Volumen des Acrylglasobjekts erfolgt nur über diese kurze Distanz. Dies erzeugt einen sehr definierten, der dünnen Stärke des Objekts angemessenen Look.

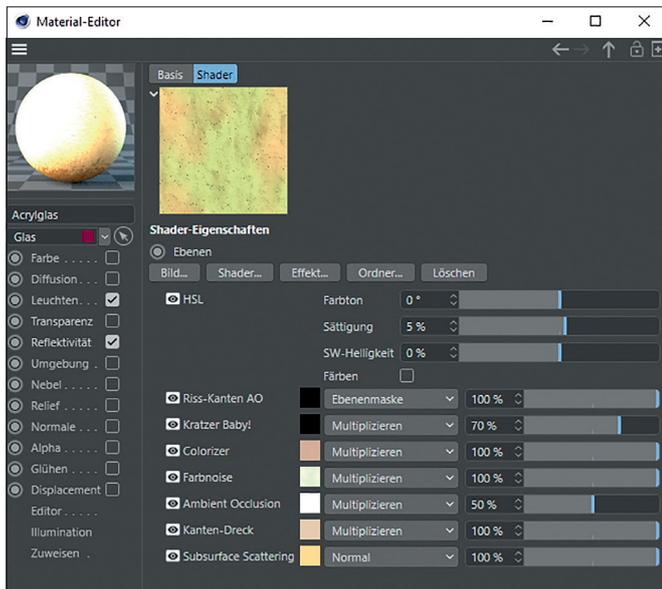


**Bild 11: Quasi ein Standardverfahren für Kanteneffekte: ein invertierter Ambient Occlusion Shader (hier benannt „Kanten-Aufhellung“) und darüber ein Noise im Ebenen-Modus „Levr“.**

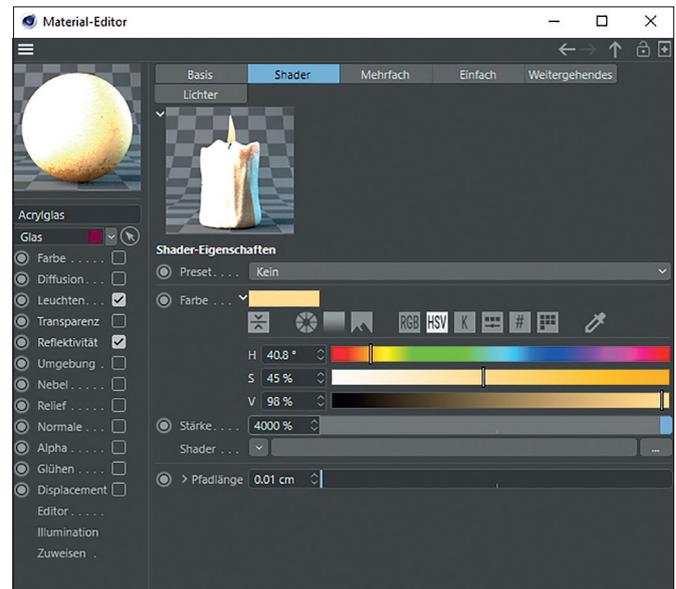
**Bild 12: Mit „Gegenrichtung“ sammelt Ambient Occlusion in die Gegenrichtung der Flächennormalen und fragt quasi: „Wo decke ich mich selbst ab?“**

Verhalten nun um. Ambient Occlusion fragt also: „Was liegt wohl hinter mir?“, sampelt in Gegenrichtung der Flächennormalen (also wo sich das Objekt selbst abdeckt) und erzeugt so einen AO-Effekt nur auf Kanten. Für helle Objektkanten habe ich den AO-eigenen Verlauf auf weiß-schwarz umgekehrt, da der Shader ja in einem Ordner im Modus „Inneinanderkopieren“ liegt. Weiß wird also dargestellt und Schwarz ignoriert.

Über dem so eingestellten AO-Shader liegt nun ein Noise (siehe Bild 11). Bei diesem handelt es sich um ein Turbulenz-Noise, angewendet im Raum „Welt“. Dadurch ignoriert es Texturprojektionen und UVs und durchdringt das Objekt räumlich. Dieses Noise liegt nun im Ebenenmodus „Levr“ über dem o.a. „Kanten-Aufhellung“. Grob vereinfacht erzeugt Levr eine Art Mittelwert von sich selbst und der Ebene darunter



**Bild 13: Das Subsurface Scattering des Acrylglases mit weiteren variierenden Komponenten innerhalb eines Ebenen-Shaders**



**Bild 14: Das Subsurface Scattering wird später durch Transparenz und Schmutz geschwächt und braucht daher eine brachiale Stärke. Die kurze Laufänge sorgt für definierte Ergebnisse.**



**Bild 15:** Das Acrylglas nur mit Subsurface Scattering – ausgeprägte Transluzenz, die aber noch blind und modulationsarm wirkt.



Im Tab sind außerdem die beteiligten echten Lichtquellen ausgeschlossen. Dieses Vorgehen stellt sicher, dass nur das Licht der Leuchtstoffröhren zum Subsurface Scattering beiträgt. Zur verwendeten Global Illumination dann später mehr. Gehen wir das Ganze einmal schrittweise mit Zwischenrenderings an.

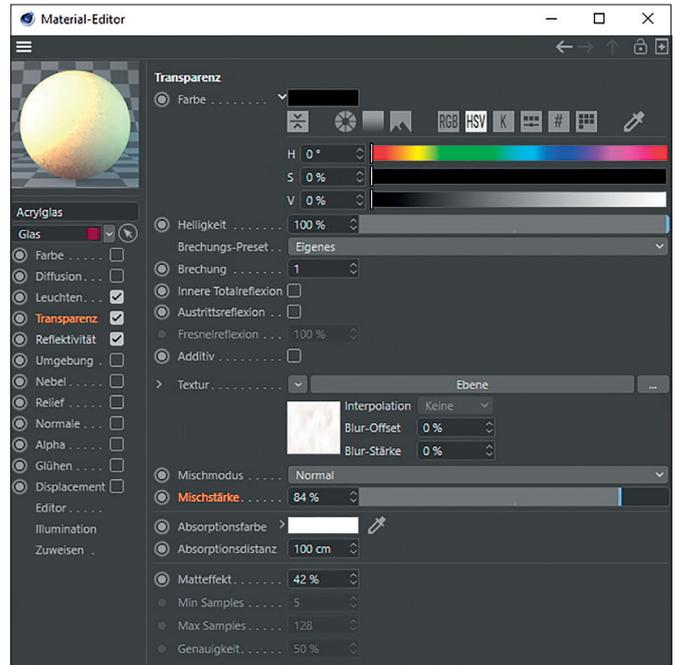
### Subsurface Scattering

Das Ergebnis nur mit SSS und farblicher Variation durch Noises ergibt eine glaubwürdige Transluzenz an den korrekten Stellen. Wo die Leuchtstoffröhren nicht so stark leuchten, ist der Effekt entspre-

chend schwächer. Allerdings wirkt das Ganze noch relativ langweilig und irgendwie blind. Eine echte gestreute Transparenz besteht nicht (siehe Bild 15).

### ... plus gestreuter Transparenz

Dazu musste ich tatsächlich den Transparenz-Kanal aktivieren und eine milchglasartige Streuung der Transparenz (in diesem Fall 42%) einstellen. Zudem senkte ich die Wirkung der Transparenz insgesamt auf 84% ab. Das heißt im Umkehrschluss: Das Objekt ist nun zu 16% opak. Und in genau diesen 16% kommt das Subsurface Scatte-



**Bild 16:** Eine gestreute milchglasartige Transparenz mit 84% Wirkung sorgt umgekehrt für 16% Opazität, in denen das Subsurface Scattering wirken kann.

ring im Leuchten-Kanal zum Tragen (siehe Bild 16).

Das so gerenderte Ergebnis mit gestreuter Transparenz und Subsurface Scattering wirkt nun deutlich definierter und greifbarer. Auch die beschädigten Bruchkanten des Acrylglases kriegen einen Leuchteffekt mit (siehe Bild 17).

### ... plus Schmutz

Das bisherige Ergebnis wirkt jedoch noch zu sauber, zu glatt. Es fehlt auf dem Acrylglas noch eine gewisse Verschmutzung, die zu dem Look der anderen Materialien passt. Nun könnte man auf die Idee kommen: „Okay, ich erstelle mir ein Material mit Alpha-Kanal, staple in dem Shader ein paar Noises und Co. und lege mir das dann als extra Textur-Tag auf mein Acrylglasobjekt.“ Und dann kommt die Enttäuschung.

In diesem Fall würden nämlich auf ein und demselben Objekt zwei Materialien liegen, die jeweils eine Transparenz enthalten: das Acrylglas mit aktivem Transparenz-Kanal und das Schmutzmaterial mit aktivem Alpha-Kanal. Und das geht schief. Denn Cinema 4D kann pro Oberflächennormale nur eine einzige Art Transparenz berechnen. Zumindest für Standard und physikalischen Renderer.

Der Ausweg aus dieser Situation ist: Man erzeugt eine Kopie seines Acrylglasobjekts, benennt es „Acrylglas Dreck“ und wendet dort das Schmutzmaterial mit Alpha-Kanal als einziges Material an. Nun hat diese Kopie allerdings dieselben



**Bild 18:** Verschmutzungen für weiteren Realismus: Eine Kopie des Acrylglases, nur mit Schmutztextur, sitzt durch ganz leichtes Displacement außerhalb des Originals.

Dimensionen wie das original Acrylglas, d.h. es würden Überschneidungsartefakte entstehen. Deswegen muss das Schmutz-Acrylglas einen Hauch über der Oberfläche des original Acrylglases sitzen. Dies bewerkstelligt man durch ein ganz leichtes Displacement, basierend auf einem reinen Weiß als Farbe-Shader und einer geringen Höhe von 0,15 Zentimetern. So sitzt das Schmutz-Acrylglas ganz leicht außerhalb der Begrenzungen des original Acrylglases. Die an den Kanten entstehenden leichten Outlines sind in Photoshop später leicht zu entfernen (siehe Bild 18).

Das so geschichtete Ergebnis sieht schon deutlich besser aus als nur mit Subsurface Scattering. Und nun erschließt sich auch, warum der Subsurface Scattering Shader mit einer Stärke von 4.000% angewendet wird: Der aktive Transparenz-Kanal und der via Kopie des Objekts erstellte Schmutz schwächen SSS natürlich in seiner Wirkung ab. Daher muss mit einem so hohen Wert gegengesteuert werden.

## Lighting

Das Lighting der Szene besteht im Grunde aus einem sehr simplen Dreipunkt-Licht-Setup. Dieses Licht soll die Beleuchtung einer nächtlichen Szene nachempfinden und daher nicht zu dominant wirken. Vielmehr muss Raum gelassen werden für das Leuchten der Leuchtstoffröhren und die Wirkung von Spiegelungen. Das Licht-Setup baut sich auf wie folgt (siehe Titelbild):

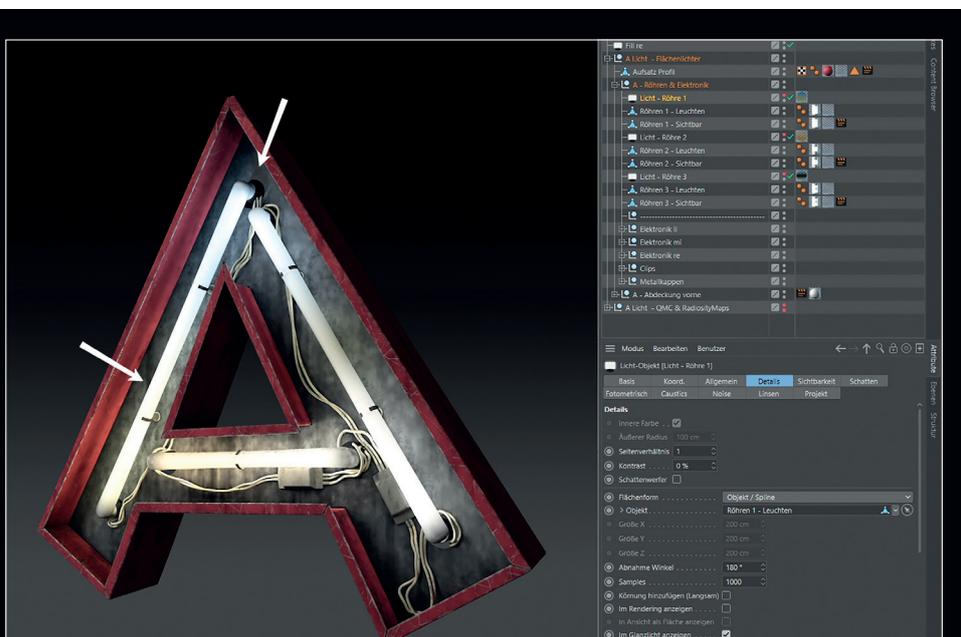
- ▷ Ein Hauptlicht, von links oben kommend, wirkt relativ stark und läuft nach rechts weich aus.
- ▷ Ein Fülllicht von rechts mildert die Schatten des Hauptlichts ab und füllt noch dunkle Bereiche mit etwas Licht.
- ▷ Von hinten mittig wirken eine Handvoll Silhouettenlichter. Diese trennen den Schriftzug besser vom dunklen Hintergrund.

## Flächenlichter vs. Polygonlichter

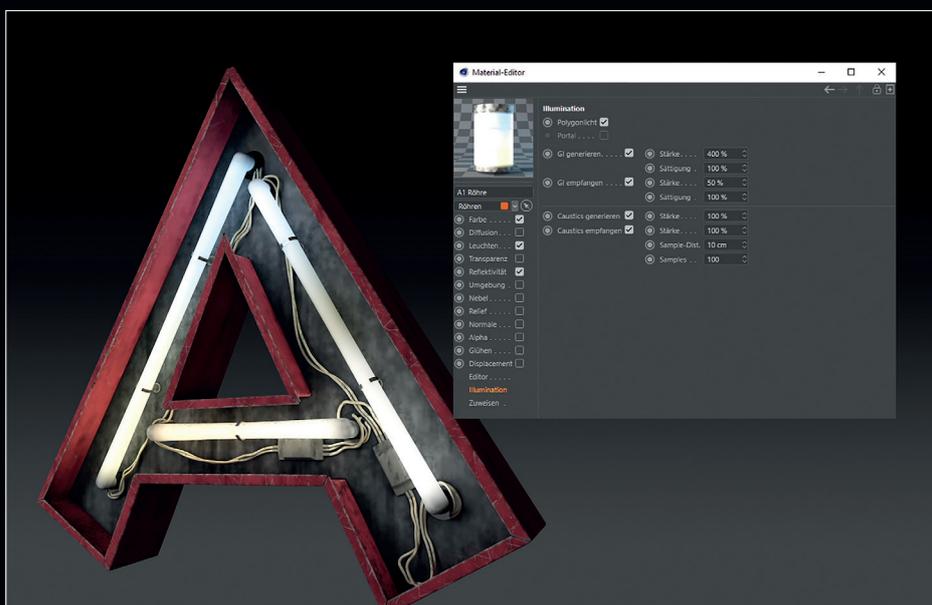
Wenn man so eine Leuchtreklame vor sich hat, stellt sich als zentrale Frage natürlich: „Wie realisiere ich die Lichtemission durch die Leuchtstoffröhren?“ In Cinema 4D existieren dafür zwei Methoden:

### Flächenlichter

Das Prinzip der **Flächenlichter**-Methode ist, dass man ein Flächenlicht hat, dem man als Quelle für „Flächenform“



**Bild 19:** Das Leuchten der Leuchtstoffröhren, realisiert durch Flächenlichter mit den Röhren als Flächenformquelle: passabler Gesamteindruck, jedoch zu konzentrischer Schattenwurf.



**Bild 20:** Das Leuchten der Röhren, realisiert durch ein selbst leuchtendes Material mit der Option „Polygonlicht“: deutlich homogenerer Lichtverlauf und diffusere Schatten dank QMC und Radiosity Maps



**Bild 21:** Das Farbkonzept des Schriftzugs: Das Rot der Gehäuse hält die Szene als wiederkehrendes Element zusammen, das Gelb des Acrylglases bildet seinen eigenen Duktus, bläuliche Akzente suggerieren räumliche Tiefe.

ein Polygonobjekt zuweist. Dies schaltet das Objekt selbst für das Rendern unsichtbar. Um dann ein entsprechendes Objekt – z.B. eine Leuchtstoffröhre – im Rendering sichtbar zu haben, muss also eine Kopie erstellt werden. Diese sollte sinnvollerweise keinen Schatten werfen und benötigt dazu ein entsprechendes Render-Tag.

Vorteil: Das Verfahren ist relativ schnell zu rendern. Nachteil: Der Lichtverlauf und der Schattenwurf sind nicht so homogen, wie man es erwarten würde, und Schatten laufen konzentrischer, als man es von einer ausgedehnten Objektlichtquelle erwarten würde (siehe Bild 19).

## Global Illumination und Polygonlichter

Kommen wir zur zweiten Methode: Global Illumination und **Polygonlichter**. Dieses Prinzip beruht darauf, dass es die Leuchtstoffröhren selbst sind, die mit einem leuchtenden Material tatsächlich Licht ausstrahlen. Dazu muss natürlich Global Illumination aktiv sein und im entsprechenden Material muss unter „Illumination“ die Option „Polygonlicht“ aktiviert werden (siehe Bild 20).

Was ist denn nun in diesem Spezialfall von Licht emittierenden Polygonobjekten besser – die Methode mit Flächenlichtern oder

die Methode mit Global Illumination? Die Flächenlichtermethode hat zwar in Sachen Geschwindigkeit die Nase vorn, hat aber den Nachteil der konzentrischen Schatten und der nicht ganz so homogenen Lichtausbreitung. Indirektes Licht könnten wir uns zwar noch durch wenige Handgriffe erzeugen, das wäre kein Problem, aber der Lichtverlauf und der Schattenwurf sind gegenüber der Lösung mit Global Illumination einfach nicht so hochwertig (siehe Bild 19). Das heißt, die Methode mit Global Illumination und Polygonlichtern ist hier das Mittel der Wahl. Das Ganze erzeugt natürlich deutlich höhere Renderzeiten. Da es sich bei diesem Projekt aber um ein Still-Rendering und nicht um eine Animation handelt, ist das zu verkraften.

## GI-Methoden

Als primäre Methode habe ich Quasi Monte Carlo (QMC) gewählt. Warum? Quasi Monte Carlo ist gegenüber Irradiance Cache deutlich genauer. Da wird nichts gecacht, nichts interpoliert, und man wird sozusagen mit der ungeschönten, langsamen GI-Wahrheit konfrontiert. In meinen Cinema-4D-Trainings nenne ich QMC gerne den „rigorosen Perfektionisten“, wohingegen Irradiance Cache eher der „intelligente Softie“ ist. So verhalten sich diese beiden Methoden im Vergleich zueinander.

QMC ist aber nicht nur die genaueste, sondern auch die langsamste Methode. Sie basiert auf stochastischem Sampling und

benötigt für einen praktikablen Einsatz die Beschleunigung durch eine sekundäre GI-Methode. In diesem Falle habe ich mich für Radiosity Maps entschieden. Diese liefern im Unterschied zu Licht-Maps zwar nur einen weiteren Bounce, das ist aber okay, da es sich bei dem Motiv um eine Nachtszene mit direktem Blick auf Lichtquellen handelt. Indirektes Licht spielt daher naturgemäß eine untergeordnete Rolle.

Radiosity Maps haben gegenüber Licht-Maps den Vorteil, dass zur Berechnung kein Prepass im Spiel ist. Das macht sich vor allem bei der Nutzung des interaktiven Renderbereichs bemerkbar: Hier verzögert der Aufbau des Prepasses die interaktive Darstellung und ist auf Dauer schlicht lästig. Aber auch was das finale Rendering der Szene angeht, sind Radiosity Maps gegenüber Licht-Maps im Vorteil: Letztere spielen ihren Geschwindigkeitsvorteil nämlich vor allem bei hohen Bounce-Zahlen aus (z.B. 16). Da aber ohnehin nur ein Bounce benötigt wird, sind Radiosity Maps hier die schnellere Alternative. Die Wahl der GI-Methoden fiel am Ende also auf Quasi Monte Carlo als primäre und Radiosity Maps als sekundäre Methode.

## Rendering

Global Illumination in Kombination mit Subsurface Scattering und gestreuter Transparenz ist natürlich ein Hattrick des langsamen Renderns. Vor allem in Print-Auflösung und vor allem mit dem physikalischen Renderer.



**Bild 23:** Das finale Ergebnis des Schriftzugs aus anderer Perspektive: Gradationskorrektur, Glow, Korn und Co. verleihen dem rohen Rendering erst den richtigen Kick und einen analogen Charakter.

Da es sich aber nur um ein Standbild handelt, lohnt sich die Geduld. Außerdem kann der Renderprozess via Cinema 4Ds Team Render ordentlich beschleunigt werden, sodass mehrere Maschinen an ein und demselben Bild rendern können. Und wer jetzt noch die Hände über dem Kopf zusammenschlagen sollte: Mein Studio ist mittlerweile auf Corona Renderer umgestiegen.

## Farbkonzept

Der gezielte Einsatz von Farben – z.B. durch Komplementärkontraste oder das gezielte Fortführen von Farben in unterschiedlichen Bildbereichen – hilft, eine Bildkomposition zu verstärken und die Atmosphäre des Renderings zu verdichten. Solche konzeptionellen Überlegungen sollten natürlich vor dem finalen Rendering geschehen. Sehen wir uns das implementierte Farbkonzept der Szene aber ruhig einmal anhand der fertigen Ergebnisse an.

Deutlich hervorstechend ist das überall vorhandene Prinzip der umlaufenden Aufsatzprofile in ihrem grauroten Farbton (siehe Titelbild). Dieser Farbton hält die Szene als wiederkehrendes Element zusammen. Das gelblich orangene Acrylglas sticht dagegen heraus. Dieses hat einen eigenen Duktus, quasi wie eine Schrift in der Schrift, und verleiht dem Ganzen im Grunde seinen eigenen Rhythmus.

Der Schriftzug wird farblich also zusammengehalten durch die grauroten Aufsatz-

profile, aufgebrochen und akzentuiert durch das Acrylglas und durch Komplementärkontraste. Denn wir haben auch die innen liegenden Blenden, auf denen die Elektronik und die Röhren montiert sind. Und diese tendieren ein wenig ins Bläuliche (siehe Bild 21 und 23). Das liegt auch daran, dass im Licht-Setup für die Szene Softboxes aufgestellt sind, wie man sie aus einem Fotostudio kennt. Diese strahlen in einem hellen, bläulichen Farbton und sorgen im Zusammenspiel mit den warmen Farbtönen für einen schönen Komplementärkontrast. In der Unterseite sieht man das noch deutlicher.

Neben dem Komplementärkontrast zu den Rot- und Orangetönen bewirkt die bläuliche Reflexion jedoch noch einen weiteren Effekt: Bläuliches wird immer als weiter entfernt wahrgenommen als Rötliches oder Gelbliches. Das liegt an der sogenannten Luftperspektive: Mit zunehmender Entfernung erscheinen Objekte weniger kontrastreich, bläulicher und entsättigter, während Objekte, die nicht so weit entfernt sind, eher warme Farben aufweisen. Der Blick aus dem Fenster beweist es. Das heißt: Alles, was irgendwie bläulich daherkommt, scheint weiter weg zu sein. Dadurch kann man den räumlichen Eindruck bzw. die subjektiv empfundene Größe der Szene verstärken.

## Dynamikumfang

Trotz Farbkonzept und der eingearbeiteten Kontraste wirkt das rohe Rendering etwas

flau – das ist aber durchaus beabsichtigt. Grundsätzlich rendere ich nämlich gerne einen leicht flauen Kontrast in meine Arbeiten, gebe diese allerdings in einem höheren Dynamikumfang, also einer höheren Farbtiefe, aus. In den Render-Voreinstellungen von Cinema 4D besteht nämlich die Möglichkeit, 8, 16 oder 32 Bit Farbtiefe pro Farbkanal (rot, grün, blau) zu wählen. So stehen bei der Wahl von 16 Bit gegenüber 8 Bit nicht nur 256 Stufen, sondern über 32.000 Stufen pro Farbkanal zur Verfügung. Es ist daher nachvollziehbar, dass ein 16-Bit-Bild sehr viel flexibler auf massive Farb- oder Gradationskorrekturen reagiert als in 8-Bit-Bild. Deswegen sind 16 Bit immer meine Wahl.

## Postproduktion

Kein Rendering ohne ordentliches Compositing. Farbkonzept der Gestaltung, smartes Shading, Lighting und Rendering sind schön und gut, entfalten aber erst in der Postproduktion ihre beabsichtigte Wirkung. Werfen wir daher einen Blick auf die in Photoshop erstellten Korrekturen des 16-Bit-Renderings (siehe Bild 23):

- ▷ generelle Gradationskorrektur,
- ▷ erhöhte Sättigung und größerer Kontrast der rötlichen Aufsatzprofile,
- ▷ Glow-Effekt für das Acrylglas,
- ▷ Variation des Acrylglases in puncto Transparenz, Transluzenz und Schmutz; extra gerenderte, variierte Layer,



**Bild 22: Das rohe Rendering des Schriftzugs aus einer weiteren freien Perspektive: Ein bewusst flauer Kontrastumfang jedoch mit 16 Bit Dynamikumfang bietet ordentlich Spielraum für eine ausgeprägte Gradationskorrektur.**

- ▷ Kanten-Lichter, die den Linienfluss nochmal betonen und das Ganze auch in kleinerer Größe besser lesbar machen,
- ▷ Glow für die Leuchtstoffröhren, teils farbig,
- ▷ Filmkorn und chromatische Aberration, erstellt in After Effects mit den Plug-ins Red Giant Magic Bullet und Renoiser. Die Qualität dieser Plug-ins kannte ich schon aus Projekten zuvor, somit lohnte sich dieser kleine Umweg.

Die gezielte Korrektur einzelner Bildbereiche erfolgte mittels Objekt-Passes, die im Multipass-Rendering als Graustufenbilder mit ausgegeben wurden und in Photoshop als Masken verwendet werden können.

In das Compositing habe ich ebenfalls ein paar fotografierte Bokeh-Flecken und Ob-

jektivverschmutzungen als Bildebene integriert, um das Ganze noch mehr in Richtung analoges Foto zu trimmen, weg von zu perfekter Computergrafik. Ebenso wie Korn und chromatische Aberration sind dies ja analoge Abbildungsfehler, die man sonst aus Fotos eliminieren möchte. Hier bin ich den umgekehrten Weg gegangen.

Der Unterschied zu den rohen Renderings ist am Ende immens (siehe Bild 22). Das Gesamtergebnis wirkt gerade durch das Zusammenspiel aus Farbkonzept, Multipass-Rendering und Postproduktion spannend und kontrastreich.

### Bleibt zu sagen ...

Die Erstellung des Titeldesigns für Jamiris neues Comic-Album war ein arbeitsreiches

Projekt mit Direktleitung zur Kunst und einer Menge Spaß dabei. Das Ergebnis kann sich sehen lassen und geht nun hoffentlich durch die Hände zahlreicher Leser. Erschienen ist „Gödel, Escher, Gott“ bei Edition 52 ([www.edition52.de](http://www.edition52.de)). > ei



Marc Potocnik ist Diplom-Designer (FH) und Inhaber des Animationsstudios renderbaron in Düsseldorf. renderbaron realisiert seit 2001 hochwertige 3D-Animationen für renommierte Kunden wie ZDF, Intel, Siemens u.a. und bietet als Maxon Authorized Training Center (ATC) Trainings zu C4D an. Marc Potocnik ist Maxon Master Trainer und Autor des Cinema-4D-Trainings „Shading, Lighting & Rendering“. Er teilt sein Wissen außerdem in Form von Fachvorträgen auf internationalen Branchen-Events wie der Siggraph, FMX, IBC etc. Außerdem ist Marc Beta-Tester für Cinema 4D. [renderbaron.de](http://renderbaron.de)

Anzeige



**CINEMA 4D**  
Release 21



**CERTIFIED PARTNER**



- 3D-Software
- Plugins & 3D-Objekte
- Schulungs-Center
- Hardware

**VISION 4D**  
Alte Landstr. 12-14  
85521 Ottobrunn  
Tel.: 089 - 69 70 86 08  
[www.vision4d.de](http://www.vision4d.de)

• **Cinema 4D Release 21 bei uns mit erweitertem Lieferumfang!**

• **Unser beliebtes DocTabs Plugin für alle unsere Cinema 4D Kunden kostenlos.**

← Bitte rufen Sie uns an oder bestellen Sie online.

