

Tutorial: Textur Ambient Occlusion mit Blender "backen"

Inhaltsverzeichnis

- [1 Aufgabenstellung](#)
- [2 Durchführung](#)
 - [2.1 AO-Textur selbst malen?](#)
 - [2.2 AO-Textur in Blender berechnen lassen](#)
 - [2.3 Separate Texturkoordinaten: Wie sieht dann die Textur aus?](#)
 - [2.4 Nun wird gemappt](#)
 - [2.5 AO generieren](#)
 - [2.6 Man sieht nix](#)

1 Aufgabenstellung

Zunächst einmal das untexturierte 3D-Objekt:



Solche automatischen Mittelpuffer-Kupplungen wie die berühmte "Schaku", die Scharfenberg-Kupplung, zeichnen sich dadurch aus, dass sie zwar ein sehr detailliertes Mesh besitzen (wegen der vielen Hebelchen, Käbelchen und Schraubchen), das meiste davon aber mit der Zeit mehr oder weniger braungrau geworden ist. Wenn man es also "nackt" texturiert, dann sieht sie ungefähr so aus:



Das sieht natürlich ziemlich bescheiden aus. Sofern die Sonne in LOTUS scheint, sieht es ein bisschen schicker aus:



Dennoch - erstens scheint die Sonne ja nicht immer und zweitens gibt's selbst dann noch Stellen, die unnatürlich hell sind/bleiben (siehe rote Pfeile).

Viel schöner wäre es, wenn mittels der Technik der Ambient Occlusion Schattierungen hinzugefügt werden, die dadurch zustande kommen, dass auch das Umgebungslicht (das "ambiente" Licht) nicht überall gut hinkommt:



Wohlgermerkt: Auf diesem Bild scheint die Sonne nicht!

2 Durchführung

Die AO-Informationen werden in diesem Fall in einer weiteren Textur gespeichert, deshalb heißt es auch "Textur-AO". Die AO-Textur selbst ist dabei einfach eine Graustufen-Bitmap, bei der "Weiß = voll beleuchtet" und "Schwarz = komplett dunkel" gilt.

2.1 AO-Textur selbst malen?

Im einfachsten Fall bastelt man diese Schwarz-Weiß-Textur selbst, in dem man mit dem virtuellen Pinsel einfach helle und dunkle Töne auf eine neue Textur dort aufbringt, wo in der Original-Textur mehr oder weniger ambientes Licht hinfallen soll.

Ein typischer Anwendungsfall: Eine nicht besonders lichtdurchflutete Wagenhalle.

Fertig konstruiert lädt man sie in LOTUS bei bedecktem Wetter und wird feststellen: Alle Flächen, ob innen oder außen, sind hell beleuchtet, weil wetterbedingt nur ambientes Licht vorhanden ist, und deshalb auch kein Schattenwurf zu sehen ist.

"OK", denkt man sich, und macht einfach die Innenwände in der Textur dunkler. Nun sieht es bei bedecktem Wetter wieder gut aus: Draußen hell, innen dunkel.

Nun wechselt man aber in die Nacht: Obwohl das Fernlicht nun direkt auf eine Innenwand fällt, ist diese trotzdem nur sehr grau, obwohl sie eigentlich weiß gestrichen ist. Oder aber am Tage bei Sonnenschein: Die Sonne scheint durch die Deckenfenster auf jene Innenwand, die aber trotzdem noch ziemlich dunkel ist. Dafür sind die restlichen Flächen, die im Schatten liegen, noch dunkler.

Die Lösung ist hier natürlich eine AO-Textur. Allerdings hat man jetzt keine Lust, sich aufwändig mit Texturkoordinaten oder Problemen beim Generieren von AO-Texturen herumzuschlagen, deshalb malt man sich einfach eine ganz simple AO-Textur: Dort, wo sich auf der Original-Textur Außenbereiche einer Wagenhalle befinden, belässt man die selbstgemalte AO-Textur weiß, auf den Bereichen, wo sich die Innenbereiche der Wagenhalle befinden, malt man sie dunkelgrau. Innenbereiche, die vielleicht dichter an den Fenstern sind, versieht man mit einem netten Farbverlauf und/oder hellt sie auf. Das geht recht schnell und kann auch schnell stimmig aussehen.

Im Resultat wird die Wagenhalle nun drinnen grundsätzlich dunkler sein als draußen - fällt aber das Licht der Sonne oder einer künstlichen Lichtquelle auf eine Innenfläche, wird diese dennoch in ihrer Original-Helligkeit erstrahlen.

2.2 AO-Textur in Blender berechnen lassen

Mit diesem Konzept kommen wird aber bei der Kupplung nicht sehr weit... daher werden wir hier einen Schritt weiter gehen und für eine realistische AO-Textur Blender bemühen.

Nun gibt es aber zwei Möglichkeiten: Entweder, die Anordnung, also das Mapping der AO-Textur, entspricht genau dem Mapping der Haupttextur, oder es wird ein komplett anderes Mapping verwendet. Da es natürlich einen erheblichen Mehraufwand gibt, separate Texturkoordinaten anzulegen, stellt sich natürlich die Frage, worin der Vorteil besteht. Dieser ergibt sich aus den recht strengen Regeln für das AO-Textur-Mapping, wenn man die AO-Textur generieren lassen will:

- Kein Teil der Textur darf mehrfach verwendet werden! Ansonsten stellt sich für Blender sofort die Frage: Wie soll er denn nun beleuchtet werden? So wie bei Anwendungsfall 1, 2, ... ?
- Jeder Teil des Objektes muss sinnvoll gemappt werden
- Keinerlei 1-Pixel-Mapping, immer einigermaßen senkrecht Mapping und einigermaßen gleichbleibende Auflösung - oder zumindest sollte jeder Teil mit einer Mindestauflösung gemappt werden.

Hieraus resultiert natürlich: Entweder muss die Haupttextur auch diesen Regeln folgen oder man braucht separate AO-Texturkoordinaten.

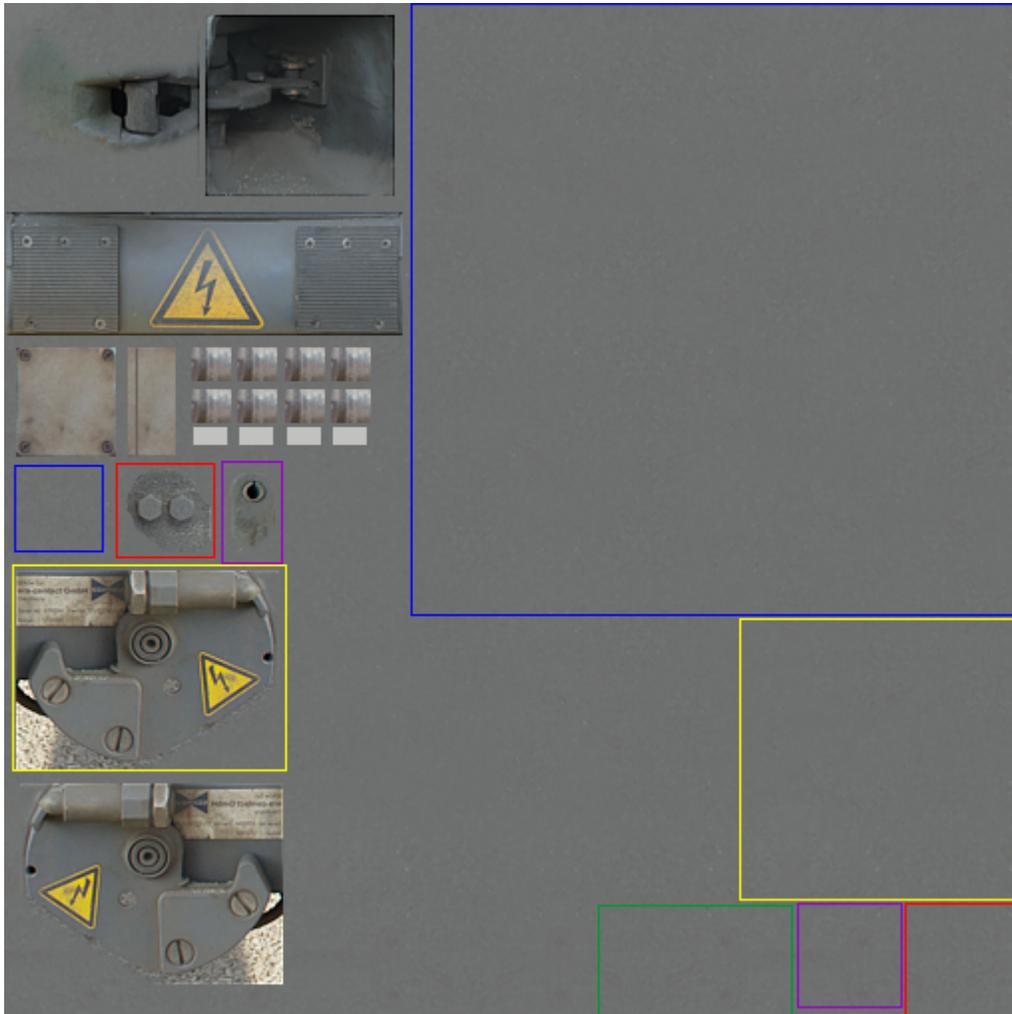
Da wir aber nur sehr wenige "wirklich" texturierte Stellen haben, wollen wir mit der Haupttextur den Regeln folgen.

2.3 Separate Texturkoordinaten: Wie sieht dann die Textur aus?

Zum Glück kann man mit Blender ein Objekt sehr einfach nach obigen Regeln automatisch mappen lassen, dafür gibt es einen eigenen Algorithmus. Es ist also jetzt keine manuelle "Fizzelarbeit" für all die vielen verwinkelten Polygone der Kupplung nötig.

Dieses automatische Mapping erzeugt am Ende immer eine gepackte quadratische Anordnung. Wichtige Info für gleich... 😊

So sieht die Grundtextur aus:



Einerseits sieht man ganz gewohnt die einzelnen Texturschnipsel für verschiedene Bauteile, die nicht einfach nur grau sein sollen. Zusätzlich gibt es aber eine große, graue Fläche. Dort kommen alle Polygone hin, die einfach nur grau sein sollen. Dafür wird deshalb so viel Platz eingeräumt, weil ja jedes Bauteil individuell gemappt wird! Mit den bunten Quadraten werden verschiedene Mapping-Bereiche reserviert, weil die Textur ja auf mehrere Unterobjekte verteilt werden soll. Interessant für dieses Tutorial ist aber vor allem das blaue Quadrat, weil es der Bereich ist, der für die nicht individuell texturierten Bereiche des Haupt-Objektes reserviert ist. Es darf ja nichts doppelt verwendet werden! Das ist auch der Grund dafür, dass die Seiten der E-Kupplung ganz brav zwei Mal vorhanden sind, und auch die vier Stecker, die zwar alle gleich aussehen, doch aber tatsächlich vier Mal vorhanden sind und zwar auch wirklich einmal von der einen und einmal von der anderen Seite.

2.4 Nun wird gemappt

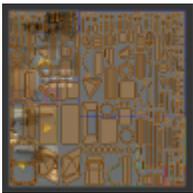
Die mit individuellen Texturen gemappten Bauteile werden ganz "normal" gemappt, aber es werden dabei

dennoch die Regeln eingehalten:

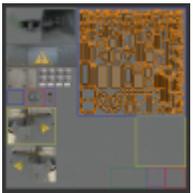


Nun werden alle anderen Polygone markiert (die Umkehr der Markierung geht übrigens automatisch mit [Strg]+[I]) und sollen gemappt werden. Hier soll Blender nun entsprechend der genannten Regeln automatisch mappen. Hierzu einfach wie üblich [U] drücken und dann aber im Menü "Smart UV Project" auswählen. Es folgen vier Einstellmöglichkeiten, die schon gut gewählt sind, bei denen nur "Island Margin" größer als 0 eingestellt werden sollte, z.B. auf 0,01. Das ist der Abstand, den die Einzel-Mappings voneinander haben sollen, und ein paar Pixel Platz können nicht schaden, damit es nachher nicht zu unsauberen Rändern kommt.

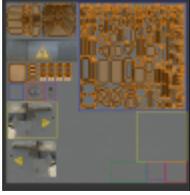
Das automatisch erzeugte Mapping sieht nun so aus:



Das muss jetzt nur noch in den reservierten Bereich skaliert und verschoben werden:

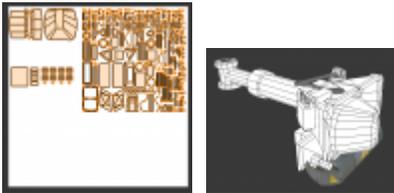


Das Mapping komplett:



2.5 AO generieren

Um nun die AO zu generieren, legen wir zunächst eine weiße Textur der gewünschten Auflösung an (diese darf von der Original-Textur abweichen!) und laden diese in Blender auf die soeben gemappten Polygone - und dementsprechend ist die Kupplung nun erstmal weiß:



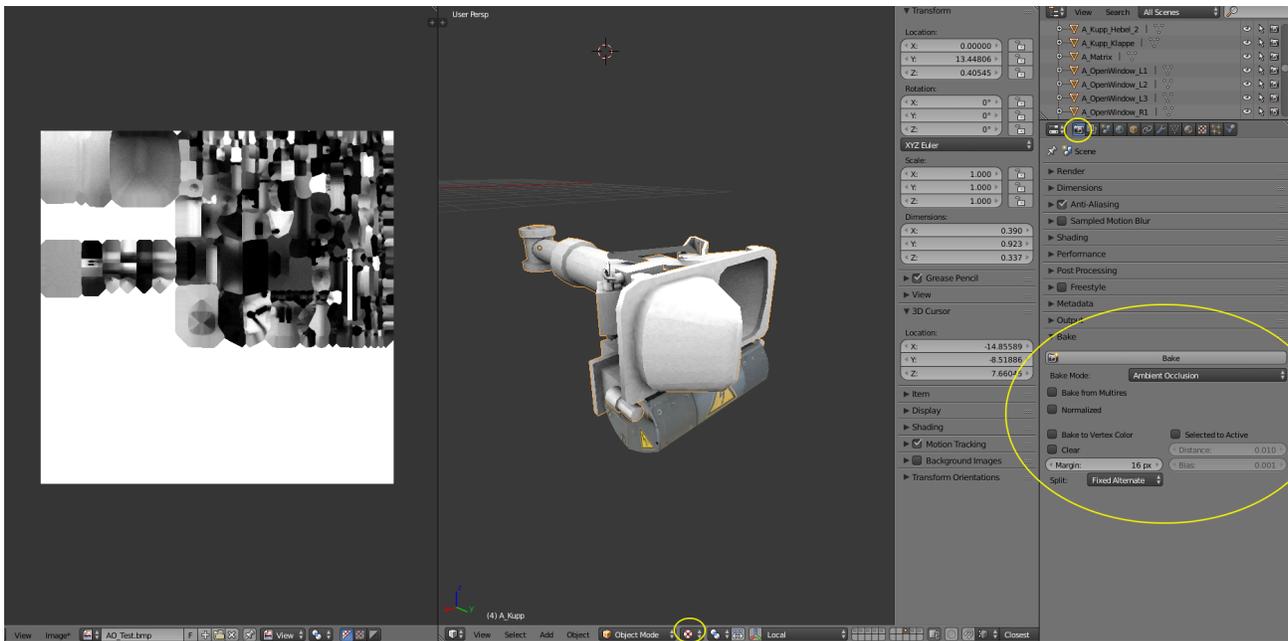
So - und damit ist nun alles vorbereitet!

Wir wechseln in den Objekt-Modus und klicken im "Properties"-Bereich auf die [Weltkugel](#). Dort gehen wir hinunter zum Abschnitt "Gather", wo die Ambient Occlusion *konfiguriert* wird. Um die Einstellungsmöglichkeiten zu testen, können wir als "Viewport Shading" "Rendered" einstellen. Im Screenshot sieht man die von mir verwendeten Werte.



Und jetzt wird's ernst: Zunächst das "Viewport Shading" auf "Texture" umstellen. Damit wird die Kupplung wieder rein weiß dargestellt. Dann geht's im Properties-Bereich auf den Fotoapparat in den Abschnitt "Bake". Dort können wieder ein paar Werte eingestellt werden (siehe wieder den Screenshot) und dann klickt man auf "Bake". Dann passiert erstmal eine ganze Weile nichts.....

Aber nach nicht allzu langer Zeit ist er dann fertig und das Fenster sieht so aus:



Im Texturfenster sieht man nun die fertig erzeugte AO-Textur, deren Wirkung man im 3D-Fenster direkt bestaunen kann. Schließlich sind nur noch drei Schritte nötig:

- Im Textur-Fenster wird die bearbeitete AO-Textur im Menü "Image" mit "Save Image" gespeichert
- Dem Objekt wieder vor dem Export als x3d-Datei die originale Textur zuweisen
- Im ContentTool das Material entsprechend konfigurieren:
 - Materialtyp "Complex"
 - "Intensität der AO-Textur" für den Anfang auf 1 stellen, danach nach Geschmack justieren
 - In den "[Optionen](#) und ganzzahligen Werten" die Haken für "AO-Textur aktivieren" und "AO nutzt Originalmapping" setzen
 - Unten bei den Texturen die AO-Textur auswählen

Fertig! 😊

2.6 Man sieht nix

"Alles schön und gut, aber man sieht ja im ContentTool (fast) nix!"

Wichtig ist zu wissen, dass die AO nur beim ambienten Licht wirksam ist! Bevor man darüber verzweifelt, sollte man erstmal links im Abschnitt "Testumgebung" die oberen beiden Licht-Regler für's direkte und indirekte Licht auf 0 ziehen und dafür den dritten Regler fürs ambiente Licht entsprechend hoch ziehen. Erst dann sieht man die völlige Auswirkung der AO-Texturen.