

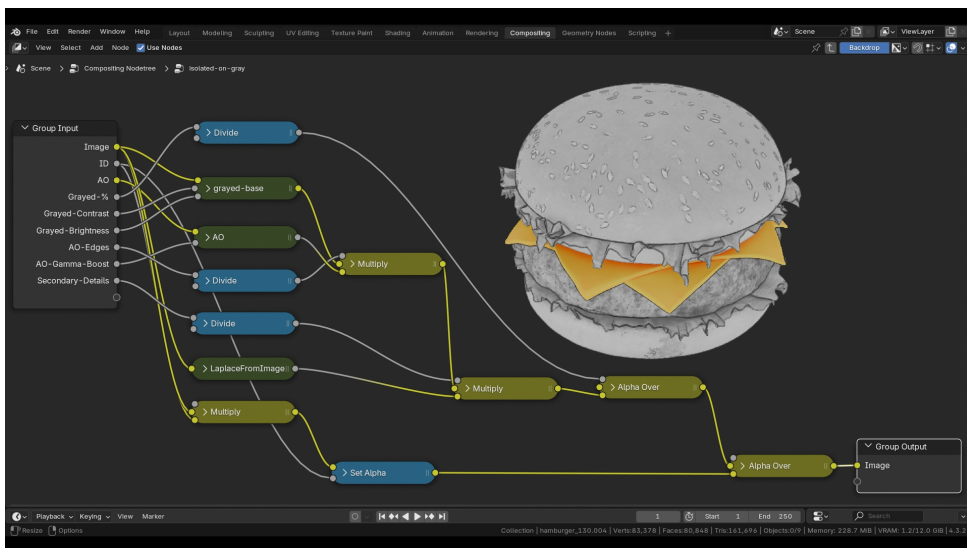
部分的にカラー、部分的にグレースケールのレンダリング | Blender 用の無料コンポジット イングノードグループ | チュートリアル 2/2

www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)

ビデオバージョン : <https://youtu.be/1Ys9O2jSp74>

この PDF は無料です。気軽にシェアしてください！

私の YouTube チャンネルを登録して、新しい動画の更新を受け取るために
通知をオンにしてください！



皆さん、こんにちは！これは、Blender 4.3 用のコンポジットングノードグループを分析し、使用方法を解説する2つのチュートリアルのうちの2つ目です。このノードグループを使用すると、特定のオブジェクトやマテリアルのみを元の色でレンダリングし、それ以外のすべてをグレースケールでマスキングしつつ、エッジやキャビティを強調することができます（オプション）。

最初のチュートリアルでは、このノードグループの使い方を説明しました。このノードグループは、説明欄のリンクから無料でダウンロードできます。この2つ目のチュートリアルでは、ノードグループの内部を見て、各エフェクトがどのように実装されているかを確認していきます。

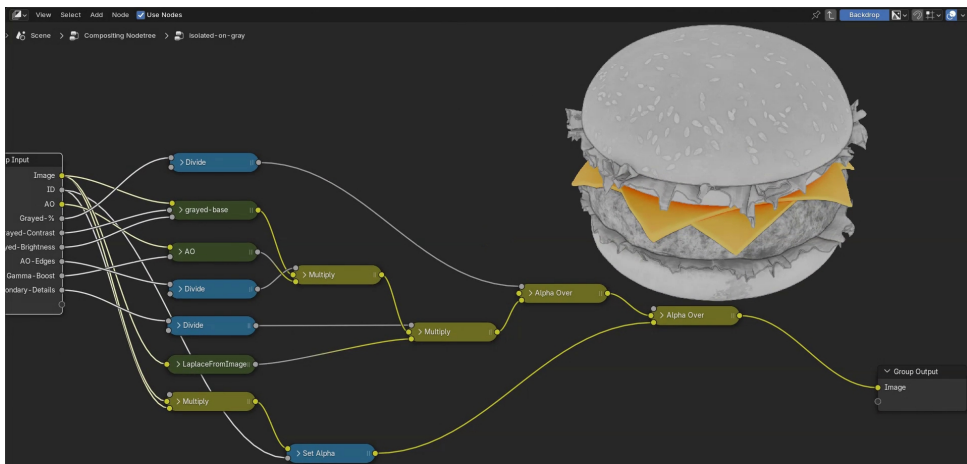
部分的にカラー、部分的にグレースケールのレンダリング | Blender 用の無料コンポジット
ィングノードグループ | チュートリアル 2/2
www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)
ビデオバージョン : <https://youtu.be/1Ys9O2jSp74>

他の多くのビデオチュートリアルと同様に、このチュートリアルの PDF 版も説明欄のリンクから無料で入手できます。ぜひ、この機会にチャンネル登録と通知をオンにして、今後のチュートリアルをお見逃しなく！

TAB キーでノードグループを開くと、エフェクトがノードの左側のコラムに実装されていることがわかります。

これらのノードの一部は、実際にはノードグループになっています。

右側のノードは、画像にエフェクトを順番に追加するために使用されています。



前回のチュートリアルで説明したように、このノードグループは基本的に3つの処理を行います。

まず、特定のオブジェクトまたは **Material** を選択し、それをカラーのまま保持しながら、その他の要素をグレースケールに変換します。

2つ目のエフェクトとして、レンダーから **Ambient Occlusion** の情報を取得し、キャビティやエッジを強調します。

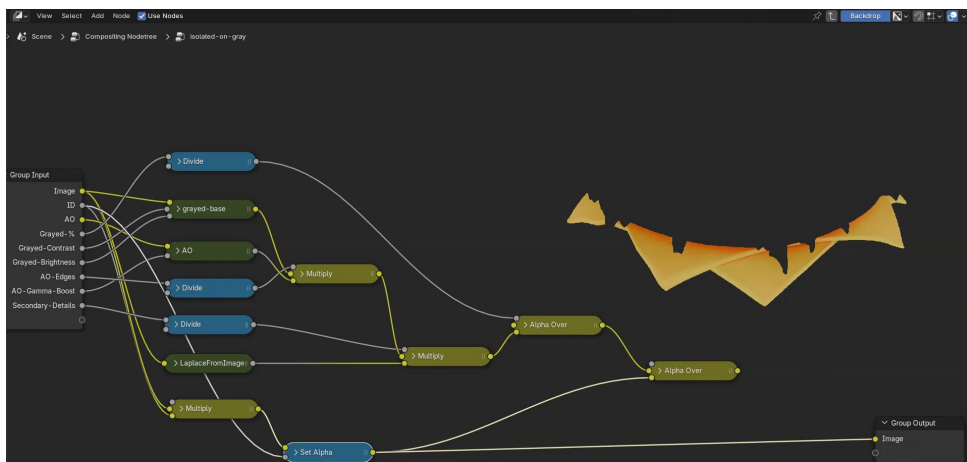
部分的にカラー、部分的にグレースケールのレンダリング | Blender 用の無料コンポジット ニングノードグループ | チュートリアル 2/2

www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)

ビデオバージョン : <https://youtu.be/1Ys9O2jSp74>

3つ目のエフェクトとして、フィルターを使用して、ジオメトリではなく Textures に含まれる細かいディテールをさらに強調します。

入力された ID と一致する要素を画像の他の部分から分離し、それらをカラーのまま保持するために、元の画像に ID 入力からのマスクを乗算します。次に、同じ ID マスクを Alpha 透過として使用することで、この段階では透明な背景上にカラーでレンダリングされた一部の要素のみが残ります。



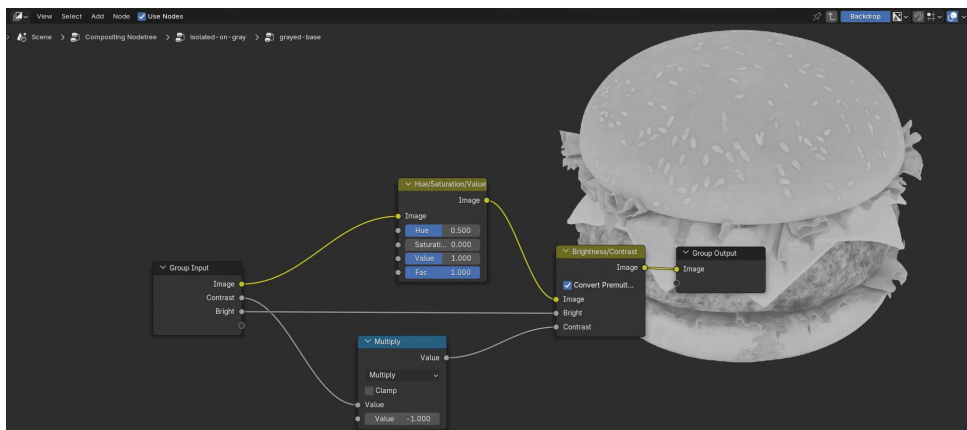
元のレンダリング全体をグレースケールで取得するために、**Image** の情報と **Grayed Contrast**、**Grayed Brightness** のデータを **Grayed Base** サブグループに送信します。

このサブグループ内では、**Hue Saturation Value** ノードを使用して画像の色度を 0 に設定し、画像をグレースケールにします。その後、**Brightness Contrast** ノードを使用して、明るさとコントラストを調整します。

部分的にカラー、部分的にグレースケールのレンダリング | Blender 用の無料コンポジット イングノードグループ | チュートリアル 2/2

www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)

ビデオバージョン : <https://youtu.be/1Ys9O2jSp74>



こうして得られた画像には、**Ambient Occlusion** や **Laplace filter** を使用して取得したディテールレイヤーが順番に適用されます。これらについては後ほど詳しく見ていきます。

追加のエフェクトを適用した後の最終的なグレースケール画像は、**Grayed** 入力によって強度がパーセンテージで制御されます。この入力は 0 から 100 の数値で表され、100 で割ることで **Alpha Over** ノードの **Factor** として使用できるようになります。

この **Alpha Over** ノードでは、最初の **Image** 入力は完全な透明画像（黒色で Alpha が 0）となっています。**Grayed** の値が 0 のときは、この完全な透明のみが使用されます。

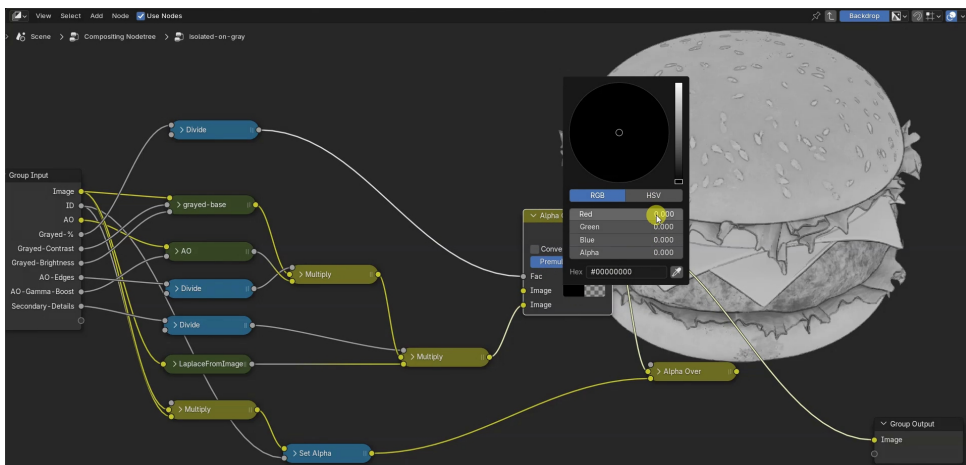
値が 100% のときは、100 で割った結果 **Factor** が 1 となるため、グレースケール画像のみが使用されます。

中間の値では、これら 2 つの間のブレンドされた画像が出力されます。

部分的にカラー、部分的にグレースケールのレンダリング | Blender 用の無料コンポジット リングノードグループ | チュートリアル 2/2

www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)

ビデオバージョン : <https://youtu.be/1Ys9O2jSp74>



この **Alpha Over** ノードの出力画像は、グレースケールのベースとして使用され、そこに ID が一致するカラー画像を適用します。これには、2 つ目の **Alpha Over** ノードを使用します。

これで処理の流れが明確になったので、次に **Ambient Occlusion** と **Laplace From Image** サブグループの処理を確認しましょう。これらはエッジやキャビティの強調に使用されます。

詳しく分析する前に説明しておく、これらのサブグループは、グレースケールのレンダリングと順番に乘算される白黒のマスクを生成します。画像を別のグレースケール画像と乗算すると、乗算された画像が暗い部分ほど元の画像が暗くなります。

乗算の Factor 入力の前にある **Divide** ノードは、入力値を 0 から 1 の範囲に戻すために使用されています。これは、操作を簡単にするためにパーセンテージで値を設定しているためです。それでは、残りの 2 つのサブグループを詳しく見ていきましょう。

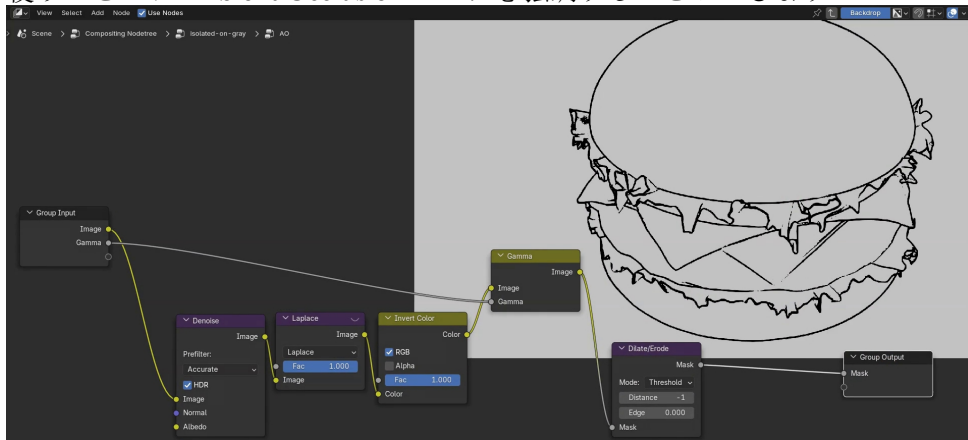
Ambient Occlusion サブグループの処理手順と出力を確認するために、その出力を直接メイングループの出力に接続します。

部分的にカラー、部分的にグレースケールのレンダリング | Blender 用の無料コンポジット イングノードグループ | チュートリアル 2/2

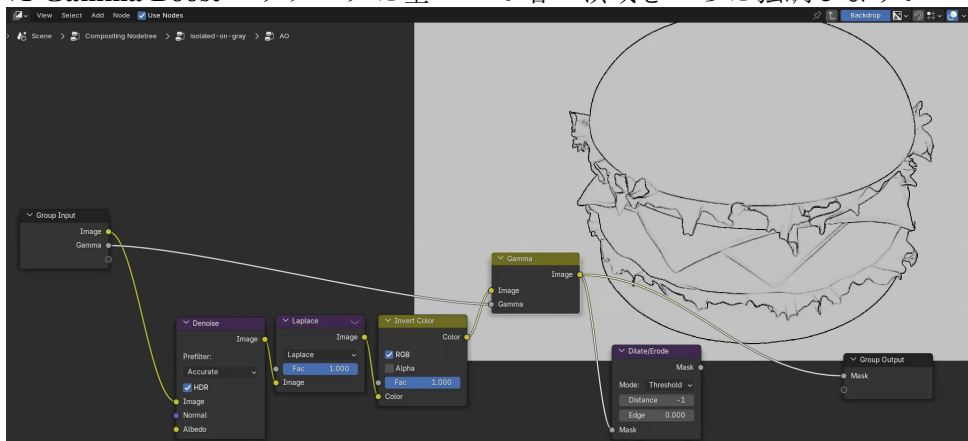
www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)

ビデオバージョン : <https://youtu.be/1Ys9O2jSp74>

Ambient Occlusion グループは、**Render Layers** からの **Ambient Occlusion** データと、**Gamma Boost** という数値入力を受け取ります。この **Gamma Boost** を使うことで、**Ambient Occlusion** マスクを強調することができます。



Ambient Occlusion データは、まず **Denoise** ノードに送られ、ノイズを低減します。その後、**Filter** ノードの **Laplace** モードを通過し、エッジを強調します。 **Laplace** の結果では、エッジが明るく、背景が暗くなっています。しかし、乗算を使用してエッジ部分をさらに暗くするためには、これを逆にする必要があります。そのため、**Laplace** の出力を **Invert Color** ノードに送り、さらに **Gamma** ノードを使って、ユーザーが設定した **Gamma Boost** パラメータに基づいて暗い領域をさらに強調します。

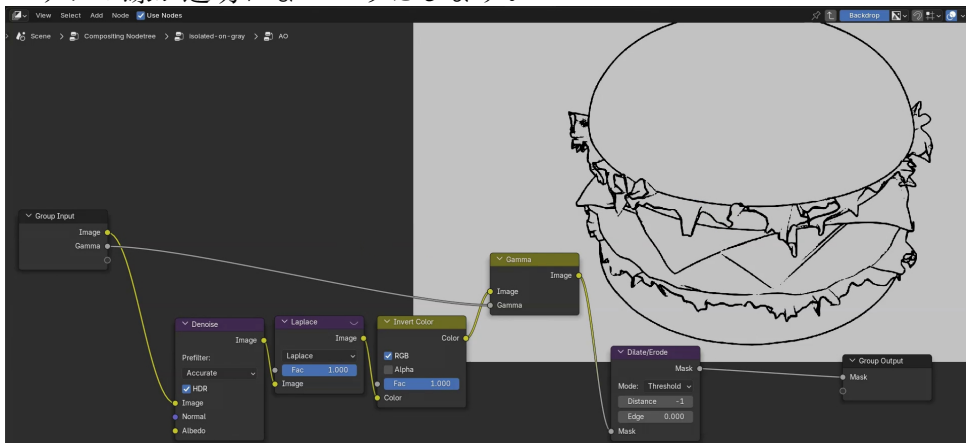


部分的にカラー、部分的にグレースケールのレンダリング | Blender 用の無料コンポジット イングノードグループ | チュートリアル 2/2

www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)

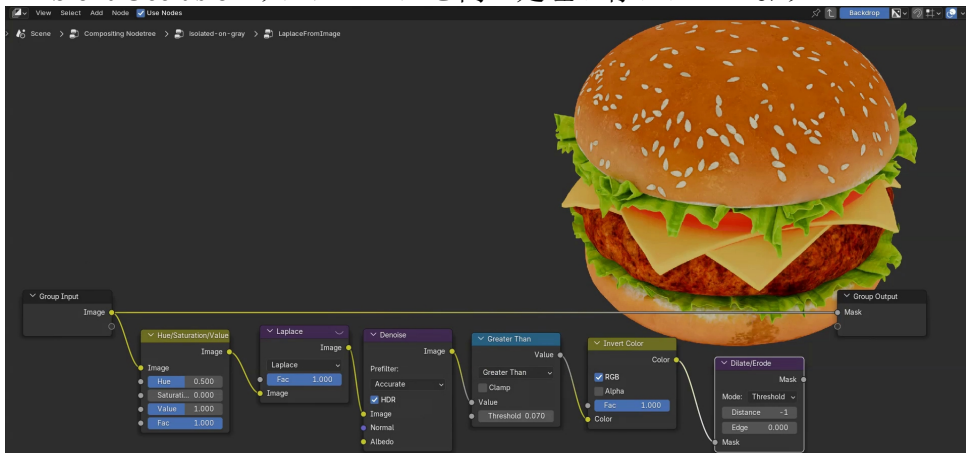
ビデオバージョン : <https://youtu.be/1Ys9O2jSp74>

次に、**Dilate Erode** フィルターを適用し、暗い領域をわずかに拡張して、エッジの線が途切れないようにします。



ここで、**Ambient Occlusion** サブグループから出て、**Laplace From Image** サブグループの出力を直接メイングループの出力に接続し、その処理手順と最終結果を確認します。

Laplace From Image サブグループの内部構造は、**Ambient Occlusion** サブグループと非常によく似ています。ここでも、**Filter Laplace**、**Denoise**、**Invert Color**、**Dilate Erode** ノードが使用されており、**Ambient Occlusion** サブグループと同じ処理が行われています。

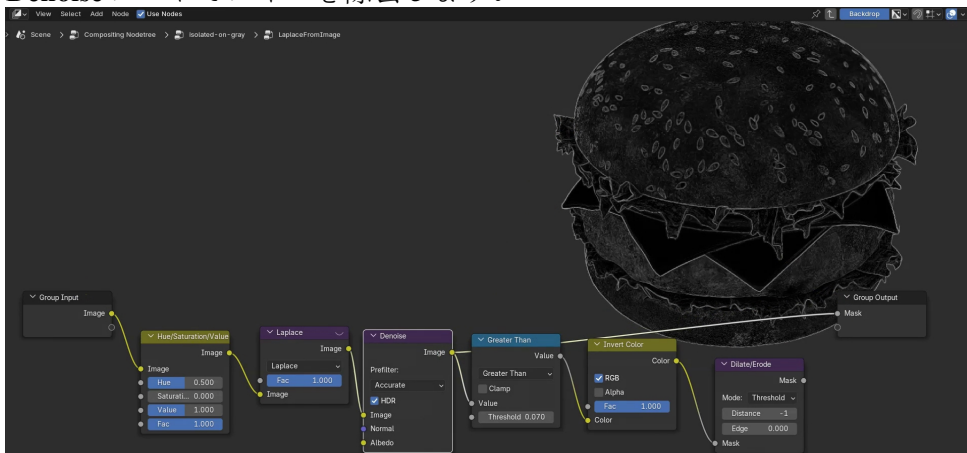


部分的にカラー、部分的にグレースケールのレンダリング | Blender 用の無料コンポジット イングノードグループ | チュートリアル 2/2

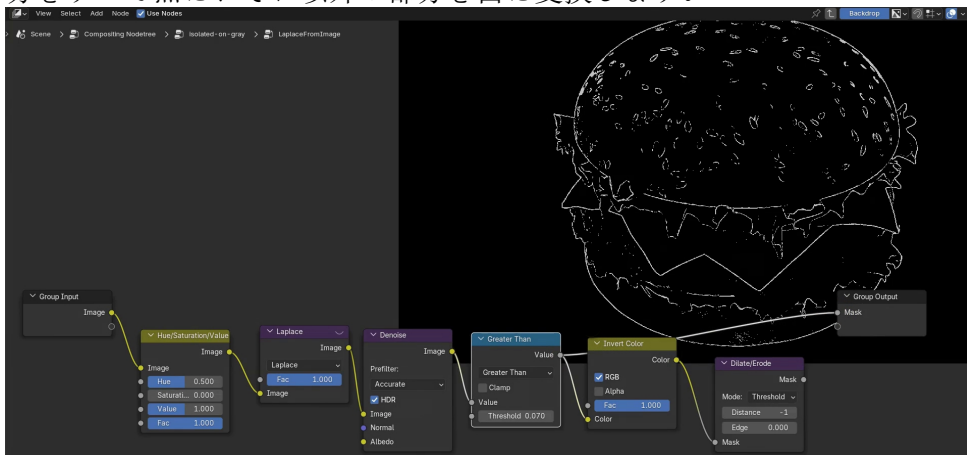
www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)

ビデオバージョン : <https://youtu.be/1Ys9O2jSp74>

ただし、この場合、最初のステップとして **Hue Saturation Value** ノードを使用し、**Saturation** を 0 に設定して入力画像をグレースケールに変換します。このステップは、**Ambient Occlusion** には不要でした。なぜなら、**Ambient Occlusion** の情報は元々グレースケールだからです。グレースケールに変換された画像は、**Laplace filter** で処理され、その後 **Denoise** ノードでノイズを除去します。

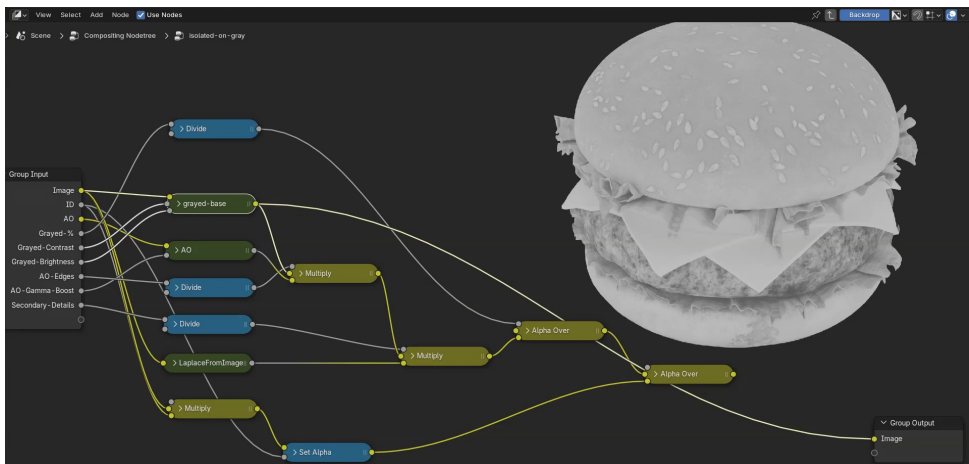
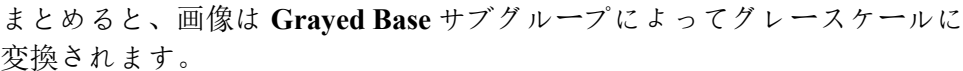


このサブグループの2つ目の特徴的な要素は、**Greater Than** ノードです。これは基本的にしきい値のように機能し、非常に低い値以下のグレー部分をすべて黒に、それ以外の部分を白に変換します。



www.francescomilane.com --- 2025 --- [GUMROAD \(PDF\)](#)

マスクは黒い背景に白いエッジを持っているため、**Invert Color** ノードを適用し、その後 **Dilate Erode** ノードを使用して、エッジが途切れないように補正します。この処理は、**Ambient Occlusion** サブグループと同じです。

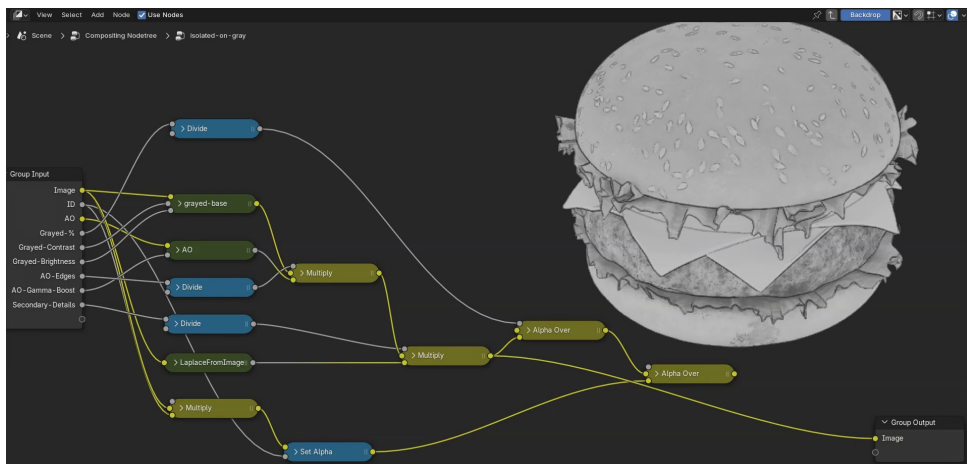
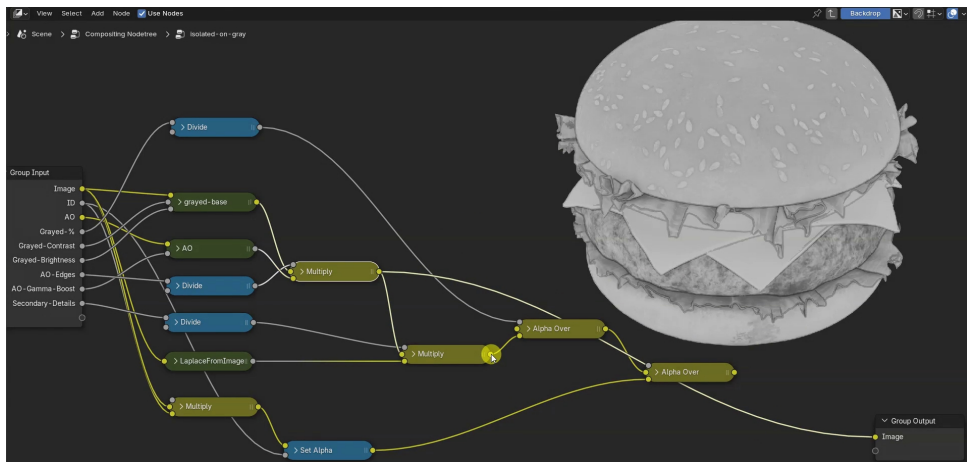


部分的にカラー、部分的にグレースケールのレンダリング | Blender 用の無料コンポジット イングノードグループ | チュートリアル 2/2

www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)

ビデオバージョン : <https://youtu.be/1Ys9O2jSp74>

その後、得られた画像に対して、**Ambient Occlusion** サブグループと
Laplace サブグループが生成したマスクを順番に乗算します。

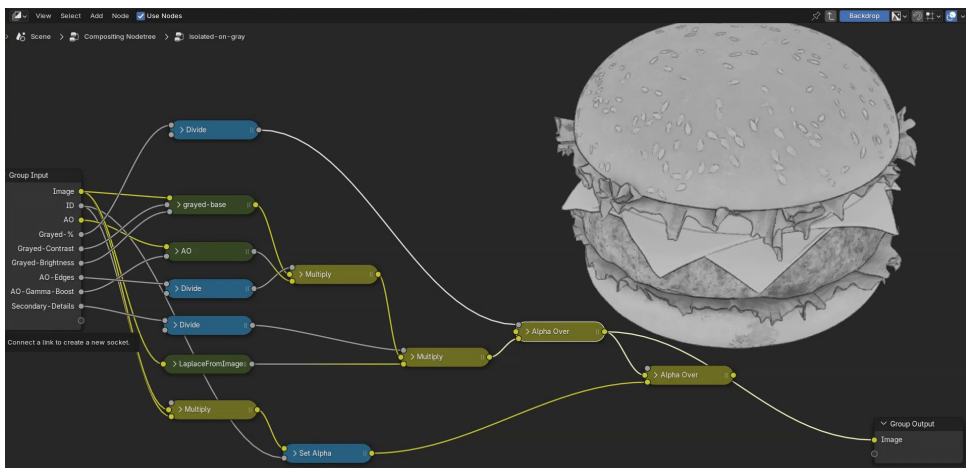


最終的に得られた画像は、**Alpha Over** ノード内で **Grayed** 入力によって透
明度が調整されます。

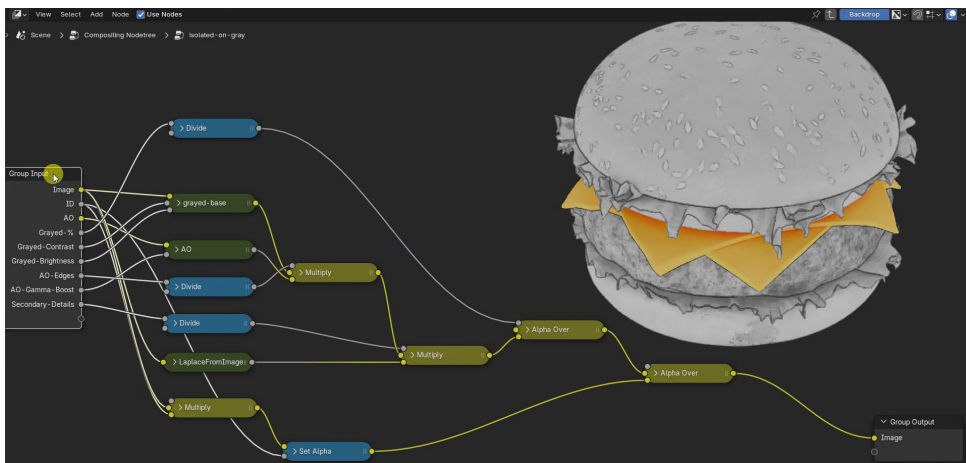
部分的にカラー、部分的にグレースケールのレンダリング | Blender 用の無料コンポジット イングノードグループ | チュートリアル 2/2

www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)

ビデオバージョン : <https://youtu.be/1Ys9O2jSp74>



正しい数値 ID を持つオブジェクトまたはマテリアルのカラー画像が、この結果の上に重ねられます。



これで今回のチュートリアルは終了です！お役に立てたなら嬉しいです。
また次回お会いしましょう！