Silicon Studio
Ideas × Art × Technology

CEDEC24

リアルタイム機械学習レンダリングのアプローチ：ポリゴン描画による形状安定性とニューラルネット画像生成による高品質性を両立する手法

シリコンスタジオ
研究開発室
ヘルツベルユ・フレドリック

２０２４年８月２３日

概要

Silicon Studio
Ideas × Art × Technology

- 機械学習を用いてリアルタイムに描画するための一つのアプローチおよびその実装を紹介
- 三角形メッシュのレンダリングと機械学習画像生成を統合
  - それぞれの特長である形状安定性と画像品質の両方を実現
- 様々な神経回路網(ニューラルネットワーク)を採用可能
- 結果の検証よりも理論と実装の詳細な議論に重点

本発表では、機械学習を用いてリアルタイムに描画するための一つのアプローチおよびその実装を紹介します。
三角形メッシュのレンダリングと機械学習画像生成を統合し、それぞれの特長である形状安定性と画像品質の両方を実現できるアプローチです。
このアプローチは、特定の神経回路網の構造に依存しないため、GAN、拡散モデルなど、様々な画像生成用の神経回路網を採用することができます。
本発表では、結果の検証よりも理論と実装の詳細な議論に重点を置いています。

用語

Silicon Studio
Ideas × Art × Technology

【ラスタライズ】
あるカメラから見た三角形メッシュをドットグリッドに換算すること

【種(Seed)】
画像生成神経回路網の入力。一般的に512個ほどの32ビット値

【文法】
種を神経回路網に入れる時の神経重みに与えられた意味

【可能性空間】
種の文法の表現力でありえることの全部

本発表で用いている用語の定義です。

ラスタライズとは、あるカメラから見た三角形メッシュをドットグリッドに換算することです。

種、シードとは、画像生成神経回路網に最初に与える入力です。一般的に512個ほどの32ビット値が用いられます。

文法とは、種を神経回路網に入れる時の神経重みに与えられた意味のことです。

可能性空間とは、種の文法の表現力でありえることの全ての集合を表します。

画像生成神経回路網のメリット

Silicon Studio
Ideas × Art × Technology

- 神経回路網による画像生成のメリット
  - ラスタライズ描画との比較
  - この研究では、これらを維持したい
- 画像生成の神経回路網の利点
  - 弾力的な表面の変形
  - 弾力的な表面の下の構造からの影響
  - sub-surface light scattering (むしろ重みによっての予測)
  - メッシュアニメーションのボディーランゲージの問題を回避
  - 統計的にありえる特徴合成の見た目（良くも悪くも）

ラスタライズ描画と比較して、神経回路網による画像生成にはいくつかの利点があり、この研究ではそれらを維持したいと考えています。
ここではそれらの利点を列挙します。

ラスタライズの場合、弾力的な表面の変形は、メッシュを細かくして各頂点座標を変更する必要があります。
また、表面の下の構造からの影響を考慮することが困難です。
表面下散乱も、近似的にしか計算できません。

神経回路網の場合、教師データとなる画像の統計的な特徴を自動的に学習するため、リアルな見た目を再現することができます。
メッシュアニメーションのボディーランゲージの問題も回避できます。

ラスタライズ描画のメリット

Silicon Studio
Ideas × Art × Technology

- ラスタライズ＋レイトレーシングの利点
  - 計算処理の効率性
  - 細かくポーズを指定でき、非常に高い安定性
  - ピクセル品質の安定性が高い
  - 細かくカメラを指定できる
  - 細かくライティングを指定できる
  - 3Dソフトやライブラリの蓄積

ラスタライズ＋レイトレーシングにも利点があります。
慣れているので自然だと思う人も多いかも知れませんが、利点を列挙します。

計算処理の効率性に優れています。
細かくポーズを指定でき、安定性が非常に高いです。
ピクセル品質の安定性も高いです。
細かくカメラやライティングを指定できます。
数十年の歴史がありますので、3Dソフトやライブラリが豊富にあります。

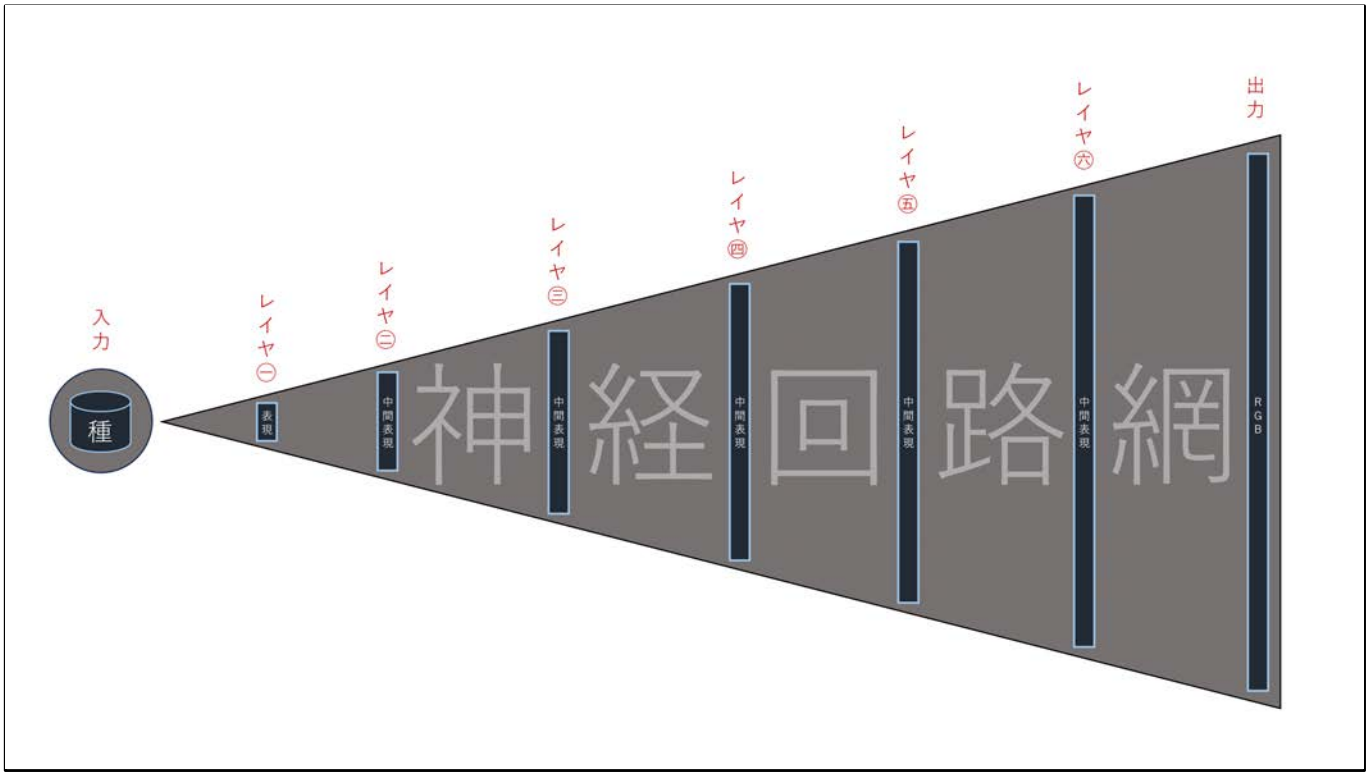
設計概念の紹介

Silicon Studio
Ideas × Art × Technology

- まずは設計の概念を図で紹介
  - 本研究で想定していることを直感的に伝えるため

まずは設計の概念を図で紹介します。

本研究で想定していることを直感的に伝えるためです。

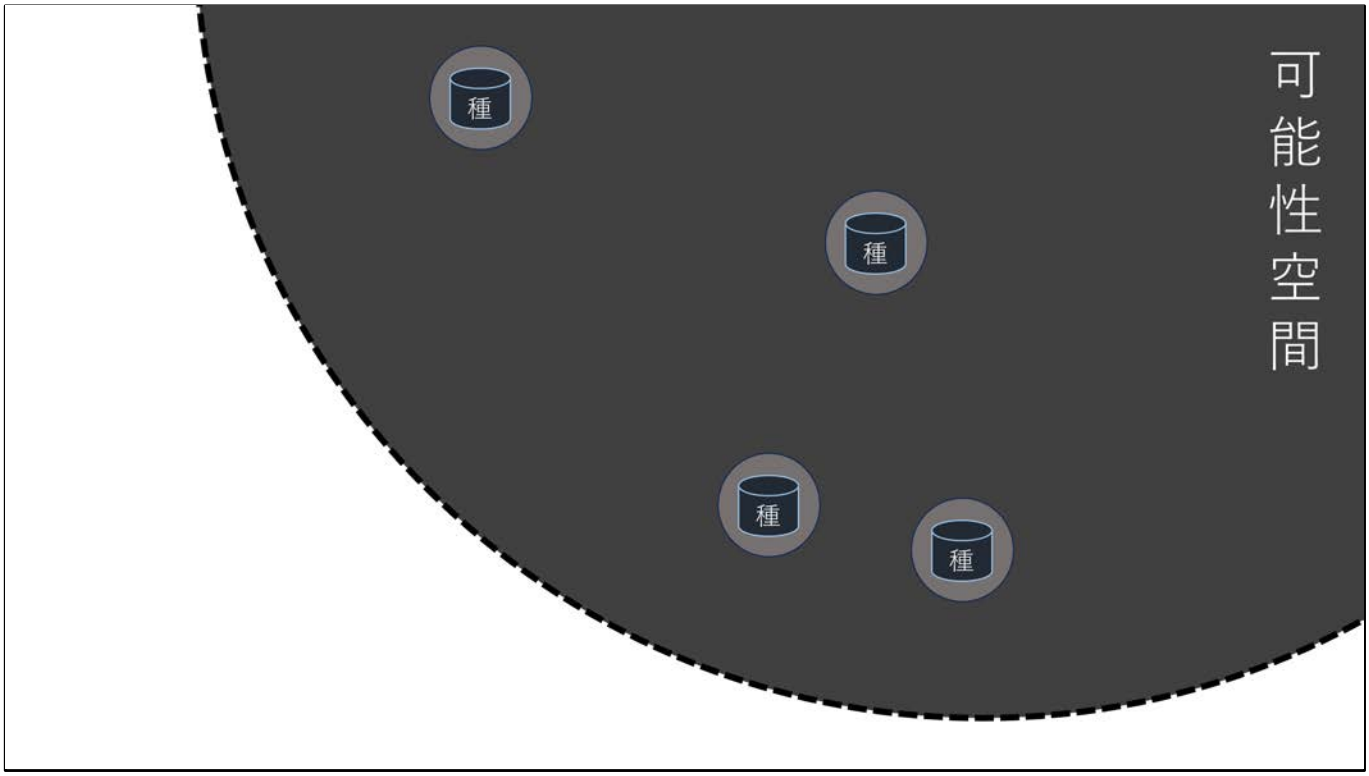


神経回路網が画像を生成するとき、入力された種からレイヤーごとに広がっていきます。

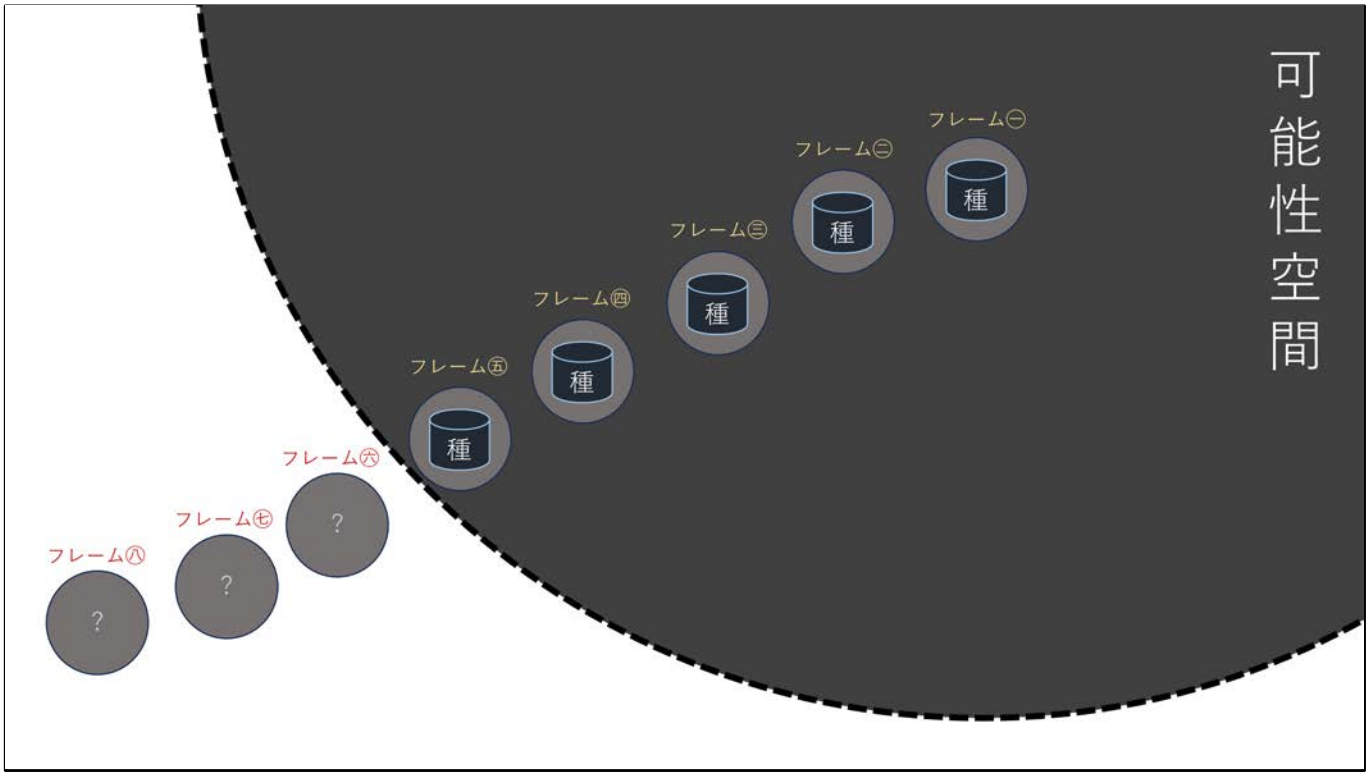
アーキテクチャによて、広がるのは解像度かも知れません、もしくは精度かも知れません。

でも、最終的にはRGBのピクセル値が出力されます。

出力されないと、画像生成ではないですね。

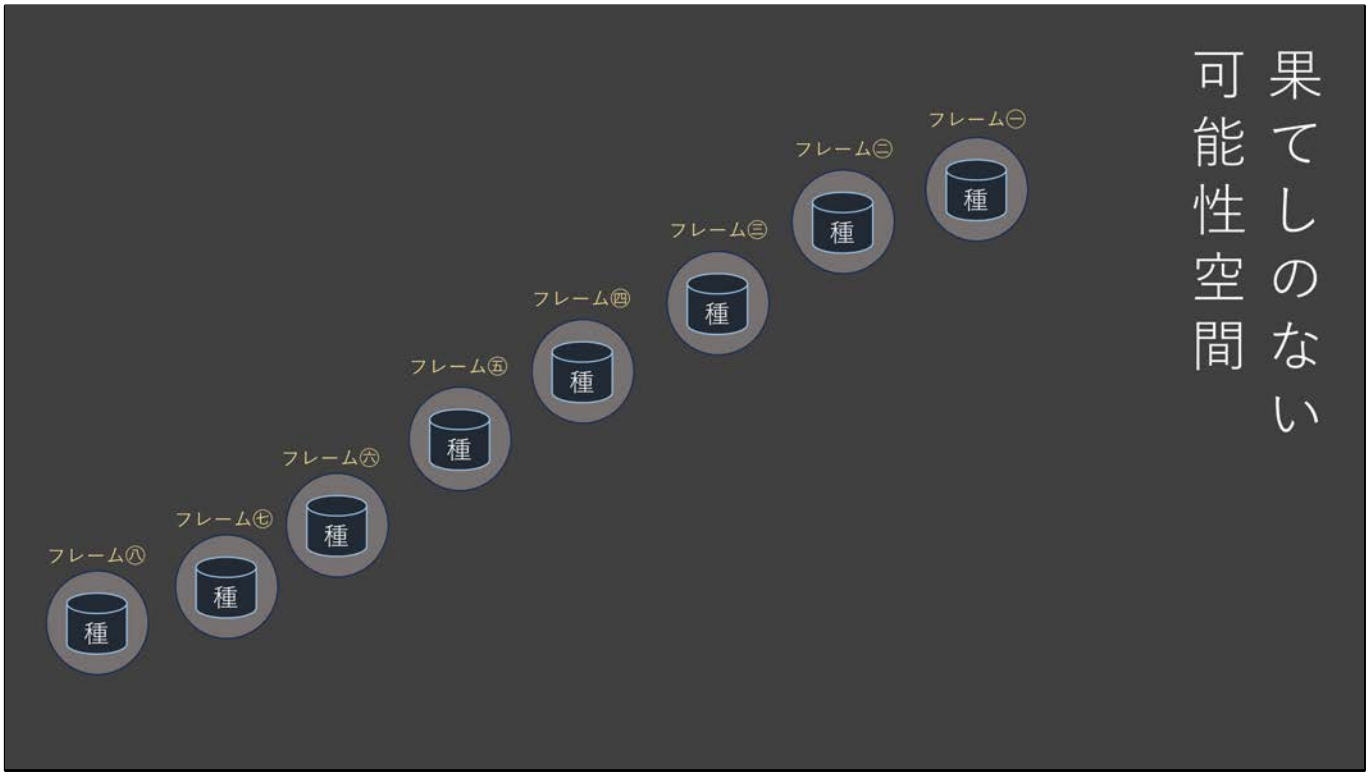


その最終的に生成される画像は、種の特徴によって最初から決定されます。
そのため、可能性空間は引数となる種の精度によって限定されています。
可能性空間の境界は、ありえること、とありえないことの区別です。
この神経回路網として。



基本的に、フレームごとに種を更新すれば、画像生成神経回路網でリアルタイム描画は可能です。
生成される画像は可能性空間の限界を超えることはできません。
この図では、フレーム⑥～⑧を生成できる種は存在しません。

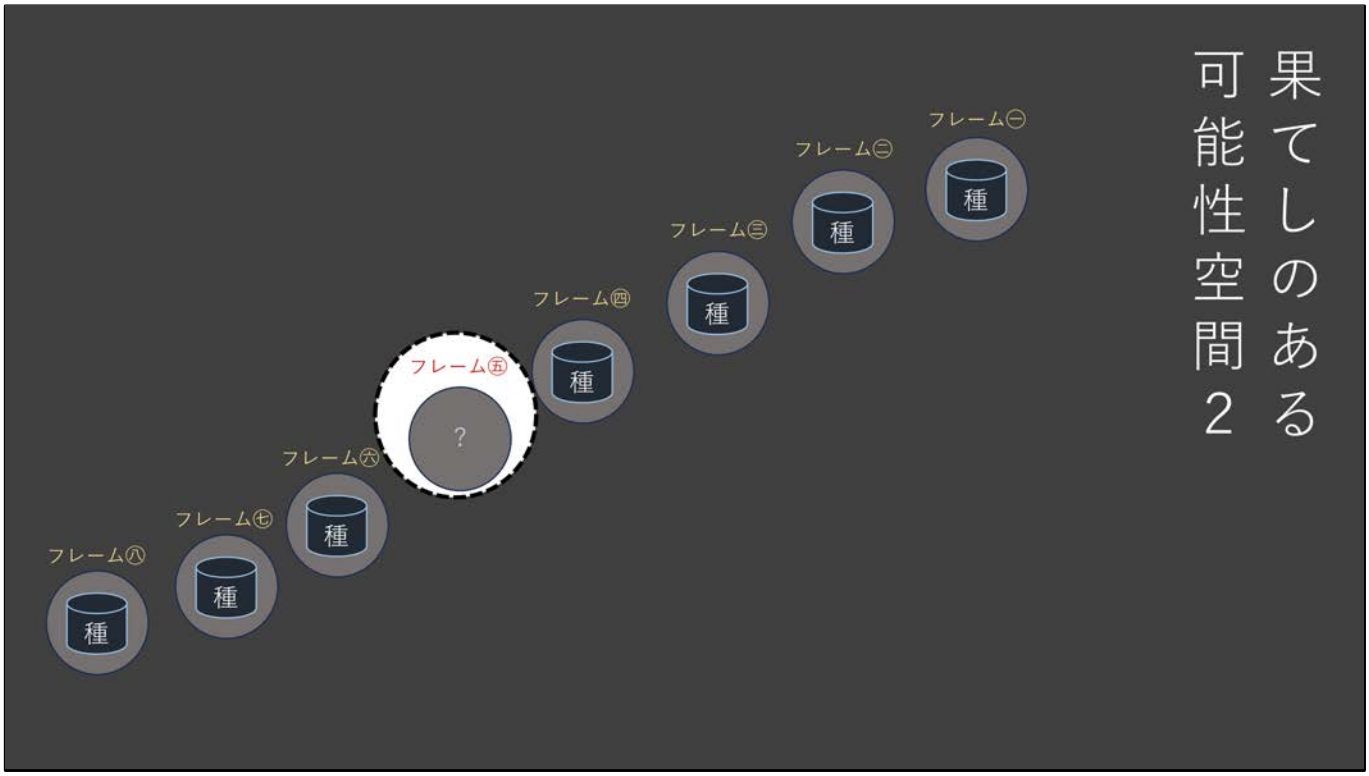
リアルタイム描画として、それは問題です。



もし、可能性空間が果てしなく広ければ、どんな画像でも生成できます。

可能性空間の文法が理解できるなら、と言うことですね。

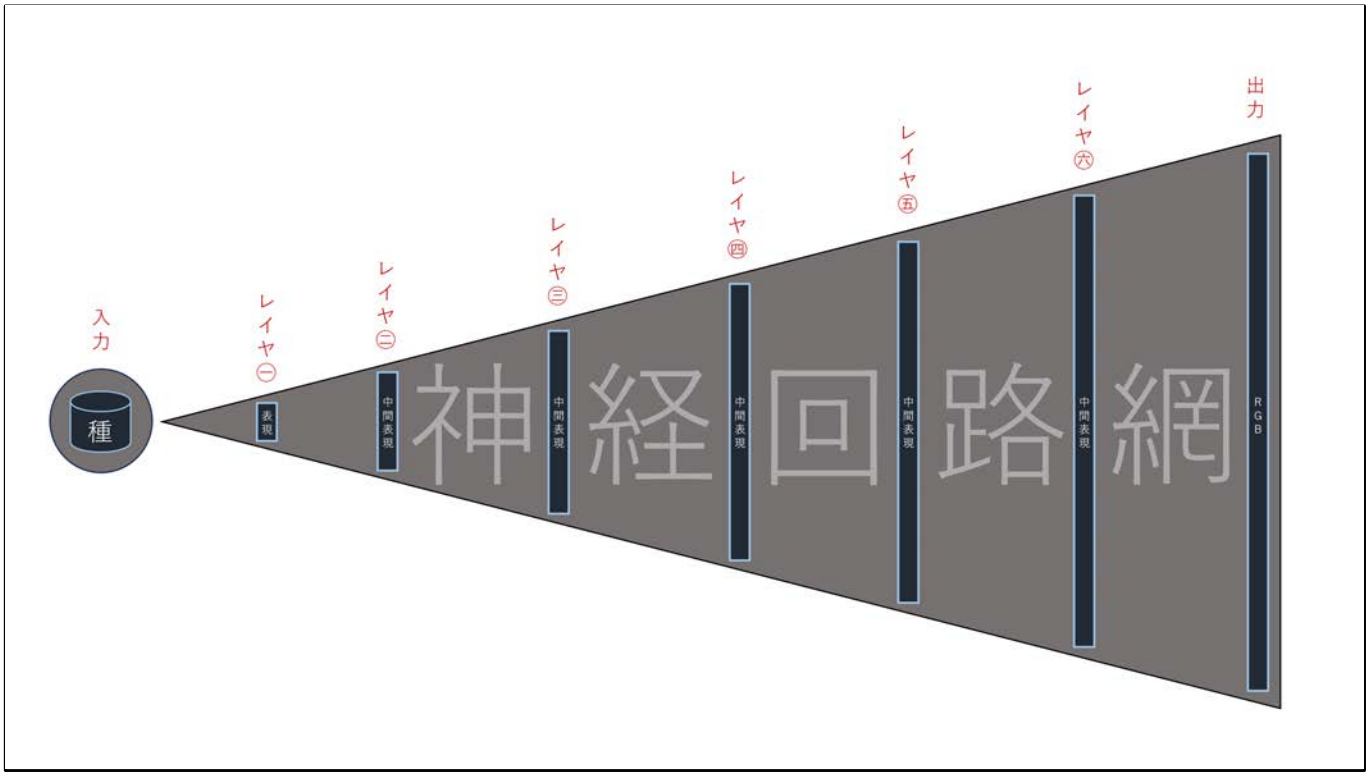
場合によって文法も難しいことです。



しかし、トレーニングのために描きたいことのすべてを網羅するサンプルを用意することは難しいです。

トレーニングの手法によってのmodality collapseもよく出ます。

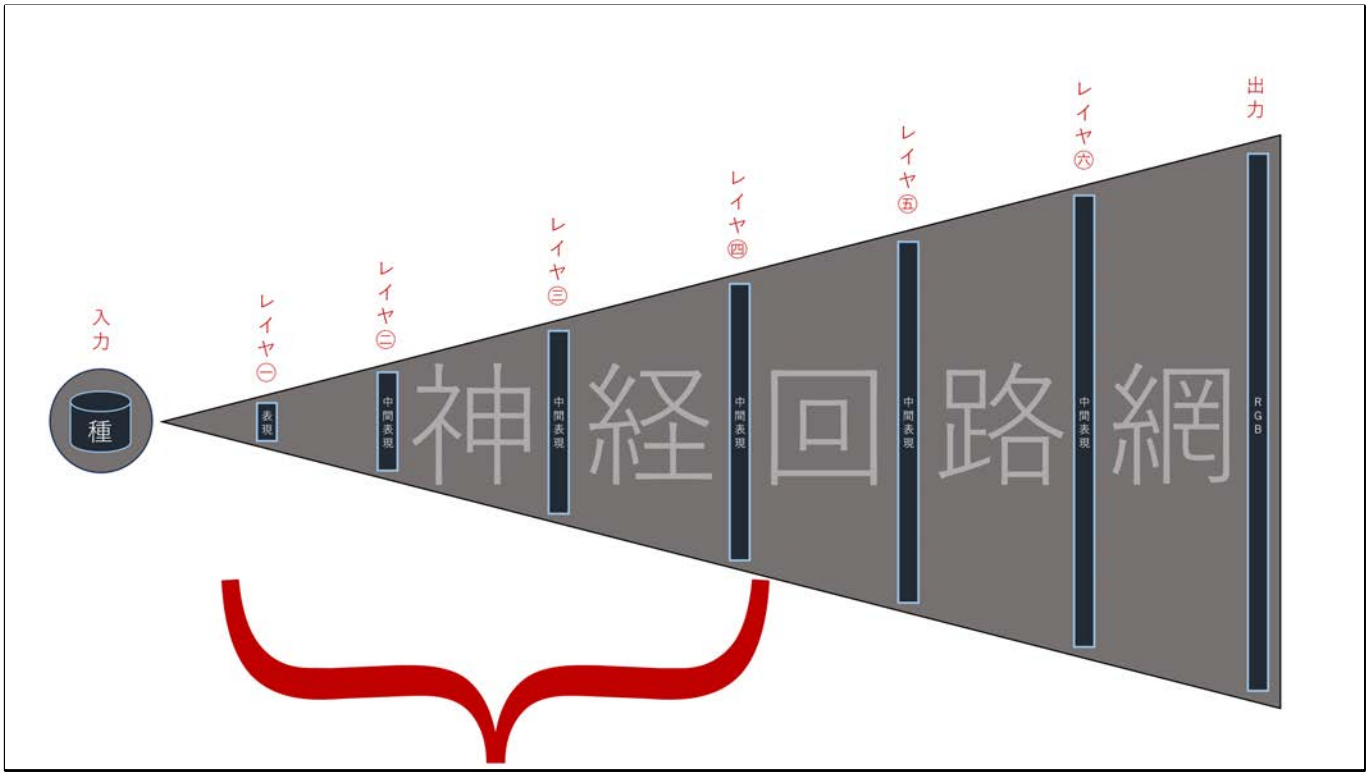
そのため、学習が不十分な領域ができてしまいます。
結局のところ、可能性空間には、どうしても限界があるらしいです。
果てしのない可能性空間は見たことがありません。



種から生成される各レイヤーの中間表現を見ていくと、

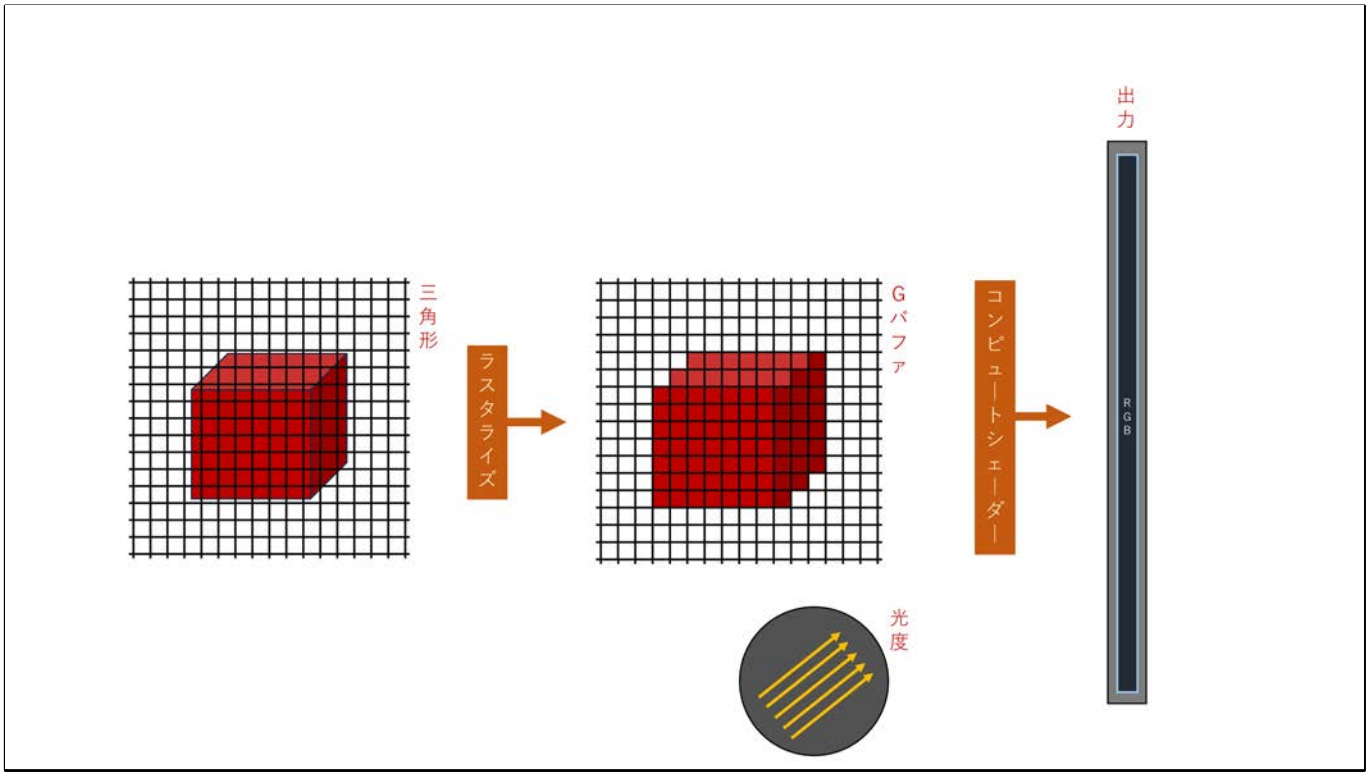
ポーズ、形状、光度：この三つは早い段階で現れます。

表面の詳細は遅い段階だけで生成されることが多いです。



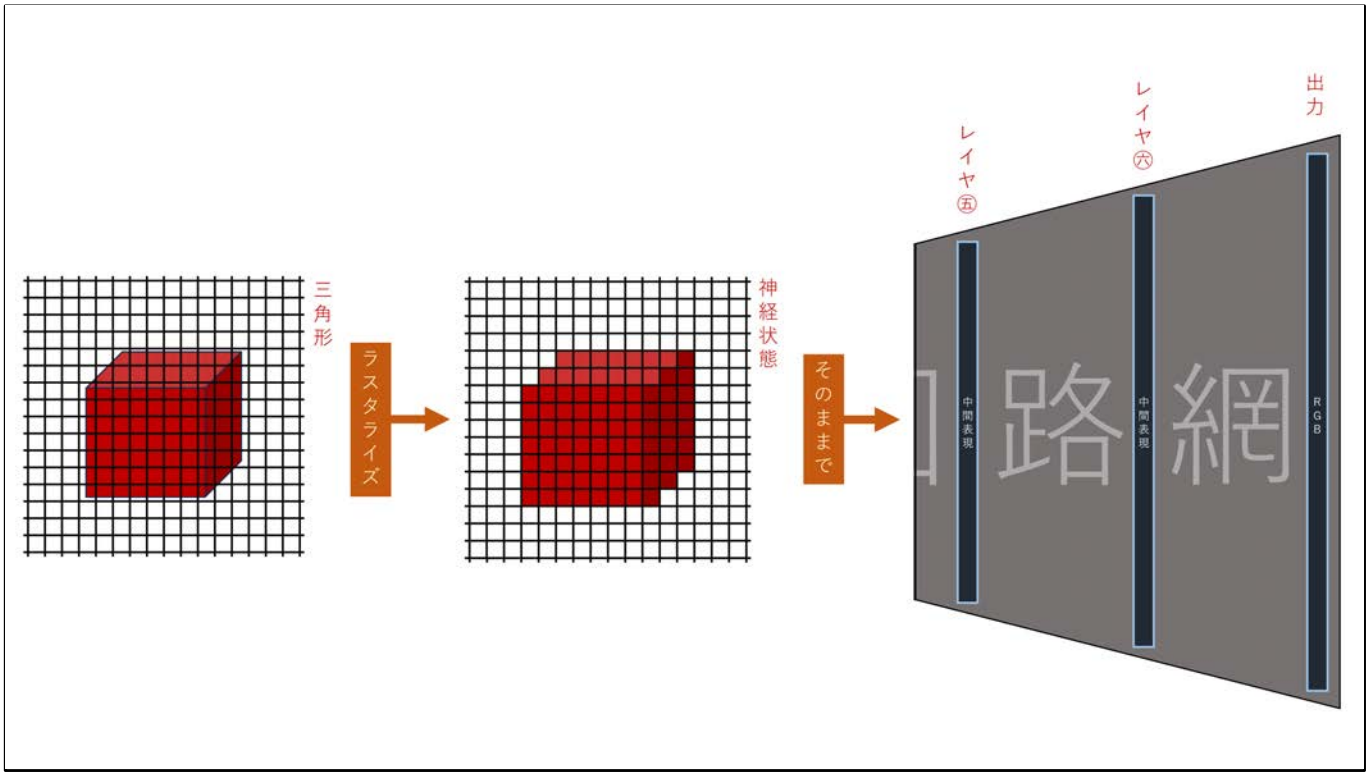
それで、可能性空間の問題点は主に前半の中間表現で出ます。

神経回路網の中間表現は、ある意味では、ブラックボックスです。
トレーニング手法の実装によっても変わり、定義が定まっていません。



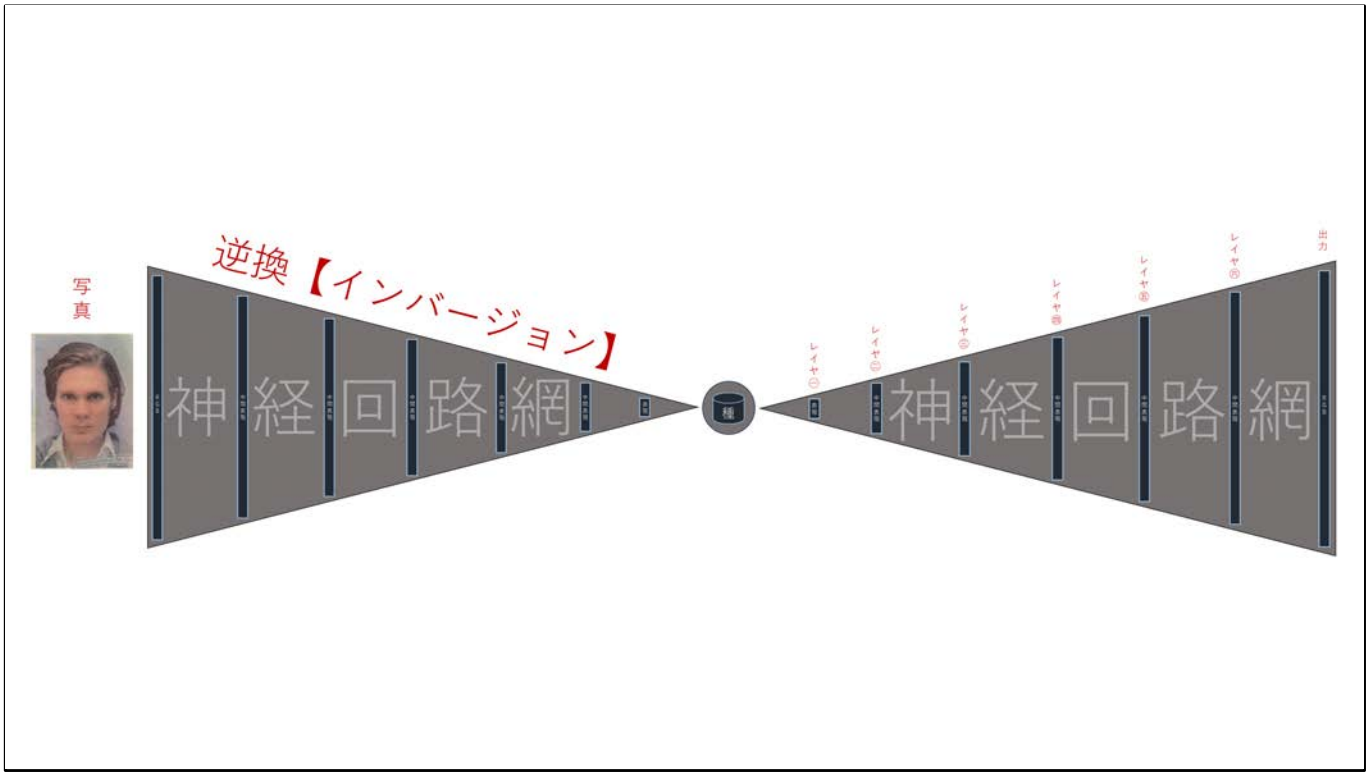
しかし、真ん中の神経レイヤーのデータ構成的な意味はある程度、整然としています。
概念的に、レンダーパイプラインのG-Bufferと似た働きをしています。

ラスタライズ描画の場合、まず三角形メッシュをラスタライズし、G-Bufferの各ピクセルに表面の法線やマテリアルなどを書き込みます。
そして、コンピュートシェーダーによって光源情報を用いて出力のRGB値を計算します。



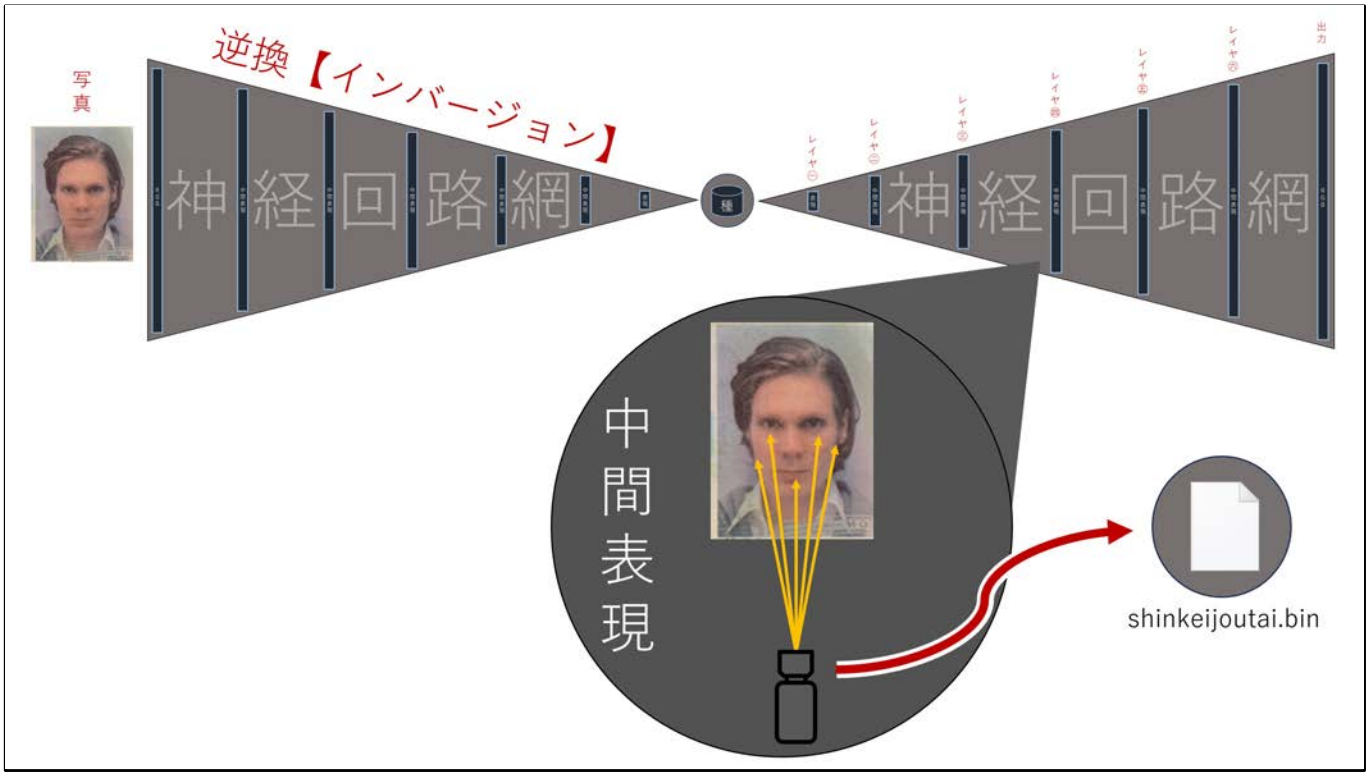
そこで、神経回路網による画像生成とラスタライズ描画を組み合わせることを提案します。

三角形メッシュをラスタライズした画像を、そのまま、中間レイヤーでの神経状態として用いて、神経回路網の後段と接続します。

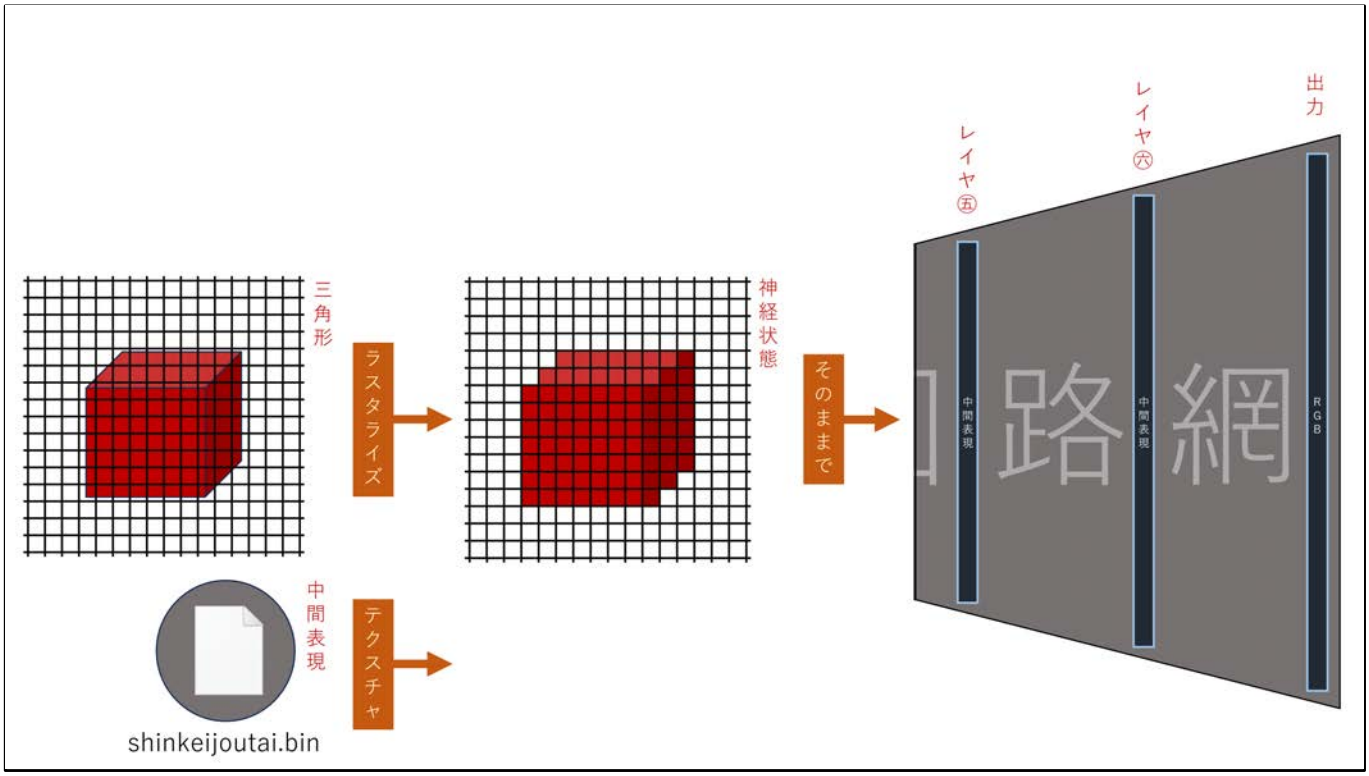


前のスライドのように神経状態をラスタライズしたいなら、神経状態の多次元テクスチャが必要です。
テクスチャを作るのはリアルタイムではなく、アセット用意のタイミングです。

作るためには写真逆換の神経回路網を利用することが可能です。
写真＋3Dスキャンのペアを利用すると良いアセットになれます。

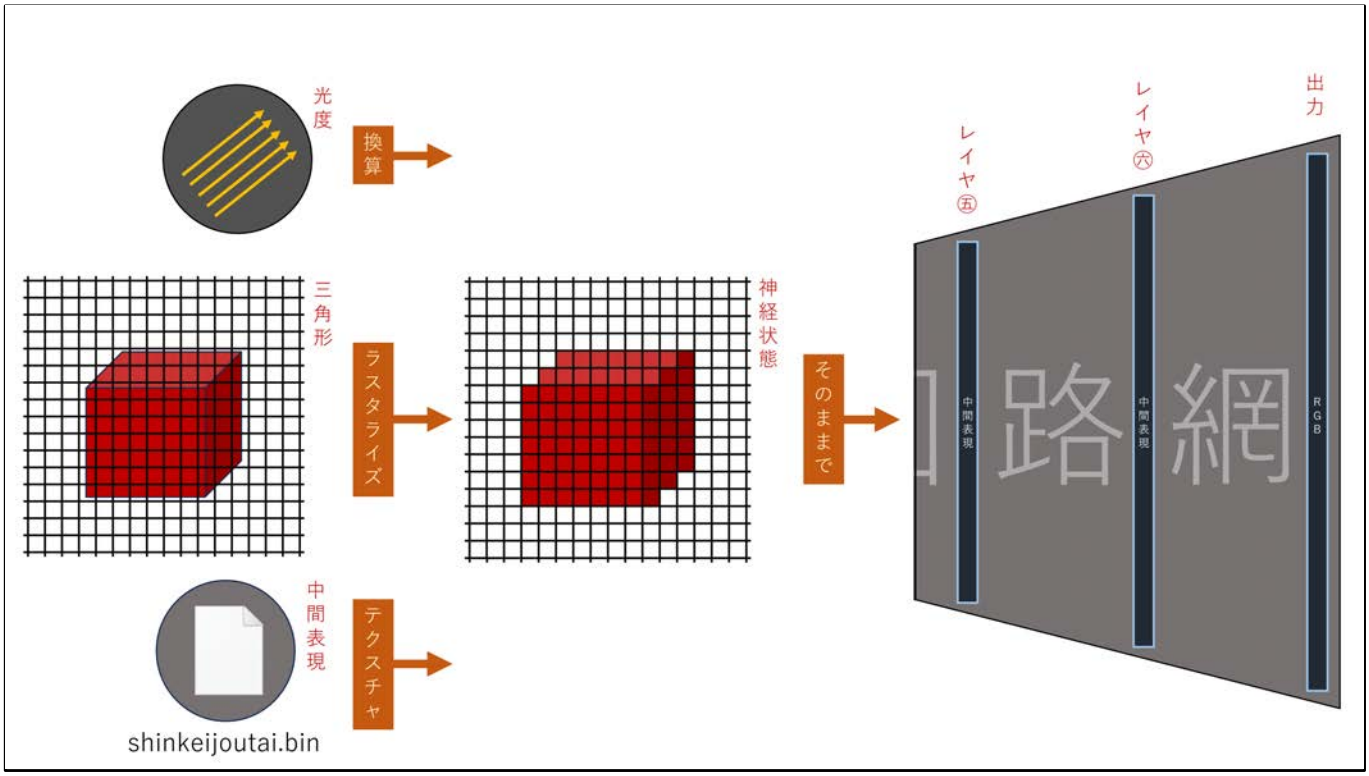


種から元の画像を復元するときに、中間レイヤーで出力された中間表現が、メッシュ表面のどこに接続されているかを見て、ファイルに書き出します。



中間表現をテクスチャとして三角形メッシュに貼り、リアルタイムにラスタライズして描画します。

これによって、メッシュの形状、姿勢、カメラ視点を直接的に反映させた神経状態を得ることができます。



ライティングの情報も、神経入力的なフォーマットに変換されて神経回路網に提供されます。

これにより、同じ種からカメラ位置、メッシュのポーズ、ライティングの異なる画像を生成できるようになります。
つまり、種の可能性空間に制約を受けずに自由度の高い画像生成が可能となります。

設計概念のまとめ

Silicon Studio
Ideas × Art × Technology

- 神経回路網による画像生成の画像は種から広がる
- 最終的な画像は種の特徴によって最初から決定され、
  種の可能性空間は引数の精度により制約される
- 生成された画像は可能性空間の限界を超えることはできない
- 仮に種の可能性空間を広げても、その中を100%満たして
  トレーニングすることはほぼ不可能

神経回路網による画像生成の画像は種から広がっていきます。
最終的な画像は種の特徴によって最初から決定され、種の可能性空間は引数の精度によって制約されます。
生成された画像は可能性空間の限界を超えることはできません。
仮に種の可能性空間を広げても、その中を100%満たしてトレーニングすることはほぼ不可能です。

設計概念のまとめ

Silicon Studio
Ideas × Art × Technology

- ポーズ、形状、光度：早い段階で出現
- 表面の詳細：遅い段階だけで生成されることが多い

- 基本的には、神経回路網の中間レイヤーからの出力のデータ形式はブラックボックス
  - トレーニング手法によっても変わるため定義できない
- しかし、真ん中の神経レイヤーの文法はある程度整然としている
  - 概念的にレンダーパイプラインのG-Bufferに似ている

ポーズ、形状、光度：その三つは早い段階で現れます。
表面の詳細は遅い段階だけで生成されることが多いです。

基本的には、神経回路網の中間レイヤーからの出力のデータ形式はブラックボックスであり、トレーニング手法によっても変わるため定義できません。
しかし、真ん中の神経レイヤーに関しては、文法（つまりデータ構造的な意味）はある程度整然としており、概念的にレンダーパイプラインのG-Bufferに似ています。

設計概念のまとめ

Silicon Studio
Ideas × Art × Technology

- そこで、この中間表現を直接ラスタライズで描画することを提案
  - メッシュのポーズ、カメラ位置やライティングは
    ラスタライズによって表現
  - 種の可能性空間の限界を超えて多様な描画が可能になる

- 任意の多次元テクスチャをメッシュに持たせ、
  ラスタライズすることは容易
  - 問題はテクスチャをどのように用意するか

そこで、この中間表現を直接ラスタライズで描画することを考えます。
これが可能であれば、メッシュのポーズ、カメラ位置やライティングはラスタライズによって表現でき、種の可能性空間の限界を超えて多様な描画が可能になります。
任意の多次元テクスチャをメッシュに持たせ、ラスタライズすることは容易であり、問題はテクスチャをどのように用意するかです。

提案手法の内容

Silicon Studio
Ideas × Art × Technology

- 提案手法は次の2つの手法から構成されます
  - リアルタイム描画
    - 毎フレーム実行される計算
    - ほぼ設計概念図の通り
  - アセット用意のパイプライン
    - 中間表現のテクスチャをどのように用意するか
    - こちらの方が重要

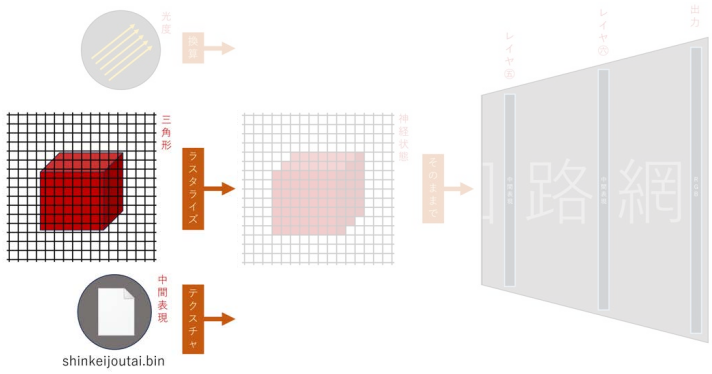
提案手法は次の2つの手法から構成されます
一つ目は、リアルタイム描画の手法です。
こちらは毎フレーム実行される計算になります。
ほぼ設計概念図の通りなので、理解しやすいと思います。

二つ目は、アセット用意のパイプラインの手法です。
中間表現のテクスチャをどのように用意するか、ということです。
こちらの方がより重要です。

リアルタイム描画の手法

Silicon Studio
Ideas × Art × Technology

1. 三角形のメッシュをラスタライズ
2. このメッシュには、神経回路網の中間レイヤーの多次元の
   神経出力値が含まれたテクスチャが貼られている

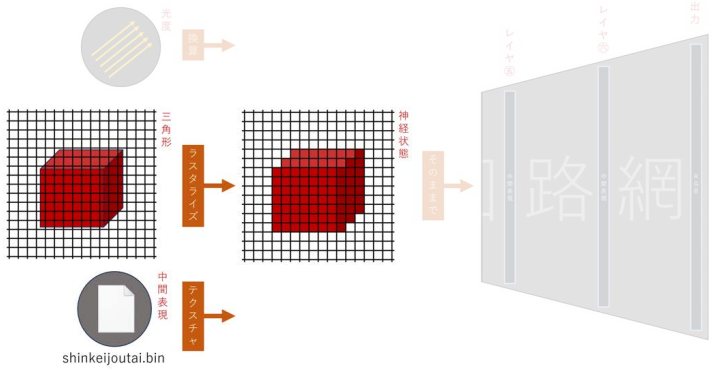


まず、三角形のメッシュをラスタライズします。
このメッシュには、神経回路網の中間レイヤーの多次元の神経出力値が含まれた
テクスチャが貼られています。

リアルタイム描画の手法

Silicon Studio
Ideas × Art × Technology

3. ラスタライズ後のオフスクリーン描画対象の画像が、神経回路網の中間レイヤーの神経状態(中間表現)に相当
   - 中間表現にはカメラ位置、メッシュのポーズの情報が反映されている

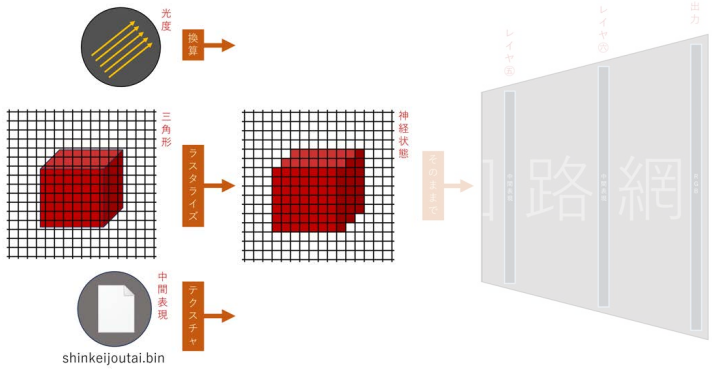


ラスタライズ後のオフスクリーン描画対象の画像が、神経回路網の中間レイヤーの神経状態(中間表現)に相当します。
中間表現にはカメラ位置、メッシュのポーズの情報が反映されています。

リアルタイム描画の手法

Silicon Studio
Ideas × Art × Technology

3. ラスタライズ後のオフスクリーン描画対象の画像が、神経回路網の中間レイヤーの神経状態(中間表現)に相当
   - 中間表現にはカメラ位置、メッシュのポーズの情報が反映されている
   - ライティング情報も神経入力的なフォーマットに変換されて神経回路網に入力される

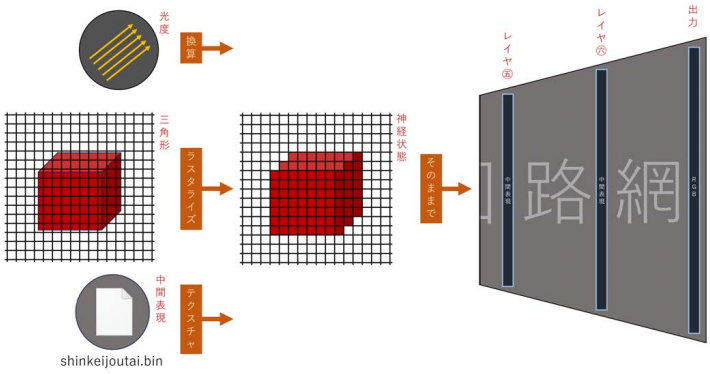


ライティング情報も、神経入力的なフォーマットに変換されて神経回路網に入力されます。

リアルタイム描画の手法

Silicon Studio
Ideas × Art × Technology

4. 元々の画像生成神経回路網の後半分を実行
5. 神経回路網の最終レイヤーの出力値はピクセル色であり、
   そのまま画面に表示される



元々の画像生成神経回路網の後半分が実行されます。
神経回路網の最終レイヤーの出力値はピクセル色であり、そのまま画面に表示されます。

アセット用意のパイプライン手法

Silicon Studio
Ideas × Art × Technology

- 各アセットから、それらの中間表現を含んだ重みテクスチャを生成する手法
  - リアルタイムに実行されることはない
  - ソースコードも製品のバイナリに含まれない

1. 対象物体の撮影
2. トレーニング
3. レイのエクスポート
4. 神経出力のエクスポート
5. レイキャスト
6. UV座標のエクスポート
7. テクスチャ生成
8. 3～7の繰り返し

各アセットから、それらの中間表現を含んだ重みテクスチャを生成する手法です。
リアルタイムに実行されることはありません。ソースコードも製品のバイナリに含みません。

アセット用意のパイプライン手法

Silicon Studio
Ideas × Art × Technology

1. 三次元スキャンによってのメッシュを用意。その際、同じ視点からの写真も撮影
2. 写真から種への逆変換（Inversion）をトレーニング
   - つまり、写真を画像生成神経回路網の種（Seed）に生成し、その種から再び同じ画像を再生成できるようにする

1. **対象物体の撮影**
2. **トレーニング**
3. レイのエクスポート
4. 神経出力のエクスポート
5. レイキャスト
6. UV座標のエクスポート
7. テクスチャ生成
8. 3～7の繰り返し

1) 三次元スキャンによってのメッシュを用意します。スキャンの瞬間に、同じ視点からの写真も撮影します。

2) 写真から種への逆変換（Inversion）をトレーニングします。
つまり、写真を画像生成神経回路網の種（Seed）に生成し、その種から再び同じ画像を再生成できるようにします。

アセット用意のパイプライン手法

Silicon Studio
Ideas × Art × Technology

3. 画像生成神経回路網内の計算処理ではレイが飛ばされている。
   そのレイ（三次元空間ベクトル）をファイルにエクスポート
4. レイが当たる座標の中間レイヤーの神経出力値を別のファイルにエクスポート

1. 対象物体の撮影
2. トレーニング
3. **レイのエクスポート**
4. **神経出力のエクスポート**
5. レイキャスト
6. UV座標のエクスポート
7. テクスチャ生成
8. 3～7の繰り返し

3) 画像生成神経回路網内の計算処理ではレイが飛ばされます。
そのレイ（三次元空間ベクトル）をファイルにエクスポートします。

4) レイが当たる座標の中間レイヤーの神経出力値を別のファイルにエクスポートします。

アセット用意のパイプライン手法

Silicon Studio
Ideas × Art × Technology

5. 別のアプリで（Vulkanなど）、メッシュにステップ3でエクスポートされたレイを飛ばす
6. レイが当たったメッシュ表面のUV座標をレイIDと共にファイルにエクスポート
7. レイIDに相当する神経出力値を多次元テクスチャのUV座標で書き込む

1. 対象物体の撮影
2. トレーニング
3. レイのエクスポート
4. 神経出力のエクスポート
5. **レイキャスト**
6. **UV座標のエクスポート**
7. **テクスチャ生成**
8. 3～7の繰り返し

5) 別のアプリで（我々はVulkanのAPIを利用して実験しています）メッシュにステップ3でエクスポートされたレイを飛ばします。

6) レイが当たったメッシュ表面のUV座標をレイIDと共にエクスポートし、ファイルに書き込みます。

7) レイIDに相当する神経出力値が多次元テクスチャのUV座標で書き込まれます。

アセット用意のパイプライン手法

Silicon Studio
Ideas × Art × Technology

8. カメラアングルを変更し、ステップ3～7を繰り返す。これにより、メッシュ表面の全面がテクスチャに書き込まれる

1. 対象物体の撮影
2. トレーニング
3. レイのエクスポート
4. 神経出力のエクスポート
5. レイキャスト
6. UV座標のエクスポート
7. テクスチャ生成
8. **3～7の繰り返し**

8) カメラアングルを変更し、ステップ3～7を繰り返します。これにより、メッシュ表面の全面がテクスチャに書き込まれます。

ラスタライズの問題点

Silicon Studio
Ideas × Art × Technology

- ラスタライズはある意味ではアナログ系のプロセス
- 信号強度が毎フレーム複数のピクセルに分割される
  - ピクセルの中心にぴったり一致して配置されることはほとんどない
- レイヤー間の信号強度伝達は線形変化に対して頑健でなければならない

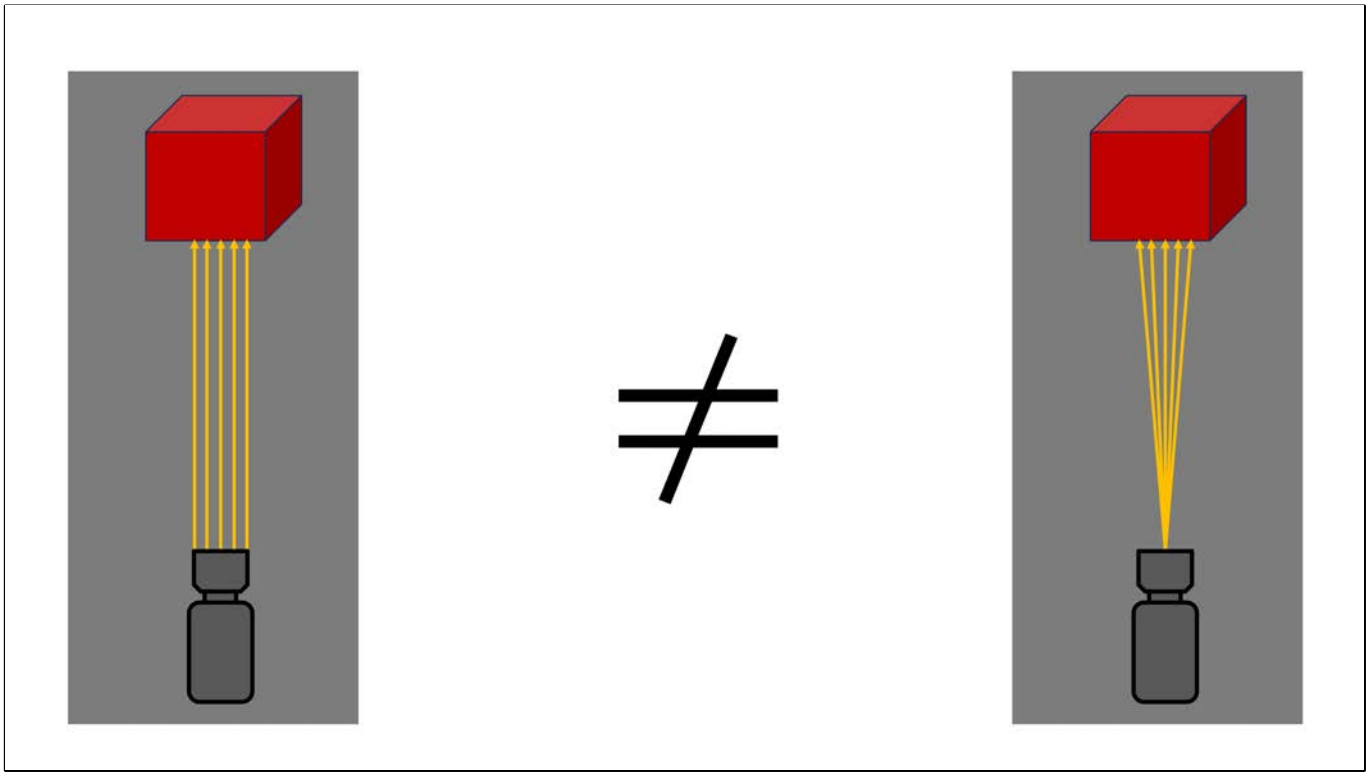
しかしながら、もうひとつの因子があります：
ラスタライズと言うのはある意味ではアナログ系のプロセスです。
信号強度が毎フレーム複数のピクセルに分割されます。ピクセルの中心にぴったり一致して配置されることはほとんどありません。
レイヤー間の信号強度伝達は線形変化に対して頑健でなければなりません。

ラスタライズの問題点

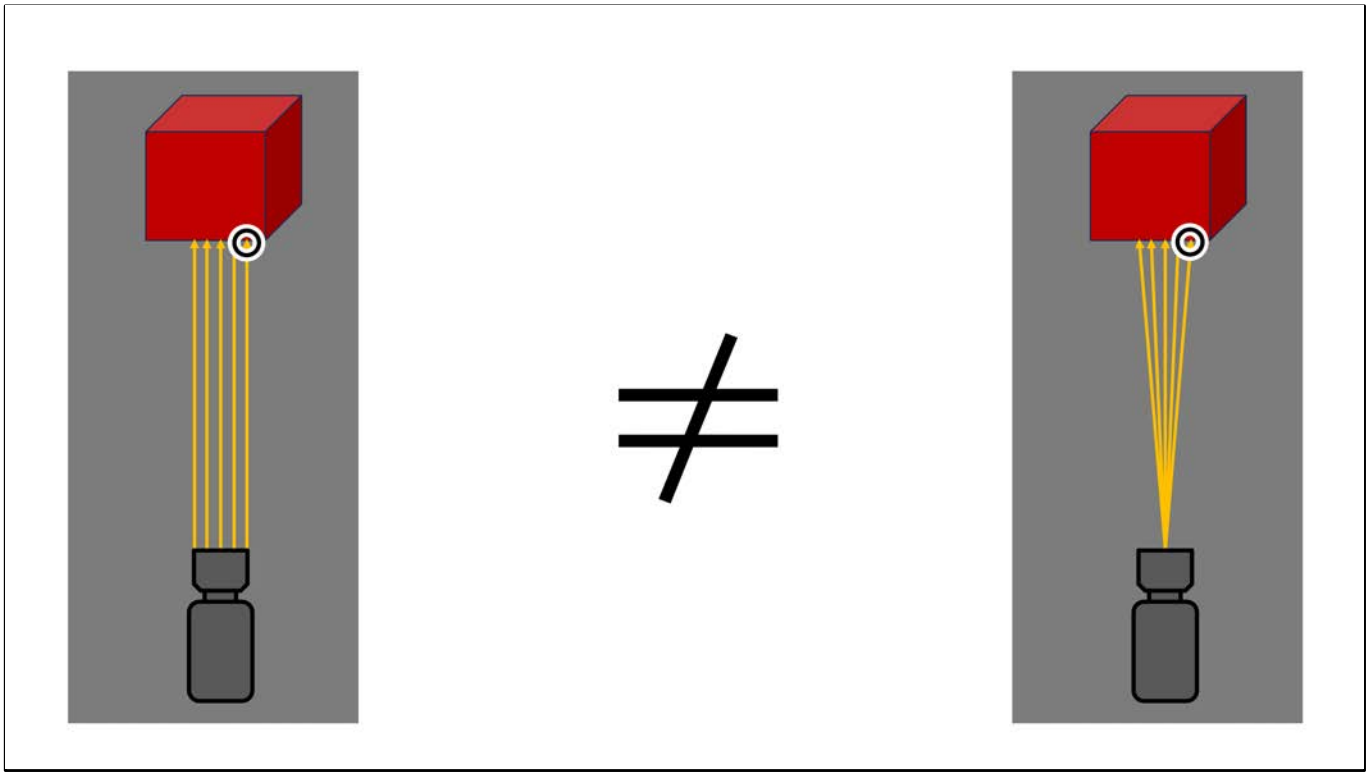
Silicon Studio
Ideas × Art × Technology

- そのため、神経回路網のトレーニングによってピクセル品質が不安定性になる場合がある
- その場合、トレーニングデータを（人工的に）増やしたほうがよいかも
- ただし、この問題はあくまで実装上のものであり、原理的に解決不能ではない

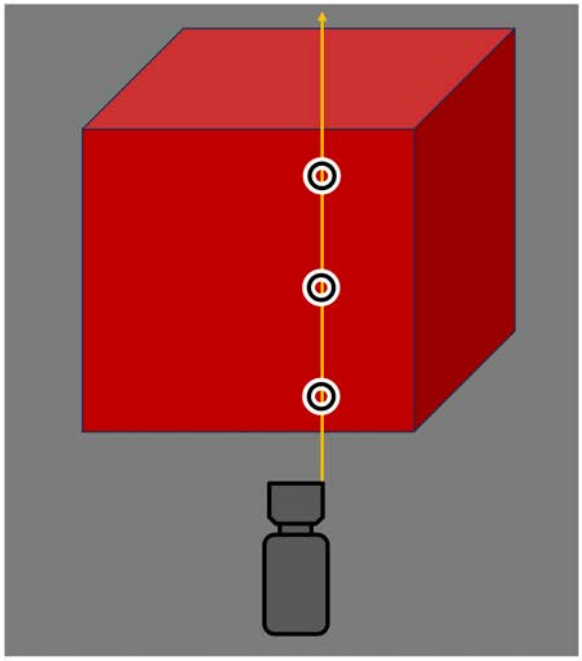
そのため、神経回路網のトレーニングによってピクセル品質が不安定性になる場合があります。
その場合、トレーニングデータを（人工的に）増やしたほうがいいかも知れません。
ただし、この問題はあくまで実装上のものであり、原理的に解決不能であるとは考えていません。



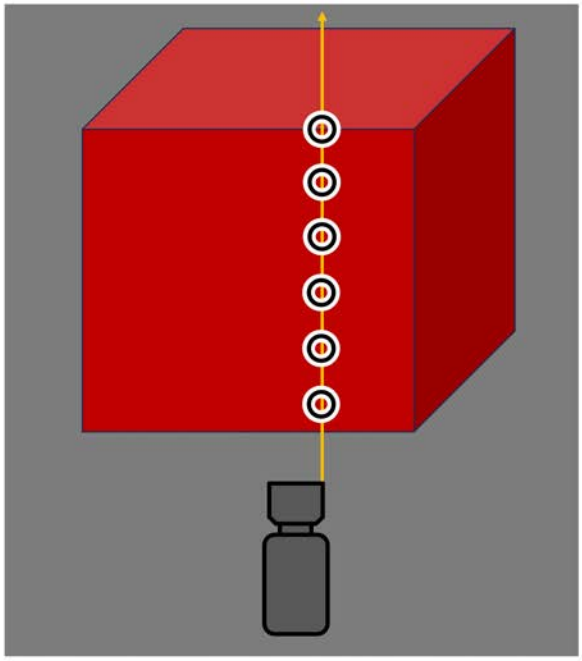
現在利用する画像生成神経回路網についてですが、似ている難点は他のimplicitモデルにも表れると思いますので、対策のいくつかを紹介します。
左と右で、レイを飛ばした描画結果は異なります。
投影方法の問題だけではないです。



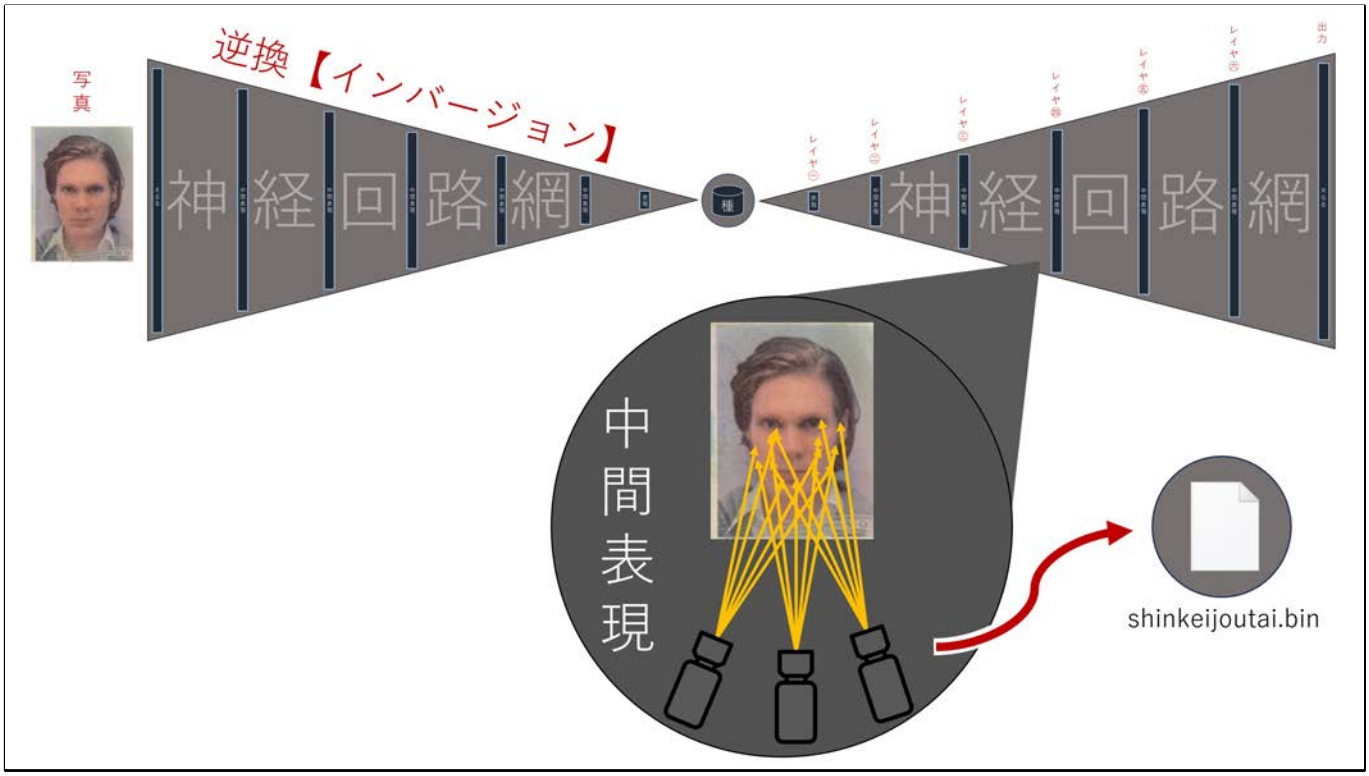
メッシュ表面に当たる座標が一致していたとしても、レイの描画結果が変わる場合もあります。
神経回路網のトレーニング頃からのバグだと考えても良いかも知れません。
何回も神経回路網を差し替えるつもりですので、いったん対策を載せた（のせた）ほうがいいと思います。
リアルタイム描画の時はレイを飛ばしませんので、アセット用意時だけの問題点です。



問題はレイ上の空間サンプリングの実装方法です。（＋トレーニング）
実際にサンプリングされる座標は、レイとメッシュが交差する点ではなく、レイ上を等間隔にサンプリングしています。
その後、神経回路網の次のレイヤーに送られるのは、レイとの交差点に近いサンプルのレイヤーで計算した出力の平均です。
結果は、このトレーニング環境に依存します。



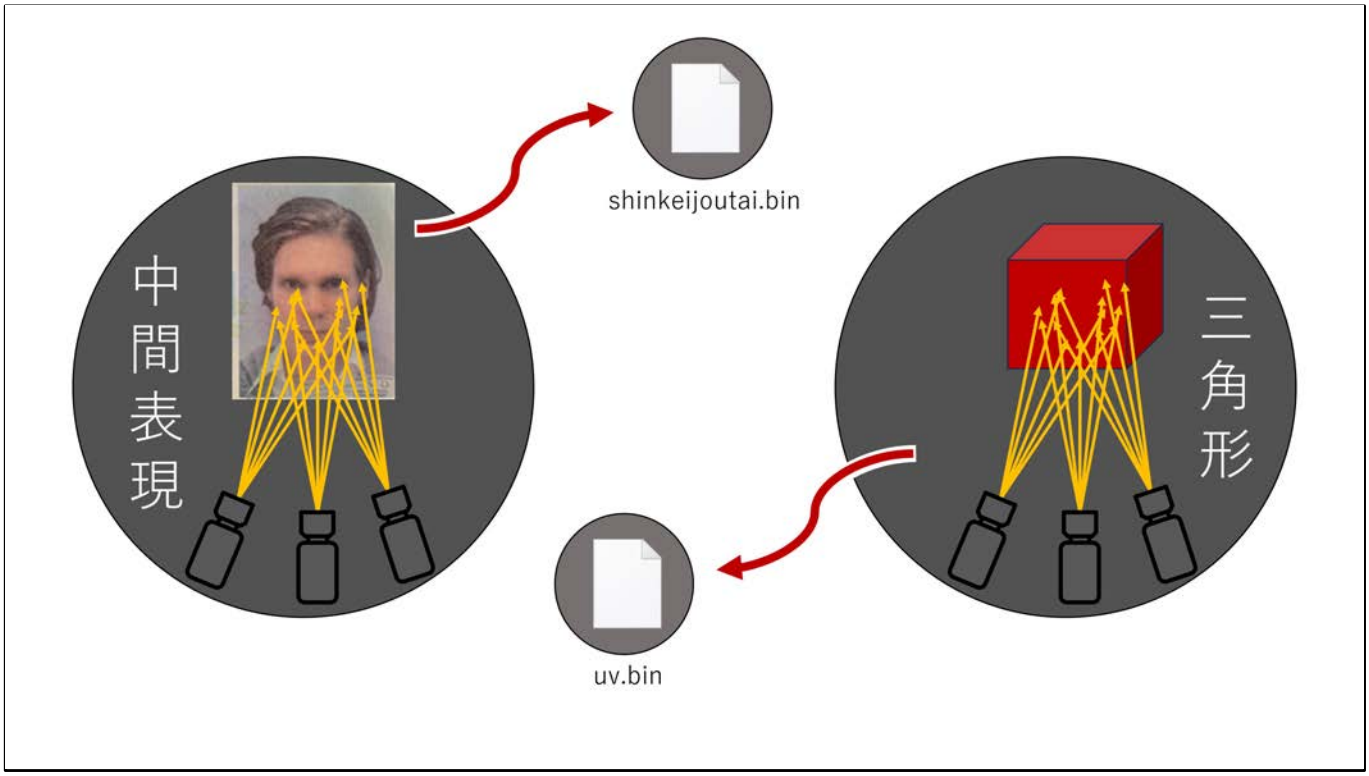
サンプル数を増やすと変化に対する頑健性が向上しますが、依然として不十分です。



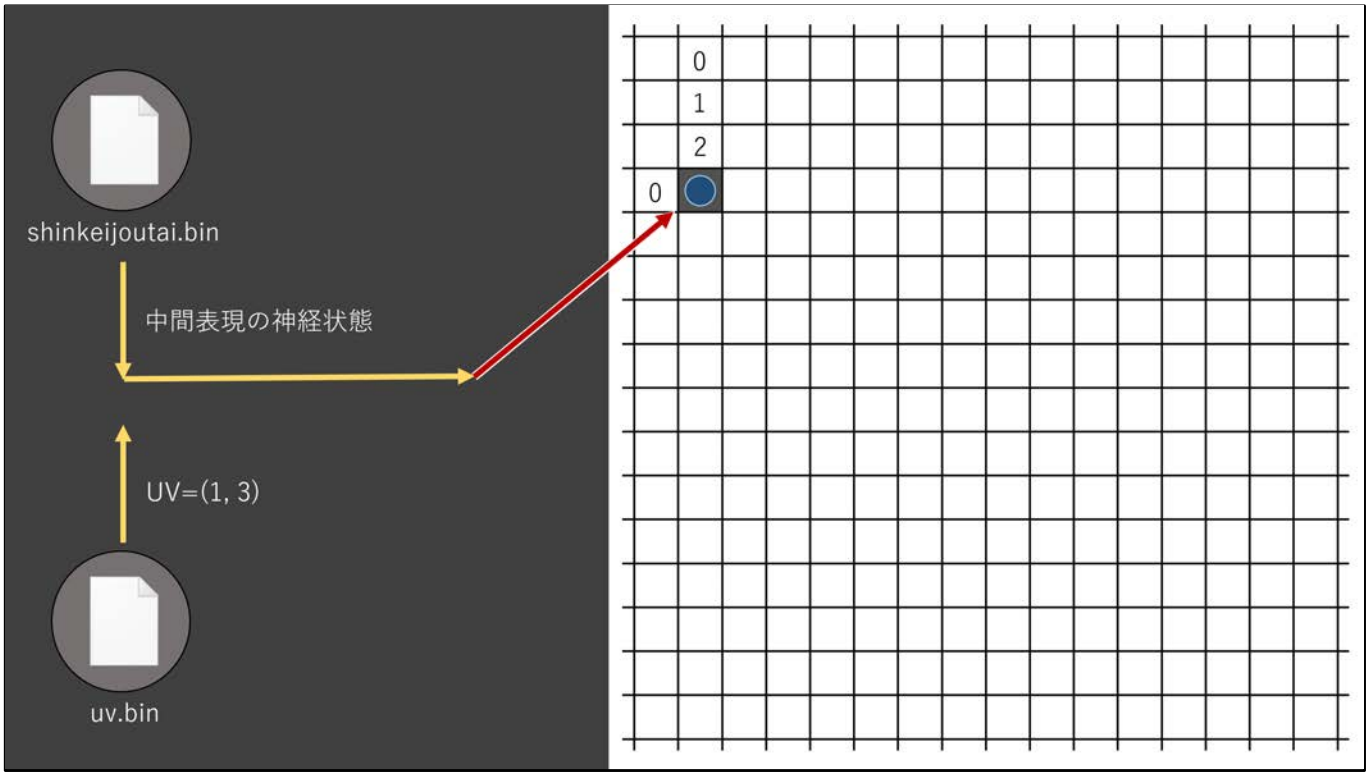
トレーニング時と同様にカメラとレイを利用したほうが安定です。

空間座標から直接サンプルできないので、複数のカメラアングルから撮影したほうがいいです。

以前のようにレイIDと、その位置での神経状態がファイルにエクスポートされます。

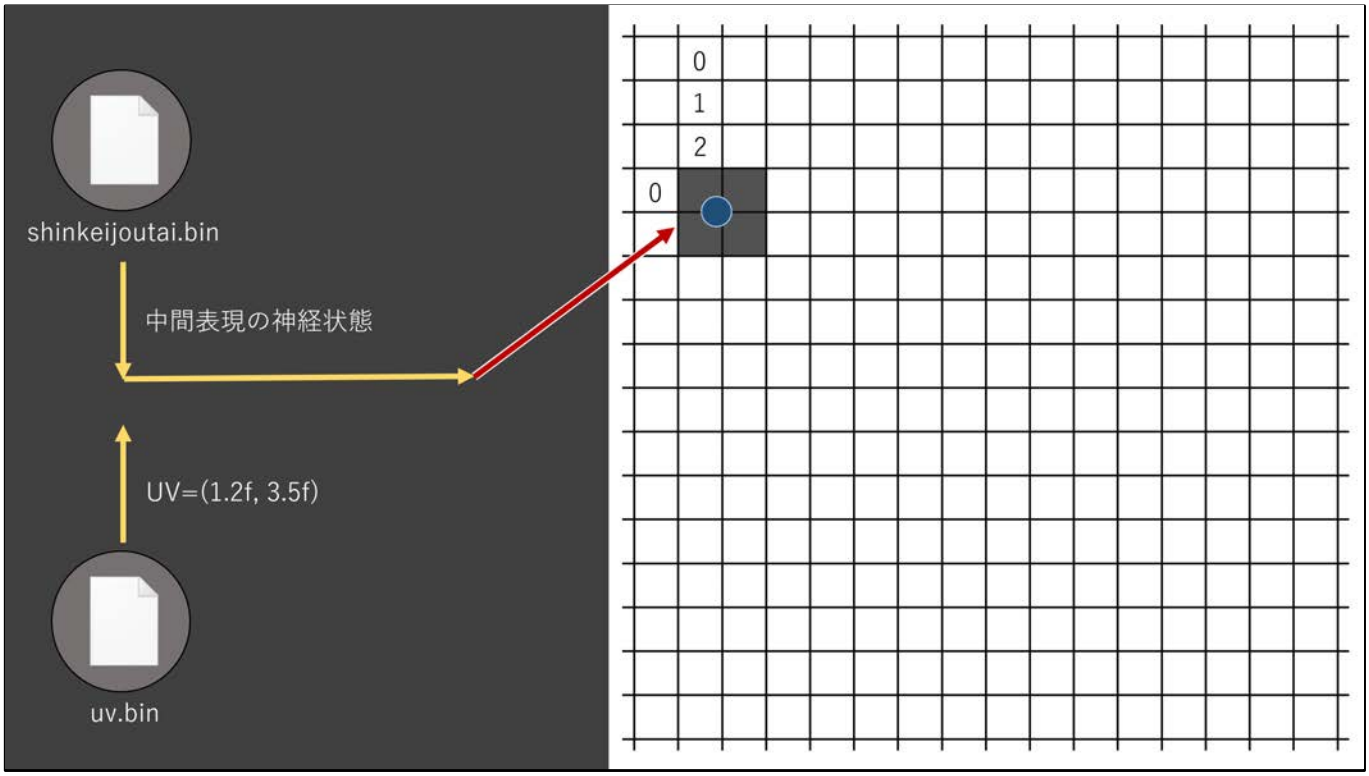


中間表現に飛ばしたレイを、三次元スキャンされたメッシュにも飛ばします。
そうするとレイあたりの【神経状態＋UV】のペアを手に入れることができます。



テクスチャを作るためは神経状態をUVの座標に書き込みます。

レイごとにこれを繰り返します。



UVはピクセル中心をぴったり指していない場合、値が分割されます。

アセット用意手法の要件

Silicon Studio
Ideas × Art × Technology

- アセット用意パイプラインの手法
  - 技術的にはこの手法は一番重要
- 満たすべき要件：
  - ３Dオブジェクトの中間表現が三角形メッシュと一致
  - 中間レイヤーの神経出力が表面座標に接続されてテクスチャに送られる
  - 表面の全部を覆うために複数のカメラアングルを統合
  - 三角形メッシュの品質が重要
- これらの条件さえ満たせばどんな手法でも十分

アセット用意パイプラインの手法についてです。

技術的にはこの手法は一番重要です。

満たすべき以下の要件があります：

3Dオブジェクトの中間表現が三角形メッシュと一致しています。

中間レイヤーの神経出力が表面座標に接続されてテクスチャに送られます。

表面の全部を覆うために複数のカメラアングルを統合します。

三角形メッシュの品質が重要です。

これらの条件さえ満たせばどんな手法でも十分です。

アセット用意手法のメリット

Silicon Studio
Ideas × Art × Technology

- 三次元スキャンによってメッシュは高品質
- 様々な物体を三次元スキャン可能
- 様々な画像生成神経回路網を利用可能
- プロセスの自動化は(ある程度)可能

本発表で提案した、アセット用意手法は以下のメリットがあります。

三次元スキャンによってのメッシュは品質高いです。
様々な物を三次元スキャンできます。
多くの画像生成神経回路網を利用できます。
プロセスの自動化はある程度は可能です。

技術的な難点（ほぼ解決されました）

Silicon Studio
Ideas × Art × Technology

- 写真インバージョンの神経回路網が必要
  - 画像生成神経回路網の三次元中間表現が弱い場合も
- 神経回路網のPythonライブラリのカメラ計算はC++にエクスポートしにくい場合がある
- 画像生成神経回路網の解像度が低い場合もある

実験を進めていく上で遭遇した技術的な難点です。
これらは、現在ではほぼ解決されています。

まず、写真インバージョンの神経回路網が必要です。
画像生成神経回路網の三次元中間表現が弱い場合もあります。
神経回路網のPythonライブラリのカメラ計算は場合によってC++にエクスポートしにくいです。
画像生成神経回路網の解像度が低い場合もあります。

将来の研究

Silicon Studio
Ideas × Art × Technology

- 中間表現フォーマットの入力形式の検討
  - 神経回路網の後半分の入力としてフレーム毎に必要になる情報
  - 今後は、利用できる情報を増やしていきたい
- 特に重要な入力は、レイトレーシングによる照明情報
  - ハイエンドGPUではリアルタイムレイトレーシングが可能
  - レイトレーシング結果を神経状態に変換するため、専用の神経回路網を用いる計画

将来の研究です。

中間表現フォーマットの入力形式の検討します。
神経回路網の後半分の入力としてフレーム毎に必要になる情報です。
今後は、利用できる情報を増やしていきたいです。

特に重要な入力は、レイトレーシングによる照明情報です。
ハイエンドGPUではリアルタイムレイトレーシングが可能です。
レイトレーシング結果を神経状態に変換するため、専用の神経回路網を用いる計画です。

将来の研究

Silicon Studio
Ideas × Art × Technology

- その次のステージは、物体内部の情報
  - 人間の顔なら皮膚下の筋肉や骨格をシミュレーション
    - これらの情報を神経回路網に入力することでより正確な描画が可能に
  - 人間の心理状態をシミュレーション
    - 顔の骨格や表情筋の動きは心理状態が反映されたものなので、心理状態を直接入力する
    - これにより、より違和感の少ない人間の描画ができるはず

その次のステージは、物体内部の情報です。
人間の顔なら、皮膚下の筋肉や骨格をシミュレーションします。
現時点は具体的な計画がないですが、それらの情報を神経回路網に入力することでより正確な描画が可能になります。

また、顔の骨格や表情筋の動きは心理状態が反映されたものなので、直接的に人間の心理状態を入力とする方法も考えられます。
これにより、より違和感の少ない人間の描画ができるはずです。

まとめ

- 神経回路網による画像生成
  - すでに広い範囲において写実的な人間の外見や状態を描画可能
  - カメラ位置やメッシュ形状などを細かく指定することはできない
- 提案手法
  - リアルタイム描画に必要な情報を、従来のラスタライズパイプラインによって生成
  - 神経回路網に入力できる形式に変換し、神経回路網の後半の層で画像を生成
- 実装は順調に進行中

神経回路網による画像生成は、すでに広い範囲において写実的な人間の外見や状態を描画することができますが、カメラ位置やメッシュ形状などを細かく指定することはできません。
そこで、リアルタイム描画に必要な情報を、従来のラスタライズパイプラインによって生成し、その後、神経回路網に入力できる形式に変換する手法を提案しました。
まだ結果の画像をお見せできる状態ではありませんが、これまでのところ、実装は順調に進んでおり、成功への根本的な障壁はありません。

Ideas × Art × Technology

技術力・表現力・発想力を兼ね備えたCGソリューションプロバイダー