

深層学習を用いた  
3D モデルのテクスチャの自動生成

東京工科大学大学院

バイオ・情報メディア研究科

メディアサイエンス専攻

チン エキトウ

深層学習を用いた  
3D モデルのテクスチャの自動生成

指導教員 渡辺 大地 教授

東京工科大学大学院  
バイオ・情報メディア研究科  
メディアサイエンス専攻

チン エキトウ

## 論文の要旨

論文題目	深層学習を用いた 3D モデルのテクスチャーの自動生成
執筆者氏名	チン エキトウ
指導教員	渡辺 大地 教授
キーワード	深層学習、敵対的生成ネットワーク、テクスチャー、U-Net

### [要旨]

近年、3D モデリングは多くの産業で革新的な変化をもたらし、特にゲーム開発とビジネスの世界において著しく、企業はこの技術を利用して、製品のプロトタイピングからマーケティングまで、多岐にわたるアプリケーションで新しい価値を創出している。例えばゲーム開発の分野では、開発者たちは没入感の高いゲーム環境を構築することに注力している。これはプレイヤーにリアルな体験を提供し、ゲーム世界に深く引き込むために不可欠である。しかし、リアリティの追求は、高品質な 3D モデルのテクスチャーを必要とし、これはしばしば膨大なコストと時間を要する作業となる。このような背景の中、深層学習を用いてテクスチャーの自動生成を考え、研究を行う。しかし従来の研究では膨大なデータを使い、長時間の学習段階をわたり、深層学習のモデルを学習させることで高品質な画像データを生成する。

本研究では深層学習を用いたテクスチャーの自動生成を目的とする。ベースとする手法 FAMOS を利用して、テクスチャーを生成するが、検証段階においてそのまま利用できない問題点があり、今回ベースとした手法にはコンテンツ画像とテンプレート画像二つの学習データを収集し、前処理を行う。またコンテンツ画像の生成にある問題点を改善し、入力と出力の部分処理した学習データに適応し、学習を行う。生成された画像の評価についてはアンケートを取り、獲得した点数を t 検定を用いて、有意差を得る。

# A b s t r a c t

Title	Using Deep Learning to Generate Textures for 3D Models
Author	Yitong Chen
Advisor	Taichi Watanabe
Key Words	Deep Learning, GAN, Textures, U-Net

## [summary]

In recent years, 3D modeling has brought about innovative changes in many industries, especially in the realms of game development and business. Companies are leveraging this technology to create new value across a wide range of applications, from product prototyping to marketing. In the field of game development, for instance, developers are focusing on creating immersive game environments. This is essential for providing players with a realistic experience and deeply engaging them in the game world. However, the pursuit of realism requires high-quality 3D model textures, which often entails considerable cost and time.

Against this backdrop, there is consideration for using deep learning to automate texture generation, conducting research in this area. Traditional studies have involved using vast amounts of data and lengthy learning phases to train deep learning models to generate high-quality image data.

This study aims to automate texture generation using deep learning, utilizing the FAMOS method as a base for generating textures. However, there are issues that make it unsuitable for direct use in the verification stage. The method we base our study on involves collecting two types of learning data: content images and template images, and performing preprocessing. Additionally, we address certain problems in generating content images, adapting the input and output parts of the processed learning data for training. The evaluation of the generated images will be conducted through surveys, and the scores obtained will be analyzed using t-tests to determine significant differences.

# 目次

第1章	はじめに	1
1.1	背景と目的	2
1.2	論文構成	6
第2章	深層学習と入力データの提案手法	7
2.1	既存手法の検討	8
2.2	学習データ	9
2.3	スタイル生成の改善	12
2.4	入力と出力	15
2.5	評価方法	15
第3章	実験	17
3.1	実験環境	18
3.2	学習データの前処理	18
3.2.1	コンテンツ画像の分割	18
3.2.2	テンプレート画像の分割	20
3.3	パラメータの調整	21
3.4	画像生成	22
3.5	評価	27
第4章	まとめ	31
	謝辞	34
	参考文献	36
	発表実績	40

# 目次

1.1	ゲーム用いる 3D モデルの実例	5
1.2	UV 展開簡単な事例	5
1.3	GAN のアーキテクチャ	6
2.1	FAMOS のアーキテクスチャ	9
2.2	ロボット型 3D モデルの UV テクスチャ	10
2.3	XDoG によりエッジ抽出後の UV テクスチャ	11
2.4	グレースケールした後の uv テクスチャ	11
2.5	FAMOS を用いたロボット型 3D モデルの UV テクスチャの自動生成画像	13
2.6	FAMOS で生成された UV テクスチャを 3D モデルに貼り付けた結果、(左と右) 生成画像を貼り付けた結果、(中) オリジナルテクスチャ	13
2.7	生成テクスチャとオリジナルテクスチャ頭部の詳細の比較	14
2.8	生成テクスチャとオリジナルテクスチャ胸部の詳細の比較	14
2.9	生成テクスチャとオリジナルテクスチャ武器の詳細の比較	14
2.10	FAMOS を基づく入力と出力を変更したアーキテクスチャ	15
3.1	エッジ抽出後のテクスチャを分割した後のテクスチャ	19
3.2	グレースケールした後のテクスチャを分割した後のテクスチャ	20
3.3	分割処理したテンプレート画像	21
3.4	エポックごとに学習回数 250 回	22
3.5	エポックごとに学習回数 10 回	22
3.6	グレースケール処理後の生成したロボット型 3D モデルのテクスチャ	24
3.7	XDoG 処理後の生成したロボット型 3D モデルのテクスチャ	25
3.8	Unity 上で描画した 3D モデル、(左) コンテンツ画像をグレースケール変換後の生成テクスチャ、(中) オリジナルテクスチャ、(右) コンテンツ画像をエッジ抽出後の生成テクスチャ	26

3.9	オリジナルテクスチャーと生成テクスチャーを 3d モデルに貼り付けた後の頭部の詳細 . . . . .	26
3.10	オリジナルテクスチャーと生成テクスチャーを 3d モデルに貼り付けた後の胸部の詳細 . . . . .	26
3.11	オリジナルテクスチャーと生成テクスチャーを 3d モデルに貼り付けた後の武器の詳細 . . . . .	27
3.12	本研究の手法を用いた生成したテクスチャーケース 1 . . . . .	28
3.13	本研究の手法を用いた生成したテクスチャーケース 2 . . . . .	28
3.14	先行手法を用いた生成したテクスチャーケース 3 . . . . .	29
3.15	先行手法を用いた生成したテクスチャーケース 4 . . . . .	29

# 第 1 章

## はじめに

## 1.1 背景と目的

3D モデルにおける 3D テクスチャの制作は複数の手順を含む詳細なプロセスである。一般的な手順はコンセプトとリファレンスの収集、ベーステクスチャーや詳細テクスチャーの追加、UV マッピング、シェーディングとマテリアル等々のステップを繰り返すことで、望むレベルのリアリズムと質を実現するアートワークを作り上げる。プロジェクトの要求や目的によっては、さらに専門的なテクニックや工程が加わることもある。それらの作業量は膨大であり、制作には時間やコストなどもかかる。図 1.1 でゲーム用いる 3D モデルの実例を示す。この 3D モデルのような近年のゲームにおいて同じ 3D モデルを使用し違う模様のモデルを利用しながら、ゲームに応用していることも多くなっている。ゲームにおける呼び方はスキンである。また GAME CREATORS 編集部の斉藤 [1] ではゲーム内でスキンを実装するメリットとしてはユーザーに新鮮な気持ちで操作することができる。またゲームを続けるモチベーションになると論じた。しかしそれらを作るのにテクスチャーの色デザインも考え直す必要があると考えられる。Su ら [2] にも 3D モデルにおけるテクスチャーの制作にはかなり時間がかかることを論じている。また小林 [3] は 3D モデルの制作を紹介した上で、金銭的成本もかかることが多くなる。そこに着目したのは UV テクスチャー遯ると Edwin ら [4] によって開発したテクスチャー座標であり、2 次元ベクトル、また 3 次元ベクトルで示され、ベクトルの成分は多い場合  $U, V, W$  で示す。近年 UV マッピングも発展し、この UV とはポリゴン、サブディビジョンサーフェスメッシュの頂点コンポーネント情報が備わった 2 次元のテクスチャーであり、つまり 3D モデルの各頂点に対応する 2D テクスチャーの位置を定義するため使用する。UV テクスチャーを使用して、2D テクスチャーを 3D モデルの表面に正確に配置することはできる。またこの利点としては詳細なテクスチャリングができる。UV マッピングを使用すると、モデルの特定の部分に詳細なテクスチャーを適用することができる。図 1.2 は UV 展開簡単な事例である。

そして本研究ではゲーム制作やその他の 3D グラフィックス関連領域での利用を想定し、3D モデルのテクスチャー自動生成技術に着目した。2014 年に登場した敵対的生成ネットワーク (GAN) [5] は、イアン・グッドフェローらにより 2014 に発表した「Generative Adversarial Nets」では。GAN ではデータ生成に用いる生成器と生成されたデータの真偽を識別する識別器の二つのニューラルネットワークで構成する生成モデルの一種である。図 1.3 で GAN のアーキ

テクチャを示す。このフレームワークに基づき、DCGAN[6]、PGGAN[7]、StyleGAN[8] などのように様々な深層学習アルゴリズムが現れた。それらの先行研究では主に莫大なデータセットを用いて、長時間の学習を経て見たことのない高品質な画像生成をすることが可能である。また画像生成する以外にも Yaniv Taigman ら [9] が開発したスタイル移行または顔写真からイラストへの変換にできる GAN がある。しかし、データセットが少ない場合ではこれらの手法は効果的ではないと考えられる。多様なコンテンツ生成において注目を集め、特に StyleGAN[8] は人間の顔の画像生成において顕著な成果を挙げているが、物体のテクスチャー生成についてはまだ十分には発展していない。

一方で、画像認識の領域では代表的な CNN(Convolutional Neural Network)[10] は深層学習の一つ畳み込みニューラルネットワークのことであり、通常のニューラルネットワークと違い、全結合層だけでなく、畳み込み層とプーリング層から構成するニューラルネットワークである。特徴としては指定したフィルタと呼ばれる小領域を畳み込み、特徴量とするため、通常のニューラルネットワークのように1ピクセルに対する特徴よりも、画像の傾向を知ることができる。代表的には Karen ら [11] が開発した VGG では 16 から 19 の重み層を持つネットワークで大規模のデータを訓練した。それを GAN モデルによって生成された画像の評価にも用いられている。また Joseph ら [12] が開発した YOLO が医療分野でよく使われている。

そして、FCN(Fully Convolutional Network)[13] はこの問題を解決し、CNN よりは畳み込み層の後に完全連結層を用い、分類のための固定長の特徴ベクトルを得るのに対して、FCN は任意のサイズの入力を受け入れ、デコンボリューション層を用いて最後の畳み込み層の特徴テクスチャーをアップサンプリングし、入力画像と同じサイズに復元し、各画素に対する予測値を生成することが可能であるが、一部の位置情報はプーリング層で失われる可能性もある。U-net では各スケールごとにエンコーダの特徴テクスチャーとデコーダの特徴テクスチャーに連結し、正しく空間情報が保たれることができ、この問題を解決した。元々は医療画像処理を目的に開発したが、現在では医療分野だけではなく、画像認識タスク全般において高いパフォーマンスを示している。これらの技術を用いて高品質な画像を生成するためには、大規模な計算資源、長時間の処理時間、そして大量のトレーニングデータが一般的に必要とする。さらに、3D モデルのテクスチャー生成では、非周期性や独自性といった特性を要求するため、これらを含むテクスチャーの

生成は、さらなる挑戦を意味する。

本研究では 3D モデルの UV テクスチャに対して、色付けの作業を効率化するため、Nikolay ら [14] の既存手法に基づき、パーツの特徴が保たれたまま、スタイル変換を行う。また UV テクスチャの特性については山崎 [15] では、3D モデルを UV 展開し、配置した後、各面にその画像を正確に貼り付けることができるの特性で、UV テクスチャをスタイル変換することで、模様だけを変える 3D モデルの制作は可能となる。そこで本研究では既存手法における問題点を改善し、3D モデルに適応できるテクスチャを自動生成する。手法としてはまず学習データの処理、既存手法で用いたデータ形式では 3D モデルのテクスチャデータには適応しないため、学習データの前処理は必要とする。また学習データの前処理による入力と出力も改善する必要がある。最後に既存手法に用いる畳み込み層やパラメータは学習データによる調整する。以上の手法を用いて、生成されたテクスチャを Unity のマニュアル [16] を参照し、3D モデルに描画する。また評価はゲームプレイヤーや 3D モデルを作成したことがある経験者にアンケートを答えてもらう形で評価を行った。アンケートの内容では 4 個のケースを作り、各組の点数を獲得し、t 検定を行った。結果として、本研究の手法と先行手法で生成したテクスチャを 3D モデルに描画した後、アンケートで点数評価を得て、t 検定で有意差があることは判明し、本研究の手法では有効であると考えられる。



図 1.1 ゲーム用いる 3D モデルの実例

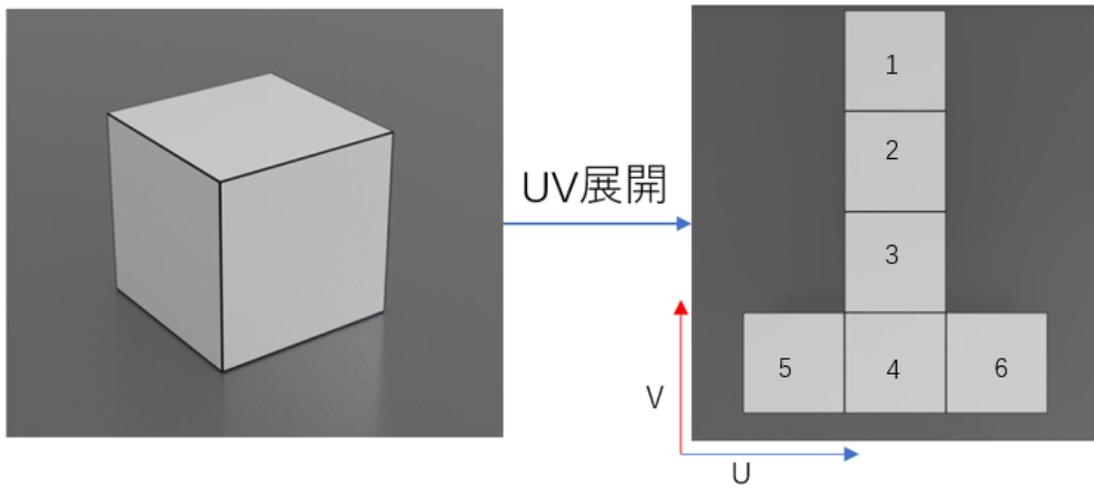


図 1.2 UV 展開簡単な事例

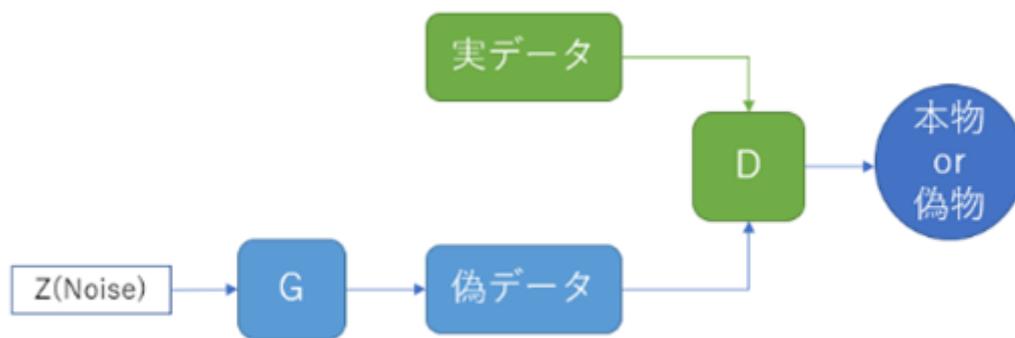


図 1.3 GAN のアーキテクチャ

## 1.2 論文構成

第 1 章で本研究の背景及び意義について説明する。第 2 章で本研究の提案手法を述べる。第 3 章で提案手法による実験を行う。第 4 章でまとめを述べる。

## 第 2 章

# 深層学習と入力データの提案手法

## 2.1 既存手法の検討

テクスチャー自動生成の技術は様々があり、Frederick ら [17] は実世界のシーンの 3D モデルにテクスチャーをマッピングする手法について研究しており、実世界のシーンからデータを収集するのは写真撮影、3D スキャン、または LiDAR 技術の使用も含むデータ収集を行い、それらの学習データを用いながら色、パターン、表面の特性などを考慮し、テクスチャー生成する。また L. Grammatikopoulos ら [18] は都市の 3D モデルを自動生成について新しい提案をした。データは衛星画像や航空調査などの写真撮影を利用し、CNN と GAN を用いながら生成した。その中 CNN の役割は撮影した都市画像の建物や道路などの識別し、3D モデル化のための基礎データを提供した。また GAN を用いてリアルなテクスチャー生成やデータ拡張に用いた。

また近年 GAN などの技術が発展することにより、数多くの画像生成手法が現れた。Elad ら [19] はテキストガイドという新たなテクスチャー生成手法を提案し、震度から画像への拡散モデルを用いる。このモデルは、事前に訓練した震度から画像への拡散モデルを活用し、3D モデルを異なる視点からペイントし、テクスチャーが 3D モデルに正確に適用する。また Gao ら [20] は微分可能な表面モデリングや微分可能なレンダリングなどの技術を用い、高品質な 3D モデルに用いるテクスチャーの自動生成も可能となった。しかしそれらの手法には莫大な学習データを使い、長時間の学習時間を経て、高品質なテクスチャーを自動生成に実現した。また画像から 3D モデルの再構築を深層学習で実現する技術も発展しており、例えば Henderson ら [21] は 2D 画像から直接テクスチャー付きの 3D メッシュを生成するため新たな深層学習フレームワークを提案し、複数枚の画像でテクスチャー付きの 3D モデルを生成することが可能となった。しかし、深層学習を用いて、3D モデルの生成はクオリティやテクスチャーのディテールなどの精度は高いとは言えないと考えられる。

本研究で主に UV テクスチャーに対するスタイル変換によって新しいテクスチャーを生成することを目的とするので、Gao らと他の研究と比べ、少ないデータから生成を可能にするため、図 2.1 で FAMOS のアーキテクチャを示す。参考になった Nikolay ら [14] の手法で GAN と U-net を用い、スタイル変換が行えた。それを実現するにはまずアーキテクチャの中に存在する U-net では 4 層のエンコーダとデコーダを使用しコンテンツ画像と座標を受け取る。位置座標を保たれまま、その次に混合配列はコンテンツ画像とテンプレートの座標を基づき計算する。この配列は

メモリテンプレートの各画像に対する重み付きを含み、これらの重みを使用して最終的なスタイルリゼーションした画像を生成する。混合配列を通じて、ネットワーク、コンテンツ画像に対して最適なテクスチャーの組み合わせを選択し、それによってリアルなテクスチャーを再現する。このプロセスは、画像生成におけるパラメトリックと非パラメトリックのアプローチの組み合わせを表しており、FAMOS の特徴的な側面の一つである。混合配列を使用することで、ネットワークは、単にパラメトリックな生成に頼るのではなく、既存のテクスチャーを効果的に活用して、よりリアルで複雑な画像スタイルリゼーションを実現することが可能である。しかし、Nikolay ら [14] の提案から用いた学習データには優れた生成ができて、本研究において用いた学習データには適用することは難しい。

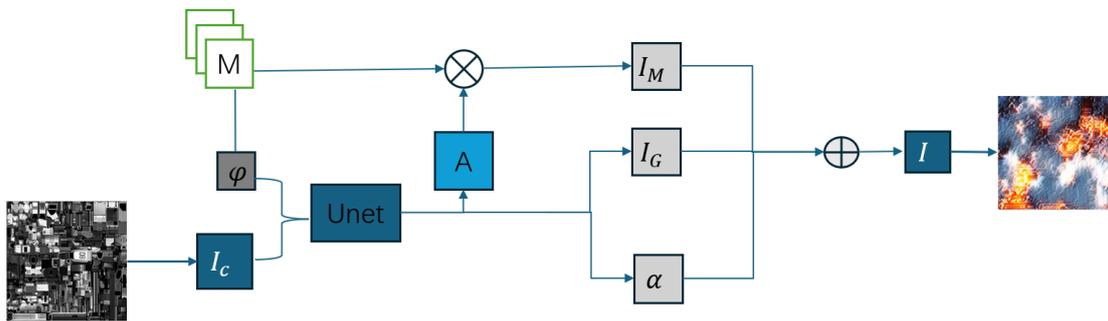


図 2.1 FAMOS のアーキテクスチャ

## 2.2 学習データ

本研究では UV テクスチャーのスタイル変換あるいは模様を変えるため FAMOS という手法を参考となった。しかし Nikolay ら [14] で提供した学習データではわかりやすい特徴がある。人物画像のデータが多数である。それらのデータを学習するに特化したモデルである故、そのまま UV テクスチャーを用いることは難しいともいえる。図 2.2 でロボット型 3D モデルの UV テクスチャーを示す。UV テクスチャーではある程度の前処理を行わないと生成に影響が出る予想が付けられるので、今回は学習データを二つの手法による事前にデータ処理を行う。まず Holger ら [22] が提案した XDoG と Canny ら [23] が提案した Canny 手法を用いて、学習データの前処理を行う。図 2.3 は XDoG によりエッジ抽出後の UV テクスチャーである。図 2.4 は RGB のチャ

ンネル数を保たれたグレースケールしたものである。これを用いて学習を行う。この処理を行う。これにより過学習を防ぐためである。モデルが関連性のない特徴やノイズの多い特徴があれば過学習につながる可能性が高くなるので、本研究に用いた UV テクスチャを以上の手法により学習データを前処理し、必要とする特徴を入力としてモデルに学習させる。

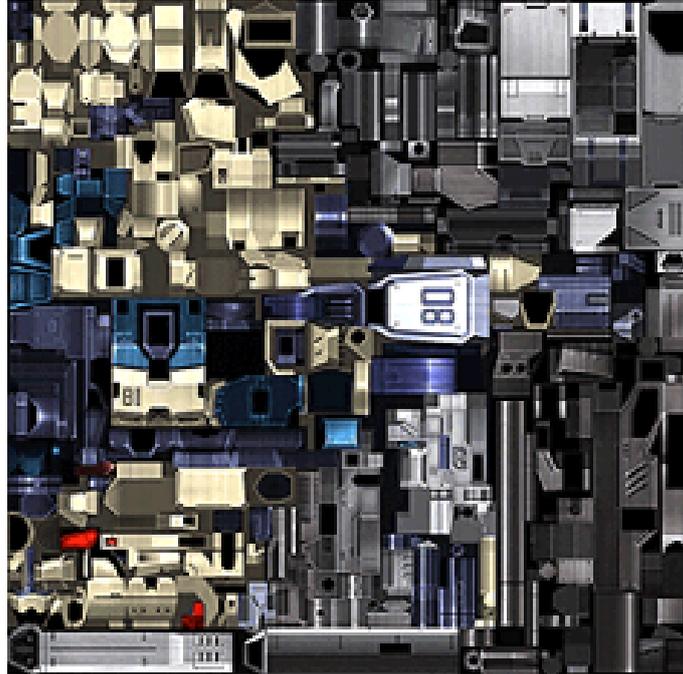


図 2.2 ロボット型 3D モデルの UV テクスチャ

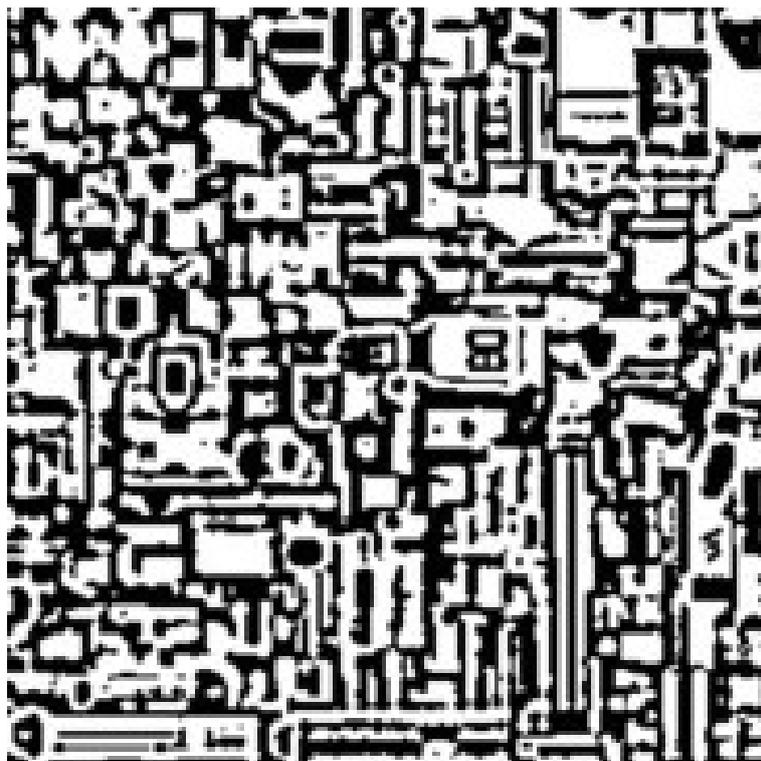


図 2.3 XDoG によりエッジ抽出後の UV テクスチャ

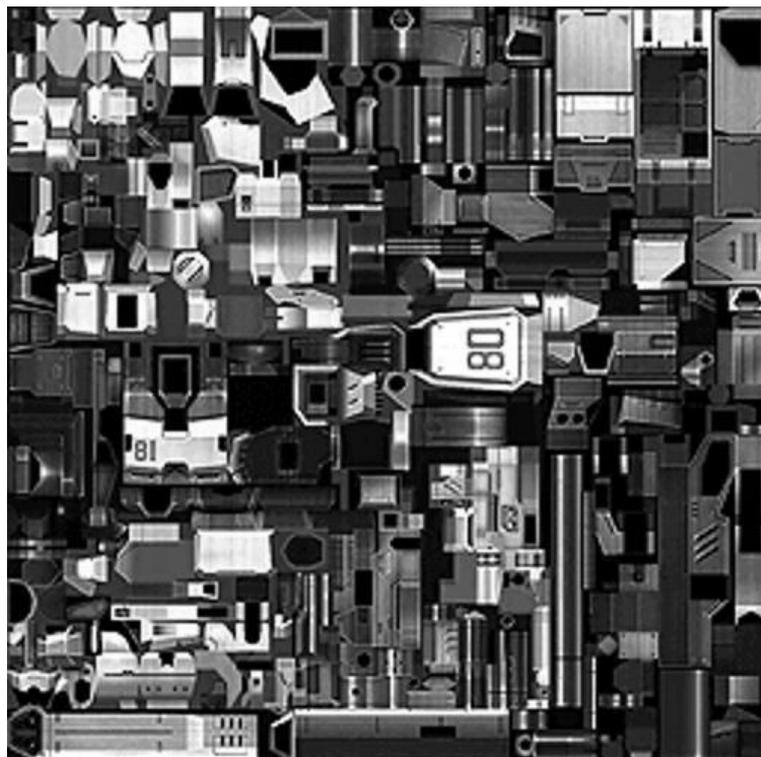


図 2.4 グレースケールした後の uv テクスチャ

## 2.3 スタイル生成の改善

検証段階では Nikolay ら [14] で提案した FAMOS 手法では UV テクスチャーに適応することは難しいと考えられ、モデルを UV テクスチャーに適応できる改善が課題となる。図 2.5 で FAMOS を用いたロボット型 3D モデルの UV テクスチャーの自動生成画像を示す。生成されたテクスチャーには重複スタイルが多く貼り付けられ、UV テクスチャーの区別の部分がわかりづらい。その中にまず一つ目のスタイルの重複である。図 2.6 で FAMOS で生成された UV テクスチャーを 3D モデルに貼り付けた結果、(左と右) 生成画像を貼り付けた結果、(中) オリジナルテクスチャーを示す。FAMOS で生成された結果はそのまま用いることは困難だと考えられる。今回使用したロボット型のオリジナルテクスチャーと比べ、FAMOS 手法による生成したテクスチャーではパーツの区切りが曖昧になり、ロボットのデザインコンセプトが損なわれてしまう。図 2.7、図 2.8、図 2.9 でそれぞれ生成テクスチャーとオリジナルテクスチャー詳細の部位の比較を示す。この現象になるのは Nikolay ら [14] 学習の段階でノンパラメトリックなメモリモジュールを用いるからだと考えられる。ノンパラメトリックなメモリモジュールにはスタイル画像からサンプリングされたテクスチャーやパターンの断片を格納することにより同じスタイル断片が複数回使用した可能性があると考えられる。スタイルの重複問題を改善するため、画像を分割処理し、部分的から学習を行うことで解決の一つとなる。そしてテンプレート画像を増やすことにもスタイル重複の問題にも改善できる。

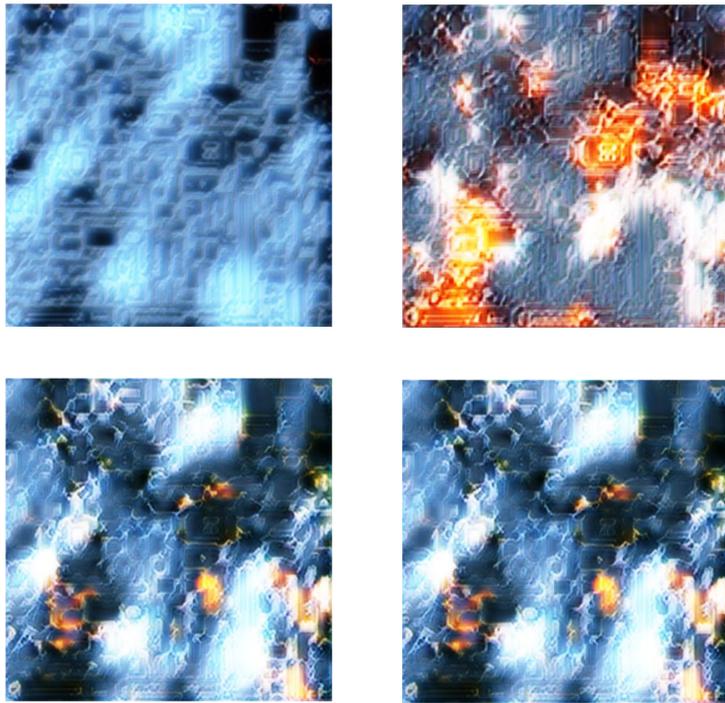


図 2.5 FAMOS を用いたロボット型 3D モデルの UV テクスチャの自動生成画像



図 2.6 FAMOS で生成された UV テクスチャを 3D モデルに貼り付けた結果、(左と右) 生成画像を貼り付けた結果、(中) オリジナルテクスチャ



図 2.7 生成テクスチャーとオリジナルテクスチャー頭部の詳細の比較



図 2.8 生成テクスチャーとオリジナルテクスチャー胸部の詳細の比較



図 2.9 生成テクスチャーとオリジナルテクスチャー武器の詳細の比較

## 2.4 入力と出力

Nikolay ら [14] 提案した FAMOS では従来の深層学習モデルと違い、コンテンツ画像とテンプレート画像二つの部分の学習がある。一枚のコンテンツ画像に対して複数枚のテンプレート画像を用いて、学習を行うが、そのままロボット型の UV テクスチャーを学習データとして利用できないことが問題点となる。図 2.5 は FAMOS を用いたロボット型 3D モデルの UV テクスチャーの自動生成画像である。改善するためにはデータセットを増やすことは重要となるが、検証段階では先行研究の深層学習モデルではコンテンツ画像の入力では複数枚を入力することができるとしても出力は一枚のコンテンツ画像しかない問題点がある。本研究ではコンテンツ画像を 9 枚に分割し、学習を行うが、先行研究のモデルをそのまま用いることはできないため、複数枚の入力と出力が望ましい、そのため、9 個のデータセットを設置し、学習を行う。図 2.10 で入力と出力を変更したアーキテクチャーを示す。

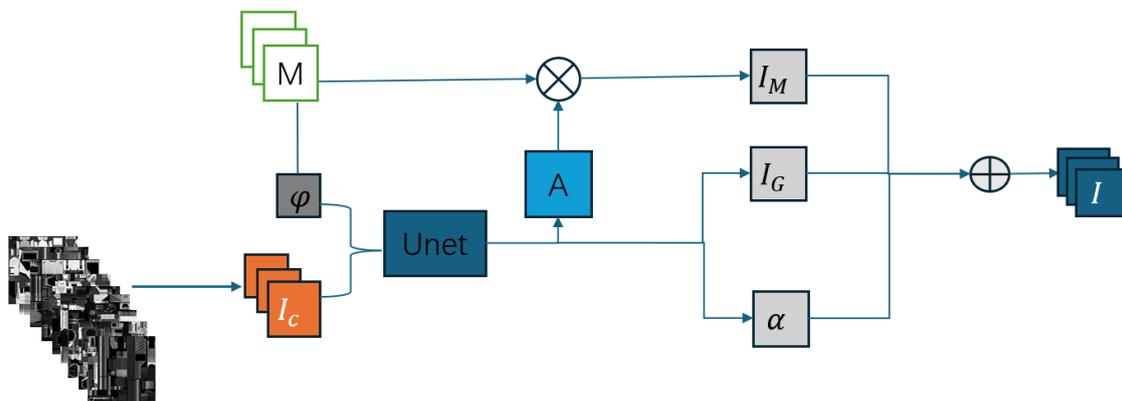


図 2.10 FAMOS を基づく入力と出力を変更したアーキテクスチャ

## 2.5 評価方法

生成型モデルあるいは生成型機械学習における評価方法はいくつにある。例え、Heusel ら [24] による FID という指標評価を提案し、主に生成された画像と実画像の分布の距離を計算し、モデルの性能を評価するが、FID は Christian ら [25] 開発した Inception v3 モデルの最後のプーリング層を用いたことにより、Inception V3 モデルでは主に人顔や物などの画像を主なデータセッ

トを使い、分類学習に用いたが、Nunn ら [26] の研究において FID を利用し、生成画像の評価も行った。結論としては FID は知覚的品質が低い画像に適切なスコアを割り当てない可能性がある」と論じた。本研究では UV テクスチャあるいは 3D モデル用のテクスチャ生成を目的とするので、それらの指標評価をそのまま用いることが困難である。

本研究で用いる評価はゲーム経験者と 3D モデルの制作者に生成したテクスチャとテクスチャを貼りつけた後の 3D モデルを示し、アンケートを答えてもらうの形で評価を行う。

## 第 3 章

# 実験

## 3.1 実験環境

表 3.1 は本研究における実験環境である。生成した uv テクスチャをゲームエンジン Unity による描画し、最後に評価を行う。

表 3.1 実験環境

開発環境	Google Colaboratory
GPU	NVIDIA A100
開発言語	Python
ライブラリ	Pytorch
ゲームエンジン	Unity

## 3.2 学習データの前処理

### 3.2.1 コンテンツ画像の分割

深層学習においては莫大なデータセットを用いて、学習を行うのが一般的である。例えば T Karras らの PGGAN や SYTLEGAN などの深層学習モデルの学習では 10 万枚以上高解像度データセットを用いることで高品質な画像生成を果たしたこともある。しかし分野の中にも医療画像の学習データは普段の車や人顔などの画像より極めて少ない場合もある。それらデータが少ない分野における深層学習モデルの学習では Mikołajczyk ら [27] が提案した画像処理の方法でデータ拡張を行うことで、モデルの精度を上げることもできると考えられるが、本研究では一つの 3D モデルに対して、テクスチャを生成するため、画像の回転や反転などのデータ拡張手法では用いることは効果が著しくない場合もあるため、画像をグレースケール処理やエッジ抽出処理した画像を分割処理する。図 3.1 と図 3.2 はそれぞれ図 2.3 で画像がエッジ抽出と図 2.4 はグレースケールした処理した後分割したコンテンツ画像である。9 枚を分割し、テクスチャーの部分的からスタイルを貼りつける。メリットとしては区分がわかりやすくなる。

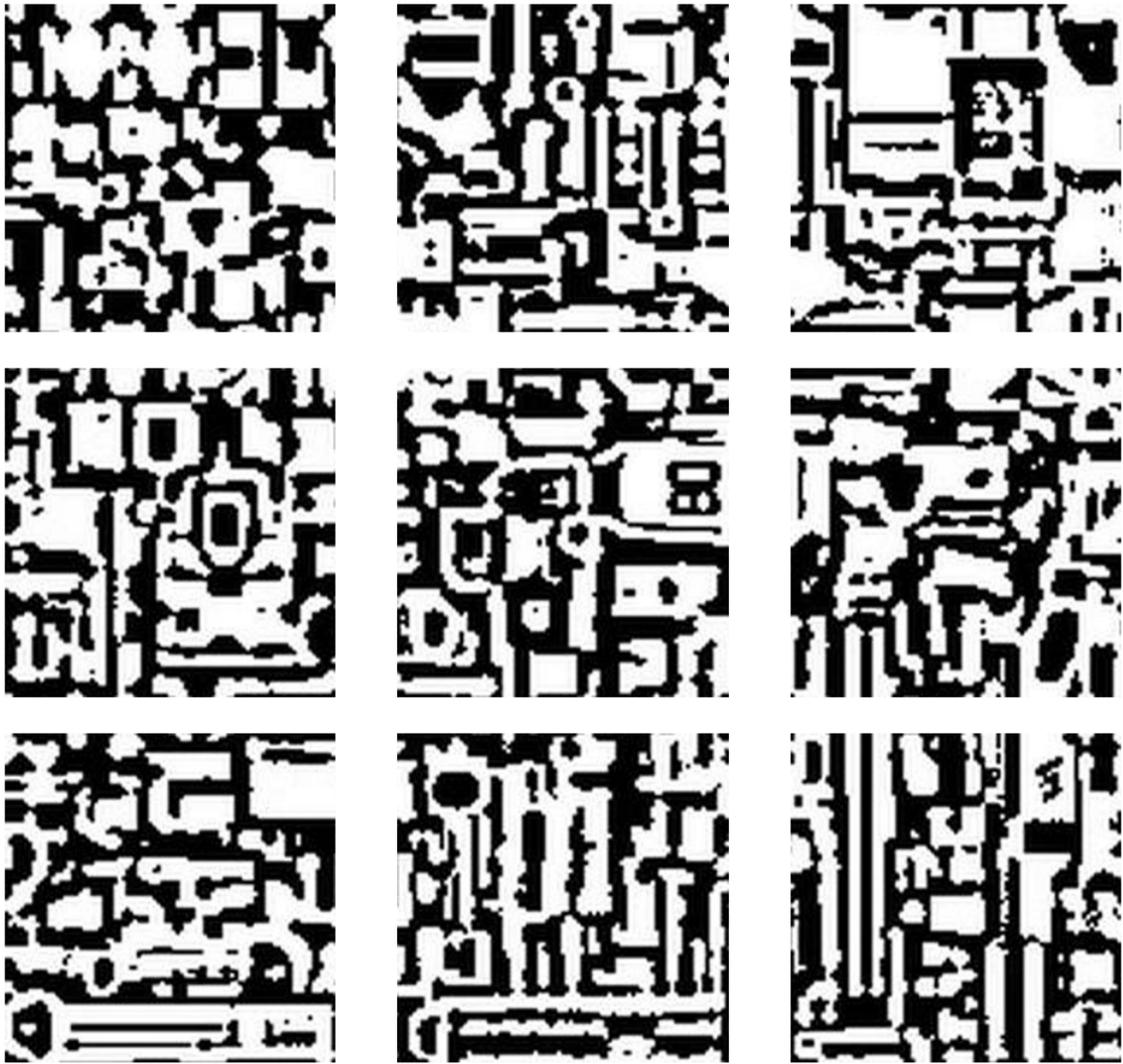


図 3.1 エッジ抽出後のテクスチャーを分割した後のテクスチャー

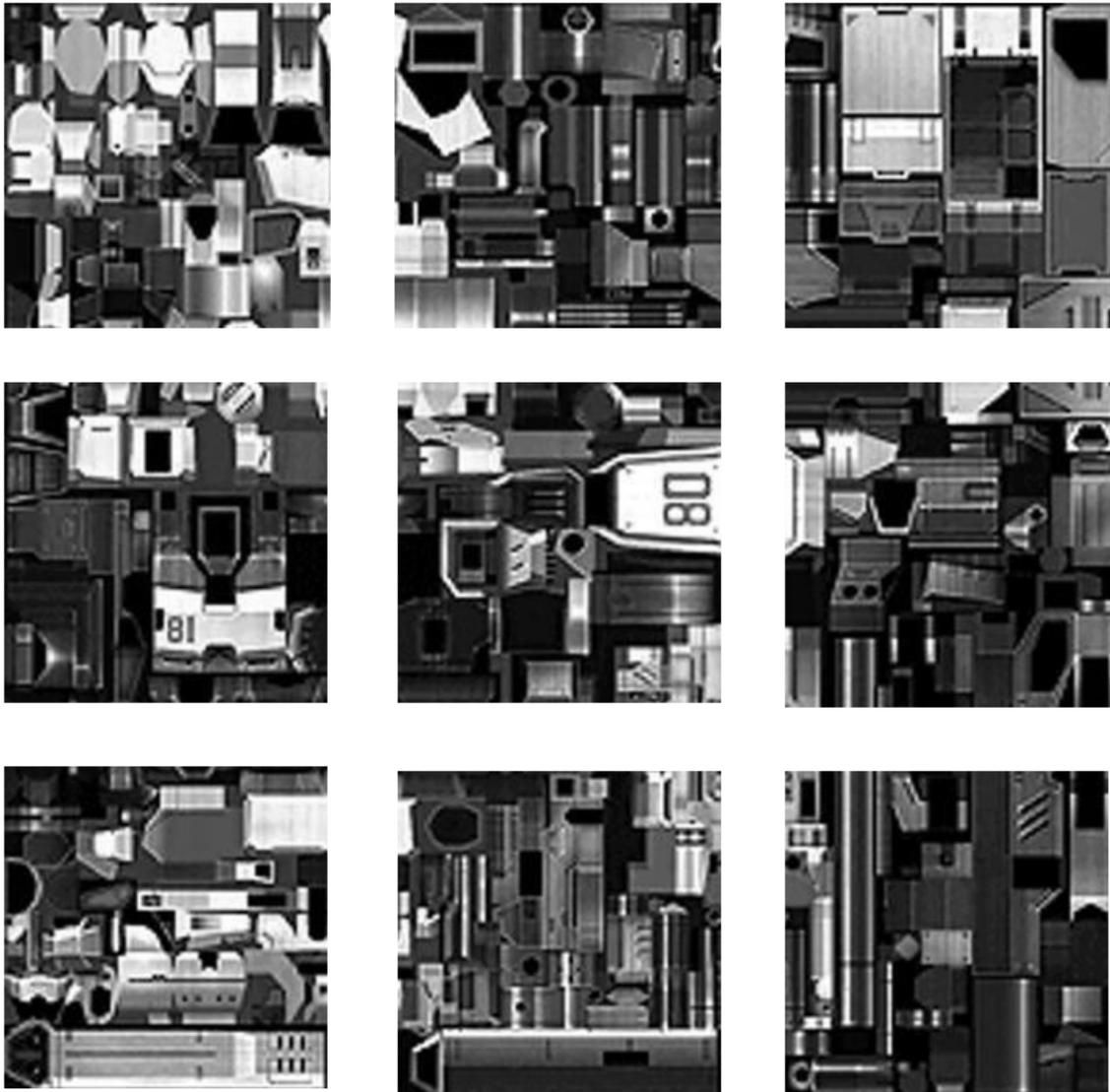


図 3.2 グレースケールした後のテクスチャーを分割した後のテクスチャー

### 3.2.2 テンプレート画像の分割

本研究で用いるテンプレート画像は主に画像からのスタイルをコンテンツ画像に貼りつくために存在する。特定あるいは主観的にスタイルを用いるため数の少ないテンプレート画像を用いる場合もあると考え、テンプレート画像にも分割処理を行う。もし一枚のテンプレート画像だけを用いる場合にはスタイルが重複されることが起こりやすくなる。学習を行うにはテンプレート画像にも分割処理をする。図 3.3 は分割処理したテンプレート画像である。



図 3.3 分割処理したテンプレート画像

### 3.3 パラメータの調整

本研究で用いるデータセットの独特性に対してはパラメータの調整も行う。今回使用したデータセットでは解像度の低いコンテンツ画像とテンプレート画像一枚ずつを用意しかなり少数なデータであり、データ拡張をした後にもかなり少数なデータでもある。それに対応するため、まず、畳み込み層を減ることで過学習のリスクを軽減する。式 (3.1) は目的関数であり、 $\lambda$  の値を常に 1 とする。これにより、元のテクスチャーあるいはコンテンツ画像に沿った画像を生成することができる。最後にエポックごとに学習量を減らす。学習量を減らすことにより、学習回数も減らす。図 3.4 と図 3.5 はエポックごとに学習回数 250 回とエポックごとに学習回数 10 回の場合の損失のグラフである。エポックごとに学習回数が上回ると過学習がしやすくなることがわかる。

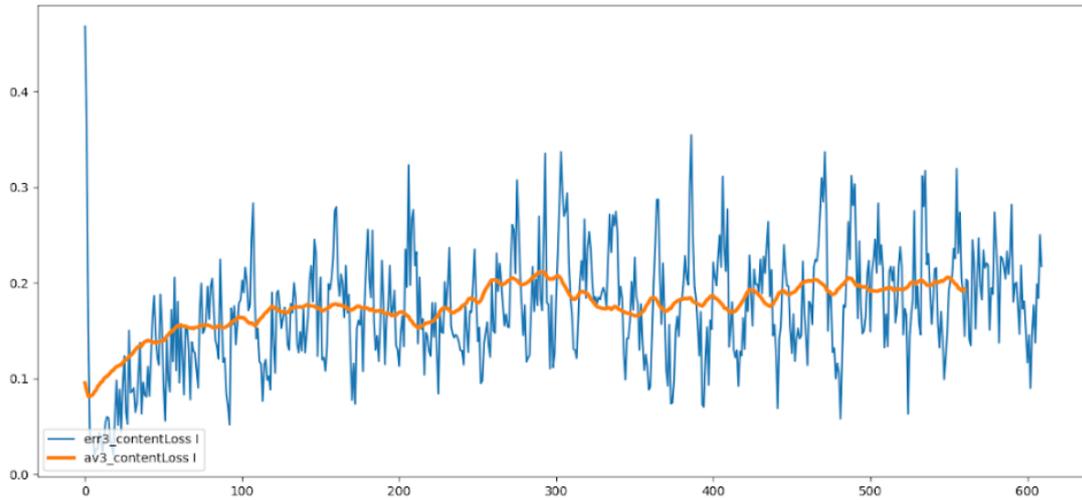


図 3.4 エポックごとに学習回数 250 回

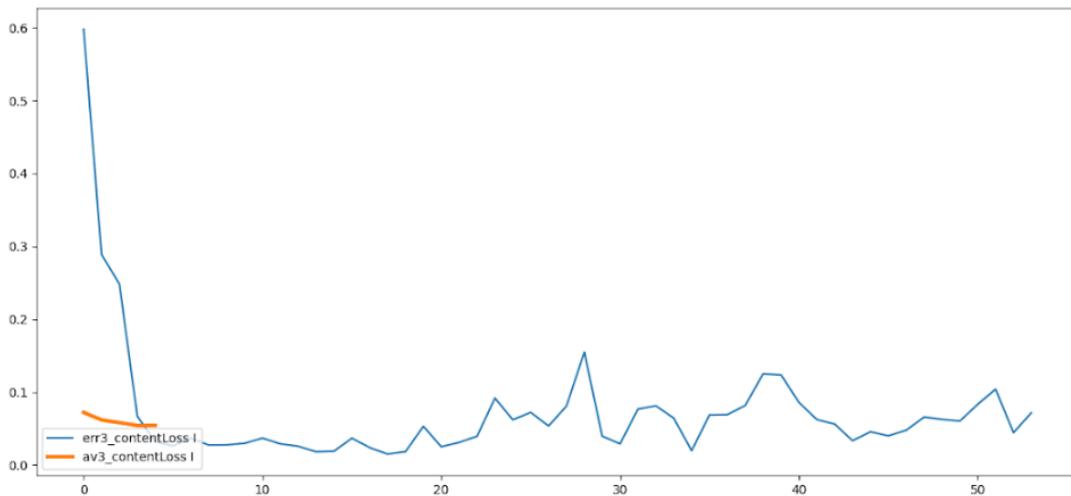


図 3.5 エポックごとに学習回数 10 回

$$G^* = \operatorname{argmin}(G)\operatorname{argmax}(D)L_{adv} + \lambda L_{content} \quad (3.1)$$

### 3.4 画像生成

本研究における画像生成では画像の分割手法と入出力の特性で、合成処理を行った。図 3.6 でグレースケール処理後の生成したロボット型 3d モデルのテクスチャーを示す。このグレースケール

ル処理後に生成されたテクスチャーを合成処理に用いる。図 3.7 は XDoG 処理後の生成したロボット型 3D モデルのテクスチャーである。この生成は代表的な二枚の UV テクスチャーを出力した。図 3.8 は Unity 上で描画した 3D モデル、(左) コンテンツ画像をグレースケール変換後の生成テクスチャー、(中) オリジナルテクスチャー、(右) コンテンツ画像をエッジ抽出後の生成テクスチャーを示すように生成した UV テクスチャーを用いて Unity 上で描画した模様である。図 3.9、図 3.10、図 3.11 でオリジナルテクスチャーと生成テクスチャーを 3d モデルに貼り付けた後の各部位の詳細を示す。このように本研究の手法による出力結果ではオリジナルテクスチャーに対して、各パーツの特徴を多く保ったまま、カラーリングを変更することは可能になった。図 3.8 は右にあるエッジ抽出後の生成テクスチャーを Unity 上で描画した画像は xDoG 手法によりカラーリングはグレースケールより色がより均一であるが、エッジ抽出により、輪郭、あるいは区分がグレースケールよりわかりずいところもある。またこれらの判断を付けるため、本研究ではアンケートの形でゲーム経験者や 3D モデルを制作した経験のある者にオリジナルテクスチャーと生成テクスチャーを用いた unity 上で描画した後の画像を用意し、点数評価を行う。

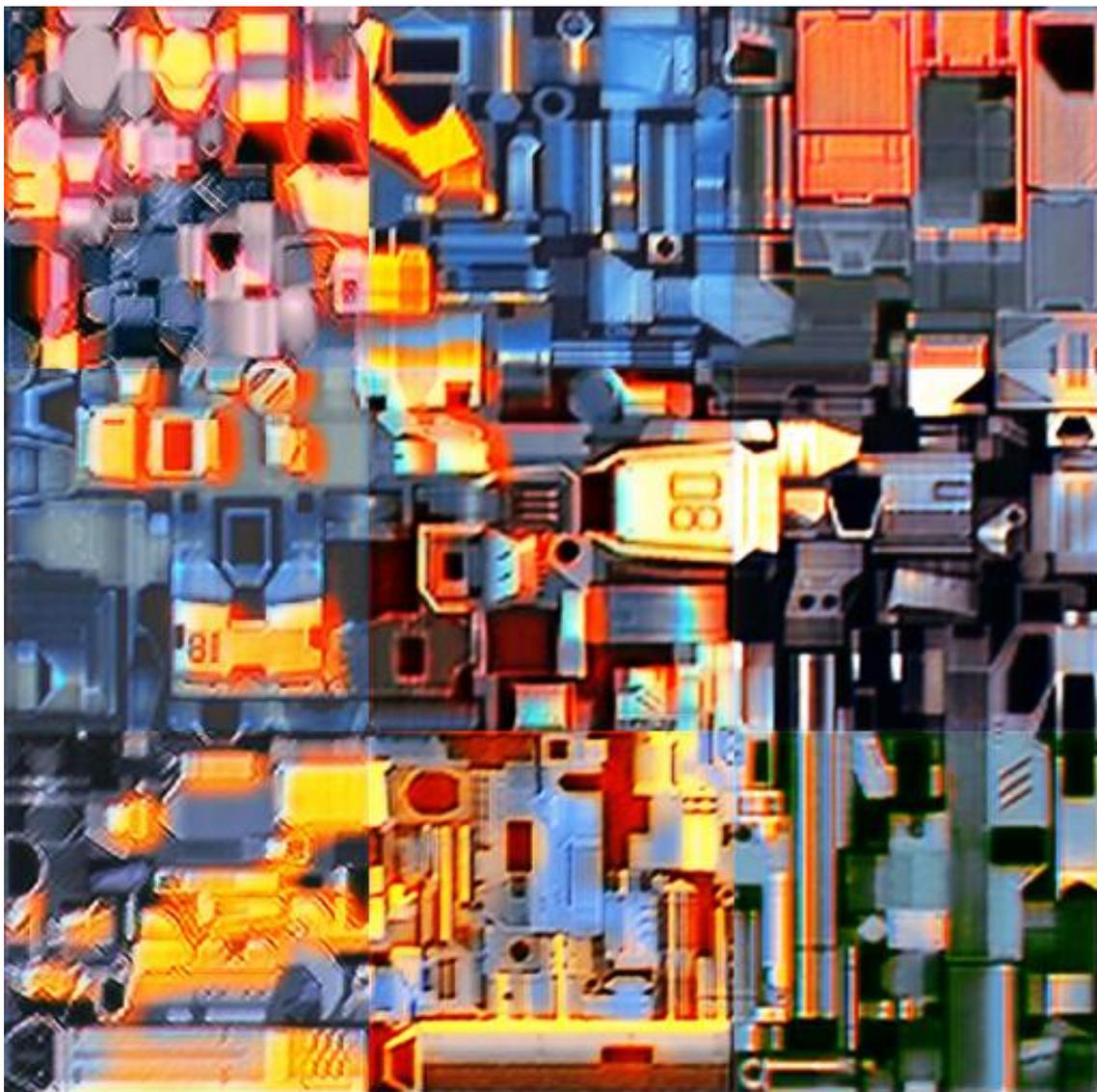


図 3.6 グレースケール処理後の生成したロボット型 3D モデルのテクスチャー

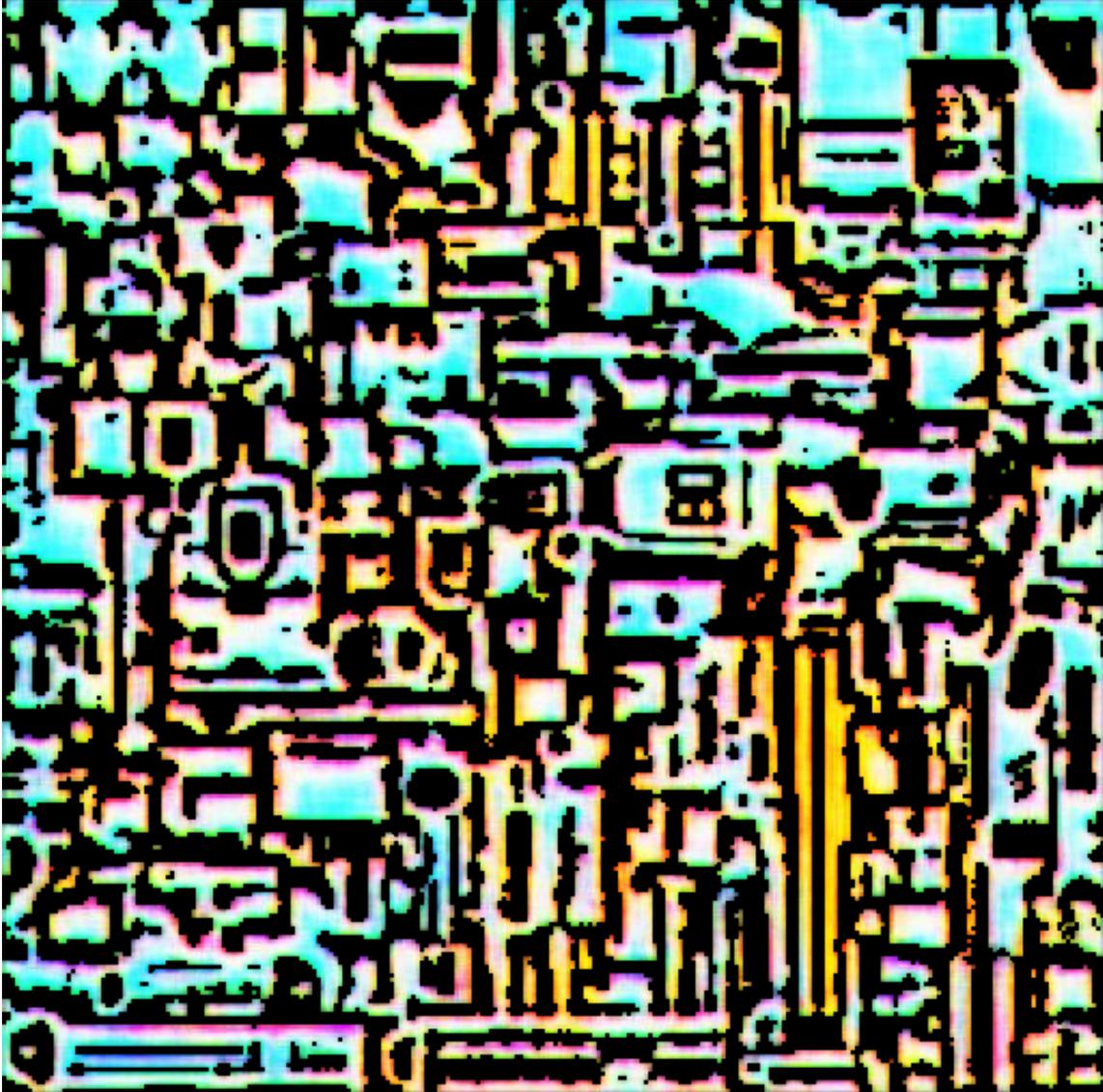


図 3.7 XDoG 処理後の生成したロボット型 3D モデルのテクスチャ



図 3.8 Unity 上で描画した 3D モデル、(左) コンテンツ画像をグレースケール変換後の生成テクスチャ、(中) オリジナルテクスチャ、(右) コンテンツ画像をエッジ抽出後の生成テクスチャ



図 3.9 オリジナルテクスチャと生成テクスチャを 3d モデルに貼り付けた後の頭部の詳細



図 3.10 オリジナルテクスチャと生成テクスチャを 3d モデルに貼り付けた後の胸部の詳細

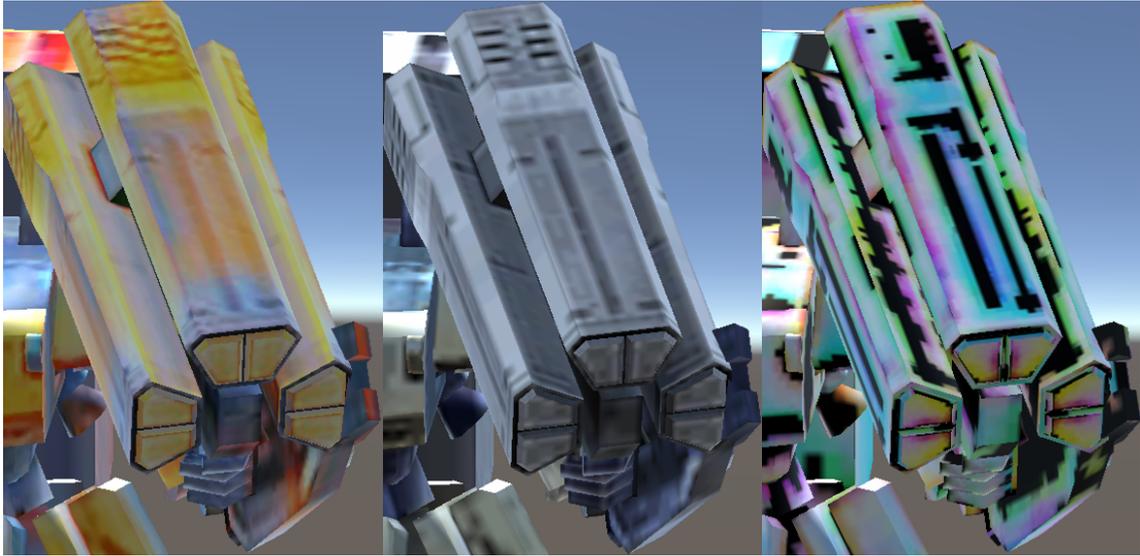


図 3.11 オリジナルテクスチャーと生成テクスチャーを 3d モデルに貼り付けた後の武器の詳細

### 3.5 評価

画像生成系深層学習は主に GAN のフレームワークを用いて様々な先行研究があったが、それらの生成画像に対して、指標評価を利用し、客観的な評価を行うことは多数存在する。また、Xu ら [28] の研究においては近年でよく用いられる指標評価を調査した。例えば Salimans ら [29] が開発した Inception Score は生成された画像の品質や多様性を評価するために多数の先行研究に用いられるが、すべての側面を完全にとらえることはできないと論じた。また Heusel ら [24] が開発した FID という指標評価では実際の画像と生成された画像を統計的な距離を計算するが、この指標評価は VGG16 を基づいで、評価が行うので、サンプルベースの原因で評価にも影響が出た。そして本研究では 3D モデルの UV テクスチャーを生成するため、普段の評価指標を用いることは困難であるため、アンケートの形で評価を行う。またアンケートについてはオリジナルテクスチャーと各手法を用いた生成された UV テクスチャーを 3D モデルに描画し、それぞれの画像に対して、どの程度パーツの特徴が保たれているかを回答者に 10 点満点で答えてもらった。図 3.12 と図 3.13 は本研究の手法により生成した UV テクスチャーを 3D モデルに描画した後の画像である。図 3.14 と図 3.15 は FAMOS 手法を用いて生成した UV テクスチャーを 3D モデルに描画した後の画像である。



図 3.12 本研究の手法を用いた生成したテクスチャーケース 1

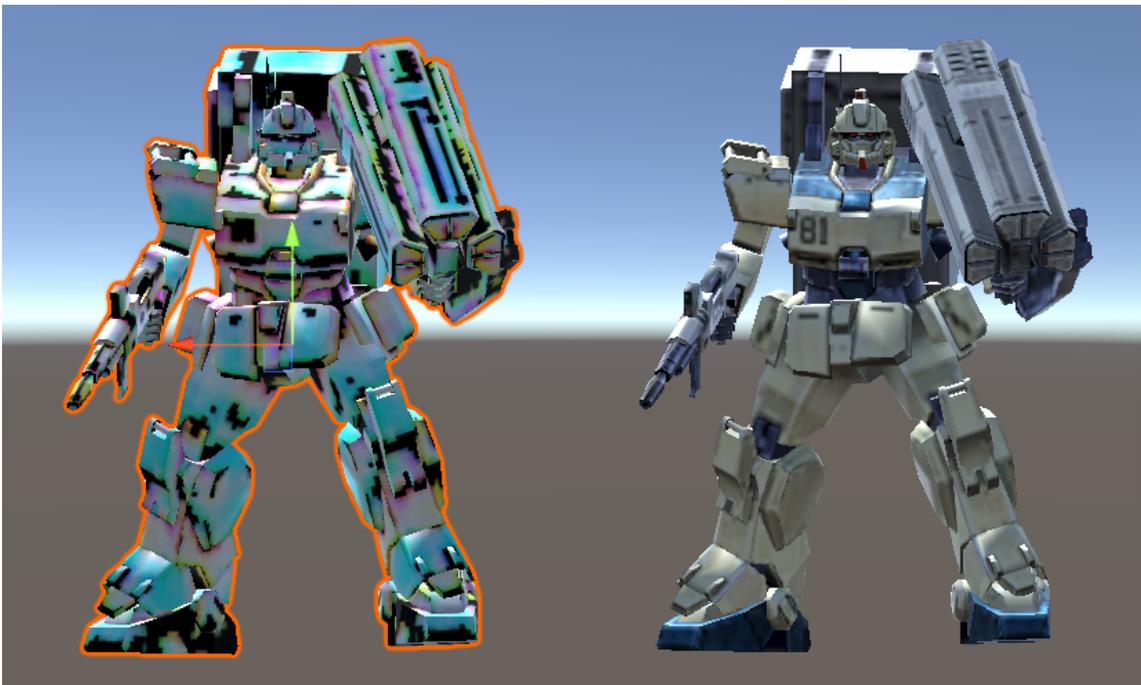


図 3.13 本研究の手法を用いた生成したテクスチャーケース 2



図 3.14 先行手法を用いた生成したテクスチャーケース 3



図 3.15 先行手法を用いた生成したテクスチャーケース 4

そして、アンケートによる収集したデータをエクセルにて計算する。表 3.2 は各ケースの回答点数である。ケース 1 とケース 2 は本研究の手法を用いた生成された画像を利用した。またケース

表 3.2 各ケースの回答点数各ケースの回答点数

ケース 1	ケース 2	ケース 3	ケース 4
8	3	1	3
8	6	2	2
8	10	6	5
10	8	7	6
9	8	7	7
8	6	3	4
8	3	0	0
9	8	7	6
10	9	8	7
8	4	3	3

表 3.3 t 検定によるアンケートデータの有意確率

有意確率	ケース 1	ケース 2
ケース 3	0.000170	0.000019
ケース 4	0.000594	0.001097

3 とケース 4 では FAMOS 手法そのまま用いた結果となる。ただし、アンケートによって取った点数を平均値だけで評価するには有意かどうかはわかりづらいと考える。馬場 [30] による論じた t 検定では多数の検定方式があり、場合による用いるべきだと論じた。今回は本研究の手法と先行研究の FAMOS 手法両方を用いて、生成した画像を比較するので、平均値の差の検定を用いることは正しいではないかと考えられる。これらのデータを用いて、t 検定を行い、データの有意差を判別する。今回実施する t 検定は一对の標本による平均の検定を用いた。まずケース 1 はケース 3 とケース 4 のデータをそれぞれ検定を行った。またケース 2 はケース 3 とケース 4 をそれぞれ t 検定を行った。表 3.3 で t 検定によるアンケートデータの有意確率の結果を示す。t 検定の結果を示したようにケース 3 とケース 4 はそのまま FAMOS 手法を用いた生成であるが、ケース 4 はケース 3 よりスタイルと区分が少し良くなっていることで、有意確率  $p$  の値も少し高くなる。

## 第 4 章

## まとめ

本研究では、ゲーム制作における 3D モデルを作るのに莫大な時間がかかる問題を着目した、制作の効率を上げるため深層学習手法を用いて、UV テクスチャーの自動生成の研究である。

従来の GAN の画像生成では画像のスタイルあるいは模様を変えたりすることが多く、学習データの中の見たことのない画像を生成することは目的となる。それに伴い、莫大な学習データを用いることになる。しかし本研究の研究対象は 3D モデルの UV テクスチャーであるため、莫大な学習データを用いることは難しいので、そこで、先行研究である FAMOS 手法では、GAN と U-net を用いて少ない学習データで画像のスタイルをコンテンツ画像に移行することができた。この特性を着目し、本研究の学習データ UV テクスチャーを用いて学習を行ったが、結果としてはスタイルの移行ができていても、3D モデルに生成したテクスチャーを描画した後パーツの特徴がほとんど失われている問題点とスタイルが単一すぎる問題点があった。原因としては、本研究を基づく先行手法 FAMOS では、ノンパラメトリックのメモリモジュールを用いて、速やかにテンプレート画像からスタイルをコピーし、コンテンツ画像に移る手法であるため、正規分布などの手法を違って、少ない学習データの中繰り返し、スタイルを用いることになってしまう。しかし正規分布や主成分分析などの統計的な分布を用いると多様かつ大量の学習データでないと、多くの特徴を捨ててしまい、うまく生成ができない問題点もあるため、本研究ではノンパラメトリックのデータ分布をそのまま利用し、解決策を考えた。このような問題を改善するため、本研究では、画像の分割手法と出力と入力、またはグレースケール処理と XDoG 処理を行って、はっきりとするパーツの特徴または豊かなスタイルを持つ UV テクスチャー生成ができた。つまり、意匠性がある程度に保たれた UV テクスチャーを生成した。そして、意匠性のあるかどうかを確認するため、評価にも必要となる。従来の研究に用いる評価方式は指標評価を用いることは多数であるが、指標評価におけるサンプルベースの原因で利用することは困難である。そのため、本研究ではアンケートの形で実際に生成した UV テクスチャーを 3D モデルに描画し、オリジナルの 3D モデルと並び、画面を見ながら回答者に 10 点満点で答えてもらった。回答した点数を  $t$  検定で有意差を計算し、結果を評価した。

今までの研究で生成した UV テクスチャーを 3D モデルに描画し、パーツの特徴をある程度に保たれた。しかし、入力的手法では一枚ずつの入力となり、スタイルはある程度豊かになったが、制御できないため、一枚だけの学習データを入力するよりは均一性がなくなってしまう。この間

題については今後の課題として検討する必要がある。また今後の展望としては、今回はロボット型のテクスチャーを対象として実験を行ってきたが、今後では人型やモンスターなどの UV テクスチャーを学習データとして、本手法の適用性を検討することも一つの課題となる。そして今回用いたロボット型の UV テクスチャーでは武器や頭部などのテクスチャーの並びは散乱であるが、もしこれらの部分を別々に分割して、学習を行うことにより、どのような結果があるのかも今後の課題になる。最後に本研究で用いる評価では回答者たち主観的な評価であり、結果としては本手法の有効性を示したが、他に客観的あるいは数学的な評価方法で評価することも課題の一つである。

# 謝辭

本研究を遂行するにあたり、多大なる支援と貴重な助言を頂いた先生方と研究室のメンバーたちに感謝を申し上げます。

修士2年間で、研究のご指導やご相談をしていただいた渡辺先生に感謝いたします。研究の最初段階では色々な不安を抱いて、中々うまくいかないところも多くありましたが、渡辺先生から貴重な助言やご指導を頂いて、研究方向にも明らかになりました。外国人留学生としてわかりづらい日本語があっても、いつも面白くわかりやすい言葉でご説明していただいて、学会投稿の時にも論文の添削やアドバイスなどのサポートを頂いた渡辺先生に心から感謝しております。

ポスター発表と予備審査にてご指摘いただいた。大淵先生、寺澤先生、飯沼先生にも感謝いたします。

修士2年間で、研究に関する助言や相談を頂いた阿部先生にも感謝いたします。最初の時研究方向に迷っていた自分を様々のご指導や助言を頂いて、とても感謝いたします。

研究室のメンバーに感謝いたします。最初に入学する時、色々わからないところがあり、親切に教えていただいた。先輩たちや同期に感謝いたします。修士一年生の時一緒に授業を受けてきた同期に色々助けをもらって、とても感謝しています。また研究に用いる3Dモデルを提供していただいた修士同期の高君にも感謝いたします。卒業した梁先輩にも感謝いたします。卒業しても深層学習について一緒に議論したり、わからないところを教えていただいたりして、とても感謝いたします。またいつも夜遅くまで一緒に作業してきた(修士一年の課題も)李君と高君にも感謝します。君らがいなければ、退屈で色々終わらないかと思う。本当にありがとう。

最後に支えてくれた両親に感謝します。留学しようとするわがままの自分へ8年間の留学生活を支えてくれて、感謝します。祖父にも感謝します。もし当時自分の留学計画を支持してくれなかったら、今までは歩けなかつたらう、本当にありがとう。

## 参考文献

- [1] 斉藤. ゲームにおける「スキン」とは？意味や課金要素としての側面を解剖！ <https://game-creators.jp/media/career/789/>, 2023.
- [2] Su Wanchao, Du Dong, Yang Xin, Zhou Shizhe, and Fu Hongbo. Interactive sketch-based normal map generation with deep neural networks. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, Vol. 1, No. 1, 2018.
- [3] 小林靖司. 3d モデルはどう作成する？手順やおすすめ作成ソフトを紹介！ <https://xrcloud.jp/blog/articles/business/2428/>, 2022.
- [4] Catmull and E. A subdivision algorithm for computer display of curved surfaces. 1974.
- [5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Advances in neural information processing systems*, Vol. 27, , 2014.
- [6] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [7] T Karras, T Aila, S Laine, and J Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.
- [8] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *Computer Vision and Pattern Recognition*, 2018.
- [9] Taigman Yaniv, Polyak Adam, and Wolf Lior. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.
- [10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *International Conference on Neural Information*, Vol. 1, pp. 1097–1105, 2012.
- [11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [12] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. pp. 779–788, 2016.

- [13] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *Computer Vision and Pattern Recognition*, 2015.
- [14] Nikolay Jetchev, Urs Bergmann, and Gokhan Yildirim. Copy the old or paint anew? an adversarial framework for non- parametric image stylization. *arXiv preprint arXiv:1811.09236*, 2018.
- [15] 山崎聡. 3d キャラクターに色を塗るにはどうすればいい？ まずは 3d モデルを uv 展開. <https://forest.watch.impress.co.jp/docs/serial/blenderwthing/1499507.html>, 2023.
- [16] Unity Documentaion. テクスチャ. <https://docs.unity3d.com/ja/2018.4/Manual/class-TextureImporter.html>, 2017.
- [17] Frederick M. Weinhaus and Venkat Devarajan. Texture mapping 3d models of real-world scenes. *ACM Computing Surveys*, Vol. 29, pp. 325–365, 1997.
- [18] Y. Takase, N. Sho, A. Sone, and K. Shimiya. Automatic generation of 3-d city models and related applications. *ISPRS*, 2003.
- [19] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, Daniel Cohen-Or, and Daniel. Texture: Text-guided texturing of 3d shapes. *arXiv preprint arXiv:2302.01721*, 2023.
- [20] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, K. Yin, Daiqing Li, O. Litany, Zan Gojcic, and S. Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, Vol. 35, pp. 31841–31854, 2022.
- [21] Henderson Paul, Tsiminaki Vagia, and Christoph H. Lampert. Leveraging 2d data to learn textured 3d mesh generation. 2020.
- [22] Holger Winnemoller, Jan Eric Kyprianidis, and Sven C. Olsen. Xdog: An extended difference-of-gaussians compendium including advanced image stylization. *Computers and Graphics*, Vol. 36, No. 6, pp. 740–753, 2012.
- [23] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, No. 6, pp. 679–698, 1986.
- [24] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp

- Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, Vol. 30, , 2017.
- [25] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. pp. 2818–2826, 2016.
- [26] Eric J Nunn, Pejman Khadivi, and Shadrokh Samavi. Compound frechet inception distance for quality assessment of gan created images. *arXiv preprint arXiv:2106.08575*, 2021.
- [27] Mikołajczyk, Agnieszka, and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. *2018 international interdisciplinary PhD workshop (IIPhDW)*, pp. 117–122, 2018.
- [28] Xu Qiantong, Huang Gao, Yuan Yang, Guo Chuan, Sun Yu, Wu Felix, and Weinberger Kilian. An empirical study on evaluation metrics of generative adversarial networks. *arXiv preprint arXiv:1806.07755*, 2018.
- [29] Salimans Tim, Goodfellow Ian, Zaremba Wojciech, Cheung Vicki, Radford Alec, and Chen Xi. Improved techniques for training gans. *Advances in neural information processing systems*, Vol. 29, , 2016.
- [30] 馬場真哉. t 検定の考え方. <https://logics-of-blue.com/t-test/>, 2020.

# 発表実績

## ポスター発表

- チンエキトウ, 阿部雅樹, 渡辺大地, 深層学習を用いた壁のテクスチャーの自動生成, 映像表現・芸術科学フォーラム, 2022.

## 口頭発表

- チンエキトウ, 阿部雅樹, 渡辺大地, 深層学習を用いた 3D モデルのテクスチャーの自動生成, 日本デジタルゲーム学会, 2023.