LLM を利用したテキスト指示による物体のレイアウト提案システム

中瀬世士 [1]　岡部誠 [1]

概要：テキストによる指示を用いて 3D モデルを自動で配置するシステムを提案する．ユーザは，3D モデリングソフトウェアを用いて配置したいオブジェクトの 3D モデルを作成し，それを仮想環境に入力する．次に，「円形に並べる」などの配置指示をテキストで入力することで，システムによりその指示を含んだプロンプトが LLM である ChatGPT に送信され，応答として 3D モデルの座標と回転を取得し，これが仮想環境に反映されることで自動的な配置が行われる．本システムを利用することで，現実でのオブジェクトの移動をせずに仮想環境において配置のシミュレーションができること，従来は手動による操作のために難しかった 3D モデルの配置を簡単にかつ自動的に行うことができることを示す．

キーワード：大規模言語モデル，プロンプト設計，3D モデル配置

1. はじめに

近年，3D モデリングは建築，エンターテインメント，教育などさまざまな分野で重要な役割を果たしており[1，2，3]，特に複数のオブジェクトを効率的に配置するレイアウト設計は課題として注目されている．また，現実世界で「身の回りの物をおしゃれに並べて居心地のいい部屋にしたい」といったような身近なオブジェクトのレイアウト設計においては，3D モデリングを活用することで実際のオブジェクトを動かす必要がなく，オブジェクトの重さや試行錯誤の過程でオブジェクトに傷がつくことなどを考慮せずに済むという利点がある．しかし，一般的なモデリングソフトウェアやゲームエンジンでは，3D モデルの配置には手動のドラッグ操作が必要であり，複雑な指示に基づく配置は高度なスキルを要求される．そのため，多様な配置を試行錯誤するには多くの時間がかかってしまう．

既存研究では，室内における家具の配置を自動生成する手法が提案されている[4，5，6，7]．しかし，これらの手法は生活するうえでの過ごしやすさや動線の確保など家具の配置に最適化されており，対象のオブジェクトは家具であるため，より一般的なオブジェクトの配置への適用は困難である．

本研究ではこれらの課題を解決するため，LLM である ChatGPT を活用し，テキストを用いた簡単な指示で 3D モデルを自動配置できるシステムを提案する．図 1 に本システムの概要図を示す．本システムでは，ユーザが「オブジェクトを円形に並べる」などのテキスト指示を入力することで，3D モデルの座標や回転が自動で計算され，その配置が仮想環境である Unity 上に反映される．具体的には，Metashape などの 3D モデリングソフトウェアを用いて生成した 3D モデルを Unity に入力し，ChatGPT API を用いて配置を決定する．本システムは，入力された 3D モデルから，その座標とバウンディングボックスの大きさを取得し，これらとユーザによるテキスト指示，および出力フォーマットの例などを含む注意事項が書かれたプロンプトを ChatGPT API への入力とし，応答としてテキスト指示に基づいた 3D モデルの座標と回転を受け取る．このアプローチにより，ユーザは直感的なテキスト指示のみで，柔軟かつ効率的に 3D モデルの配置を行うことが可能となる．また，これらの配置は従来の室内の家具の配置のみにとどまらず，ゲームなどの仮想環境内での建物の配置から現実でのフィギュアやぬいぐるみの配置まで，オブジェクトの配置という点において，本システムを様々な用途で汎用的に用いることができる．

1 静岡大学
Shizuoka University

2. 関連研究

2.1 3D モデルの自動配置

仮想環境における 3D モデルの自動配置に関する研究は，主に建築設計やメタバース・ゲーム開発の分野で行われている．Sun らの研究では，初期状態として現在の家具の配置を含む室内シーンを入力すると，強化学習により人が快適に過ごすことを目的として最適化された家具の配置を含む室内シーンを出力するシステムを提案している[6]．また，Feng らの研究では，部屋の種類やサイズと使用可能な家具のリストを入力として LLM を利用した家具の配置を出力するシステムを提案している[7]．これらはいずれも室内の家具の配置を決定することに特化しており，家具以外のオブジェクトの配置の決定に適用するのは難しい．よって，本システムではオブジェクトの種類や大きさ，環境などによらず汎用的に使用可能なオブジェクトの配置を提案することを目指す．

2.2 テキスト指示による LLM の活用

近年，LLM の発展により ChatGPT のような自然言語処理技術を活用したインタラクティブなシステムの開発が進められている[7，8，9，10，11]．その中でも，テキスト指示による画像編集を行うシステムとして Zhang らの SGEdit が提案されている[12]．SGEdit は，ChatGPT と画像生成

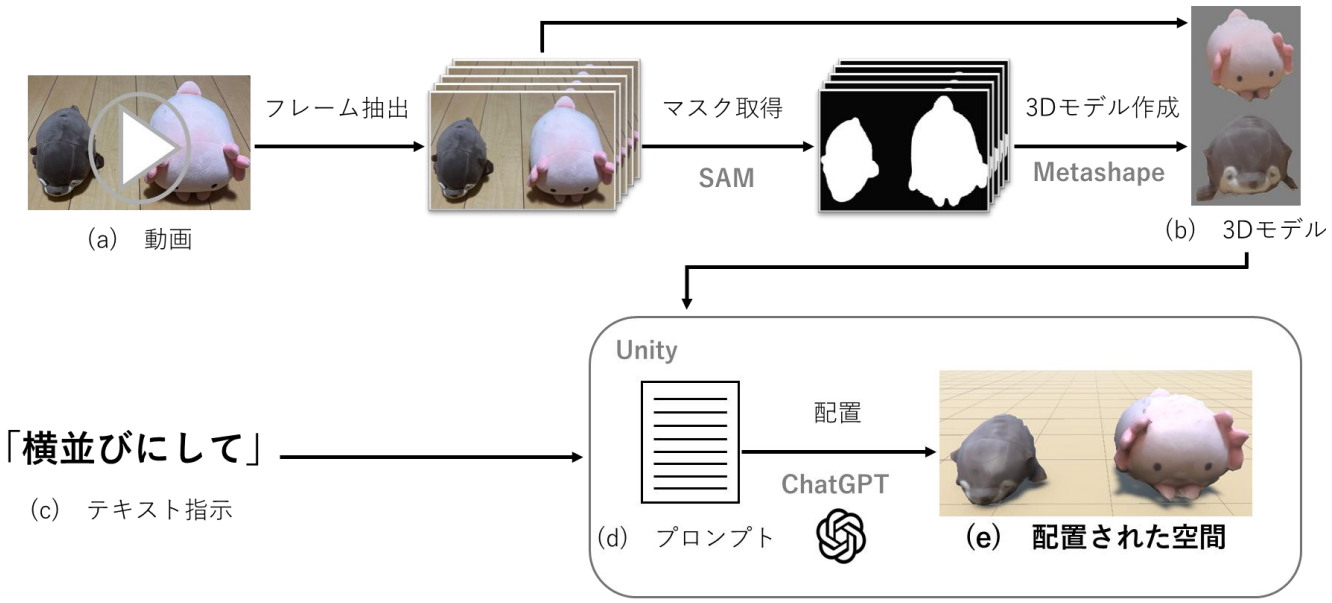


図 1　提案システムの概要図．ユーザは配置したいオブジェクトを 360 度から撮影した動画(a)をシステムに入力してフレーム抽出とマスクの取得を行い，それらを Metashape に入力して 3D モデル(b)を作成し，配置を行う仮想環境である Unity に入力する．そして，Unity 内の入力欄に配置指示を書いたテキスト(c)を入力して実行することで，そのテキスト指示がプロンプト(d)に加えられ，それが ChatGPT に送信されて応答として 3D モデルの座標と回転が返ってくる．さらに，システムがそれらの情報を 3D モデルに反映することで，指示に沿った配置がされた空間(e)が出力される．これにより，ユーザはテキストによる指示で 3D モデルの配置を行うことができる．

AI である Stable Diffusion[13]を活用し，画像内のオブジェクトの位置関係を表したシーングラフを生成・編集することで，オブジェクトの消去や追加，移動を行うシステムである．SGEdit のシステムでは，ChatGPT に対して画像と図 2 のようなプロンプト（画像解析のための質問，出力フォーマットの例，注意事項を含む）を入力とし，出力としてシーングラフと各オブジェクトの詳細な注釈を得る．その後，得られた情報を基に既存のセグメンテーションモデルを使用してオブジェクトのマスクを作成し，Stable Diffusion をファインチューニングして利用することで，画像の編集を実現している．これにより，ユーザはシーングラフを編集するだけで画像内のオブジェクトを直感的に操作し，画像編集をすることができる．

SGEdit の研究から，ChatGPT を利用することで直感的操作が可能になること，適切なプロンプト設計により出力結果の一貫性を高める工夫を取り入れることが有効であることがわかる．これらのことを踏まえ，本研究では SGEdit のアプローチを参考にしながら，ChatGPT を用いた，テキスト指示に沿って 3D モデルを自動で配置するシステムを設計する．

```
# Example
- Example 1:
                    :
Question2: What are the main instances in this image?
Answer2: man in red shirt, man in gray jacket, sidewalk· ·
Question3: what is the relationship between the main
  instances? Given me the answer in scene graph format.
Answer3:
      1. Man in red shirt -> standing on -> Sidewalk
      2. Man in gray jacket -> standing on -> Sidewalk
                    :
# Guidelines for answering the Question2.
1. Excluding accessories; Excluding accessories and
detachable parts (such as saddles, bridles, ropes, helmets, · ·
                    :
Now answer the Question2 and Question3 following on the
guidelines'"
```

図 2　SGEdit のプロンプトの例．上から順に出力フォーマットの例，注意事項，ChatGPT への指示が含まれる．ChatGPT に一定の流れで考えさせるため，また出力フォーマットを統一するためにその例が記載されている．真ん中の注意事項では，Question2 に答える際に ChatGPT に守らせるべきルールを記載している．これらをプロンプトに含めることでシーングラフを安定して出力する．

3. 提案手法

3.1 システムの概要

本研究では，テキストによる指示に基づいて，3D モデルを自動的に配置するシステムを提案する．ユーザはまず配置したいオブジェクトを 360 度から撮影した動画をシステムに入力し，フレーム抽出とマスクの取得を行い，それら

を Metashape に入力して 3D モデルを作成する．次に，作成した3Dモデルと配置のテキスト指示をUnityに入力し実行すると，システムは我々が作成したプロンプトに配置のテキスト指示を追加して ChatGPT API に送信し，応答としてテキスト指示に沿った配置に適切な座標と回転を取得し，3D モデルが自動で配置される．

3.2 3D モデルの作成

本研究では，フォトグラメトリーに基づく 3D モデリングソフトウェアである Metashape[14]を用いて 3D モデルを作成する．このとき，オブジェクトを撮影した動画のみをMetashape に入力して3D モデルを作成することもできるが，床の除去などといった手作業を減らすため，先にオブジェクトのマスクを取得し，各フレームとペアとなるマスク画像を Metashape に入力する．その後，一定の流れに沿ったいくつかのステップを経ることで 3D モデルを作成することができる．本節では，Metashape を用いた 3D モデルの作成について説明する．

まず，ユーザはオブジェクトを 360 度から撮影した動画をシステムに入力する．するとシステムは，フレーム数が 100 枚程度になるようにフレーム抽出を行い，既存のセグメンテーションモデルである Segment Anything[15]を用いて床が黒，床以外が白となるようにマスクを取得する（図 3）．ここで，オブジェクト自体ではなく床のマスクを取るのは，3D モデル作成の際に主に手作業で削る必要があるのは床の部分であり，オブジェクトは様々な種類があるためそれらをすべて指定してマスクを取るのは難しいが，床を指定してマスクを取るのは容易だからである．よって，床のマスクを取ってそれを 3D モデル作成の際に除去するようにすれば，手作業の時間や労力を大幅に減らすことができる．マスクの取得後，ユーザは先ほど抽出したフレームとマスクを Metashape に入力する．

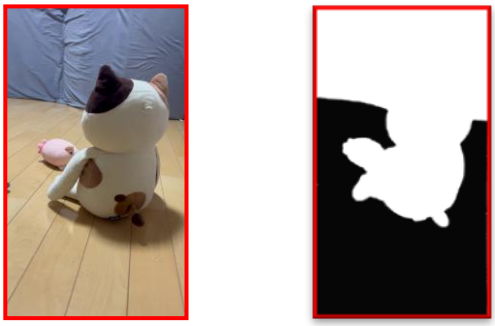
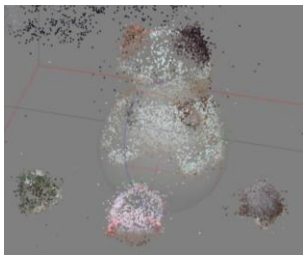
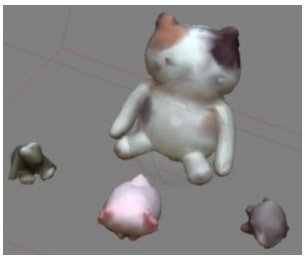


図 3　フレーム（左）とそのマスク（右）の例．マスクが見えやすいように画像を赤枠で囲っている．床が黒で，床以外が白になっていることがわかる．

次に，抽出した画像のアラインメントを行う．アラインメントとは，複数枚の画像の共通点を検出し，それらの位置関係を求める処理である．これにより，各画像に写るオブジェクトの 3D 座標が推定され，図 4 (a)のような疎な点群（タイポイント）が得られる．これはオブジェクトの大まかな形状を示す．その後，タイポイントからより密度の高い点群を作成し，図 4 (b)のようなポリゴンメッシュ（3D 形状）を生成する．この時点でオブジェクト以外の無駄な部分があれば，投げ縄ツールを使ってその部分を囲んで消すなどして手作業により省く．最後に，これに対して撮影した画像を基に，テクスチャを適用することで図 4 (c)のように 3D モデルを作成する．作成した 3D モデルを出力し，Unity 内のプロジェクトに入力することで Unity 上での 3D モデルの配置や操作が可能となる．



(a)　タイポイント



(b)　ポリゴンメッシュ

(c)　3D モデル

図 4　Metashape での 3D モデルの作成例

3.3 オブジェクトの配置プロセス

本研究では，作成した 3D モデルを Unity に入力し，ユーザの指示に基づいてオブジェクトを自動で配置する．オブジェクトの配置は，ChatGPT API（GPT-4o）を用いた自然言語処理により，座標と回転を取得し，Unity 上に反映させることで実現する[16]．本節では，Unity への 3D モデルの入力から配置までのプロセスについて説明する．

まず，作成した 3D モデルを Unity に入力する．Unity は物理エンジンを備えており，オブジェクト同士の衝突や重力の影響をシミュレーションできる[17]．本研究の時点では，ChatGPT の計算結果をもとに直接オブジェクトを配置するが，第 5 章で述べる今後の発展として，安定性も考慮したオブジェクトの配置も可能と考え，仮想環境として Unity を採用した．次に，空のオブジェクトを 2 つ作成し，1 つはオブジェクトの情報を管理しやすくするために入力した全ての 3D モデルの親オブジェクトとし，もう 1 つは我々が作成したスクリプトをアタッチする．このスクリプトを実行することで，以下のプロセスで配置が行われる．

① 3D モデルの情報（名前，座標，バウンディングボックスの大きさ）を取得する

② 3D モデルの情報とユーザによるテキスト指示，出力フォーマットの例，注意事項を含めたプロンプトを ChatGPT API へ送信する
③ ChatGPT API から 3D モデルの新しい座標と回転を含めた応答を得る
④ ③で得た座標と回転を反映する

これによりユーザは，従来の手動によるドラッグ操作の必要がない，テキスト入力のみによる操作で 3D モデルの配置が可能となる．

3.4 プロンプト設計

本システムでは，ChatGPT にプロンプトを送信し，その応答として 3D モデルの配置を得る．ゆえに，適切な配置を得るためにはプロンプト設計が重要である．本節では，具体的なプロンプトの構造や設計方針について説明する．

ChatGPT に送信するプロンプトは，3D モデルの情報，配置指示，出力フォーマットの例，および注意事項で構成される．これは，図 2 の SGEdit のプロンプト設計を参考にしている．以下に，プロンプトの要素を示す．また，図 5 に各要素の例を示す．

```
"objects": [
    {
      "name": "Chicken",
      "center": [
        3.1996038,
        -1.1920929E-07,
        0.5737834
      ],
      "size": [
        0.469131,
        1.0,
        1.0
      ]
    },
（以降同様の形式で
全オブジェクト分記載）
]
```

(a) 3D モデルの情報

道路沿いに並んでいる建物のように，等間隔に直線上に横並びにして．計算して各オブジェクトの center と rotation を教えて．

(b) テキスト指示

・center はオブジェクトの中心の座標を示し，size はオブジェクトの各軸の方向のバウンディングボックスの大きさを示す．
・オブジェクトどうしがぶつかったり貫通したりしないようにして．
（以降続く）

(c) 注意事項

図 5 プロンプトの要素の例

3D モデルの情報：3D モデルの名称，座標，およびバウンディングボックスの大きさを JSON 形式で示す．

配置指示：ユーザが希望する配置をテキストで入力する．

出力フォーマットの例：ChatGPT の出力の一貫性を確保するため，正しい出力フォーマットの例を提示する．これにより，スクリプト内での，ChatGPT からの出力に含まれる 3D モデルの情報の扱いが容易になる．このフォーマットは，図 5 (a)の”size”を”rotation”にしたものと一致する．

注意事項：配置を正しく行うため，またフォーマットに則った出力を行うために ChatGPT に守らせるべきルールを明示する．これにより，システムが安定したパフォーマンスを発揮する．スペースの都合上，図 5 (c)にはその一部を示す．

4. 実験

本章では，まず 4.1 節にて現実のオブジェクトと本システムを利用して配置された 3D モデルのオブジェクトの様子を比較することによって，本システムの結果が現実での様子と近く，現実のオブジェクトの配置のイメージをすることに利用できることを示す．次に 4.2 節にて，手動での配置と本システムによる自動での配置との時間と見た目を比較し，本システムの利用により効率的に多くの配置を試すことができることを示す．

4.1 作成した 3D モデルを用いた自動配置

Metashape を用いてオブジェクトを 360 度撮影した動画から 3D モデルを作成して Unity に入力し．さらにテキスト指示を入力して実行することで自動配置を行った．なお，配置は現実のオブジェクトを映した図 6（左）と同様になるようにテキスト指示を入力した．その結果とテキスト指示をそれぞれ図 6，図 7 に示す．

図 6 現実のオブジェクトとシステムを利用して配置した 3D モデルとの比較．（左：現実のオブジェクト，右：3D モデル）

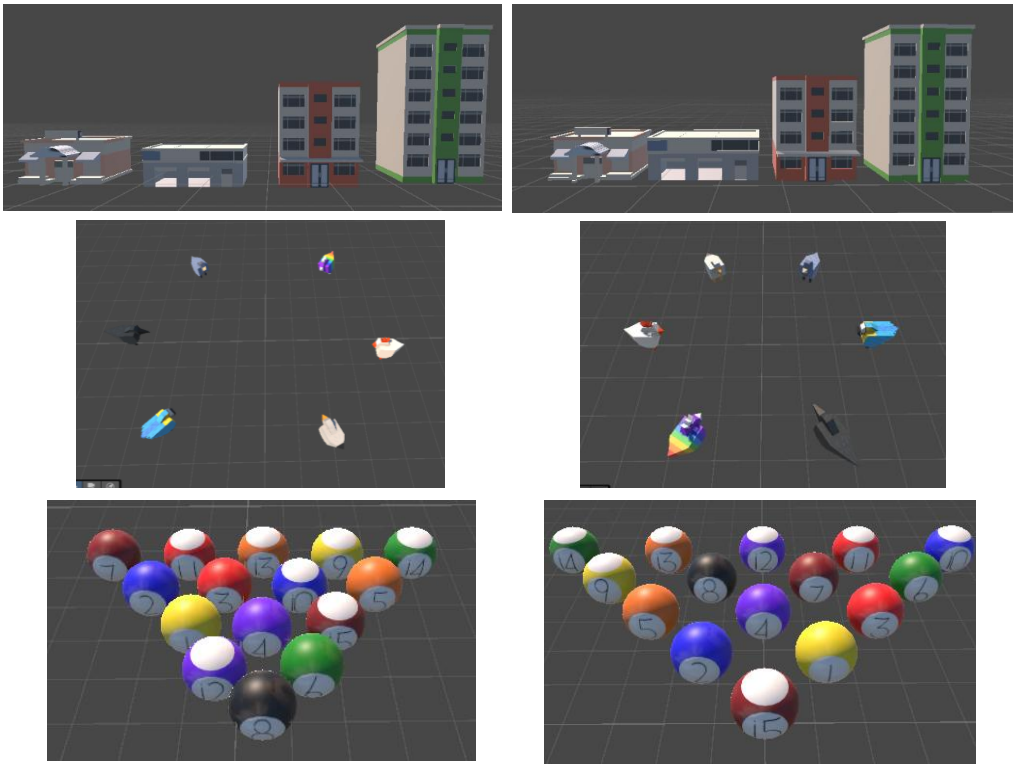
上のオブジェクトを横一列に並べたい．
二つとも前を向いてほしい．
計算して各オブジェクトの center と rotation を教えて．

図 7 システムによる配置に使用したテキスト指示．center, rotation はそれぞれ座標，回転を表す．

図 6 から，システムを利用して配置された 3D モデルは現実で同様に配置されたオブジェクトをよく再現できていることがわかる．これにより本システムを利用することによって，現実のオブジェクトを実際に動かすことなく，また図 7 のような簡単なテキスト指示によってオブジェクトの配置をイメージすることができるといえる．

4.2 手動での配置との比較

仮想環境における 3D モデルの配置において，システムの有用性をより明確にするため，Unity Asset Store から入力した 3D モデルを用いて，システムによる自動配置の結果を手動による配置の結果と比較した[18, 19, 20]．目的の配置は以下の 3 通り用意し，システムが多様なオブジェクトに対応できることを確かめるため，それぞれにおいて使用する 3D モデルを変更した．



(a) 手動による配置　(b) システムによる自動配置　(c) テキスト指示

図 8 手動配置とシステム利用による自動配置の比較. 上から順に ①建物の 3D モデルを一定間隔で直線状に横並びにする, ②鶏の 3D モデルを円周上に並べ, それぞれが円の中心を向くように並べる, ③ビリヤードの玉の 3D モデルを三角形に並べる の配置の結果を示す. (c)テキスト指示では, 本来は 3 つ全てに「計算して各オブジェクトの center と rotation を教えて.」という 3D モデルの座標と回転を出力するためのテキストが含まれるが, 見た目が煩雑になるため省略している.

① 建物の 3D モデルを一定間隔で直線状に横並びにする
② 鶏の 3D モデルを円周上に並べ, それぞれが円の中心を向くように並べる
③ ビリヤードの玉の 3D モデルを三角形に並べる

表 1 手動配置とシステムによる自動配置の所要時間

	手動配置	自動配置
(a)	92.97s	18.80s
(b)	203.68s	127.50s
(c)	376.17s	79.71s

手動配置, 自動配置ともに 3D モデルの初期位置・初期回転は同じ状態とし, そこから指示通りの配置までにかかった時間を計測した. システムによる自動配置はテキスト指示を入力する段階から計測し, 1 回の実行でうまく配置できなかった場合は適宜テキスト指示を変えながら実行し, 指示通り配置できるまでの時間を計測した. 結果の所要時間を表 1 に, 配置後の様子を図 8 に示す.

表 1 の手動配置と自動配置での所要時間の比較から, (a)から(c)のいずれにおいてもシステムを利用するほうが配置にかかる時間が短かった. 特に, 3D モデルの数が多い(c)のタスクでは, 所要時間の差は顕著であった. また, 配置の正確性に関しては, 配置の種類によって手動と自動とで得意な領域が異なることがわかった. 例えば, (a)や(b)のように一定の間隔で配置するような規則的なパターンでは, 自動配置のほうが高精度に並べることができた. 一方, (c)のように 3D モデル間の距離を微調整するような細かい作業においては, 手動のほうが柔軟に対応することができた. また, (b)では自動配置において, 何度も中心以外を向く 3D モデルがあり, ChatGPT は回転を計算するのが苦手な傾向にあることがわかった. これらを受けて, 自動配置のあとに位置の微調整や回転を手動で調整することにより, 自動・手動配置の利点をともに活かしたシステムにすることができると考える. また, (c)のような ChatGPT がすでに知識を持っている配置では,「よくあるビリヤードの最初のように玉を三角形に並べたい.」というシンプルなテキスト指示のみで目的の配置が実現できた. これは, ChatGPT がすでに「ビリヤードのスタート配置」という知識を持っているため, 追加の詳細な指示なしに適切な座標を生成できたことを示している. よって, このような配置に関しては, より直感的な指示で短時間での配置が可能となるということがわかった.

5. まとめ

本研究では，テキストによる指示で 3D モデルを自動配置するシステムを提案し，現実と仮想環境の配置のギャップと仮想環境内におけるシステムによる配置の有用性を確かめる実験を行った．その結果，規則的な配置において本システムは正確にかつ高速に配置できることを示した．一方で, 3D モデル間の微調整や回転などのタスクにおいては, 手動操作のほうが柔軟に対応できることも明らかとなった. また，ChatGPT が事前知識を持つ配置については，シンプルなテキスト指示だけで適切な配置が可能であることがわかった.

以降，今後の展望について述べる．本研究では，1 回のみの配置が可能なシステム設計であるがゆえに，位置は正しいのに向きが間違っていることで再度実行しなおすことが多かった．そこで，改善策として前回の配置情報を保存したまま，テキスト指示を可能にするという方法がある．これを行えば，位置は前回のままで向きだけを指示して変更することができる．また，似た状況として，ほとんどの 3D モデルはうまく配置できているのに, 一部が間違っていることで再度実行しなおすこともあった. これについても，配置に成功しているものは移動せずにほかの 3D モデルのみ再配置できるようなシステム設計にすることが考えられる．また，Unity が持つ物理エンジンを活かし，出力された配置に対して少し力を加えたり，重力を与えたりするなどしてその配置が安定しているかどうかを判断する機能を加えることを検討している．これらの機能を拡張することで，より柔軟な対応が可能で効率的な物体のレイアウト提案システムへと発展させることを目指す.

参考文献

1 K.-H. Chang, C.-Y. Cheng, J. Luo, M. Nourbakhsh, Y. Tsuji, “Building-GAN: Graph-Conditioned Architectural Volumetric Design Generation”, in ICCV 2021.
2 J. Guo, T. Shen, Z. Wang, W. Chen, K. Yin, D. Li, O. Litany, Z. Gojcic, S. Fidler, “GET3D: a generative model of high quality 3D textured shapes learned from images”, in NIPS 2022.
3 S. Lombeyda, S. G. Djorgovski, A. Tran, J. Liu, “An Open, Multi-Platform Software Architecture for Online Education in the Metaverse”, in Web3D 2022.
4 W.R.Para, P. Guerrero, N. Mitra, P. Wonka, “COFS: COntrollable Furniture layout Synthesis”, in SIGGRAPH 2023.
5 D. Paschalidou, A. Kar, M. Shugrina, K. Kreis, A. Geiger, S. Fidler, “ATISS: autoregressive transformers for indoor scene synthesis”, in NIPS 2021.
6 J. Sun, J. Yang, K. Mo, Y. Lai, L. Guibas, L. Gao “Haisor: Human-aware Indoor Scene Optimization via Deep Reinforcement Learning” ACM Transactions on Graphics, vol. 43, no.2, jan 2024.
7 W. Feng, W. Zhu, T. Fu, V. Jampani, A. Akula, X. He, S. Basu, X. E. Wang, W. Y. Wanga, “LayoutGPT: Compositional Visual Planning and Generation with Large Language Models”, in NIPS 2023.
8 Y. Li, H. Shi, B. Hu, L. Wang, J. Ahu, J. Xu, Z. Zhao, M. Zhang, “Anim-Director: A Large Multimodal Model Powered Agent for Controllable Animation Video Generation”, in SIGGRAPH Asia 2024.
9 S. Fang, Y. Wang, Y.-H. Tsai, Y.Yang, W. Ding, S. Zhou, M.-H. Yang,“Chat-Edit-3D: Interactive 3D Scene Editing via Text Prompts”, in ECCV 2024.
10 A. Gunturu, Y. Wen, N. Zhang, J. Thundathil, R. H. Kazi, R. Suzuki, “Augmented Physics: Creating Interactive and Embedded Physics Simulations from Static Textbook Diagrams”, in UIST 2024.
11 N. Jennings, H. _Wang, I. Li, J. Smith, B. Hartmann, “What's the Game, then? Opportunities and Challenges for Runtime Behavior Generation”, in UIST 2024.
12 Z. Zhang, D. Chen, J. Liao “SGEdit: Bridging LLM with Text2Image Generative Model for Scene Graph-based Image Editing” ACM Transactions on Graphics, vol.43, no.6, nov2024.
13 R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models”, in CVPR 2022.
14 Metashape, Agisoft llc .[オンライン]. 入手先: https://www.agisoft.com/ (参照 2025-02-07).
15 A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, R. Girshick, “Segment Anything”, arXiv:2304.02643.
16 OpenAI, “GPT-4o System Card”, arXiv:2410.21276.
17 Unity, Unity Technologies. [オンライン]. 入手先: https://unity.com/ (参照 2025-02-07).
18 Billiard Balls, langvv. Unity Asset Store, Unity Technologies. [オンライン]. 入手先: https://assetstore.unity.com/packages/2d/textures-materials/billiard-balls-6353 (参照 2025-01-31).
19 Simple Poly City - Low Poly Assets, VenCreations. Unity Asset Store, Unity Technologies. [オンライン], 入手先: https://assetstore.unity.com/packages/3d/environments/simplepoly-city-low-poly-assets-58899 (参照 2025-01-31).
20 3D Birds Prototype Pack, K. Grygoryev. Unity Asset Store, Unity Technologies. [オンライン], 入手先: https://assetstore.unity.com/packages/3d/props3d-birds-prototype-pack-150502 (参照 2025-01-24).