**Literature review on**

**3D object reconstruction from images using AI-based methods**

**Bridge AI**

**Shweta Varadarajan**

**13th February 2024**

**Table of Contents**

[***1***](#_heading=h.30j0zll) ***Introduction 3***

[**1.1**](#_heading=h.1fob9te) **Problem overview 3**

[**1.2**](#_heading=h.3znysh7) **Objective 3**

[***2***](#_heading=h.tyjcwt) ***Literature review 4***

[**2.1**](#_heading=h.3dy6vkm) **Geometrical approach 4**

[**2.2**](#_heading=h.1t3h5sf) **AI-based approaches 4**

[2.2.1](#_heading=h.4d34og8) Encoder-Decoder architecture 4

[2.2.2](#_heading=h.3rdcrjn) Generative AI approach 7

[2.2.3](#_heading=h.lnxbz9) Other approaches 9

[2.2.4](#_heading=h.35nkun2) Recent commercial approaches and resources 12

[***3***](#_heading=h.1ksv4uv) ***Summary of methods 13***

[**3.1**](#_heading=h.44sinio) **Input/Output 13**

[**3.2**](#_heading=h.2jxsxqh) **Models and Architectures 13**

[**3.3**](#_heading=h.z337ya) **Datasets 13**

[**3.4**](#_heading=h.1ci93xb) **Overall summary and implementation guidelines 14**

[***4***](#_heading=h.2bn6wsx) ***Conclusions 16***

[**4.1**](#_heading=h.qsh70q) **Summary 16**

[**4.2**](#_heading=h.1pxezwc) **Conclusion 16**

[***5***](#_heading=h.2p2csry) ***References 18***

[***6***](#_heading=h.46r0co2) ***Appendix A 20***

[**6.1**](#_heading=h.2lwamvv) **Terms and definitions 20**

[**6.2**](#_heading=h.3l18frh) **Volumetric scene understanding 20**

[6.2.1](#_heading=h.206ipza) Shape from silhouettes 20

[6.2.2](#_heading=h.2zbgiuw) Shape from Photo-consistency 21

[**6.3**](#_heading=h.1egqt2p) **Differential surface modelling and NeRF 21**

[6.3.1](#_heading=h.3ygebqi) Differential surface modelling 21

[6.3.2](#_heading=h.2dlolyb) NeRF and NeRF models 21

[***7***](#_heading=h.sqyw64) ***Appendix B 22***

[**7.1**](#_heading=h.3cqmetx) **Approach 22**

[**7.2**](#_heading=h.1rvwp1q) **Experiments on pre-trained model MCC Meta 22**

# Introduction

Our brain's natural ability to comprehend dimensions, depth, colours, textures, and object viewpoints enables the visual recognition of 3D objects in 2D images. This innate ability holds significant relevance across industries such as commerce and fashion, where compelling product visualization plays a pivotal role in guiding consumer decisions. As the world increasingly embraces virtual worlds and the metaverse, the importance of 3D visualization becomes even more pronounced - bridging the gap between physical products and digital experiences. Consequently, extensive research is underway in the realm of 3D reconstruction using 2D images.

## Problem overview

Y- Entertainment is a start-up in the creative industry developing an AI-driven platform “Shakesphere” to help brands from luxury fashion, arts, and media to digitize their creative IP into digital assets. These assets involve creating 3D models for virtual worlds, AR/VR etc.

With a focus on the fashion industry for digital assets, the company is looking to turn the physical artefacts into premium NFTs. To enable this, they are researching developing a quick rendering tool for converting 2D images captured on a handheld device to high fidelity 3D models which can then be a part of a virtual world such as Metaverse etc. They intend for the users to be able to produce 3D models for tangible artefacts such as a “ Burberry purse”. Their goal is to connect the world’s biggest technologies such as AI and blockchain in a more accessible manner for people to use in their day-to-day lives.

## Objective

The objective of this literature review is to research techniques and methods in AI for 3D object reconstruction for creating high-fidelity and robust 3D models. To this extent, we review different deep learning architectures and try to summarise their advantages and disadvantages. This would in turn be the first step and help in creating a road map for development and implementation for the company’s wider objectives listed below.

The broader objectives of the company include (adapted from [20]):

1. To develop an AI model for creating an HD 3D digital twin of meaningful physical artefacts
2. To develop a mobile application for a fully connected B2B2C service.
3. To mint digital twins as premium NFTs.
4. To employ AI to generate variations of premium NFTs at scale.

# Literature review

3D reconstruction has been a long-standing challenge in the field of computer vision. It finds many crucial applications in [healthcare](https://en.ru.is/bne/3d/), [robotics](https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-3932/6095_read-45162/#:~:text=In%20robotics%2C%20usually%20complete%20object,task%20for%20the%20human%20operator.) and more recently [augmented/virtual reality](https://ikarus3d.com/media/3d-blog/power-of-ar-and-vr-3d-modeling-for-immersive-experiences/#:~:text=3D%20Assets%20for%20AR%20and%20VR&text=Augmented%20reality%20involves%20overlaying%20a,objects%20and%20digital%203D%20environments.). The following sections provide an in-depth literature review of the recent research that has been published. It also explores some commercially available solutions.

## Geometrical approach

Historical methods approached this problem using plain geometry by mathematically deriving 2D to 3D projection. Since the methods were mathematical, these required the image to be captured in a controlled set-up, so that factors such as lighting, background etc. did not vary. One such method is described in [1] which uses silhouette-based volume intersection[[1]](#footnote-0) for reconstruction under a controlled image environment. They use a multi-image calibration (images from multiple camera view angles) approach and volumetric scene modelling[[2]](#footnote-1). While these methods have proven to be useful in computer vision research, the handheld devices (mobile phones) would not have a controlled environment and therefore, such methods would be very difficult to implement practically.

Recent research has seen vast advancement in machine learning and artificial intelligence where the predictive power of AI is leveraged to compensate for the controlled environment set-up. Further sections explore a few interesting research papers and techniques of 3D reconstruction using AI and deep learning models.

## AI-based approaches

The basic idea of 3D object reconstruction is to approximate the rough structure in 3D space without knowing all the information about the object. AI techniques namely deep learning models that can predict complex unknown characteristics have led to a new generation of methods for 3D object reconstruction. These rely on training data for approximation of the 3D structure of the object. This reduces and, in some cases, eliminates the need to have a controlled camera calibration and environment.

Literature suggests that the state-of-the-art image-based 3D reconstruction can be broadly classified based on architecture as follows:

* Encoder-Decoder architecture
* Generative AI
* Large Transformer based models

In the following sections, we dive into the details of these architectures using a few specific publications which are relevant to the problem statement.

Each architecture is explored from the perspective of our use case with a discussion at the end and also has a link to the project code online.

### Encoder-Decoder architecture

**3D R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction**

The model seen in paper [2] takes in one or more images from arbitrary viewpoints and outputs a voxelized[[3]](#footnote-2) reconstruction of the object in 3D. It is based on the idea of having a 3D prior along with input images for reconstruction.

Architecture: The architecture of the 3D R2N2 consists of an encoder (2D-CNN), a recurrence unit (3D LSTM) and a decoder (3D-DCNN). The main idea is to leverage the power of LSTM to retain the previous observations and recurrently update the output reconstruction. The encoder is a standard feed-forward CNN network which converts images to features. The LSTM network is a modified LSTM[[4]](#footnote-3) to localise prediction according to the position of the voxel in the grid. After receiving the input image sequence, LSTM passes the hidden state to a decoder which increases the resolution using 3D convolutions to reach target output resolution.

A diagram of a conversational conversation

Description automatically generated with medium confidence

*Figure 1: 3D-R2N2 Architecture, adapted from [2]*

Dataset used:

* Training: ShapeNet [14] 3D CAD models rendered to images and augmented using PASCAL VOC 2012.
* Testing: ShapeNet, PASCAL 3D+[19]

Discussion:

* This approach combines a single-view and multi-view approach with supervision during training. The training uses publicly available datasets.
* The reconstructed objects are voxelized occupancy grids with no real information of colour or texture.
* The network requires 3D supervision (3D CAD annotation) which is not readily available in all datasets.
* A dense voxel occupancy grid requires high memory compute

Project code: <https://github.com/chrischoy/3D-R2N2>

**Multiview Compressive coding (MCC – Meta)**

The problem of 3D supervision and improvement on the prediction of colour and texture from the above research is overcome in the MCC’s transformer encoder-decoder architecture mentioned in the paper [6]. This also uses point clouds to improve memory computing.

The framework introduced by Meta operates on 3D points[[5]](#footnote-4) of single objects or whole scenes coupled with Category-agnostic large-scale training[[6]](#footnote-5) from diverse RGB-D(depth) inputs.

Architecture:

Encoder: The input to the encoder is a single RGB-D image. The depth component is used to unproject[[7]](#footnote-6) and convert it into positions in point clouds. The encoder uses transformer architecture for ERGB and Exyz in Figure 2. The outputs of the transformer are concatenated and projected into suitable format as an input to the decoder alongside the queries (N samples from 3D point clouds as “positive” or “negative” based on radial distance from ground truth). The ground truth is defined as the union of all unprojected points from all RGB-D views of the scene.

Decoder: The decoder inputs a query and output of the encoders in a transformer architecture which is then passed through a binary classifier that predicts the query’s occupancy and a 256-way classifier which predicts the RGB colour.

A diagram of a graphing process

Description automatically generated

*Figure 2: MCC Architecture, adapted from [6]*

Dataset: CO3D - v2: 37k short videos of 51 objects.

Training: Training is done for 150k iterations with Adam using 32 GPUs. It took the authors 2.5 days to train the model. Further details can be found in the paper [6]

Discussion:

* Generalisation is better with this architecture because the training is category-agnostic.
* The scalability and compute time are well explained in the paper [6] and the code is available on GitHub to retrain/use for development.
* This is a good example of implementing a transformer-based encoder-decoder architecture for 3D object reconstruction.

Project code: <https://github.com/facebookresearch/MCC>

### Generative AI approach

Another approach to 3D reconstruction is using Generative AI. With GANs' ability to produce 2D images from word descriptions or other media cues, generative artificial intelligence (Gen AI) has been increasingly popular in recent works. The generated images have resolutions and clarity at par with real images. Some notable examples include DALL-E.

Expanding the scope of Gen AI to 3D reconstruction, the recent research in this field has led to leveraging the power of Gen AI to develop models which can create high-fidelity 3D models from images by mapping them to a variable in the latent space.

**Get3D – Nvidia**

In the research as published in the paper [3], Nvidia introduces a Generative AI model which generates explicit 3D meshes with complex topology and high fidelity.

Architecture: Generative Model

The basic architecture of a GAN consists of a generator – responsible for the generation of images and in this case a 3D model, and a discriminator which is responsible for comparing it with ground truth which is then used to update the optimizer or loss function.

Generator: The generative process is divided into two parts: the geometry branch and the texture branch. The geometry branch outputs a surface mesh in the form of SDF (Signed distance field) [[8]](#footnote-7) and the texture branch outputs a texture field which can be queried to produce colours.

Discriminator: The 3D texture field and surface mesh are rendered into a 2D RGB image and its silhouette using Nvdiffrast[[9]](#footnote-8). The 2D images are then used to supervise the network and optimize the loss function using two discriminator architectures as in StyleGAN [5]. The optimization function and regularization are explained in section 3.2 of the paper.

A blue car with a diagram

Description automatically generated

*Figure 3: Get3D Architecture, adapted from [3]*

Dataset: ShapeNet [14], Animal dataset

Training: The training involves splitting the dataset into Train (70%), test 20%), and validation (10%). A separate model is trained for each category. The training computes requirements and GPUs are mentioned [here](https://github.com/nv-tlabs/GET3D).

Discussion:

* There may be a need to train a separate model for each category to achieve better accuracy.
* 2D silhouettes and camera distribution is needed for the discriminator therefore not all datasets and real-world data can be directly used.
* Generalisation would depend on the categories in the dataset used to train the model.

Project code – Pretrained model:

<https://github.com/nv-tlabs/GET3D/tree/master/pretrained_model>

**3D-VAE-GAN**

Another Gen AI-based architecture that takes a single image input as opposed to the Get3D architecture seen above is the 3D-VAE-GAN [7]

Architecture:

The GAN network is preceded by an image encoder network which takes a 2D image as input and outputs the latent representation vector. The latent representation is fed as input to the generator network which maps it into a 64 X 64 X 64 cube representing an object in 3D space. The discriminator then outputs a confidence value whether the 3D object is real or synthetic.

The architecture of the convolutional layers in the generator and discriminator as well as the model training details are mentioned in section 3.1 of the paper.

A diagram of a chair

Description automatically generated

*Figure 4: 3D -VAE-GAN architecture, adapted from [7]*

Datasets: ShapeNet database – training

IKEA database - testing

Training: Training 3D-VAE-GAN requires both 2D images and their corresponding 3D models.

Discussion:

* The model is needed to be trained on each category either jointly or separately which means that the dataset needs to have the appropriate classes for the fashion industry application.
* The output of this model is a voxelized 3D shape with no information of colour or texture. Therefore, would need rendering after reconstruction or further extension in training to include texture information.
* The model needs the 3D and 2D information both on the dataset. Since it is a generative AI architecture, the output would depend on the dataset. Therefore, if the custom dataset is to be used to train the model, it should have good coverage of the classes.

Project code: Pretrained model: <https://github.com/zck119/3dgan-release>

Ikea Dataset: <http://3dgan.csail.mit.edu/data/IKEA_imgs_shapes.zip>

### Other approaches

**PiFU: Pixel-aligned Implicit Function**

PiFU seen in [9] is a deep learning-based method for 3D reconstruction of highly intricate shapes within an image with predictive learning for unseen areas of the image. PiFU was originally developed for digitizing highly detailed clothed humans that can infer both 3D surfaces and texture from a single image and optionally multiple input images.

Architecture:

The architecture is divided into two parts surface reconstruction and texture inference. The surface reconstruction predicts the continuous inside/outside probability field of a clothed human and the texture inference outputs RGB value at 3D positions.

A diagram of a structure

Description automatically generated

*Figure 5: PiFU architecture, adapted from [8]*

Dataset: RenderPeople, DeepFashion

Training: The training is done sequentially, first for surface reconstruction and then for texture inference. Further details of the training are mentioned in the supplementary section of the paper.

Discussion:

* PiFU handles single-view images and multi-view images with the same architecture but a different training procedure.
* The output has texture and colour information and is not voxelized.
* It is primarily designed for human body reconstruction which could be retrained for fashion industry-related objects.
* The code is available online to retrain on objects-based datasets.

Project code: <https://github.com/shunsukesaito/PIFu>

**LRM: Large Reconstruction model**

LRM is a transformer-based architecture seen in [11] which predicts the neural radiance field (NeRF) from the input image. The method adopts a large transformer-based encoder-decoder architecture for learning 3D representations of objects from a single image in a data-driven manner.

Architecture:

Encoder:

* Given an RGB image as input, the LRM applies a pre-trained transformer model ViT [10] to encode the image to feature tokens.

Decoder:

* The encoder is followed by a large transformer decoder network to project the features into the 3D triplane via cross-attention[[10]](#footnote-9).
* This is followed by a multilayer perceptron[[11]](#footnote-10) to predict the point colour and density for volume rendering.

The architecture also considers the camera parameters such as the focal length and principal point which helps in the reconstruction of occluded areas of the image during training.

A diagram of a computer program

Description automatically generated with medium confidence

*Figure 6: LRM architecture, adapted from*

Dataset: Training: 3D data from Objaverse, MVImgNet

Testing: Objaverse, MVImgNet, ImgNet, Real-world data etc.

Training:

The authors have mentioned training the LRM on 128 GPUs for 3 days. However, the inference is less computationally heavy and can be downstreamed for practical applications. Further training details and loss functions are described in sections 3.1 and 4.2 of the paper [11].

Discussion:

* The architecture is a large transformer model which is trained on more than a million data points which makes the generalisation accuracy high and therefore category specific data is not needed while training. The datasets used to train the model are quite recent and therefore the consistency of the model is better than the other methods seen in the report.
* Even though the architecture is a complex transformer model, the working of the model is quite similar to a normal encoder-decoder architecture which makes it easy to understand. The results further show that it is scalable and can be further adapted to make it efficient for practical application.
* The robust solution, however, such architecture does need a higher computing power to train if needed to build from scratch. The architecture also needs camera parameters in training and additionally, assumes camera parameters during testing which may/may not align with the ground truth.

Project code: <https://github.com/3DTopia/OpenLRM>

**Zero1-to-3: Zero-shot One image to 3D object**

This architecture in [13] provides novel view synthesis given just a single RGB image. This model primarily is to control the viewpoint of the image but can be extended to 3D model reconstruction by training a NeRF. The research capitalizes on StableDiffusion to create novel views and DreamFusion for 3D reconstruction. The training is done on the Objaverse dataset. The paper explains the accuracy of the model and compares the results with Meta’s MCC. Although the model is generative AI-based, the generalisation accuracy of the model is demonstrated to be higher than other GAN-based approaches.

Project link: <https://github.com/cvlab-columbia/zero123?tab=readme-ov-file>

### Recent commercial approaches and resources

**Commercial Websites and demos:**

* [Hugging face demo for Zero123](https://huggingface.co/spaces/cvlab/zero123-live)
* [Hugging face demo for OpenLRM](https://huggingface.co/spaces/zxhezexin/OpenLRM)
* [Hugging face demo for Point E](https://huggingface.co/spaces/openai/point-e)
* [Alpha3d](https://www.alpha3d.io/)
* Threestudio [implementation](https://github.com/threestudio-project/threestudio?tab=readme-ov-file#zero-1-to-3-)

**Online Resources – Code, platforms, research:**

* [InstantNerf](https://developer.nvidia.com/blog/getting-started-with-nvidia-instant-nerfs/)and**:** Platform for 3D pipeline reaslisation
* [Zero123](https://github.com/cvlab-columbia/zero123): Single shot to 3D model codebase
* [Point-E](https://github.com/openai/point-e)**:** Text to 3D model codebase
* [Step by Step guide to use Shap-E](https://lablab.ai/t/shape-e-tutorial-how-to-set-up-and-use-shap-e-model)

**Google Phorum –** [Research Paper](https://phorhum.github.io/static/assets/alldieck2022phorhum.pdf) (2022)

**Neuralangelo –** [Research paper](https://arxiv.org/pdf/2306.03092v2.pdf) **and**: 3D scene reconstruction from multiple images (2023)

* [Kaolin](https://kaolin.readthedocs.io/en/latest/)and [PyTorch3D](https://pytorch3d.org/) **:** Python libraries for 3D representation

**Technical blogs/videos:**

* <https://www.decasonic.com/perspectives> : Collection of blogs on mixed reality, AI, virtual worlds.
* <https://www.alpha3d.io/3d-file-formats/> : 3D model output file formats and other blogs
* [Blog on commercial tools by Google, OpenAI etc](https://blog.paperspace.com/3d-generation-deep-learning-models/) : Semi-technical blog explaining DreamFusion, Point E etc
* [A detailed primer on 3D Computer vision and Machine learning](https://www.kaggle.com/discussions/getting-started/426952)
* [Code Mental videos on 3D deep learning](https://www.youtube.com/playlist?list=PL3OV2Akk7XpBPBEw1jekpxDJYIRPEHbUi) : Easy YouTube videos to follow recent developments in 3D deep learning

# Summary of methods

The research publications and commercially available solutions that we have seen above can be summarised based on input-output types, architecture, datasets and results. The following sections summarise the methods in a comparative format.

## Input/Output

| **Input type** | **Single-view image**  (Single image either synthetic, dataset or clicked on the phone) | **Multiview- images**  (Images from multiple – views) | **Single-view or multi-view images with 3D prior**  (Single or multiple RGB images with 3D model as ground truth) | **Latent vector, videos, 3D prior, synthetic images, text**  (Representation of image in the mentioned formats above -not in scope of this report) |
| --- | --- | --- | --- | --- |
| **Examples** | Zero123, Shap-e, Image2Mesh, DeepSDF, Pixel2Mesh | Multiview-3D-VAE-GAN, PiFU | 3D R2N2 | Get3D, 3D-GAN |

| **Output** | **Voxelised occupancy grid**  (Volumetric output usually without texture or color information) | **3D-point cloud**  (Surface 3D points in space having texture and color information | **3D mesh**  (Surface SDF, Deformable tetrahedral) | **NeRF** |
| --- | --- | --- | --- | --- |
| **Examples** | 3D-R2n2, | MCC, DeepFashion3D | Get3D, PiFU | **Zero1-to-3** |

## Models and Architectures

| **Input type** | **Encoder-Decoder**  (CNN, | **Generative AI**  (GANs, VAE-GANs) | **Large Language model/Transformer** |
| --- | --- | --- | --- |
| **Examples** | 3D R2N2, MCC | Get3D, 3DGAN, 3D-VAE-GAN, Zero123, Point-E, Shap-E | LRM, MCC |

## Datasets

Following is a list (Link) of datasets that are extensively used in 3D reconstruction:

1. [ShapeNet](https://shapenet.org/) [14]
2. [Objaverse](https://objaverse.allenai.org/objaverse-1.0) [16]
3. [DeepFashion3D](https://github.com/GAP-LAB-CUHK-SZ/deepFashion3D) [15]
4. [CO3D-v2](https://ai.meta.com/datasets/CO3D-dataset/) [17]
5. [IKEA Dataset](https://ikeaasm.github.io/) [18]
6. [PASCAL 3D+](https://cvgl.stanford.edu/projects/pascal3d.html)[19]
7. [ModelNet](https://modelnet.cs.princeton.edu/)
8. [ObjectNet3D](https://cvgl.stanford.edu/projects/objectnet3d/)
9. [DeepFashion](https://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html)

## Overall summary and implementation guidelines

The above-mentioned architectures give a basic overview of how the problem is approached. Further experiments such as evaluating a pre-trained model and retraining it for another dataset would result in a better customised solution for the problem statement. A few additional resources are captured in this report in Section 2.2.4 which may provide more up-to-date information on the techniques apart from academic papers. To summarise,

Overall idea:

The idea should be to minimize data dependencies like 3D priors and camera settings for training the model. We would also want the model to have good generalisation accuracy. Having said that, since the application is fashion industry related, we could experiment category-wise and improve the accuracy of the model mainly for the fashion industry domain categories.

Hardware recommendation:

Generally, diffusion models are architecture-heavy and need GPUs for training. Even the pre-trained model sometimes would need GPU to render faster. So, it is recommended to have access to good hardware facilities before starting to experiment with the pre-trained model. More on GPUs and understanding requirements can be found [here](https://www.znetlive.com/blog/how-to-choose-a-gpu-for-machine-learning/).

Proof-of-concept:

It is important to understand the basic architectures and terminologies for which this report provides an overview. Once a primer on the domain knowledge is available, experimentation will be easier.

We could try to experiment with 1) Category agnostic model such as Meta MCC which also has translation from iPhone captures to 3D models

2) We could experiment with Generative AI-based models such as Zero123, and Point-e which might be focused on the fashion industry.

Although most of these models are open source for research applications, particular attention should be given to their licensing when used for commercial purposes. The details of this can usually be found on GitHub implementation.

Once a few experiments are done on the models, it would be easy to determine which would suit the most for our use case. If a fashion industry-based dataset is not available, we could use one among the ones mentioned in 3.3.

Testing:

All models should have a good evaluation framework and suitability check. These could involve metrics like speed of rendering, accuracy, output format etc. The technical evaluation of the AI model is done using metrics such as F-score, MSE values, and IoU values, all of which should be explored when retraining/training a diffusion model.

# Conclusion

The approaches discussed above provide perspective on different use cases and ways in which 3D reconstruction from 2D images could be achieved. While all the reviewed methods are seeing intensive research and progress, a few of the AI methods do stand out for Y-Entertainment’s problem statement.

## Summary

As a general summary, it is important to note that **single-shot** image reconstruction involves the prediction of novel views of the object. For such an application, Generative AI could be used to achieve high-fidelity 3D models. A good starting point is retraining a Generative AI model such as Zero 1-2-3, Get3D or Pointe using datasets such as **DeepFashion3D [15**]**, Objaverse [16]** or others if deemed fit. Performing experiments on the architecture and analysing the results. Achieving accuracy on online datasets and then creating a custom dataset category-wise would enable the model accuracy to improve incrementally. This will enable development for a specific application domain such as the fashion industry.

Alternatively, for single and multi-view images, a transformer based architecture could be explored. The use of transformer architecture allows training category agnostic AI models which will have good generalisation as seen in LRM and Meta’s MCC. Codebases for both these architectures are available and have pre-trained weights that can be used for the first experiments. Appendix B contains further details. It is important to note that using a transformer architecture would potentially minimise the need for a custom dataset. However, the same approach would possibly need a bigger generic dataset to achieve higher accuracy. To increase the suitability of a general-purpose model for the fashion industry, experiments could be performed focusing on the categories in the generic dataset that are more directly relevant. Another interesting feature, as seen in the reviewed research, is that a transformer or LLM-based AI model could handle multi-view images and single-shot images using the same architecture. However, only further experiments can determine how difficult training such a model is in practice.

Diffusion models tend to require higher computing and therefore, it would be advisable to make sure that compute requirements for training an AI model are considered. The architectures seen above require multiple GPUs with single or multiple nodes to train a model. Cloud resources by Google on GoogleColab could be used, however, they are not always very reliable for continuous intensive usage (training for days together) with limited allowances.

## Conclusion

In the past, creating 3D content required experts in 3D modelling. However, the use of AI enables high-quality 2D and 3D content creation, by making it as simple as entering text or uploading an image. At the time of writing, AI models do not yet reach the quality of the 3D content generated in software such as Blender, manually. However, the ability of AI to reach such high fidelity in the future can not be questioned. This would enable the creative audience to engage with asset creation in a much easier way thereby increasing the use of technologies such as AR/VR. Therefore, to achieve the goals of Y-Entertainment, it is crucial to understand the various aspects related to the role of AI with respect to Blockchain and Mixed Reality.

Technology such as Blockchain and NFTs enable creators to own, track and monetize their creations. Therefore, envisioning a workflow where the creator uploads their work which may be as raw as a sketch or text even, which then can be translated into 3D models using AI is a vision with substantial potential.

In addition, overlaying such creations in the user’s personal surroundings or a virtual environment would create an immersive experience. Virtual luxury clothing/accessories to visualise the user’s style helps in digitizing the entire pipeline from creation to showcase. Therefore, it becomes crucial to understand the landscape of Web3, mixed reality and blockchain before diving directly into developing models. In this respect, an interesting review of the landscape with the key players in the market can be found [here,](https://www.decasonic.com/post/web3-ai-market-map) in Decasonic’s blog.

Therefore, the pipeline that follows the generation of the 3D model itself needs an equal amount of attention as the AI architecture, dataset and training. Various platforms such as Alpha3D, Meta, Nvidia etc focus on different aspects of this pipeline and it is important to understand the combined potential of the various technologies at play.

As such, it is important to develop a model which is scalable and adaptable to such an ecosystem.

Planning the inputs and outputs, and designing the generation, visualisation and deployment aspects of the pipeline are all aspects that need to be considered in the early stages. For instance, the development of the AI model for 3D reconstruction needs to be informed of the details of the further pipeline, so that the output from the model can be translated to the correct format.

In conclusion, for such a novel and upcoming technology, keeping up to date with the AI techniques in 3D model reconstruction, Web3, mixed reality and market competitors is essential for strategizing the goals. End-to-end research on these would help in designing the framework, creating a niche and eventually becoming experts in the domain. To ensure quality, it is quintessential to collaborate with experts from the domain who can proofread outcomes and help in the planning. Therefore, a holistic approach for designing and diving deep into the basic blocks in a step-by-step way would help in the overall development of the platform “Shakesphere”.

# References

Research papers:

1. Mulayim AY, Yilmaz U, Atalay V. Silhouette-based 3-D model reconstruction from multiple images. IEEE Trans Syst Man Cybern B Cybern. 2003;33(4):582-91. doi: 10.1109/TSMCB.2003.814303. PMID: 18238208.
2. Choy, C. B., Xu, D., Gwak, J., Chen, K., & Savarese, S. (2016). *3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction*.Wu, C.-Y., Johnson, J., Malik, J., Feichtenhofer, C., & Gkioxari, G. (2023). *Multiview Compressive Coding for 3D Reconstruction*.
3. Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z., & Fidler, S. (2022). GET3D: A Generative Model of High Quality 3D Textured Shapes Learned from Images. *Advances In Neural Information Processing Systems*.
4. Karras, T., Laine, S., & Aila, T. (2019). *A Style-Based Generator Architecture for Generative Adversarial Networks*.
5. Han, X.-F., Laga, H., & Bennamoun, M. (2021). Image-Based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(5), 1578–1604. https://doi.org/10.1109/tpami.2019.2954885
6. Wu, C.-Y., Johnson, J., Malik, J., Feichtenhofer, C., & Gkioxari, G. (2023). *Multiview Compressive Coding for 3D Reconstruction*.
7. Wu, J., Zhang, C., Xue, T., Freeman, W. T., & Tenenbaum, J. B. (2017). *Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling*.
8. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., & Li, H. (2019). *PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization*.
9. Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2020). *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*.
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*.
11. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., & Tan, H. (2023). *LRM: Large Reconstruction Model for Single Image to 3D*.
12. Bottino, A., Jaulin, L., & Laurentini, A. (2003). Reconstructing 3D Objects from Silhouettes with Unknown Viewpoints: The Case of Planar Orthographic Views. 2905, 153–162. <https://doi.org/10.1007/978-3-540-24586-5_18>
13. Liu, R., Wu, R., Hoorick, B. V., Tokmakov, P., Zakharov, S., & Vondrick, C. (2023). *Zero-1-to-3: Zero-shot One Image to 3D Object*

Datasets:

1. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., & Farhadi, A. (2022). *Objaverse: A Universe of Annotated 3D Objects*.
2. Heming, Z., Yu, C., Hang, J., Weikai, C., Dong, D., Zhangye, W., Shuguang, C., & Xiaoguang, H. (2020). Deep Fashion3D: A Dataset and Benchmark for 3D Garment Reconstruction from Single Images. *Computer Vision – ECCV 2020*, 512–530.
3. Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., & Yu, F. (2015). *ShapeNet: An Information-Rich 3D Model Repository* (Techreport arXiv:1512.03012 [cs.GR]). Stanford University — Princeton University — Toyota Technological Institute at Chicago.
4. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., & Novotny, D. (2021). *Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction*.
5. Ben-Shabat, Y., Yu, X., Saleh, F. S., Campbell, D., Rodriguez-Opazo, C., Li, H., & Gould, S. (2020). The IKEA ASM Dataset: Understanding People Assembling Furniture through Actions, Objects and Pose. *ArXiv Preprint ArXiv:2007.00394*.
6. Xiang, Y., Mottaghi, R., & Savarese, S. (2014). Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild. *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Websites:

1. https://www.canva.com/design/DAFvp3sz1G8/cNXrZvpwYxUuRt5MbGxCPg/view?utm\_content=DAFvp3sz1G8&utm\_campaign=designshare&utm\_medium=link&utm\_source=viewer

# Appendix A

## Terms and definitions

1. Voxel:  A Voxel represents a value on a regular grid in three-dimensional space. Like a pixel in a 2D image, a voxel can contain a specific location inside a 3D grid and has a color value assigned to it. All individual voxels are the same size, however voxels can then be combined in the same grid at different positions with different colors to create what is known as a voxel model. More on voxel modelling can be found [here](https://blog.spatial.com/the-main-benefits-and-disadvantages-of-voxel-modeling).
2. Point cloud: A point cloud is a discrete set of data points in space. The points may represent a 3D shape or object. Each point position has its set of Cartesian coordinates (X, Y, Z). Good visualisations and overview can be found [here](https://www.dronegenuity.com/point-clouds/).
3. Mesh: A polygon mesh is a collection of vertices, edges and faces that define the shape of a polyhedral object. The faces usually consist of triangles (triangle mesh), quadrilaterals (quads), or other simple polygons (n-gons). Details of basic elements are summarised in this [blog](https://3dstudio.co/polygon-mesh/).
4. Signed distance field: An SDF is just a function which takes a position as an input, and outputs the distance from that position to the nearest part of a shape. A particularly good explanation of the concept for rendering 3D SDFs can be found [here](https://jasmcole.com/2019/10/03/signed-distance-fields/).

## Volumetric scene understanding

### Shape from silhouettes

Volume intersection: As explained in [12], many algorithms are based on occluding contours or silhouettes. The main approach is a volumetric intersection which consists of building the volume R shared by the regions Ci (see Figure 7) obtained by back projecting each silhouette Si from the corresponding viewpoint. The generic shape of the back projection shape is a cone. This simple reconstruction technique is called Volume Intersection It requires the 3D positions of silhouettes and viewpoints.

The region R is called the visual hull of the object which is the object that can be obtained by the volumetric intersection using all the viewpoints that belong to a viewing region.

A diagram of a plane

Description automatically generated

*Figure 7: Silhouettes based reconstruction*

### Shape from Photo-consistency

* Photo-consistency determines whether a given voxel is occupied. A voxel is photo-consistent when its colour is similar to all the cameras that can see it.
* Shape from Photo-consistency methods uses additional photometric information i.e. colour for the reconstruction. 3D reconstruction based on photo consistency requires camera parameters for each used view and a model for the object’s surface reflectance.

## Differential surface modelling and NeRF

### Differential surface modelling

Differential surface modelling makes use of a technique in 3D reconstruction where the object’s underlying surface is converted into an explicit mesh with a Marching Tetrahedra (MT) algorithm that is differentiable. The goal of the algorithm is to generate isosurfaces, which are surfaces representing points in space where the scalar field has a constant value. The "marching" part of the algorithm refers to systematically traversing through the 3D grid of tetrahedra. At each step, the algorithm determines the configuration of the isosurface within the tetrahedron and generates triangles that approximate the isosurface within that tetrahedron (more on the algorithm can be found [here](https://www.wikiwand.com/en/Marching_tetrahedra))

### NeRF and NeRF models

NeRF models are novel view synthesis methods which use volume rendering with implicit neural scene representations via MLP to learn geometry and lighting. They do not need 3D/depth supervision however typically need multiple images from different views. It works by predicting colour and light in any direction from any point in the 3D space. The technique can work around occlusions.

# Appendix B

## Approach

A lot of experimentation is needed with an AI model that best suits the application, adheres to the compute limitations and is scalable for the future. It is important to leverage the resources already present therefore using pretrained models would be a good starting point rather than reinventing the wheel and creating a new AI model architecture. Experiments may result in changes in architecture further on. Please note the licensing around commercial usage of pre-trained models.

The development steps potentially would involve the following:

1. Focus on experimenting with one architecture at a time. Research pre-trained models from this report and otherwise if needed (Please note the licensing around commercial usage of the pre-trained model)
2. Checking the codebase (GitHub) and requirements for running the model.
3. Check data sources used in the model, most pre-trained models would not need a dataset.
4. If the model results are satisfactory, try to retrain the model on a small subset of the dataset keeping in mind the compute requirement.
5. If the necessary accuracy is achieved, try to modify the architecture and parameters to improve accuracy.
6. Increase the dataset size, retrain the model and evaluate the output.

## Experiments on pre-trained model MCC Meta

Steps followed to download and run the pre-trained model: (This model is only for the experiment, not very suitable for our use case)

1. Clone the GitHub Repo.
2. Download the pre-trained model checkpoint from the GitHub repo trained on the CO3D v2 dataset
3. Keep all test images in the demo folder of the Github repo
4. Modify the code to run on CPU ( if CUDA is not available)
5. Open the terminal and cd into the MCC folder
6. Run the command with appropriate paths :

python demo.py --image demo/quest2.jpg --point\_cloud demo/quest2.obj --seg demo/quest2\_seg.png \

--checkpoint [path to model checkpoint] \

1. The model will run and process an HTML file which can be found in the demo folder

Note: For testing on other images, the Record3d App on iOS devices is needed to capture RGB-D images and a segmentation model (private) for removing the background etc. GitHub repo contains details[[12]](#footnote-11).

After experimentation, we do need extra software and apps to run the model and it takes a fair bit of time to run on CPU. Therefore, it is scalable and can be improved very easily but is not most suitable for our use case

1. Refer Appendix Section 6.2.1 [↑](#footnote-ref-0)
2. Refer Appendix Section 6.2.2 [↑](#footnote-ref-1)
3. Refer to Appendix section 6.1 for the definition of voxel [↑](#footnote-ref-2)
4. https://colah.github.io/posts/2015-08-Understanding-LSTMs/ [↑](#footnote-ref-3)
5. Refer Appendix section 6.1 [↑](#footnote-ref-4)
6. Category-agnostic large-scale training refers to training machine learning models on datasets that contain diverse and numerous categories without focusing on a specific predefined set of categories [↑](#footnote-ref-5)
7. Unprojection : Process of converting 2D image pixels into 3D points in the scene [↑](#footnote-ref-6)
8. SDF : Signed distance field – refer appendix A, 6.1 [↑](#footnote-ref-7)
9. https://nvlabs.github.io/nvdiffrast/ [↑](#footnote-ref-8)
10. [Blog on cross-attention](https://medium.com/@geetkal67/attention-networks-a-simple-way-to-understand-cross-attention-3b396266d82e) [↑](#footnote-ref-9)
11. [Multilayer Perceptron definition](https://scikit-learn.org/stable/modules/neural_networks_supervised.html) [↑](#footnote-ref-10)
12. \*\*It is not tested on custom images due to shortage of time on the project [↑](#footnote-ref-11)