POLITECNICO
MILANO 1863



LOUVEN+

Ali Elsayed

POLITECNICO MILANO 1863 | SCUOLA DI ARCHITETTURA URBANISTICA INGEGNERIA DELLE COSTRUZIONI

SCHOOL OF ARCHITECTURE, URBAN PLANNING AND CONSTRUCTION ENGINEERING
MASTER OF SCIENCE IN BUILDING AND ARCHITECTURAL ENGINEERING

LOUVEN+

From Text to Fabrication: Integrating AI Generative Modelling, Freeform Surface Reconstruction, and Band Strips Discretization for Automated Architectural Facade Fabrication

Author
Ali Ayman Ibrahim Mahmoud Elsayed
Matricola: 245273

Supervisor: Prof. Pierpaolo Ruttico
Co. Supervisor: Imane El Bakkali
Academic Year: 2025/2026



In the Name of Allah
the Most Gracious, the Most Merciful.

Abstract

The increasing complexity of free-form architectural facades presents significant challenges in translating conceptual designs into fabrication-ready building components while maintaining geometric accuracy, constructability, and efficient information exchange. This thesis proposes an integrated computational workflow that connects AI-assisted conceptual design, computational modelling, digital fabrication, and Building Information Modelling (BIM) within a continuous design-to-production process.

The methodology begins with the generation of façade concepts using text-to-image, image-to-image, and image-to-3D artificial intelligence models. Selected concepts are reconstructed as NURBS-based free-form surfaces and processed through a fabrication-aware workflow developed in Grasshopper. The workflow automatically discretizes double-curved surfaces into interconnected near-developable metal strip components while integrating structural backbone generation, automated connection detailing, bending information extraction, component unrolling, material nesting, and fabrication preparation.

To support interoperability within the Architecture, Engineering, and Construction (AEC) industry, the computational model is integrated with Autodesk Revit through Rhino.Inside.Revit, enabling the automatic generation of native BIM families and fabrication documentation while maintaining consistency between geometric and information models.

The methodology is validated through the fabrication and assembly of a 1:20 physical prototype, confirming the feasibility of the proposed workflow and enabling the refinement of the computational model through practical fabrication feedback.

Developed as LOUVEN+, the proposed framework extends the original LOUVEN facade system initiated by INDEXLAB at Politecnico di Milano by replacing direct surface discretisation with a fabrication-oriented reconstruction based on near-developable strip surfaces, while automating fabrication detailing, documentation, and BIM integration, providing a robust foundation for future AI-assisted design-to-production workflows for free-form architectural facade systems.

Keywords: artificial intelligence (ai); generative design; free-form architecture; computational Design; band strips; surface discretisation; near-developable surfaces; digital fabrication; Building Information Modelling (bim); adaptive Facade

Riassunto in lingua italiana

La crescente complessità delle facciate architettoniche a forma libera pone notevoli sfide nella trasformazione dei concept progettuali in componenti edilizi pronti per la fabbricazione, garantendo al contempo accuratezza geometrica, costruttibilità ed efficienza nello scambio delle informazioni. Questa tesi propone un flusso di lavoro computazionale integrato che connette la progettazione concettuale assistita dall'intelligenza artificiale, la modellazione computazionale, la fabbricazione digitale e il Building Information Modelling (BIM) all'interno di un processo continuo dal progetto alla produzione.

La metodologia ha inizio con la generazione di concept di facciata mediante modelli di intelligenza artificiale text-to-image, image-to-image e image-to-3D. I concept selezionati vengono successivamente ricostruiti come superfici NURBS a forma libera e sviluppati attraverso un flusso di lavoro orientato alla fabbricazione implementato in Grasshopper. Il processo discretizza automaticamente superfici a doppia curvatura in componenti metallici interconnessi costituiti da fasce sviluppabili, integrando la generazione della sottostruttura portante, la definizione automatica dei dettagli costruttivi, l'estrazione delle informazioni di piegatura, lo sviluppo piano dei componenti, il nesting dei pezzi e la preparazione alla fabbricazione.

Per favorire l'interoperabilità nel settore Architecture, Engineering and Construction (AEC), il modello computazionale è integrato con Autodesk Revit tramite Rhino.Inside.Revit, consentendo la generazione automatica di famiglie BIM native e della relativa documentazione di fabbricazione, mantenendo la coerenza tra il modello geometrico e il modello informativo.

La metodologia è validata mediante la realizzazione e l'assemblaggio di un prototipo fisico in scala 1:20, confermando la fattibilità del flusso di lavoro proposto e consentendo il perfezionamento del modello computazionale attraverso il riscontro derivante dal processo di fabbricazione.

Sviluppato come LOUVEN+, il framework proposto estende il sistema di facciata LOUVEN, originariamente avviato da INDEXLAB presso il Politecnico di Milano, sostituendo la discretizzazione diretta della superficie con una ricostruzione orientata alla fabbricazione basata su superfici a fasce sviluppabili. Il workflow automatizza inoltre la definizione dei dettagli costruttivi, la documentazione di fabbricazione e l'integrazione BIM, fornendo una solida base per futuri flussi di lavoro assistiti dall'intelligenza artificiale destinati alla progettazione e alla produzione di facciate architettoniche a forma libera.

parole chiave: intelligenza artificiale (ia); generative design; architettura free-form; computational design; band strips; discretizzazione delle superfici; superfici sviluppabili; fabbricazione digitale; building information modelling (bim); facciata adattiva

Content

Table of Content

Abstract 1

Riassunto in lingua italiana 2

Content 4

1. Introduction 6

1.1. Research Problem 6

1.2. Research scope and goals 8

1.3. Thesis structure 10

2. Literature Review 11

2.1. Geometry Exploration 11

2.1.1. AI text to Image 11

2.1.2. AI image to 3D model 23

2.1.3. ComfyUI Platform 32

2.1.4. Deferential & Discrete Deferential Geometry 37

2.1.5. Surface Typologies and Developability 39

2.1.6. Strips discretization of surfaces 47

2.1.7. 3D Geometry Representation Mesh, SubD & NURBS 50

2.1.8. Computational Design & Grasshopper 56

2.2. Digital Fabrication Overview 57

2.2.1. Unrolling and Nesting for Fabrication 59

2.2.2. BIM Automatize Documentation 61

3. Methodology 63

4. Project 66

4.1. AI 3D Model Generation (Comfy UI) 66

4.1.1. Text to Image 66

4.1.2. Image to Image 68

4.1.3. LORA image training 70

4.1.4. Image to 3D mesh model 75

4.2. Geometry Build-up 77

4.2.1. High Poly Mesh to single smooth surface 77

4.2.2. Gaussian curvature: principal understanding 83
4.2.3. Surface Band Strips-discretisation (Main Grasshopper Logic) 91
4.2.4. Geometry Detailing 102
4.2.5. Unrolling and Nesting for Fabrication 105
4.3. BIM Integration and Automated Documentation 111
4.4. Physical Prototype Validation 113
5. Conclusion & Future Developments 116
5.1. Discussion 116
5.2. Future Research and Development 117
5.3. Conclusion 119
6. Bibliography 120
List of Figures 124
List of Graphs 126
List of Tables 126
Acknowledgment 127

1. Introduction

1. Introduction

1.1. Research Problem

The increasing adoption of free-form and organic architectural design has significantly expanded the geometric complexity of contemporary building envelopes. While advances in computational modelling enable architects to create highly expressive double-curved geometries, translating these forms into fabrication-ready building components remains a considerable challenge. Such geometries are not only difficult to model and document but also require computational rationalisation before they can be manufactured using conventional sheet-based fabrication techniques. In practice, free-form surfaces often need to be reconstructed as developable or near-developable geometries or discretised into manufacturable components while preserving the intended architectural form. **Consequently, there is a need for computational workflows that integrate AI-Assisted geometric modelling, fabrication-aware rationalisation, detailing, and documentation within a unified process.**

Recent advances in artificial intelligence have introduced new opportunities for conceptual architectural design. Generative AI models are now capable of producing a wide range of architectural concepts from textual descriptions or reference images and, more recently, converting these concepts into preliminary three-dimensional models. **However, current AI-generated geometry lacks the geometric precision, topological consistency, and parametric structure required for computational analysis and digital fabrication.** As a result, AI-generated models cannot yet be incorporated directly into architectural production workflows and require extensive computational reconstruction before they become suitable for engineering, fabrication, and documentation.

The growing complexity of architectural projects further challenges existing workflows within the Architecture, Engineering, and Construction (AEC) industry. Many fabrication-oriented tasks—including surface discretisation, component segmentation, structural detailing, connection definition, unrolling, and fabrication preparation—remain highly dependent on manual modelling or disconnected software environments. These fragmented workflows increase modelling time, introduce opportunities for human error, and reduce the ability to rapidly evaluate design alternatives throughout the design process.

In parallel, fabrication documentation has become increasingly important as architectural production shifts toward prefabrication and digital manufacturing. Manufacturing sheet-based façade systems requires not only accurate geometric representations but also fabrication-specific information, including connection details,

bending instructions, component identification, nesting layouts, and assembly logic. **Maintaining this information consistently between computational models, fabrication layouts, and Building Information Modelling (BIM) environments remains a significant interoperability challenge.** Although computational modelling and BIM platforms have individually matured, seamless integration between design generation, fabrication preparation, and BIM documentation is still limited.

Accordingly, this research addresses the broader problem of establishing a continuous computational design-to-production workflow capable of transforming AI-generated free-form architectural concepts into fabrication-ready and BIM-integrated façade systems. The workflow aims to minimise manual intervention while maintaining geometric consistency, fabrication feasibility, and information continuity throughout the entire process.

The research problem can therefore be summarised through the following key challenges:

- **Integration of Artificial Intelligence into Architectural Design:** Establishing a workflow in which AI-generated conceptual designs can be transformed into computationally editable architectural geometry suitable for further development.
- **Bridging AI-Generated Geometry and Fabrication Workflows:** Developing computational methods that reconstruct AI-generated models into precise parametric surfaces compatible with fabrication-oriented modelling.
- **Computational Rationalisation of Complex Free-Form Surfaces:** Addressing the geometric challenges associated with modelling and fabricating double-curved surfaces through computational discretisation into interconnected near-developable band-strip systems.
- **Automation of Fabrication Preparation:** Reducing manual modelling by automating surface discretisation, structural detailing, component identification, bending information extraction, unrolling, and material nesting within a unified computational workflow.
- **Integration of Fabrication and BIM Documentation:** Establishing interoperability between computational modelling and BIM environments to automate fabrication documentation while maintaining consistency between geometric models, fabrication information, and project documentation.

1.2. Research scope and goals

This research investigates the integration of **artificial intelligence**, **computational design**, **digital fabrication**, and **Building Information Modelling** (BIM) into a **unified computational workflow** for the design and production of **free-form architectural facade systems**. The scope of the research extends from the early conceptual design stage, where artificial intelligence assists in generating architectural forms, to the automated preparation of **fabrication-ready components** and **BIM-based documentation**. Rather than focusing on a single computational technique, the research explores how multiple digital technologies can be integrated into a continuous design-to-production workflow that minimizes manual intervention while maintaining geometric consistency, fabrication feasibility, and information continuity.

The primary goal of the research is to develop an integrated computational framework capable of transforming AI-generated architectural concepts into fabrication-ready free-form facade systems based on interconnected band strips. To achieve this objective, the research combines recent advances in generative artificial intelligence, computational geometry, fabrication-aware modelling, digital fabrication, and BIM interoperability within a single parametric workflow.

The first objective is to develop an AI-assisted conceptual design workflow using the ComfyUI platform. The workflow integrates text-to-image, image-to-image, and image-to-three-dimensional generative models to support the rapid generation and exploration of free-form architectural concepts from designer-defined prompts.

The second objective is to establish a Grasshopper-based computational workflow for extracting, refining, and remodelling the selected geometry into a continuous NURBS surface suitable for computational processing. This stage converts AI-generated concepts into editable parametric geometry that supports geometric analysis and subsequent fabrication-oriented operations.

The third objective is to develop a computational methodology for generating an interconnected band-strip facade system from free-form surfaces. The workflow reconstructs the reference surface into a network of near-developable band strips connected through a continuous branching logic, preserving the intended architectural form while satisfying the geometric requirements of sheet-based fabrication.

The fourth objective is to transform the continuous band-strip geometry into manufacturable components through automated structural detailing and fabrication preparation. This includes the generation of triangulated structural backbones, connection plates, fastening system, bending information, component unrolling, and material nesting, producing fabrication-ready assemblies compatible with digital sheet-metal fabrication processes.

The final objective is to establish an interoperable BIM documentation workflow by integrating Grasshopper with Autodesk Revit through Rhino.Inside.Revit. This enables the automatic generation of native BIM families, fabrication documentation, drafting views, and associated metadata, ensuring consistency between the computational model, fabrication information, and project documentation.

Accordingly, the scope of this research is limited to the development, implementation, and validation of the proposed computational framework through digital modelling and the fabrication of a physical prototype. The research focuses on the integration of AI-assisted conceptual design, computational geometry, fabrication-aware modelling, digital fabrication, and BIM documentation. While topics such as environmental optimisation, robotic fabrication, and automated construction are considered beyond the scope of the present study and are identified as opportunities for future research.

1.3. Thesis structure

Abstract

It provides a clear overview of the study's scope.summary of the entire research, outlining the main objectives, methodology, findings, and conclusions.

1. Introduction

Introduces the research topic, research problems and outline thesis scope and gaols and gives an overview to the thesis structure.

2. Literature Review

Reviews the state of the art in existing research, theories, and methodologies related to AI-assisted design generation, computational geometry, surface optimization, digital fabrication, and BIM-based documentation, establishing the theoretical and academic foundation for the proposed research.

3. Methodology

Describes processes used to achieve the study's objectives from the form generation to the final documentation. It explains the workflow, design decisions software platforms, parametric algorithms and analytical or experimental methods employed to develop and test the proposed framework.

4. Project

Presents the practical application of the research through an experimental design model. This section demonstrates the implementation of the proposed workflows or systems within a hypothetical architectural context.

5. Conclusion and Future developments

Analysis and showcases the outcomes of the study, reflecting on their implications, limitations, and relevance to the research questions. It concludes with key findings, contributions to knowledge, and recommendations for future research.

2. Literature Review

2. Literature Review

Reviews the state of the art in existing research, theories, and methodologies related to AI-assisted design generation, computational geometry, surface optimization, digital fabrication, and BIM-based documentation, establishing the theoretical and academic foundation for the proposed research.

2.1. Geometry Exploration

Introduction The exploration and representation of architectural geometry have undergone significant transformation with the emergence of artificial intelligence, computational design, and advanced geometric modelling techniques. Contemporary design workflows increasingly integrate AI-assisted conceptual generation with parametric modelling and computational geometry, enabling architects to explore complex free-form forms while maintaining control over their geometric properties and fabrication feasibility. Consequently, geometry is no longer regarded solely as a final design outcome but as a computational entity that can be analysed, manipulated, rationalised, and translated into manufacturable components.

This chapter reviews the principal concepts and technologies that underpin the proposed computational workflow. It begins by examining recent advances in AI-assisted geometry exploration through text-to-image, image-to-image, and image-to-three-dimensional generation techniques, followed by an overview of the ComfyUI platform as a flexible environment for integrating generative AI models. The discussion then shifts to the mathematical foundations of computational geometry, introducing differential and discrete differential geometry together with the principal methods of three-dimensional geometric representation, including polygon meshes, subdivision surfaces, and Non-Uniform Rational B-Splines (NURBS). Finally, the chapter reviews surface typologies, surface discretisation strategies, and the role of computational design within the Grasshopper visual programming environment.

Collectively, these topics establish the theoretical foundation required to understand how conceptual free-form geometries can be represented, analysed, rationalised, and prepared for the computational and fabrication-oriented workflow developed in the subsequent chapters.

2.1.1. AI text to Image

Text-to-image (TTI) generation is a branch of generative artificial intelligence concerned with synthesizing visual content directly from natural language descriptions. A user supplies a text prompt — ranging from a single noun to a detailed paragraph — and a trained model produces a corresponding image.

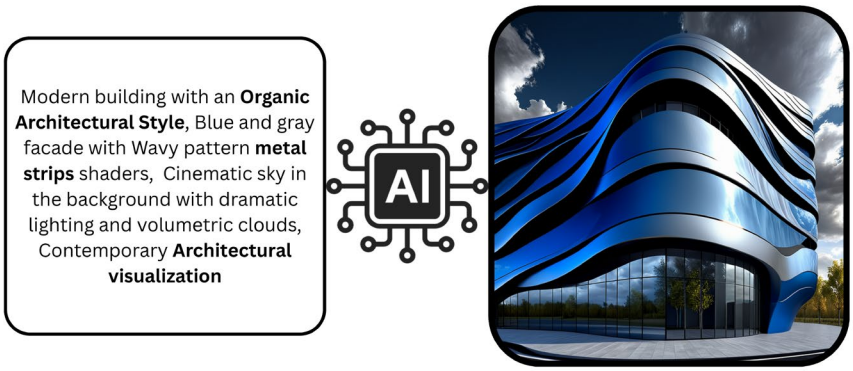


Figure 1: Text Prompt to Image Generation

The field has undergone remarkable transformation over the past decade. What began as a research curiosity producing coarse, low-resolution outputs has become a huge rapidly developing industry, with platforms such as Midjourney, DALL-E, and Stable Diffusion reshaping creative workflows not only in architecture but also in design, marketing, filmmaking, and scientific communication.

This review is structured as follows: Section 1 traces the historical development of TTI. Section 2 explains the core machine learning concepts. Section 3 surveys current commercial platforms.

2.1.1.1. Historical Development of Text-to-Image Generation

As noted in the survey RenAIssance: A Survey into AI Text-to-Image Generation in the Era of Large Mode: "Text-to-image generation using neural networks could be traced back to the emergence of Generative Adversarial Networks (GAN), followed by the autoregressive Transformer. Diffusion models are one prominent type of generative model used for the generation of images through the systematic introduction of noises with repeating steps" [1]

1) Pre-GAN Era (Before 2014)

Prior to Generative Adversarial Networks, generative models for images relied on Boltzmann machines, deep belief networks, and Variational Autoencoders (VAEs). These approaches suffered from computational intractability or training instability and could not reliably produce high-fidelity image. The outputs were blurry and lacked semantic coherence, making text-guided generation practically infeasible.

2) The GAN Revolution (2014–2019)

The field was transformed on June 10, 2014, when Ian Goodfellow and colleagues published Generative Adversarial Nets on[2]. Goodfellow conceived the idea at a bar in Montreal and coded the first working prototype within hours that same night.

GANs introduced a game-theoretic framework pitting two neural networks against each other: a Generator (G) that produces synthetic images from random noise, and a Discriminator (D) that attempts to distinguish real from fake images. Training is a minimax game in which G tries to fool D while D improves its discrimination ability.

Early GAN variants relevant to TTI include:

- Conditional GAN (cGAN, 2014): Extended GANs to condition generation on class labels or text, enabling class-directed synthesis.
- DCGAN (2015): Used deep convolutional networks for more stable and higher-quality image synthesis.
- StackGAN (2017): Stacked two GANs in sequence — Stage-I to generate a low-resolution sketch, Stage-II to refine it. This was among the first credible text-to-image GAN systems.
- BigGAN (2018–2019): Demonstrated large-scale high-resolution image generation, establishing quality benchmarks for subsequent work.

3) The Transformer Era and CLIP (2017–2021)

Originally invented by Google researchers for text translation in 2017, the Transformer architecture [3] is the core foundational engine behind all modern AI which opened a path to learning joint text-image representations. The key leap came in January 2021 with two parallel breakthroughs from OpenAI:

DALL-E (January 2021): OpenAI's first TTI system used a discrete VAE to tokenize images into a grid of visual tokens, then trained a 12-billion parameter autoregressive Transformer to jointly model text and image tokens. It could generate surprisingly coherent imagery from creative prompts but was limited to 256×256 resolution.

CLIP (January 2021): Contrastive Language-Image Pretraining learned to embed both images and text in a shared latent space using 400 million image-caption pairs scraped from the internet. CLIP became the backbone for aligning textual descriptions with visual concepts and was incorporated into virtually every major TTI system that followed. [4]

4) The Diffusion Model Breakthrough (2020–2022)

Diffusion Probabilistic Models (DPMs) are a class of generative machine learning models that generate data by iteratively removing noise from a distribution. They were first introduced in 2015 by Jascha Sohl-Dickstein and his team in their seminal research paper, "Deep Unsupervised Learning using Nonequilibrium Thermodynamics."

While GANs dominated early image synthesis, they suffered from training instability and mode collapse. DPMs offered a more stable alternative. Jonathan Ho, Ajay Jain, and Pieter Abbeel's Denoising Diffusion Probabilistic Models [5] established the modern diffusion framework: a forward process gradually corrupts an image by adding Gaussian noise across many timesteps, and a reverse process learns to denoise step by step, ultimately generating a high-quality image from pure noise.

The DDPM approach proved dramatically more stable to train than GANs and was more suitable to conditioning on text embeddings. Key subsequent milestones include:

- GLIDE (December 2021): OpenAI combined a diffusion model with classifier-free guidance and showed that text-guided diffusion significantly outperformed DALL-E v1 at the same scale.
- DALL-E 2 / CLIP (April 2022): Used CLIP image embeddings as conditioning for a diffusion model. A "prior" network maps CLIP text embeddings to CLIP image embeddings, which then conditions the diffusion decoder. This produced markedly higher-quality and more semantically faithful images[6]
- Imagen (May 2022, Google): Swapped a custom text encoder for a frozen T5-XXL large language model (4.6B parameters) and applied cascaded diffusion up sampling, achieving photorealistic images with deep language understanding.
- Stable Diffusion / Latent Diffusion Models (August 2022): [Rombach et al. 2022] moved diffusion into a compressed latent space using a VQ-GAN encoder, dramatically reducing computational cost while maintaining quality. This was released as an open-source model, enabling an explosion of community adoption and derivatives [7]

5) Large Model Scaling and Multimodal LLMs (2023–2026)

Since 2023, TTI has converged with large language model (LLM) scaling. Models are now trained on billions of image-text pairs, incorporate advanced attention mechanisms, and are increasingly integrated into multimodal conversational systems.

Major developments include:

- DALL-E 3 (2023): Dramatically improved prompt-following by training on DALL-E 3-generated captions, achieving far greater semantic fidelity to complex prompts.
- Midjourney v5/v6 (2023–2024): Closed-source model celebrated for its aesthetic quality and stylistic range.

- Stable Diffusion 3.x / FLUX (2024–2025): Introduced Multimodal Diffusion Transformers (MMDiT) and rectified flow matching, producing cleaner prompt adherence and improved text rendering in images.
- GPT Image 1/2 (2025–2026): Integrated TTI directly into conversational LLMs, enabling iterative refinement through dialogue.
- Imagen 4 / Veo (Google, 2025–2026): Extended high-fidelity generation to video, marking the beginning of the text-to-video era.

2.1.1.2. Core Machine Learning Concepts and Definitions

1) Generative Models

A generative model is a statistical model that learns the underlying probability distribution P(X) of a dataset X and can draw new samples from that distribution. In the context of TTI, the model learns P (image I text_prompt) — the distribution of plausible images given a textual description.

Generative models fall into several families:

- Explicit density models: Directly parametrize and maximize P(X). Examples: VAEs, autoregressive models (DALL-E v1, PixelCNN).
- Implicit density models: Learn to sample from P(X) without explicitly representing it. Example: GANs.
- Score-based / Diffusion models: Model the gradient of the log probability (score function), enabling stable, high-quality generation.

2) Generative Adversarial Networks (GANs)

Introduced by [2], a Generative Adversarial Networks (GANs) are a framework for estimating generative models through an adversarial process, where two models are trained simultaneously in a minimax two-player game.

The system consists of two competing neural networks:

The Generative Model (G): This model captures the data distribution. Its goal is to map random noise (sampled from a prior distribution) into the data space to create synthetic samples.

The Discriminative Model (D): This model estimates the probability that a given sample came from the actual training data rather than from the generator.

3) Variational Autoencoders (VAEs)

Variational Autoencoders (VAEs) are a type of directed probabilistic model designed to perform efficient approximate inference and learning with large datasets. [8]

They are categorized as likelihood-based models that approximate a target data distribution explicitly.

A VAE consists of two primary neural network components:

Probabilistic Encoder (Recognition Model): This component maps input data into a low-dimensional "latent space". Instead of a fixed code, it produces a distribution over latent variables, typically an isotropic multivariate Gaussian.

Probabilistic Decoder (Generative Model): This component takes a sample from that latent distribution and attempts to reconstruct the original input.

A VAE encodes input data into a latent distribution, samples, then decodes these samples back into reconstructed data. Unlike deterministic autoencoders, the stochastic latent space enables controlled generation by sampling new values.

VAEs are critical to TTI systems: in Latent Diffusion Models (Stable Diffusion), a VQ-VAE or VQ-GAN first compresses images into a compact latent representation, and diffusion is performed in that lower-dimensional space, vastly reducing computation.[2]

4) Diffusion Probabilistic Models (DPMs)

Diffusion Probabilistic Models are a class of generative machine learning models that learn to produce data (such as images, audio, or text embeddings) by modelling a gradual denoising process. They are widely used in modern generative AI systems, especially image synthesis.

Diffusion models define a forward process (Forward Diffusion) that progressively adds Noise is progressively added to training data over many time steps. After sufficient steps, the data distribution approaches pure Gaussian noise.

A neural network (typically a U-Net) then learns the reverse denoising process (Reverse Diffusion), starting from random noise, the model iteratively removes noise to generate realistic samples.

As [Ho et al. 2020] note: "We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics" [5]. The score-based diffusion approach is more stable and easier to train than GANs and does not suffer from mode collapse.

5) Classifier-Free Guidance (CFG)

CFG (Ho & Salimans, 2022) is a technique for amplifying the influence of the text conditioning signal during diffusion sampling, without needing a separate classifier. The model is jointly trained with and without the text condition. At inference, the score estimate is:

$$\hat{\varepsilon} = \varepsilon_{uncond} + w \cdot (\varepsilon_{cond} - \varepsilon_{uncond})$$

where w is the guidance scale. Higher w produces images more faithful to the prompt but less diverse. CFG is now standard in virtually all modern TTI diffusion systems.

6) CLIP and Contrastive Learning

Contrastive Learning is a self-supervised or representation learning technique where a model learns meaningful feature representations by:

- Pulling similar samples closer in embedding space
- Pushing dissimilar samples farther apart

The objective is to learn discriminative latent representations without requiring extensive labelled data.

CLIP for Contrastive Language-Image Pre-training is a multimodal model developed by OpenAI that learns joint representations of images and natural language using contrastive learning.

CLIP [4] (Contrastive Language–Image Pretraining) trains an image encoder and a text encoder together using contrastive learning. The model increases the cosine similarity between correct image–caption pairs and decreases it for incorrect pairs. Trained on about 400 million image-text pairs collected from the web, CLIP learns a shared embedding space where images and text with similar meanings are placed close together.

These embeddings are widely used in modern generative AI systems. For example, CLIP embeddings help condition diffusion models in DALL·E 2, filter low-quality training data, and evaluate generated images using the CLIP Score, which measures how well an image matches a text prompt.

7) Transformers and Attention

The Transformer architecture [3], introduced by Ashish Vaswani and colleagues in 2017, replaced recurrent neural networks (RNNs) in many sequence modelling tasks. Its main innovation was the self-attention mechanism, which allows the model to measure relationships between all words or tokens in a sequence at the same time. Unlike RNNs, which process information step by step, Transformers can process sequences in parallel, making training faster and more effective for capturing long-range dependencies.

Self-attention works by assigning different importance weights to different tokens in the input sequence. For example, when processing a sentence, the model can learn which words are most relevant to each other in understanding meaning and context.

Transformers became the foundation of modern large language models (LLMs) and many multimodal AI systems.

In text-to-image (TTI) generation systems, a related mechanism called cross-attention is especially important. Cross-attention allows image-generation layers to interact directly with text embeddings produced by language encoders such as CLIP or T5. During image generation, the model repeatedly attends to different parts of the text prompt while constructing different spatial regions of the image.

For example, if the prompt is “a red car under a blue sky,” cross-attention helps the model associate:

“red” with the car region,
“blue” with the sky region,
and “car” with the object structure itself.
This mechanism enables the generated image to remain semantically aligned with the text description throughout the diffusion process.

8) Latent Diffusion / Stable Diffusion Architecture

The Latent Diffusion Model (LDM), introduced by Robin Rombach and colleagues in 2022[7], improves the efficiency of diffusion models by performing image generation in a compressed latent space instead of directly operating on full-resolution pixels.

The architecture can be understood in three main stages:

Stage 1 – Perception Compression

First, an encoder compresses a high-resolution image into a smaller latent representation. For example, a 512×512 RGB image can be compressed into a latent tensor of size 64×64×4.

This process greatly reduces the amount of information the diffusion model must process:

- the spatial size becomes 8× smaller in both width and height,
- while still preserving important visual details and semantic information.

The encoder is trained so that the compressed latent representation keeps the perceptual quality of the original image as much as possible. This means humans can still recognize the reconstructed image with little visible quality loss.

Stage 2 – Diffusion in Latent Space

Instead of adding and removing noise directly on image pixels, the diffusion process is performed in the compressed latent space.

A U-Net neural network learns:

- how noise is gradually added to latent representations (forward diffusion),
- and how to remove the noise step by step (reverse diffusion).

During generation, the model is conditioned on text embeddings produced by language models such as CLIP or T5. This conditioning is applied through cross-attention layers, allowing the model to continuously relate image regions to words in the prompt.

For example:

- the word “cat” guides object structure,
- “white” guides color,
- and “snowy background” influences scene composition.

Because diffusion happens in compressed latent space rather than full pixel space, computation and memory costs are greatly reduced — often by tens of times compared to pixel-space diffusion models.

Stage 3 – Image Decoding

After the denoising process is complete, a decoder converts the clean latent representation back into a full-resolution image.

The decoder reconstructs the final 512×512 image while restoring fine visual details such as:

- textures,
- edges,
- lighting,
- and colors.

This final stage transforms the compact latent features into a realistic and visually coherent image.

Overall, the Latent Diffusion Model achieves high-quality image generation while being much more computationally efficient than earlier diffusion models that operated directly on pixels.

2.1.1.3. Current Commercial Platforms and Market Landscape (2026)

The commercial TTI landscape has matured considerably since 2022. As one recent analysis notes, there is now "no single best model for all cases" — the right choice depends on the specific use case, from artistic generation to commercial production to developer workflows [9]

Midjourney

Developer: Midjourney, Inc. (independent research lab)

Architecture: Proprietary; believed to use diffusion-based approaches with custom aesthetic fine-tuning

Strengths: Widely regarded as the benchmark for aesthetic output quality. "Midjourney remains the benchmark for



Figure 2: Midjourney Logo

aesthetic output in 2026... produces images that look intentional rather than accidental" [9] Midjourney v7 (released April 2025) extended artistic quality further.

Access: Discord-based interface (web UI no longer in beta); subscription plans from $10/month. Closed source.

DALL-E 3 / GPT Image 2 (OpenAI)

Developer: OpenAI

Architecture: CLIP-conditioned diffusion model; DALL-E 3 trained on re-captioned data for enhanced prompt fidelity; GPT Image integrates generation directly into the LLM

Strengths: Best-in-class prompt following. "DALL-E 3 holds its position because... it actually does what you tell it" [9] GPT Image 2 also achieves reliable text rendering within images.

Figure 3: Chat GPT Logo

Access: Integrated into ChatGPT (free and Plus tiers); API available via OpenAI platform.

Stable Diffusion / FLUX (Stability AI & Black Forest Labs)

Developer: Stability AI (Stable Diffusion 3.5); Black Forest Labs (FLUX)

Architecture: Latent Diffusion Models (SD 3.5: Multimodal Diffusion Transformer; FLUX: Rectified Flow Matching)

Strengths: Open source; self-hostable on consumer GPUs; maximum technical control; FLUX 1.1 Pro produces photorealistic imagery competitive with closed models [10]

Access: Model weights freely available on Hugging Face; commercial APIs via Stability AI and Replicate.

Adobe Firefly

Developer: Adobe Inc.

Architecture: Proprietary diffusion model trained exclusively on licensed and copyright-cleared content

Strengths: Only major platform with fully commercially safe content pipeline. Deep integration with Adobe Creative Suite (Photoshop, Illustrator, Express). "Adobe Firefly is the only fully copyright-safe option" [11]



Figure 4: Adobe Firefly Logo

Access: Included in Adobe Creative Cloud subscriptions; standalone Firefly web app with free tier.

Ideogram

Developer: Ideogram AI

Architecture: Proprietary diffusion model with specialized text-rendering capability

Strengths: Unmatched in rendering legible text within generated images — a longstanding weakness across all other TTI models. "Ideogram remains the easiest choice for text-heavy image designs" [10]

Figure 5: Ideogram Logo

Access: Web platform; free tier with 40 images/day; paid plans for higher resolution and commercial use.

Google Imagen / Veo

Developer: Google DeepMind

Architecture: Cascaded diffusion with frozen T5-XXL text encoder (Imagen); Diffusion Transformer for video (Veo)

Strengths: Deep language understanding from T5 pretraining; Imagen 4 offers strong text rendering. Veo 2 leads in AI video generation quality.

Access: Available via Google Gemini (Imagen integration); Vertex AI API for developers.



Figure 6: Google Gemini Logo

Platform Comparison Summary

The table below summarizes key differentiators across leading platforms:

Midjourney	DALL-E 3 / GPT Image 2	FLUX / Stable Diffusion	Adobe Firefly	Ideogram	Imagen 4
Best for artistic quality, aesthetic work, editorial and fantasy images.	Best for complex prompt following; beginner-friendly; conversational iteration.	Best for photorealism, technical control, open-source flexibility.	Best for commercial/brand-safe work; professional designer workflows.	Best for text-in-image designs, typography, posters.	Best for deep language understanding; strong text rendering.

Table 1: Brief Text-to-Image Platforms Comparison

Modern TTI systems mainly rely on diffusion models, but each platform focuses on different strengths, The “best” model depends on the user’s needs rather than overall capability alone.

2.1.2. AI image to 3D model

The ability to reconstruct a three-dimensional model of an object or scene from photographs is a long-standing goal in computer vision. Classical photogrammetry — using multiple calibrated cameras and geometric algorithms like **Structure-from-Motion** (SfM) and **Multi-View Stereo** (MVS) — achieves this for objects with rich texture and known camera poses but requires dozens to hundreds of input images and fails on reflective, transparent, or featureless surfaces.

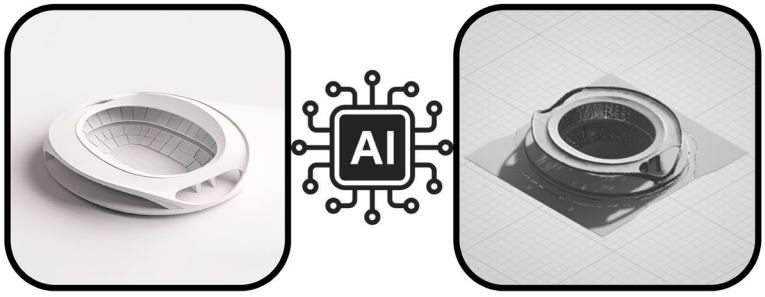


Figure 7: Image to 3D Model Generation

The emergence of deep learning and, subsequently, neural implicit representations fundamentally transformed this landscape. Beginning with the publication of **NeRF** in 2020 and accelerating through **DreamFusion** (2022), **3D Gaussian Splatting** (2023), and large feed-forward reconstruction models (2023–2025), researchers have achieved what was previously impossible: generating high-quality, textured 3D assets from a single photograph in seconds.

The connection to Previous section TTI of this review is direct and deep. The same diffusion model priors (Stable Diffusion, CLIP) that power text-to-image generation also serve as the generative backbone in most modern image-to-3D pipelines — either through Score Distillation Sampling, as multi-view diffusion priors, or as texture generators for raw geometry. The 3D field is, in many ways, built atop the 2D generation revolution described previously.

2.1.2.1. Classical Photogrammetry and Multi-View Stereo

Before deep learning, 3D reconstruction relied on two classical pipelines:

- Structure-from-Motion (SfM): Extracts key points (SIFT, ORB) from multiple images, matches them across views, and estimates camera poses and a sparse 3D point cloud via bundle adjustment. COLMAP is a popular open-source photogrammetry and 3D reconstruction system developed by Johannes Lutz Schönberger and collaborators. (Schonberger & Frahm, 2016) [12]It implements both:
  - SfM (camera pose estimation + sparse point cloud)
  - MVS (dense reconstruction)
- Multi-View Stereo (MVS): Given known camera poses from SfM, MVS densifies the sparse point cloud into a dense depth map and surface mesh using photometric consistency between views.

These pipelines work well for textured, Lambertian surfaces — A Lambertian surface is an ideal diffuse surface that reflects light equally in all viewing directions. — with many input views, Classical SfM/MVS methods require multiple overlapping views and therefore cannot reconstruct a complete 3D model from a single image alone. As well as for reflective materials, thin structures, and objects lacking key points. They also require minutes to hours of computation for a single model.

2.1.2.2. The Deep Learning Shift

Deep convolutional networks changed the paradigm starting around 2018. Systems like Pixel2Mesh (Wang et al., 2018) [13] learned to deform a template sphere into an object shape by mapping image features to mesh vertices. Occupancy Networks (Mescheder et al., 2019) [14] and DeepSDF (Park et al., 2019) [15]learned implicit continuous representations of 3D geometry using neural networks.

However, the most transformative development came with neural rendering — specifically, with NeRF in 2020 — which provided a differentiable, photorealistic 3D representation that could be optimized directly from posed 2D images without any explicit 3D supervision.

2.1.2.3. Core Machine Learning Concepts and Definitions

Neural Radiance Fields (NeRF)

NeRF (Neural Radiance Fields) is a deep-learning-based method for creating a 3D representation of a scene from a collection of images taken from different viewpoints.

Introduced by (Mildenhall et al. 2020) at ECCV [12], NeRF represents a 3D scene as a continuous 5D function: F (x, y, z, θ, ϕ) → (RGB color, volume density σ). Here (x, y, z) is a 3D spatial coordinate and (θ, ϕ) is the viewing direction. The function F is parameterized by a multi-layer perceptron (MLP).

To render an image from a given camera viewpoint, NeRF uses classical volume rendering: rays are cast from the camera through each pixel, N 3D points are sampled along each ray, the MLP predicts (colour, density) at each point, and the pixel color is the accumulated weighted sum along the ray. Because this rendering process is differentiable, the MLP weights can be optimized directly from a set of 2D photos with known camera poses using photometric reconstruction loss.

Key properties: NeRF can generate highly realistic images of a scene from new viewpoints, including effects such as reflections and shiny surfaces that change with the viewing angle. Instead of storing a scene as a traditional 3D mesh, NeRF stores the scene inside a neural network, with the 3D information encoded in the network's learned parameters (weights).

Limitations: Training a single NeRF scene requires 100–200 posed images and hours of optimization. Inference (rendering) is slow due to iterative MLP queries per ray sample. However, converting a NeRF into a traditional 3D mesh for further applications is not straightforward.

[1] 3D Representations: An Overview

AI 3D generation systems use several different 3D representations, each with trade-offs:

- Voxel Grids: Regular 3D pixel arrays. Simple but cubic memory cost (N^3). Limited resolution at feasible memory budgets.
- Point Clouds: Sets of (x, y, z) points, optionally with normals and colors. Flexible and simple but lack explicit surface connectivity.
- Meshes: Triangulated surfaces — the standard representation for rendering pipelines (Unity, Unreal, Blender, 3D printing). Explicit and efficient but difficult to optimize directly with gradient descent due to discrete topology.

- Implicit Functions (NeRF, SDF, Occupancy): Continuous volumetric representations encoded in neural network weights. Differentiable and topology-flexible but require conversion to mesh for downstream use.
- 3D Gaussian Splatting: Scenes represented as sets of 3D Gaussian primitives with position, covariance, color, and opacity. Enables real-time rendering via rasterization and differentiable optimization from 2D supervision.

[2] 3D Gaussian Splatting (3DGS)

Introduced by (Kerbl et al. 2023) [16] and published in ACM Transactions on Graphics, 3DGS replaced NeRF's implicit MLP with an explicit set of 3D Gaussian primitives. Each Gaussian is defined by center position (x, y, z), a full 3D covariance matrix (encoding shape and orientation), opacity α, and colour (as spherical harmonic coefficients for view-dependence).

Rendering is performed via splatting: each 3D Gaussian is projected onto the 2D image plane as a 2D Gaussian, and they are composited front-to-back in a differentiable rasterization pipeline. This enables real-time rendering at 30–140 fps for complex scenes — dramatically faster than NeRF's iterative volume rendering.

Training follows the same paradigm as NeRF: given posed 2D images, minimize the difference between rendered and real images. The Gaussians are initialized from a sparse SfM point cloud and densified/pruned adaptively during training.

Impact on image-to-3D: 3DGS became the dominant 3D representation in generative methods from 2023 onward because it is both differentiable (enabling optimization from 2D losses) and renderable in real time (enabling production use). DreamGaussian, GaussianObject, and many subsequent works build on 3DGS.

[3] Score Distillation Sampling (SDS)

Proposed by (Poole et al. 2022) in DreamFusion [17], Score Distillation Sampling is the key technique for lifting 2D diffusion model priors into 3D generation. The core insight is rather than requiring a 3D dataset, we can use a pretrained 2D diffusion model as a critic.

The SDS gradient tells a 3D representation (NeRF or 3DGS) how to update so that its renderings look more like plausible samples from the 2D diffusion model conditioned on the input prompt or image. Concretely:

- Render the current 3D representation from a random camera angle to get image x.

- Add random noise to x to get noisy image X_t at a random timestep t.
- Ask the diffusion model: "what noise should be removed to make X_t look more realistic?" The difference between the diffusion model's noise prediction and the actual added noise gives the update gradient.
- Backpropagate this gradient through the differentiable renderer to update the 3D representation parameters.

SDS is powerful because it requires no 3D training data and leverages the rich visual priors learned by billion-parameter diffusion models. Its limitations are the "Janus problem" (multiple faces on a single object), over-saturation, and slow optimization (30–60 minutes per object).

[4] Large Reconstruction Models (LRM)

A different paradigm from SDS-based optimization is feed-forward reconstruction: train a large neural network on millions of image-3D pairs so that at inference time, a single forward pass produces a 3D model in under one second. (Hong et al. 2024) proposed LRM (Large Reconstruction Model) [18], which uses a Transformer encoder-decoder architecture: the image is encoded with a DINO-pretrained ViT, and a Transformer decoder predicts the parameters of a NeRF triplane representation. Trained on Objaverse's 800K+ 3D objects, LRM generalizes to in-the-wild images remarkably well.

This feed-forward paradigm was the basis for TripoSR (Tochilkin et al. 2024) [19], which generates a textured mesh from a single image in under 0.5 seconds — making real-time image-to-3D workflows industrially viable.

[5] Multi-View Diffusion

A complementary approach uses 2D diffusion models fine-tuned on multi-view image datasets to generate multiple consistent views of an object from a single input image. Zero-1-to-3 (Zero-1-to-3 Liu et al. 2023) [20] was the first major work: it fine-tuned Stable Diffusion on Objaverse's rendered multi-view images to produce novel-view synthetic images given one reference view and a camera pose delta. One-2-3-45 (One-2-3-45 Liu et al.,2024) [21] then used Zero-1-to-3's multi-view outputs as input to a multi-view reconstruction pipeline, achieving 3D mesh generation in 45 seconds without per-shape optimization.

More recent multi-view diffusion models (SyncDreamer, Wonder3D, Zero123++) have improved multi-view consistency by adding 3D-aware attention mechanisms that synchronize noise predictions across views during the denoising process.

2.1.2.4. Current Commercial Platforms (2025–2026)

The image-to-3D commercial landscape has matured rapidly. What required specialized hardware and hours of compute in 2022 can now be done in seconds via browser-based tools. As one 2026 platform review notes: "For 90% of use cases, single-image input produces excellent results in 30–120 seconds" [22].

The following survey covers the leading commercial and open-source platforms for AI-driven 3D asset generation, spanning feed-forward reconstruction, diffusion-based modelling, and photogrammetric capture.

[1] Tripo AI (Tripo3D)



Figure 8: Tripo 3D Logo

Developed by Tripo AI in collaboration with Stability AI, Tripo3D is built on the TripoSR v2.5+ architecture — a Transformer-based feed-forward reconstruction model extended with quad-mesh generation and multi-view input support. Its primary advantage is generation speed, producing textured 3D assets in under one second with PBR material output, animation-ready quad-mesh topology, and a one-click Universal Rig system. Native plugins for Blender, Unity, and ComfyUI further integrate it into professional pipelines. The platform reportedly enables creators to complete the full 3D pipeline — modelling, texturing, retopology, and rigging — up to 50% faster than comparable tools [23]. Output formats include GLB, FBX, OBJ, USDZ, and STL. A free tier is available under CC BY 4.0; commercial plans begin at approximately $15/month, with API access priced at $0.20–$0.40 per generated model.

[2] Meshy AI



Figure 9: Meshy AI Logo

Meshy is a proprietary diffusion-based platform employing a multi-stage pipeline that separates geometry generation from AI-driven texture synthesis. It is distinguished by high texture fidelity, particularly in preserving photographic detail, surface logos, and fine material patterns from input images [24]. Its Blender plugin is widely regarded among the strongest available for any AI 3D tool, and generation times consistently fall under one minute. The platform has found strong adoption among game developers and digital artists. Supported formats include FBX, GLTF/GLB, OBJ, and USDZ. A limited free trial is available; the Pro plan begins at $20/month.

[3] Luma AI (Genie / Imagine 3D)

Developed by Luma AI — also responsible for the Dream Machine video generation system — this platform combines NeRF-based and 3D Gaussian Splatting reconstruction with a text-to-3D capability through its Genie model. It is particularly well-suited to environmental and product visualization, offering high photorealism and direct export pipelines to Unity and Unreal Engine. Its NeRF reconstruction from multi-view video capture positions it as a practical alternative to traditional photogrammetry workflows, delivering near-instant production-grade results [25].

Figure 10: Luma AI Logo

Access is available via web application and API, with a free tier and paid Plus/Unlimited plans for commercial use.

[4] Rodin AI (DeemosTech / Hyper3D)

Rodin AI, developed by DeemosTech, is grounded in the CLAY and Rodin model architectures, both of which received Best Paper nominations at SIGGRAPH 2024 and 2025. The platform targets high-fidelity character and organic object generation, producing production-ready meshes with physically accurate PBR materials. Its CAST model extends this capability to full scene reconstruction from a single photograph, generating geometrically coherent and physically plausible environments [26]. Pricing is positioned at the premium end of the market, at approximately $99/month, with an accessible entry point via 3D AI Studio at $29/month.

Figure 11: Hyper 3D Rodin AI Logo

[5] Kaedim

Kaedim AI distinguishes itself through a hybrid pipeline that combines AI-driven geometry generation with a human-in-the-loop quality assurance stage. This approach ensures clean mesh topology and accurate PBR texturing — qualities that purely automated systems frequently fail to guarantee. The platform is specifically designed to convert 2D concept art and sketches into production-ready 3D assets, making it well-suited to professional game development and film VFX pipelines where topological correctness is a hard requirement. Pricing ranges from approximately $120–$150/month, with enterprise contracts available for high-volume production use.



Figure 12: Kaedim Logo

[6] Polycam

Polycam occupies a distinct category among the platforms surveyed: rather than generating 3D assets from generative priors, it reconstructs them from real-world physical capture. The platform leverages on iPhone 12 Pro and later devices, alongside photogrammetry and multi-view NeRF reconstruction from video. LiDAR integration yields exceptional geometric accuracy for architectural elements, manufactured products, and heritage objects. While not a generative AI tool in the strict sense, Polycam serves as a critical complement to generative workflows — providing high-fidelity real-object digitization where photographic accuracy is required. A free tier is available; the full plan is priced at approximately $10/month.



Figure 13: Polycam Logo

[7] Hunyuan3D 2.0 (Tencent AI Lab)

Hunyuan3D 2.0 is a large-scale open-source image-to-3D system developed by Tencent AI Lab, available freely via Hugging Face. It employs a diffusion-based architecture trained at significant scale, producing highly detailed textured meshes that are broadly competitive with leading commercial platforms. It has been identified as the quality leader among free and open-source image-to-3D tools in 2025 comparative evaluations [22] For practitioners requiring production-quality 3D generation without licensing costs, Hunyuan3D 2.0 represents the strongest available open-source option.

Figure 14: Tencent Hunyuan 3D Logo

2.1.2.5. Brief image-to-3D Platforms Comparison

Tool	Main Strength	Input Type	Output Quality	Best Use Case
Luma AI	NeRF + Gaussian Splatting scene reconstruction	Video / images	Very high realism	Real-world buildings, interiors, environment capture
Polycam	LiDAR + photogrammetry scanning	Phone scan / LiDAR / images	High geometric accuracy	Architectural surveying, as-built capture
Rodin AI	High-quality generative 3D meshes	Single image	Very detailed	Concept architecture, stylized or realistic structures
Meshy AI	Strong texture + asset pipeline	Single image / text	Good visual quality	Game/visualization-ready architectural assets
Tripo AI	Extremely fast generation pipeline	Single image	Medium-high	Rapid architectural prototyping
Hunyuan3D	Open-source high-quality diffusion-based 3D	Single image / text	High (strong geometric detail)	Free alternative for architectural modelling & research prototypes
Kaedim AI	Human-in-the-loop mesh correction	Image / concept art	High (clean topology)	Production-ready game/film/architectural assets requiring clean meshes

Table 2: Brief image-to-3D Platforms Comparison

Best for **speed** (<1 sec): TripoSR / Tripo3D v2.5

Best **textures**: Meshy AI

Best **photorealism / characters**: Rodin AI

Best **scene reconstruction**: Luma AI/Rodin

Best for **game animation** (rig-ready): Tripo AI (Universal Rig), Masterpiece Studio

Best **concept-art-to-3D**: Kaedim AI

Best **free open-source**: TripoSR, Hunyuan3D 2.0

Best for **scanning real objects**: Polycam (LiDAR)

Synthesized from platform benchmark reviews: (Modelfy 2025) [22], (3D AI Studio 2026) [24], (Tridimensi 2025) [25], and (WaveSpeed AI 2025) [27]. See also (Tochilkin et al. 2024) [19], for TripoSR open-source benchmarks.

2.1.3. ComfyUI Platform

What is ComfyUI?

ComfyUI is an open-source, node-based workflow platform used for building and running generative artificial intelligence (AI) pipelines. It supports tasks such as text-to-image generation, image editing, image-to-3D conversion, video generation, and audio synthesis.

Figure 15: ComfyUI Logo

ComfyUI was first released on GitHub in 2023 by the developer comfyanonymous. Since then, it has become one of the most widely used open-source workflow systems for diffusion-based AI models.

Unlike simplified AI platforms such as Midjourney or DALL·E, ComfyUI exposes the entire generation pipeline visually. Users build workflows by connecting independent processing blocks called nodes on a canvas. Each node performs a specific task, such as:

- Text encoding
- Image generation
- Diffusion sampling
- LoRA model loading
- ControlNet conditioning
- Background removal
- 3D mesh reconstruction

The workflow structure follows a Directed Acyclic Graph (DAG) design, where data flows step-by-step between connected nodes. This architecture gives users full control over the generation process.

Workflows are stored as portable JSON files and can also be embedded inside PNG image metadata, allowing workflows to be shared and reproduced easily.

One important technical advantage of ComfyUI is its partial re-execution system. When a workflow is modified, only the changed branches are recomputed instead of rerunning the entire pipeline. This significantly reduces processing time during experimentation.

ComfyUI supports Windows, Linux, and macOS systems and can run locally on consumer GPUs with relatively low VRAM requirements. Cloud-hosted deployment options also became available through ComfyUI Cloud in 2025.[28]

Position of ComfyUI Among Other Platforms

Here is a list of some ComfyUI Alternatives and a summary of their feathers and constraints:

- Automatic1111 (A1111): The prior dominant WebUI. Tab-based linear UI — quick to learn but opaque, no node graph, slow for multi-stage experimentation.
- InvokeAI: Node-based like ComfyUI, cleaner consumer UI, but less extensible for custom model integration.
- Fooocus: High-quality simplified SD frontend. Zero workflow flexibility — single fixed pipeline.
- Midjourney / DALL-E / Meshy: Closed, cloud-only platforms. No pipeline access, no local execution, no training capability.

To conclude, ComfyUI is currently considered one of the most flexible open-source interfaces for diffusion-based AI generation. Compared with Automatic1111, which uses a simpler tab-based interface, ComfyUI provides a fully visual node graph system that is more suitable for advanced and multi-stage workflows. Compared with InvokeAI and Fooocus, ComfyUI offers greater customization and broader support for research models and experimental pipelines. Unlike cloud-only systems such as Midjourney and DALL·E, ComfyUI enables local execution, custom model integration, and full workflow transparency [28].

A key technical advantage is its selective re-execution engine, which processes only modified branches of the workflow graph rather than re-running the entire pipeline — significantly reducing iteration time in complex multi-stage tasks [28]. For pipelines chaining image generation with 3D reconstruction, this architecture makes ComfyUI particularly well-suited compared to any available open-source alternative.

Importance of ComfyUI for Text-to-3D Pipelines

ComfyUI is especially important for modern text-to-3D workflows because it supports combining multiple AI models inside a single pipeline. Models such as SDXL, FLUX.1, ControlNet, LoRA adapters, TripoSR, TRELLIS, and Hunyuan3D can all be integrated using custom nodes. This modular design allows researchers and developers to build highly customized pipelines for image generation and 3D reconstruction.[29]

Another important advantage is transparency. Every intermediate stage of the pipeline, including masks, latent representations, depth maps, and multi-view outputs, can be inspected directly. This makes debugging and optimization much easier than in closed commercial systems. In addition, workflows can be grouped into reusable modules, allowing large pipelines to be reused efficiently.

ComfyUI also has a large community ecosystem with many shared templates and plugins available through GitHub, RunComfy, and OpenArt. Many complete text-to-3D

workflows are already publicly available. Since the software can run locally, it is also useful for organizations that require privacy and protection of intellectual property.[28]

Workflow Templates in ComfyUI for Generative image and 3D model

ComfyUI offers an extensive library of ready to use workflow templates spanning several generative AI domains, including text-to-image, image-to-image, and image-to-3D reconstruction. Through its modular architecture, these workflows can be interconnected, modified, and extended to form complex processing pipelines that address specific application requirements. This flexibility allows users to design customized solutions that align with particular objectives, ranging from image synthesis and editing to automated 3D asset creation and visualization.

Text-to-Image Workflows

Text-to-image (TTI) workflows form the foundation of most ComfyUI applications. These workflows generate images directly from textual prompts using diffusion or flow-based generative models. The basic SD 1.5 / SD 2.1 workflow consists of a CLIP text encoder, a KSampler for latent image generation, and a Variational Autoencoder (VAE) decoder for image reconstruction. This pipeline serves as the introductory workflow for understanding node-based image generation in ComfyUI [30].

More advanced workflows include SDXL + Lightning, which employs a two-stage architecture consisting of a base model and a refiner model to produce high-quality 1024 × 1024 images. The Lightning variant significantly accelerates inference by reducing the number of sampling steps, enabling near real-time image generation. Similarly, FLUX.1-dev / FLUX.1-schnell represents a new generation of open-source models based on rectified flow matching, offering superior image quality and efficiency. FLUX.1-dev prioritizes visual fidelity, while FLUX.1-schnell focuses on faster generation, and both are widely regarded as among the best open-source text-to-image models available in 2025–2026 [31].

Additional enhancements can be achieved through ControlNet, IP-Adapter / InstantID, and Stable Cascade (unCLIP) workflows. ControlNet enables structural guidance using depth maps, edge maps, or human pose information, making it valuable for architecture, engineering, and product visualization. IP-Adapter and InstantID allow reference images to influence style or preserve facial identity, while Stable Cascade introduces a multi-stage generation process that improves controllability and image variation through CLIP Vision conditioning [30],[31].

Image-to-Image Workflows

Image-to-image (I2I) workflows modify existing images while preserving their underlying structure. A standard Basic Img2Img pipeline encodes an input image into latent space, applies partial denoising through a KSampler, and reconstructs the final

image using a VAE decoder. The degree of transformation is controlled primarily through denoising strength and prompt guidance, allowing users to balance preservation and modification of the source image [30].

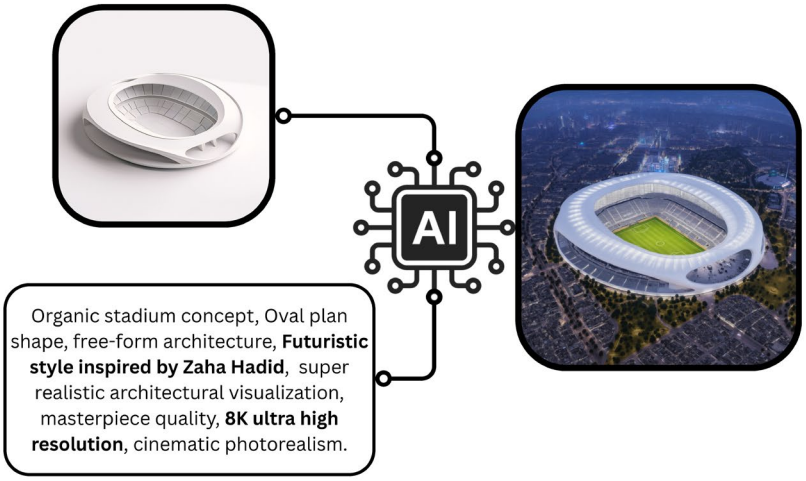


Figure 16: Image To Image Generation

Among the most commonly used image-editing techniques are Inpainting / Outpainting. Inpainting regenerates selected image regions using masks, whereas outpainting extends images beyond their original boundaries. These approaches are widely employed for object replacement, scene completion, and background modification.

LayerDiffuse further enhances image-to-image workflows by generating images with transparent alpha-channel backgrounds, which are particularly useful for product design and 3D asset creation. When combined with SDXL and ControlNet-based pipelines, it provides clean subject isolation without manual masking, simplifying downstream 3D reconstruction workflows.

Another notable development is IP-Adapter Plus Blending, which supports blending information from multiple reference images. This capability allows users to merge artistic styles, combine visual characteristics, and create controlled compositions from diverse image sources while maintaining consistency in the final output [30].

Image-to-3D and Text-to-3D Workflows

The rapid growth of immersive technologies has increased demand for workflows capable of converting images and text prompts into three-dimensional assets. Within the ComfyUI ecosystem, the ComfyUI-3D-Pack framework serves as the primary integration platform for modern image-to-3D and text-to-3D models. These workflows generate meshes, Gaussian splats, and other 3D representations suitable for applications in gaming, simulation, virtual reality, and digital content creation.

One of the most accessible solutions is TripoSR, which uses a streamlined workflow consisting of background removal followed by 3D reconstruction. The model can directly

generate GLB or OBJ files from a single image and can easily be integrated with text-to-image pipelines to create complete text-to-3D workflows [32], [19]. A widely adopted community workflow combines SDXL Lightning + LayerDiffuse + TripoSR, allowing users to generate a high-resolution image with a transparent background and subsequently reconstruct it into a 3D mesh. On modern consumer GPUs such as the RTX 3080, this process can be completed in less than ten seconds [19].

Recent advances have introduced more powerful reconstruction models. TRELLIS-2, developed by Microsoft, utilizes approximately four billion parameters to generate physically based rendering (PBR) meshes at resolutions up to 512^3 while achieving generation times of only a few seconds on high-performance hardware [33].

Similarly, Hunyuan3D 2.0, developed by Tencent, provides an open-source large-scale image-to-3D solution integrated into ComfyUI through the ComfyUI-3D-Pack framework [32].

Alternative approaches such as Wonder3D / CRM Multi-View, Large Multiview Gaussian Model (LGM), and Triplane Gaussian Transformer focus on improving reconstruction quality and speed. Multi-view methods generate several synthetic viewpoints from a single image before reconstructing the final mesh, leading to improved geometric consistency. Gaussian-based approaches instead generate 3D Gaussian representations that can later be converted into meshes, with some methods producing results in under ten seconds on consumer-grade hardware [32].

Intake

The review of recent advances in generative artificial intelligence highlights the growing potential of node-based workflows for integrating multiple generative models within a unified computational environment. Among the available platforms, ***ComfyUI*** *provides a highly flexible and transparent framework that supports the combination of* ***text-to-image****,* ***image-to-image****, and* ***image-to-3D models*** *through modular and reproducible workflows. Its* ***node-based architecture****, selective re-execution mechanism, and compatibility with a broad ecosystem of open-source models make it particularly suitable for research-oriented applications that require iterative experimentation and customization.*

Recent developments in open-source generative models, particularly the ***FLUX*** *family of models, have significantly improved the visual quality, prompt adherence, and semantic consistency of AI-generated imagery compared with earlier diffusion-based approaches. Furthermore, the integration of custom Low-Rank Adaptation (****LoRA****) models enables the specialization of these foundation models toward specific architectural languages or design intentions through lightweight fine-tuning, allowing the generation process to be guided toward targeted geometric characteristics while preserving the general capabilities of the base model. In parallel,* ***image-to-3D models***

*such as **Hunyuan3D 2.0** demonstrate the increasing maturity of AI-driven three-dimensional reconstruction, enabling the efficient transformation of conceptual imagery into editable geometric assets.*

Intake

*Collectively, these developments establish a robust foundation for **AI-assisted conceptual design**, where image generation, controlled style adaptation, and preliminary three-dimensional reconstruction can be combined within a **single computational environment**. This integrated approach significantly accelerates the early design exploration process while maintaining flexibility, reproducibility, and compatibility with subsequent computational modelling workflows.*

2.1.4. Deferential & Discrete Deferential Geometry

Traditional differential geometry describes the geometry of smooth surfaces through a function (f) that map a region in a 2D plane into a 3D subset in space. The differential (df) of this mapping function explains how to stretch or push and pull vectors from that plane onto the curved surface. This field allows designers to translate familiar operations from flat Euclidean space – such as differentiating and integrating – into more complex, curved 3D spaces.

In fact, Computers cannot store the infinite amount of information contained in a smooth surface; instead, they must store a finite amount of "integrated" data. As a core theme of Discrete Differential Geometry there is no single "correct" way to discretize a geometric quantity; rather, different methods are chosen based on which essential structures (such as topological or homological properties) need to be preserved at the discrete level. Coming from this brief introduction we can identify:

Differential Geometry (smooth case)

Differential geometry is a science that studies smooth curves and surfaces using calculus. It focuses on objects (Curves, surface) that are continuous and differentiable. And their Intrinsic properties like Curvature, distances and angles

As the previous mentioned example parametrization of a surface where the basic core idea of it is having a function that map a 2D element in 3D or vice versa:

$$f: M \subset \mathbb{R}^2 \to \mathbb{R}^3$$

where:
$\mathbb{R}^2$ is the 2D plane (points (u, v))
$\subset$ means that M is a subset of $\mathbb{R}^2$
M is a region inside that plane
(for example: a square, disk, or any open set)

While $\mathbb{R}^3$ is the 3D space

The function

$$f: M \to \mathbb{R}^3$$

assigns to each point (u, v) an point in 3D.

which is usually written as:

$$f(u,v) = (x(u,v),\ y(u,v),\ z(u,v))$$

Discrete Deferential Geometry (DDG)

Discrete differential geometry studies the same ideas of deferential geometry but on discrete objects like: Meshes (combinatorial surfaces), polygons, triangulated surfaces.

The basic simplifying assumption of Discrete differential geometry is that the shapes we want to study are manifold and finite pieces.[34]

Intake

The review of differential and discrete differential geometry identifies the principal concepts adopted in the subsequent computational methodology. Differential geometry provides the mathematical framework for representing and analysing continuous free-form surfaces through parameterisation, normal vectors, and curvature evaluation. Discrete differential geometry complements this framework by introducing methods for representing continuous geometry as finite interconnected elements while preserving geometric continuity and topological relationships. These principles underpin the discretisation of free-form surfaces, the generation of triangulated structures, and the approximation of developable geometries used throughout the computational workflow.

Note: The concepts introduced in this section are revisited and applied in Section 5.2.2 Gaussian Curvature: Principal Understanding, where their role in surface developability assessment and computational surface discretisation is discussed in greater detail. As Gaussian curvature and related geometric properties constitute fundamental concepts of differential and discrete differential geometry.

2.1.5. Surface Typologies and Developability

Double curved surfaces

A surface that curves in two different directions at any given point.
“A surface is, roughly speaking, the “outer shell” of a shape—for instance, you can think of a whole orange as a solid ball; its peel describes a spherical surface (especially if we consider an idealized peel with zero thickness)”.[34]

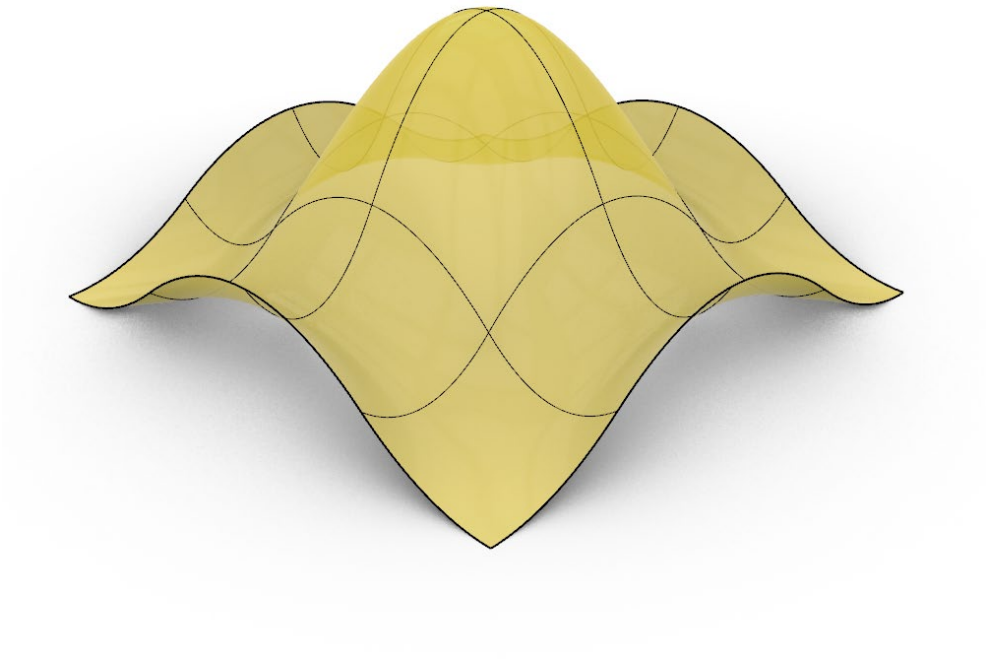


Figure 17: Double curved surface

Ruled Surfaces [35]

Linear Segment Discretization

Ruled Surfaces are geometric surfaces generated by moving a straight line (or generator) along one or more guiding curves. This technique is particularly valuable in architecture and engineering, as surfaces formed from linear elements can be manufactured and assemble more easily, reducing complexity in construction.

Ruled surfaces have the unique property that at every point on the surface, there exists at least one straight line that lies entirely on it. This property enables the creation of complex three-dimensional forms using simple linear elements.

Lofting

Lofting involves connecting two or more guiding curves with straight lines between corresponding points, creating a smooth transition between shapes.

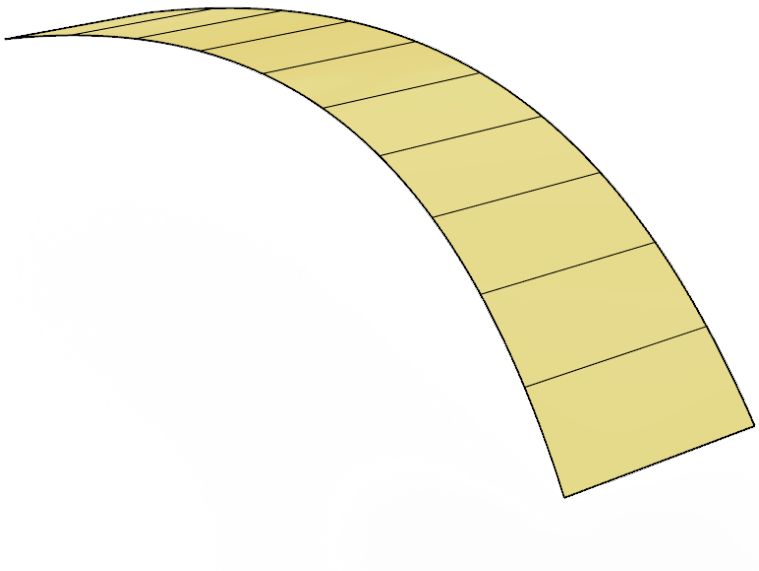


Figure 18: Surface Lofting

Rotating a Frame Around an Axis (Helicoid)

This method involves rotating a frame around a fixed central axis while simultaneously moving or translating it along that axis. Each point on the frame follows a helical or spiral path around the axis, resulting in a uniform spiral structure. The frame maintains a consistent rotational relationship with the central axis, creating a predictable and symmetrical helical surface.

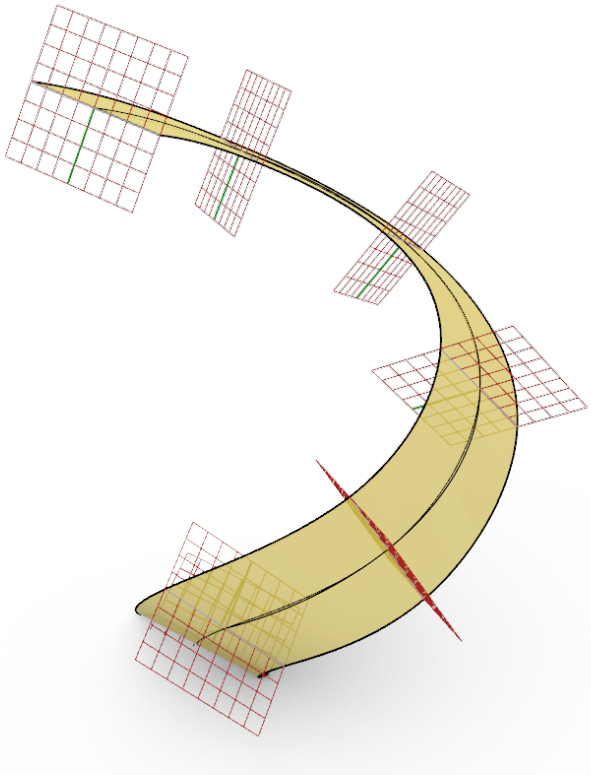


Figure 19: Rotating a Frame Around an Axis

Sweeping a Line Along a Curve

In this method, a straight line (or genera tor) is moved along a guiding curve (or directrix) while maintaining a fixed orientation relative to the curve. The line does not twist, tilt, or rotate as it follows the curve, creating a surface with consistent orientation. This results in simpler ruled surfaces where the line remains aligned in one direction throughout the sweep.

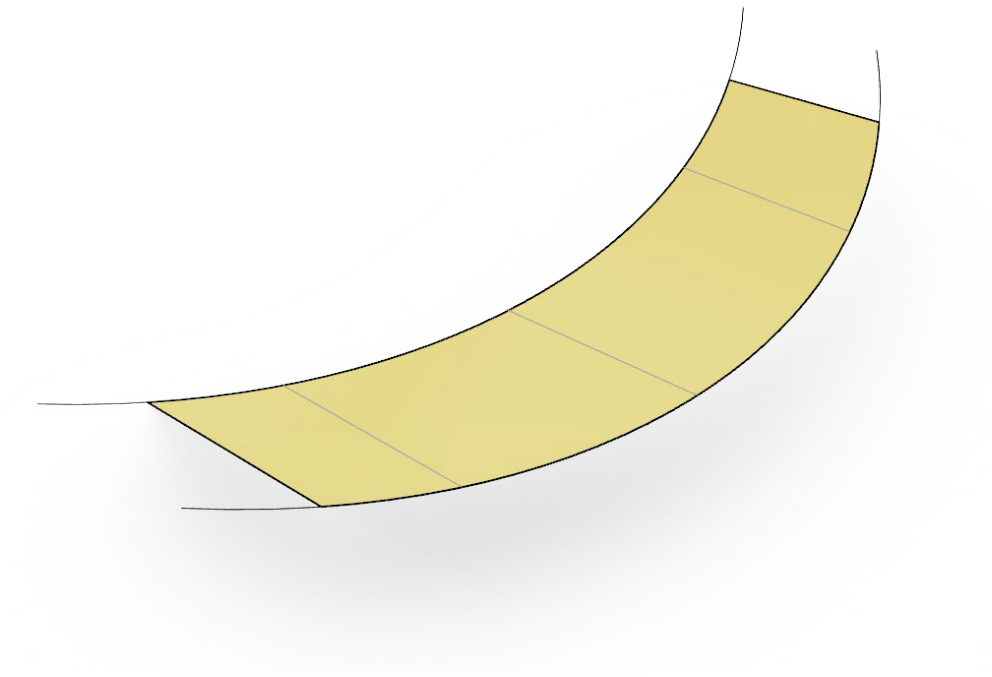


Figure 20: Sweeping a Line Along a Curve

Warped (or Twisted) Surface

A straight line moves along a guiding curve but changes its orientation gradually. creating a twisted or warped effect.

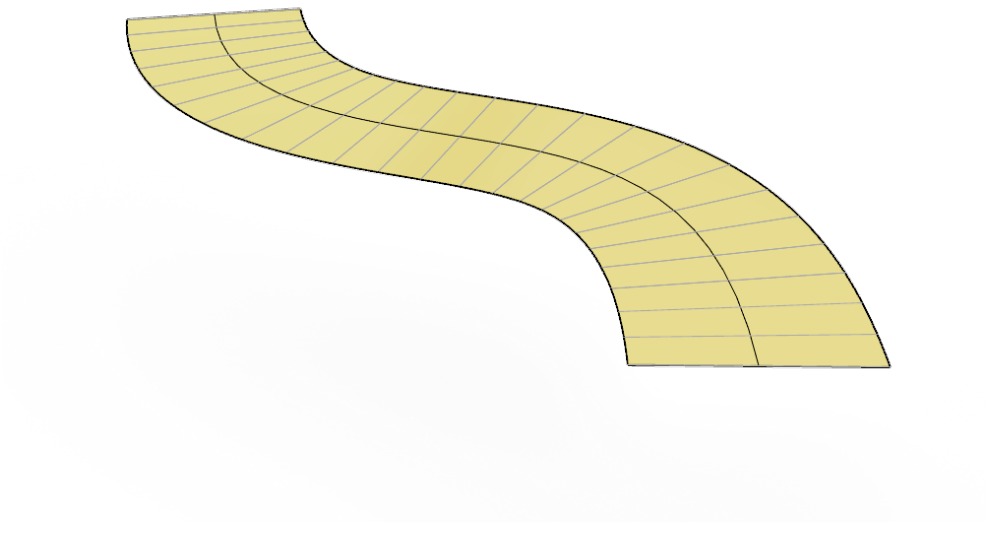


Figure 21: Warped (or Twisted) Surface

Hyperbolic Paraboloid

A hyperbolic paraboloid has a saddle shape, created by moving a straight line along two families of skew lines. It's a doubly ruled surface, meaning there are two distinct straight lines through each point on the surface.

The equation for a Hyperbolic Paraboloid is:

$$z = \frac{x^2}{a^2} + \frac{y^2}{b^2}$$

Where 'a' and 'b' control the curvature in the 'x' and 'y' directions, respectively.

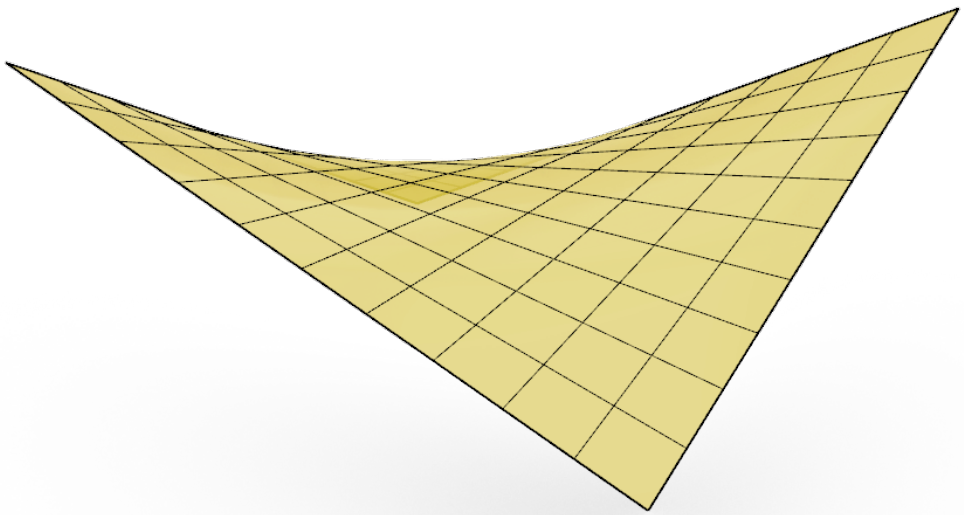


Figure 22: Hyperbolic Paraboloid Surface

Developable Surfaces [35]

Flattenable Panel Discretization

Developable surfaces are three-dimensional forms that can be unfold- ed into a two-dimensional plane without deformation, meaning they can be flattened without stretching, compressing, or twisting, Mathematically, these surfaces have zero Gaussian curvature (K-0) at every point. This property is particularly valuable in fields like architecture. industrial design, and naval and aerospace engineering, as it enables the creation of complex shapes using flat materials. Once assembled. these materials maintain their strength and functionality without deformation

Cylindrical Developable

This method involves forming surfaces by extruding a profile curve along a direction parallel to itself. In a cylindrical develop able surface, the rulings (straight lines that define the surface's structure) are parallel and follow the main axis of the cylinder. The surface can be flattened along its rulings, which makes it developable without any deformation. In the development (un- folded form), these rulings are represented as parallel lines, maintaining the right angle with the profile curve, which ensures that it can be easily unrolled.

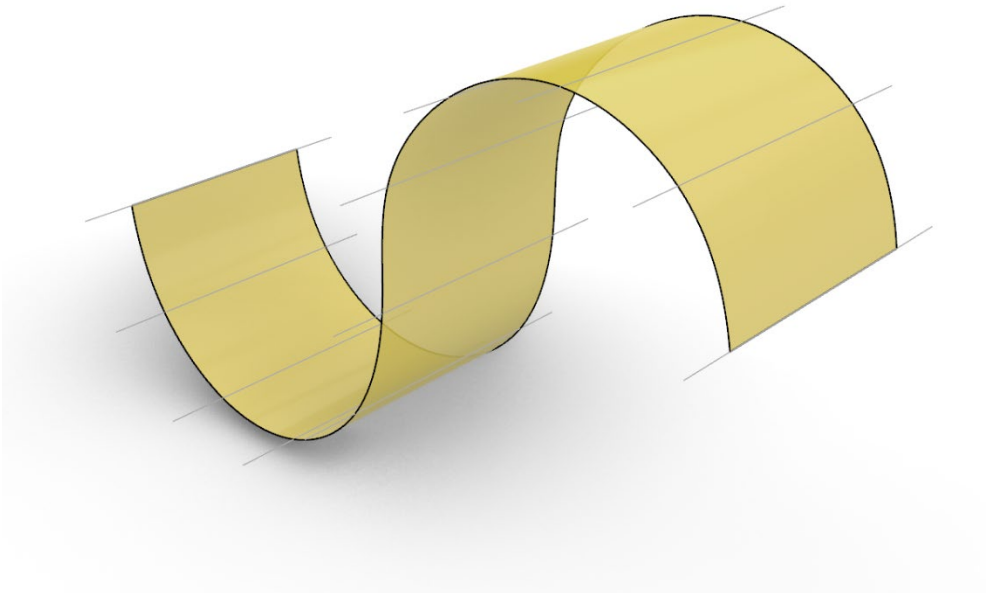


Figure 23: Cylindrical Developable Surfce

Conical Developable

Conical developable surfaces extend from a central point (vertex) and radiate outward, similar to the structure of a cone or a pyramid. In a discrete model, this can be represented by polygons that form a pyramid shape, as these polygons refine, the shape approaches a smooth cone. The development of a conical surface results in a radial, circular pattern cantered on the vertex, maintaining the total lengths of the curves.

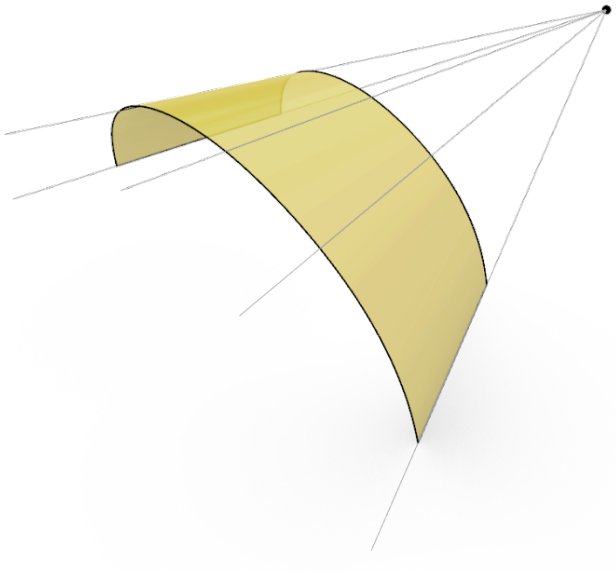


Figure 24: Conical Developable Surface

Tangential Developable

Tangential developable surfaces are generated by sweeping along a spatial curve with the surface rulings being tangential to the curve. This type of developable surface is highly adaptable to complex and organic shapes, making it especially suited for situations where the surface needs to follow an irregular, non-uniform path while remaining developable.

in the discrete case, this is approximated by a strip model, where planar quadrilateral strips (PQ strips) are used, with vertices along the curve. The PQ strips serve as a discrete model, where consecutive edges of quadrilaterals intersect at points along the curve, forming a tangent surface. This surface remains developable as it follows the tangency constraints, enabling more flexible and complex designs.

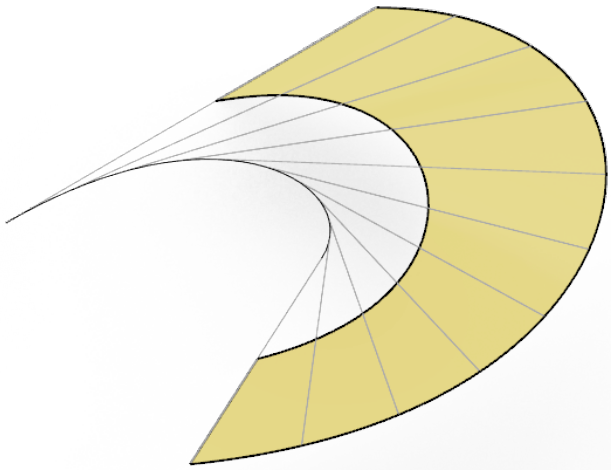


Figure 25: Tangential Developable Surface

2.1.6. Strips discretization of surfaces

Strips discretization of surfaces is a geometric modelling approach where a continuous surface is approximated or represented by a set of strip-like elements instead of traditional meshes (like triangles or quads). Commonly used in the fields of Computational, architectural geometry, and digital fabrication, and is especially useful when surfaces need to be constructed from developable or nearly developable materials like metal sheets – as adapted in the project of this thesis –, wood planks, or fabric.

Strips discretization relies on concepts from differential geometry, particularly where Gaussian Curvature is zero typically in the scale of product design or nearly zero commonly less than 10^{-4} in architectural scales, where these small values does not affect the locally at the points of the analysis. Many strip discretization layouts align with principal curvature directions of the surfaces where Rulings (straight lines on surfaces) approximate the surface between adjacent curves.

Curvature aligned stripes

This method involves decomposing the surface into thin, continuous strips that follow the main curvature of the form, maintaining tangential continuity across the surface. While this approach can achieve smooth cohesive lines along simple curves. It may have limited adaptability to high variable curvatures and complex geometries

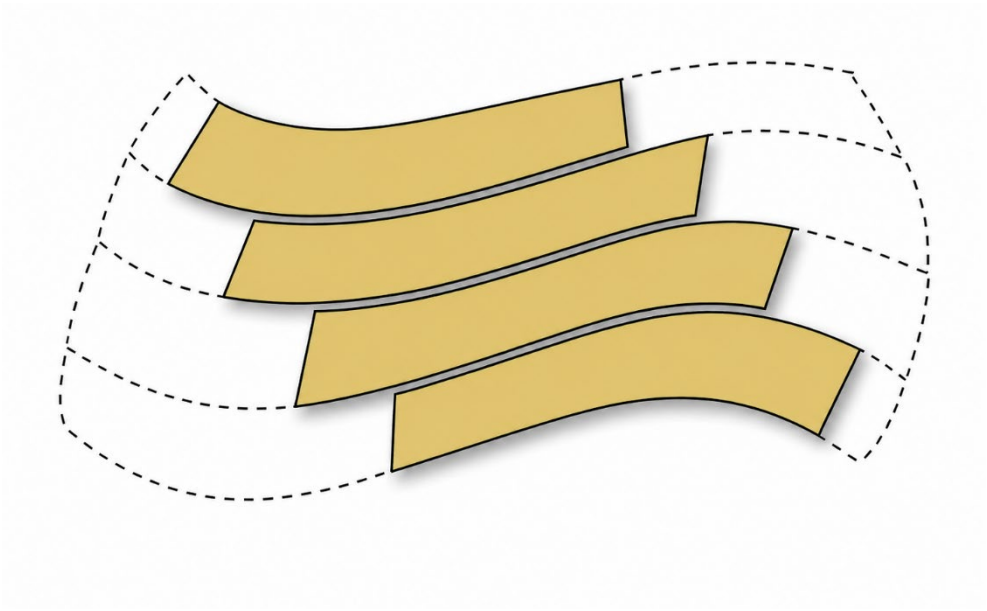


Figure 26: Curvature Aligned Stripes

Band-Strips[35]

This approach uses wider strips, or bands combined with Narrower strips to cover larger sections of surface. this method is useful when fewer joints or seams are desired, or when a more continuous appearance is needed, this method can also involve leaving wide empty spaces between the strips. It is important to mention that this approach is the one adapted in the project proposed in this thesis.

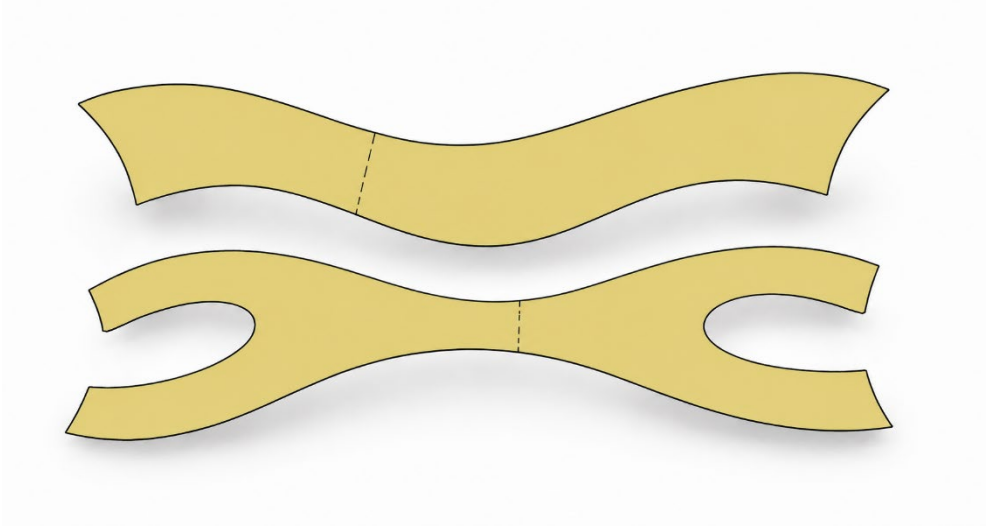


Figure 27: Band Strips

Cross Strips

Cross lamination uses strips arranged in a pattern where one set of strips runs in One Direction, and another set runs perpendicularly or at an angle. This method increases the structural stability of the laminated surface and enhances the ability to adapt to more complex geometries.

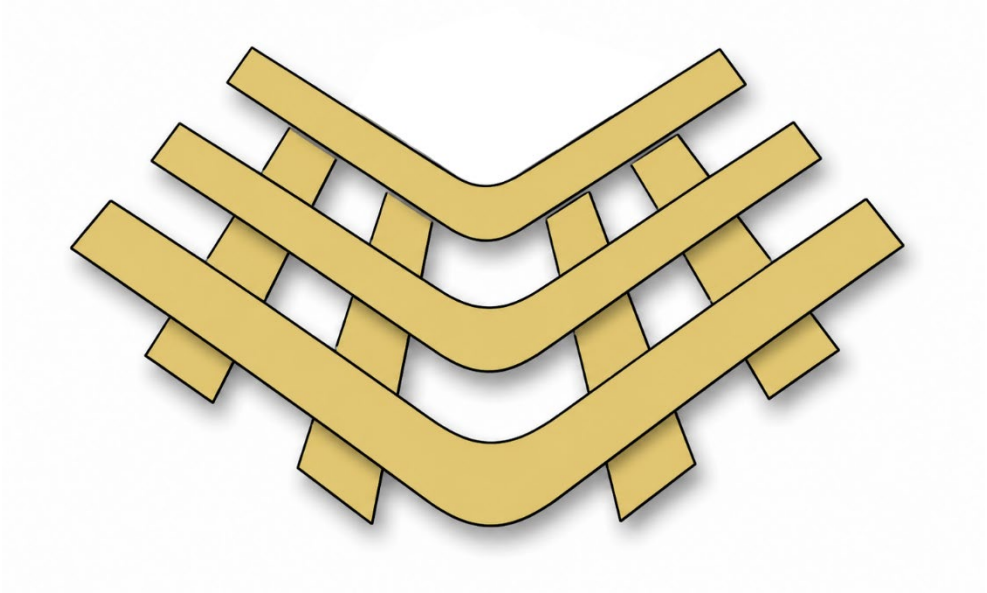


Figure 28: Cross Strips

MeshGraph-based Strips Network

This method uses a mesh graph, applying graph theory algorithms to mesh geometry, to guide the stripe Layout by representing the surface as a network of nodes and edges. Strips align along the mesh edges, following the natural curvature and contours of the form. Graph algorithms define the stripe Paths ensuring that the strips fit precisely along the surface's shape, reducing material waste and avoiding unnecessary overlaps.

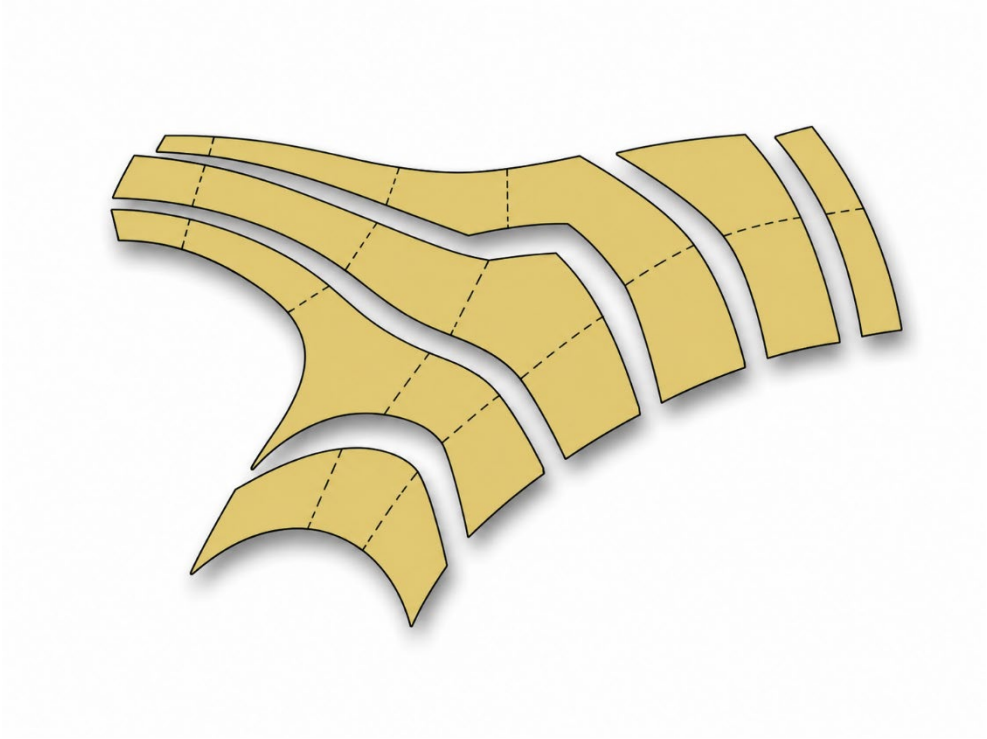


Figure 29: MeshGraph-based Strips Network

Intake

The review of strip-based surface discretisation highlights the advantages of representing continuous free-form surfaces through interconnected strip systems rather than conventional mesh discretisations. Among the reviewed approaches, band-strip discretisation offers an effective balance between geometric continuity, structural integration, and fabrication feasibility, particularly for sheet-based architectural applications. Furthermore, the alignment of strip networks with the local geometric behaviour of the surface, together with the use of near-developable strip geometries, provides a practical strategy for rationalising complex double-curved forms while preserving their overall architectural intent. These principles establish the theoretical foundation for the computational generation of interconnected band-strip systems developed in the subsequent chapters.

2.1.7.3D Geometry Representation Mesh, SubD & NURBS

In the field of digital geometry processing, the representation of geometric objects typically focuses on the 2D boundary surface of 3D solid bodies. And in the following a listing of the main methods.

Polygon Mesh[36]

A polygon mesh or combinatorial surfaces is a piecewise linear representation of a surface and has become a fundamental standard in digital geometry processing. At its core, a mesh - most commonly a triangle mesh - consists of two interrelated components. The first is a topological structure, typically formalized as a simple complex composed of vertices, edges, and faces, which defines how elements are connected. The second is a geometric embedding, whereby each vertex is assigned a specific position in three-dimensional space.

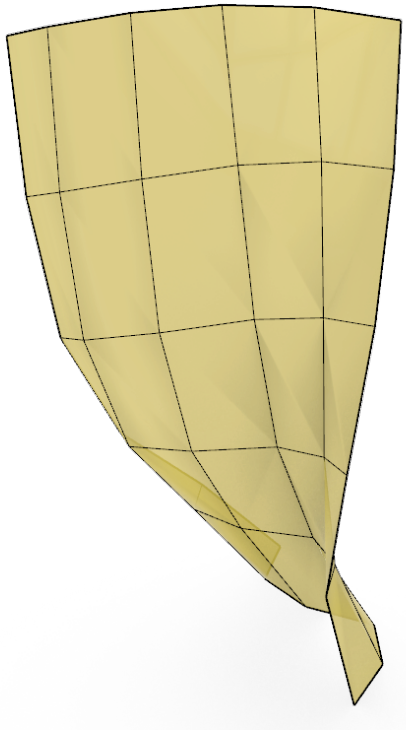


Figure 30: Polygonal Mesh

In another way, a mesh describes the boundary of a three-dimensional object, defining the transition between its interior and exterior. This dual nature - combining abstract connectivity with spatial realization - makes meshes both mathematically tractable and visually expressive.

Types of Meshes

Meshes can be classified according to both the type of elements they employ and the overall structure of their connectivity.

From the perspective of element type, **triangle meshes** are the most widely used due to their simplicity and robustness. **Quadrilateral meshes**, or quad-dominant meshes, are also common, particularly in applications such as simulation and spline fitting, where their tensor-product structure is advantageous. Additionally, meshes may be described as isotropic or anisotropic: **isotropic meshes** show uniform elements, while **anisotropic meshes** contain elongated elements aligned with features such as principal curvature directions, allowing for more efficient representation of complex forms.

From a structural standpoint, meshes range from **irregular**, with no consistent connectivity pattern, to **highly regular** configurations in which most vertices share the same valence. Intermediate forms include **semiregular** meshes, often generated through the subdivision of a coarse base mesh, and **fully regular** meshes, which can be organized as two-dimensional arrays and are sometimes referred to as geometry images.

Advantages and Limitations

Polygon meshes offer a number of practical advantages that explain their widespread adoption. Their simplicity of their concept supports the development of flexible and efficient computational algorithms, while their structure allows for local adaptation of vertex density, enabling detailed representation in regions of high curvature and more basic sampling in flatter areas. Furthermore, meshes are capable of representing surfaces with arbitrary topology, making them highly adapted across applications such as computer graphics, animation, and simulation.

However, these strengths are accompanied by notable limitations. Overall meshes are approximations of smooth surfaces, and achieving visual smoothness often requires very high vertex densities, which in turn leads to large and computationally demanding datasets. Additionally, when meshes are derived from higher-level representations—such as CAD models—they may exhibit artifacts including gaps, overlaps, or self-intersections. Finally, operations that alter topology, such as merging or cutting, can be complex and hardly interpretable compared to alternative representations.

Other Methods of 3D Representation

Beyond polygon meshes, geometric modelling includes several alternative paradigms, each with deferent conceptual and practical implications.

Parametric surface representations, such as spline-based models, describe geometry through continuous mathematical functions. Among these, NURBS have become the

standard in computer-aided design, as they provide a unified framework for representing both analytic and free-form shapes. Subdivision surfaces extend this approach by enabling the representation of complex topologies through the iterative refinement of a coarse control mesh.

Subdivision (SubD) [37]

Subdivision is a method in computer graphics and geometric modelling that allows a complex shape to be defined as the limit of an infinite sequence of increasingly faceted polyhedral. It can be viewed mathematically as a synthesis of two distinct modelling approaches: piecewise polynomial functions and fractals (iterative transformations).

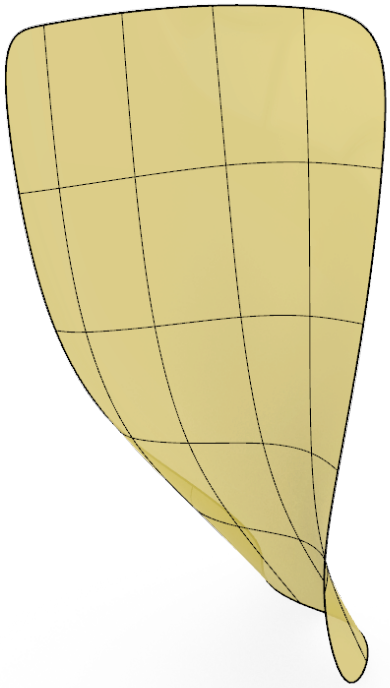


Figure 31: Subdivision (SubD) Surface

Subdivision operates directly on **polyhedral meshes** through an iterative two-step process:

- **Topological Subdivision:** This step modifies the mesh connectivity by splitting each face of a coarse mesh into a set of subfaces (for example, splitting one triangle into four). This process introduces new "ordinary" vertices while increasingly isolating any extraordinary vertices (vertices with irregular valence) within the mesh.

- Geometric Positioning: New vertex positions are computed as weighted combinations of the positions of the original vertices, often using a subdivision matrix or simple repeated averaging rules.

As this process is repeated, the discrete mesh converges toward a smooth continuous limit surface.

Importance of subdivision in Geometric Representation

Subdivision has become a fundamental tool in geometric design for several critical reasons:

- Arbitrary Topology: Noticeably traditional B-spline surfaces have many limitations: they can only represent a limited set of topologies, such as disks, cylinders, and tori. Subdivision surfaces overcome this by naturally allowing arbitrary surface topologies modelling, opening a new paradigm for the representation of complex, interconnected shapes.
- Computational Efficiency: Many subdivision schemes rely on repeated averaging, which is a fast and numerically stable method for generating dense approximations of smooth shapes.
- Ease of Implementation: Unlike some parametric methods, subdivision is relatively easy to implement because it works exclusively with discrete polygonal shapes rather than complex global equations.
- Unified Representation: It captures the best of both functional and fractal methods, providing a rich vocabulary for describing shapes that are expressively complex yet mathematically well-understood through linear analysis tools.

NURBS[38]

Non-Uniform Rational B-Splines (NURBS) have become aa common industry standard for representing, designing, and exchanging geometric information. They evolved from the pioneering work of Pierre Bézier and the subsequent development of B-splines, serving as a unified mathematical foundation for both precise analytic shapes and complex free-form geometries.

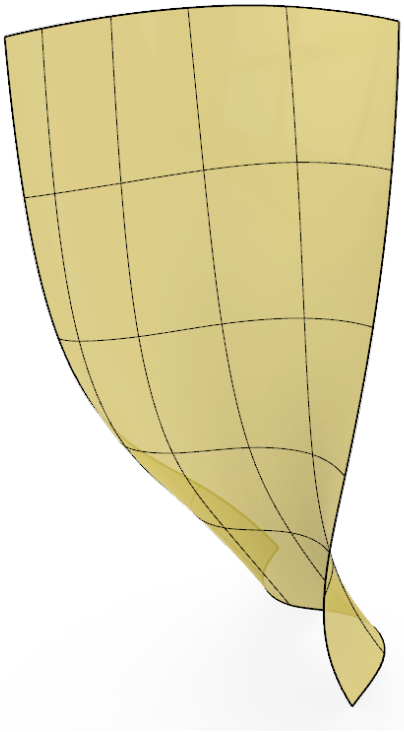


Figure 32: NURBS Surface

This method well be adapted in complex mesh simplification and reconstruction to reach a single continues NURBS surface

While other modelling methods, such as meshes, represent geometry as collections of discrete linear segments or polygons, NURBS offer several critical advantages for high-precision 3D modelling:

- Unified Mathematical Basis: NURBS can accurately represent both analytic shapes (e.g., circles, ellipses, cylinders, and spheres) and free-form geometries (e.g., car bodies or ship hulls) within a single mathematical framework.
- Precision and Accuracy: Meshes approximate shapes and often require high densities to appear smooth. In contrast, NURBS use rational functions to represent conic sections exactly—something standard nonrational polynomials or linear mesh segments cannot achieve.

- Local Shape Control: NURBS allow for "local" modifications, meaning that adjusting a control point or changing a weight affects only a specific, predictable portion of the curve or surface.
- Geometric Invariance: NURBS are invariant under common transformations, including translation, rotation, and parallel or perspective projections, ensuring that the model maintains its integrity throughout design iterations.
- Compact Data Storage: Defined mathematically rather than by thousands of individual polygons, NURBS enable efficient processing and compact storage of complex shapes.
- Numerical Stability: Algorithms for processing NURBS are generally fast and numerically stable, making them ideal for rigorous engineering and fabrication applications.

NURBS surface based on an organized network of points will be the initial step representing a generic building skin that will host the proposed shaders design.

Intake

The review of three-dimensional geometric representations highlights the strengths and limitations of the principal modelling paradigms used in computational design. Polygon meshes provide a flexible and efficient discrete representation suitable for visualization, simulation, and complex topologies; however, they approximate smooth geometry and therefore lack the precision required for high-accuracy geometric analysis and fabrication. Subdivision (SubD) modelling offers an effective intermediate representation by combining the flexibility of polygonal meshes with the generation of smooth limit surfaces, making it particularly suitable for reconstructing and simplifying complex mesh-based geometries.

Among the reviewed representations, NURBS surfaces provide the most appropriate mathematical framework for the objectives of this research. Their continuous parametric representation enables the precise evaluation of surface points, tangent vectors, normal vectors, and curvature while maintaining local geometric control and numerical stability. These characteristics are particularly advantageous for computational operations involving surface analysis, discretisation, and geometric reconstruction. Consequently, NURBS surfaces are adopted as the primary geometric representation throughout this research, providing a robust and accurate foundation for the subsequent computational workflow.

2.1.8. Computational Design & Grasshopper

Architectural practice has faced a significant evolution from traditional 2D manual drafting to 3D Building Information Modelling (BIM) and, ultimately, to Computational Design. Historically, digital tools initially replicated traditional methods, but the 21st-century shift toward algorithmic and generative eras allowed architects to move beyond being a "form-givers" to becoming the controllers of generative processes. This shift enables the exploration of unpredictable outcomes through rule-based logic.

Form this point of view some essential definitions could be framed in relation to the topic of this thesis:

Computational Design: The broad field of designing with computers, In other word the use of digital tools, computation, and data-driven methods in the design process.

Parametric Design: A design approach in which geometry is defined by parameters and relationships. Where Geometric properties are treated as variable network of relations and dependencies to give the final shape.

Generative Design: A particular use of algorithms to create and produce multiple design solutions. Within defined rules, goals, or constraints, and sometimes incorporating optimization process to reach the best option.

Modern tools like Grasshopper facilitate this by providing visual programming environments to manage these types of design. Along with Ai generation tools grasshopper will be used as the main programming platform to host the pipeline from building skin surface until the preparation for fabrication and furthermore as a programming script the manage manufacturing robots.



Figure 33: Grasshopper Logo

2.2. Digital Fabrication Overview

Digital fabrication has fundamentally transformed the relationship between architectural design and construction by establishing a direct link between computational models and manufacturing processes. Unlike conventional workflows, where design drawings are manually interpreted before production, digital fabrication enables geometric information generated within Computer-Aided Design (CAD) environments to be transferred directly to fabrication systems through Computer-Aided Manufacturing (CAM) technologies. This transition catalysed in the 1990s by pioneering architects including Frank Gehry and Zaha Hadid, has reduced the separation between design and production, allowing complex architectural geometries to be realised with higher precision, improved efficiency, and reduced information loss throughout the construction process. It moved the discipline away from drawings as the primary medium of construction and toward digital models that encode both geometric intent and fabrication logic simultaneously [39].

The integration of computational design with digital fabrication has further expanded these capabilities. Rather than considering fabrication as a post-design activity, contemporary computational workflows incorporate manufacturing constraints directly into the design process. Parameters such as material behaviour, assembly logic, connection details, tolerances, and fabrication limitations can be embedded within parametric models from the early design stages. Consequently, modifications to the computational model automatically propagate through the entire workflow, maintaining consistency between geometry, fabrication information, and construction documentation [39].

Within architectural applications, digital fabrication encompasses a broad range of manufacturing approaches. Among these, sheet-based fabrication has become particularly significant due to its compatibility with lightweight construction systems, façade engineering, and prefabricated building components. Sheet materials offer several practical advantages, including high dimensional accuracy, efficient material utilisation, repeatable manufacturing, and relatively simple transportation and assembly. As a result, metal sheet systems have become widely adopted in free-form façade construction, where complex geometries must be rationalised into manufacturable components while maintaining the intended architectural expression [40].

The fabrication of free-form architectural envelopes introduces additional geometric challenges that differ substantially from those encountered in conventional planar construction. Since most sheet materials are manufactured as flat elements, doubly curved surfaces generally cannot be fabricated directly without specialised forming processes. Consequently, computational rationalisation techniques have become an essential stage of the digital fabrication pipeline. These techniques transform continuous free-form surfaces into discrete fabrication components while considering developability, material limitations, bending behaviour, assembly requirements, and production efficiency. The resulting fabrication-aware models establish a direct relationship between geometric design and manufacturing feasibility, reducing manual intervention during downstream production stages [40].

Recent developments in parametric modelling environments, particularly Grasshopper, have further strengthened the integration between computational geometry and fabrication planning. Through visual programming, designers can generate adaptive geometric systems whose fabrication information—including segmentation, connection logic, bending data, and material organisation—is automatically updated as the design evolves. This capability has shifted digital fabrication from a geometry translation process toward an integrated design-to-production methodology in which geometric modelling and fabrication planning are developed simultaneously rather than sequentially.

Intake

The review of digital fabrication highlights the evolution from conventional CAD/CAM workflows toward integrated computational design-to-production methodologies in which fabrication constraints are embedded directly within the geometric modelling process. For sheet-based architectural systems, this requires computational approaches capable of transforming complex free-form surfaces into manufacturable components while preserving geometric continuity, assembly logic, and fabrication efficiency. These principles establish the theoretical foundation for the fabrication-oriented computational workflow developed in the subsequent chapters.

2.2.1. Unrolling and Nesting for Fabrication

The fabrication of free-form architectural components from sheet materials requires the transformation of three-dimensional geometry into two-dimensional cutting patterns while preserving the geometric and assembly information necessary for manufacturing. This process is generally divided into two consecutive stages: **geometric unrolling**, which converts three-dimensional surfaces into flat patterns, and **material nesting**, which optimizes the arrangement of these patterns on fabrication sheets. Together, these operations establish the link between computational design and digital manufacturing by translating complex geometries into production-ready fabrication layouts [39], [40]

Geometric Unrolling

Unrolling, also referred to as **surface development or flattening**, is the process of transforming a three-dimensional surface into a two-dimensional representation while preserving its intrinsic geometric properties. For **developable surfaces**, such as cylinders and cones, this transformation can be performed exactly because the surface possesses **zero Gaussian curvature** and can be unfolded without stretching or compression. In contrast, **free-form architectural surfaces** frequently contain regions of **double curvature**, making exact unrolling mathematically impossible without introducing geometric distortion or material deformation [40].

To address this limitation, contemporary computational fabrication workflows employ **surface rationalisation** techniques that approximate complex free-form geometries using developable or **nearly developable** components. These approaches minimise geometric distortion while maintaining the intended architectural form within **acceptable fabrication tolerances**. Depending on the material behaviour and manufacturing process, flattening algorithms may preserve edge lengths, minimise strain energy, or optimise local geometric deviations to generate practical fabrication patterns suitable for bending or assembly [40].

Beyond geometric flattening, modern fabrication workflows increasingly recognise that fabrication components consist of more than surface boundaries alone. Assembly information, including connection holes, reference points, bending lines, identification labels, and construction annotations, must remain associated with each component throughout the unrolling process. Preserving these relationships is essential for ensuring consistency between the digital model, fabrication drawings, and physical

assembly, particularly in computational workflows where fabrication data is generated automatically from parametric models.

Material Nesting

Once fabrication components have been flattened, they must be arranged efficiently on standard sheet materials prior to manufacturing. This process, known as nesting, seeks to **maximise material utilisation** while respecting **fabrication constraints** such as sheet dimensions, component spacing, orientation requirements, cutting tolerances, and production sequencing. Because material cost often represents a significant proportion of fabrication expenses, effective nesting directly influences both manufacturing efficiency and project economics.

Material nesting is commonly formulated as a two-dimensional optimisation problem. Numerous computational strategies have been proposed, including bottom-left placement methods, no-fit polygon algorithms, simulated annealing, and genetic optimisation techniques. Although these methods differ in computational complexity and optimisation strategy, they share the common objective of reducing material waste while generating practical layouts compatible with digital fabrication equipment.

Within computational design environments, several software implementations provide automated nesting capabilities. Among these, **OpenNest** has become one of the most widely adopted solutions for **Grasshopper-based workflows** due to its seamless integration with parametric modelling environments and its ability to preserve geometric relationships throughout the nesting process. Such integration enables fabrication layouts to remain fully associative with the original computational model, allowing modifications to the design geometry to be propagated automatically to the fabrication sheets without manual rearrangement [41].

The combination of automated unrolling and nesting represents a fundamental stage within digital fabrication workflows. Rather than serving solely as geometric conversion processes, these operations establish the basis for fabrication documentation by organising component geometry, manufacturing information, and material layouts into production-ready datasets suitable for CNC-based sheet fabrication.

Fabrication preparation extends beyond the geometric development of surfaces to encompass the preservation of fabrication attributes and the optimisation of material utilisation. Consequently, computational workflows should maintain the correspondence between three-dimensional components, fabrication information, and two-dimensional layouts while ensuring that all geometric and assembly data remain consistent throughout the transition from design to production. These principles form the theoretical basis for the automated fabrication preparation workflow presented later in this research.

2.2.2. BIM Automatize Documentation

Building Information Modelling (BIM) has transformed architectural documentation from a drawing-based workflow into an information-centric process in which geometric representation, semantic information, and project documentation are generated from a unified digital model. Unlike conventional CAD systems, where drawings are produced independently, BIM environments establish associative relationships between model elements and their corresponding plans, elevations, schedules, quantities, and construction details. Consequently, modifications to the model are propagated automatically throughout the documentation, reducing coordination errors while improving consistency and information management across the project lifecycle.

The increasing adoption of computational design has introduced new opportunities and challenges for BIM workflows. Parametric modelling environments such as Grasshopper enable designers to generate complex geometries that would be difficult to model manually within conventional BIM software. However, transferring these computational models into BIM environments has historically required exporting simplified geometry, often resulting in the loss of parametric relationships, semantic information, and fabrication data. This disconnect has created a gap between computational design and Building Information Modelling, requiring significant manual effort to recreate computationally generated elements as native BIM objects suitable for documentation and coordination [42].

To address this limitation, recent research has focused on improving interoperability between computational design platforms and BIM environments. Rather than exchanging only geometric representations, contemporary workflows increasingly seek to preserve both geometry and associated information throughout the design process. This includes the automatic transfer of object parameters, structured information,

identification codes, material properties, fabrication attributes, and assembly information, allowing computational models to function as information-rich BIM components rather than purely geometric references.

One of the most significant developments in this area is Rhino.Inside.Revit, introduced through the collaboration between McNeel and Autodesk. Unlike traditional file exchange methods based on formats such as DWG, SAT, or IFC, Rhino.Inside.Revit enables Rhino and Grasshopper to operate directly within the Autodesk Revit environment using Revit's native Application Programming Interface (API). This approach allows computational geometry to be generated, modified, and managed directly inside a BIM model while preserving native Revit elements, parameters, and project relationships[43].

The integration provided by Rhino.Inside.Revit has significantly expanded the role of computational design within BIM workflows. Grasshopper-generated geometry can be converted directly into native Revit families, adaptive components, model elements, and annotation objects while maintaining their associated parameters and metadata (Structured information provides context for other data). In addition, repetitive documentation tasks—including family generation, parameter assignment, drafting views, schedules, and sheet creation—can be automated through computational workflows. These capabilities substantially reduce manual modelling and documentation efforts while improving consistency between computational design, fabrication data, and BIM documentation.

Recent research and industrial applications increasingly employ Rhino.Inside.Revit as an interoperability platform connecting computational geometry, environmental analysis, structural optimisation, digital fabrication, and BIM documentation within a unified workflow. Rather than functioning solely as a geometric modelling interface, it establishes a bidirectional exchange of information that supports collaborative design, project coordination, and digital construction processes. As a result, Rhino.Inside.Revit has become one of the most widely adopted frameworks for integrating parametric design methodologies into Building Information Modelling environments.

Intake

BIM interoperability has evolved beyond the exchange of geometric models toward the integration of computational design and information management within a unified digital environment. Contemporary interoperability frameworks enable computationally generated geometry to retain its associated parameters, metadata, and fabrication information throughout the documentation process, supporting automated model generation, coordinated project documentation, and improved consistency between design, fabrication, and construction information. These developments provide the theoretical foundation for integrating computational modelling with BIM-based documentation workflows.

3.
Methodology

3. Methodology

Starting from LOUVEN A project by INDEXLAB – Politecnico di Milano in collaboration with Cremonini. Louven is an innovative system for creating shading devices or free-form walls, suitable for both indoor and outdoor use. Generative artificial intelligence is employed during the design phases and combined with computational processes for the fabrication stages, which include laser cutting, CNC bending, and laser welding.[44]

This research extends the LOUVEN methodology by introducing a more comprehensive and automated computational workflow, addressing several stages that were not fully resolved in the initial project. In particular, the work focuses on refining the strip-based discretization strategy of complex surfaces. Modifications are applied to improve geometric consistency, Strips developability alignment, and detailed fabrication logic, enabling a more controlled translation from continuous surfaces into manufacturable components.

A key contribution is the development of a fully parametric workflow using Grasshopper. Within this environment, multiple critical aspects of the process are automated:

- Detailed geometric modelling for fabrication, ensuring that each strip element contains precise information required for manufacturing.
- Continuity of geometry data extraction, including bending angles and direction, fabrication tolerances and bending notches which are essential for CNC bending accuracy.
- Surface unrolling and nesting optimization, allowing efficient material usage and direct preparation of fabrication files.

Beyond geometry processing, the workflow establishes interoperability between Rhinoceros 3D and Autodesk Revit. Fabrication-ready geometries generated and nested in Rhino are automatically translated into Revit, where executive drawings are produced. This includes the automated creation of views, sheets, and component labelling systems with adaptive numbering and naming conventions, ensuring consistency, traceability, and scalability across large assemblies.

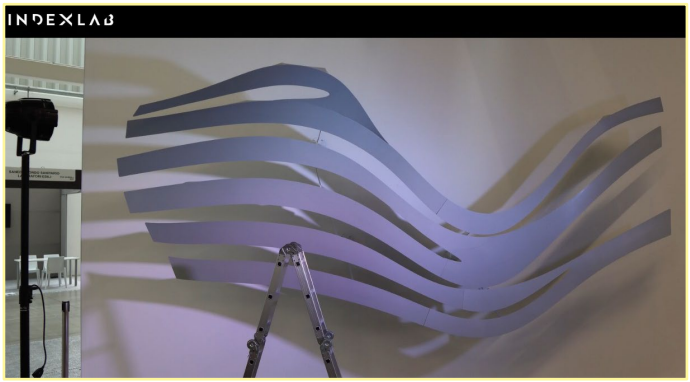


Figure 34: Aluminium Physical model - LOUVEN by INDEXLAB

In parallel, the research explores the integration of artificial intelligence at the early stages of façade design. AI tools are used to generate alternative shell geometries from text prompts or image inputs. These outputs are then processed and reconstructed into functional 3D models, enabling their seamless integration into the parametric workflow and subsequent fabrication pipeline.

By combining generative AI with computational design and digital fabrication, this approach establishes a continuous pipeline from concept generation to construction documentation. It effectively bridges the gap between creative exploration and technical realization, reducing the need for designers to manually manage complex construction details while maintaining full control over design intent and production logic.

METHODOLOGY ROADMAP

From Text to Fabrication: Integrating AI Generative Modelling, Freeform Surface Reconstruction, and Band Strips Discretization for Automated Architectural Facade Fabrication.

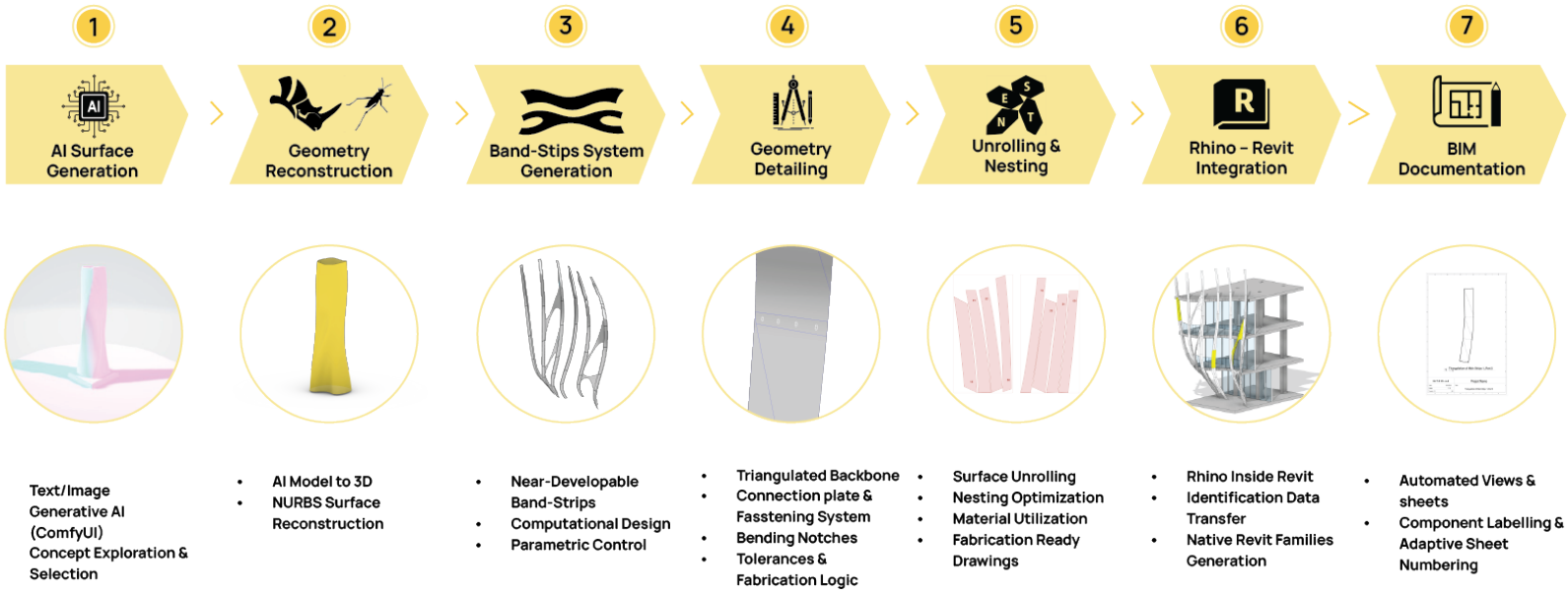


Figure 35: Methodology Roadmap

LOUVEN+

4.
Project

4. Project

4.1. AI 3D Model Generation (Comfy UI)

4.1.1. Text to Image

The initial stage of experimentation focused on text-to-image generation using **ComfyUI** and a selection of publicly available models. Due to hardware limitations and the need for efficient inference, the **FLUX FP8** model was selected, employing the basic text-to-image workflow illustrated in **Figure 35**. The FP8 version offered a suitable balance between image quality and computational performance, enabling iterative experimentation within the available GPU resources.

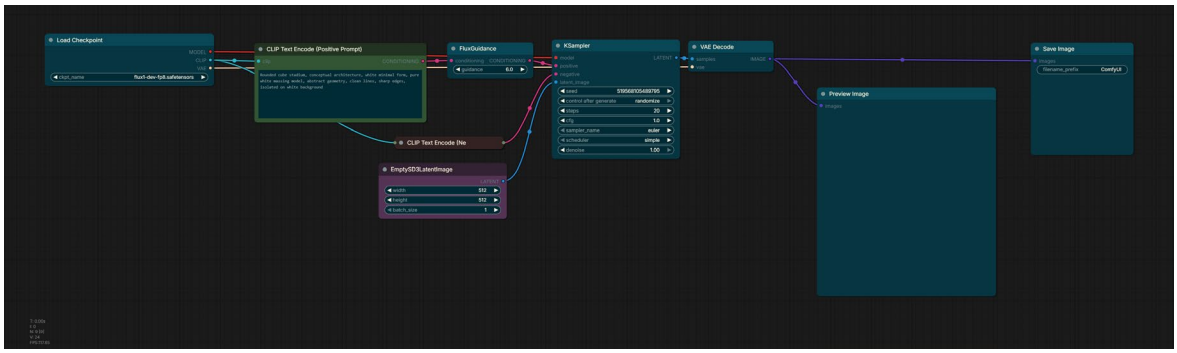


Figure 36: Text to Image ComfyUI Workflow

The generated results, together with the corresponding prompts, are presented below. Unlike traditional Stable Diffusion models, the FLUX model relies primarily on **positive prompts**, without requiring extensive negative prompting. The prompts were intentionally designed to produce forms resembling stadiums and large-span cultural buildings, as these represent relatively simple geometries that can be interpreted as continuous single surfaces. Such forms are particularly suitable for subsequent geometric processing, as their meshes can be more easily traced and reconstructed into NURBS surfaces, ensuring compatibility with the following stages of the workflow.

The choice of stadium-like forms was motivated by their inherent characteristics of continuity, smooth curvature, and structural simplicity, which closely align with the objectives of generating fluid architectural envelopes and shell structures. This approach establishes a foundation for later transformations and geometric refinements, ultimately facilitating the conversion of AI-generated meshes into parametric and manufacturable three-dimensional models.

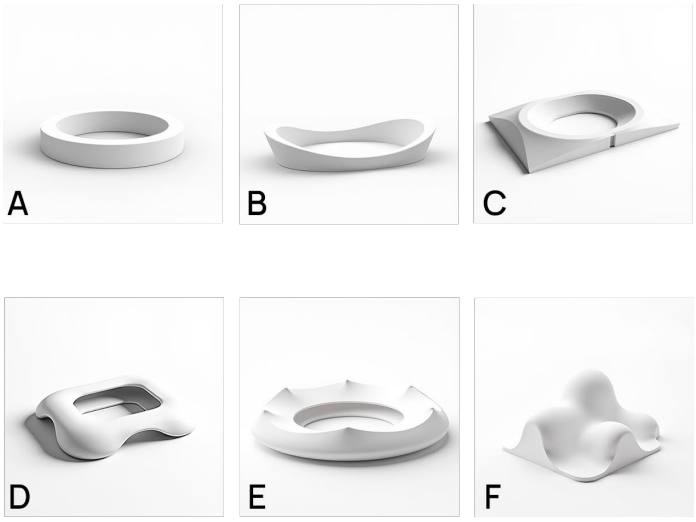


Figure 37: Text to Image Selected Outputs

Prompts for pictures as follows:

A. " *Rounded cube stadium, conceptual architecture, white minimal form, pure white massing model, abstract geometry, clean lines, sharp edges, isolated on white background"*

B. "Rounded corners stadium concept, wavy form, one surface, conceptual architecture, white minimal form, pure white massing model, abstract geometry, clean lines, sharp edges, isolated on white background"

C. *"Rounded corners stadium concept, wide wavy form, one surface, conceptual architecture, white minimal form, pure white massing model, abstract geometry, clean lines, smooth edges, isolated on white background"*

D. " *Rounded corners stadium concept, wide wavy form, one surface, bird's-eye view of conceptual building form, urban scale, no creases, white Fabric roof structure, white minimal form, pure white massing model, abstract geometry, clean lines, smooth edges, isolated on white background, clean and tidy, masterpiece, 8k"*

E. *"Rounded corners stadium concept, wide wavy form, one surface, bird's-eye view of conceptual building form, urban scale, no creases, white Fabric roof structure, white minimal form, pure white massing model, abstract geometry, clean lines, smooth edges, isolated on white background, clean and tidy, masterpiece, 8k"*

F. " *Rounded corners building concept, wide wavy form, one surface, bird's-eye view, no creases, white minimal form, pure white massing model, abstract geometry, clean lines, smooth edges, isolated on white background, clean and tidy, masterpiece, 8k"*

The six prompts show a gradual shift from simple geometric forms to more fluid and continuous architectural surfaces. Prompt A produced a basic circular stadium shape with limited variation, while B and C introduced wavy edges and smoother transitions, resulting in more dynamic forms. However, the stadium constraint kept the generated shapes relatively similar and restricted the model to ring-like configurations.

Among the stadium-based prompts, D produced the most successful result. The addition of terms such as bird's-eye view, urban scale, and fabric roof structure led to a smoother and more continuous shell geometry. Nevertheless, the explicit reference to a stadium still limited the design freedom, making the result architecturally weaker and less fluid than desired.

In contrast, F was the only prompt that did not include a stadium as a constraint. This allowed the model to generate a more organic and continuous form with smoother double curvature and no noticeable creases. As a result, F achieved the highest level of continuity and fluidity among all the generated images. Together, D and F were the most successful results, with D being the strongest stadium-based design and F showing the greatest potential as a free-form architectural envelope suitable for later reconstruction into NURBS geometry.

4.1.2. Image to Image

The Flux 2 image-to-image pipeline can further help more through the integration of Depth ControlNet and Canny ControlNet, which provide additional structural guidance during image generation. Unlike conventional image-to-image generation that relies primarily on the input image and text prompt, ControlNet introduces spatial constraints that enable the model to preserve specific visual characteristics while allowing semantic modifications.

Depth ControlNet utilizes a depth map extracted from the input image to preserve the three-dimensional spatial structure of the scene. This guidance enables the model to maintain object placement, perspective, and relative distances while generating new visual content. Consequently, it is particularly suitable for tasks requiring consistent scene geometry, such as architectural visualization, landscape transformation, and style transfer, where maintaining spatial relationships is essential.

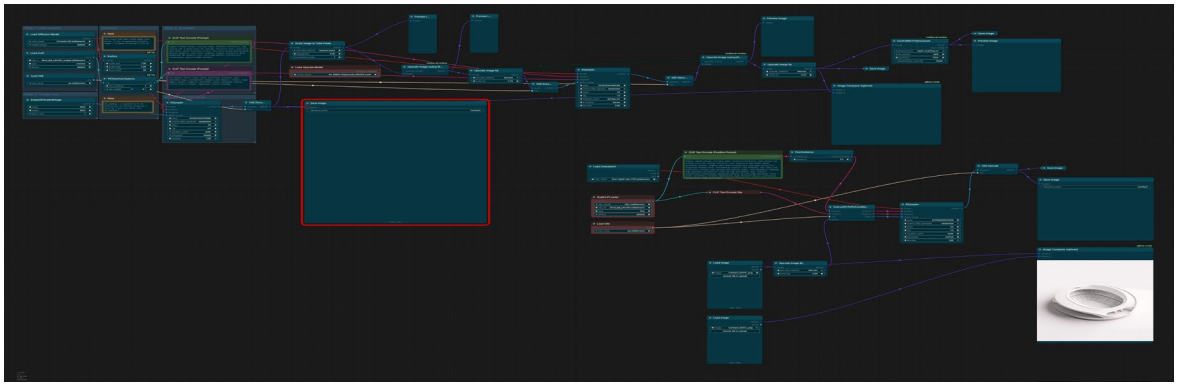


Figure 38: Flux 2 Depth ControNet workflow + Upscaling and render addition in ComfyUI

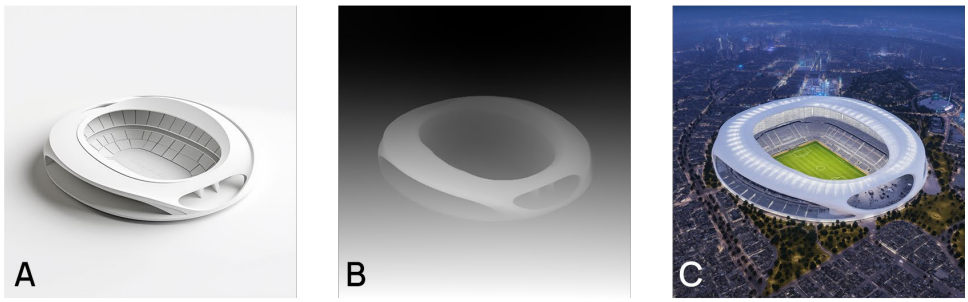


Figure 39: Flux 2 Depth ControNet outputs

Canny ControlNet, in contrast, employs edge maps generated using the Canny edge detection algorithm to preserve the structural outlines and contours of objects. By constraining the generation process according to these edge features, the model maintains the overall composition and object boundaries while allowing changes in texture, colour, lighting, and artistic style. This approach is especially effective for preserving fine structural details in illustrations, product images, and character designs.

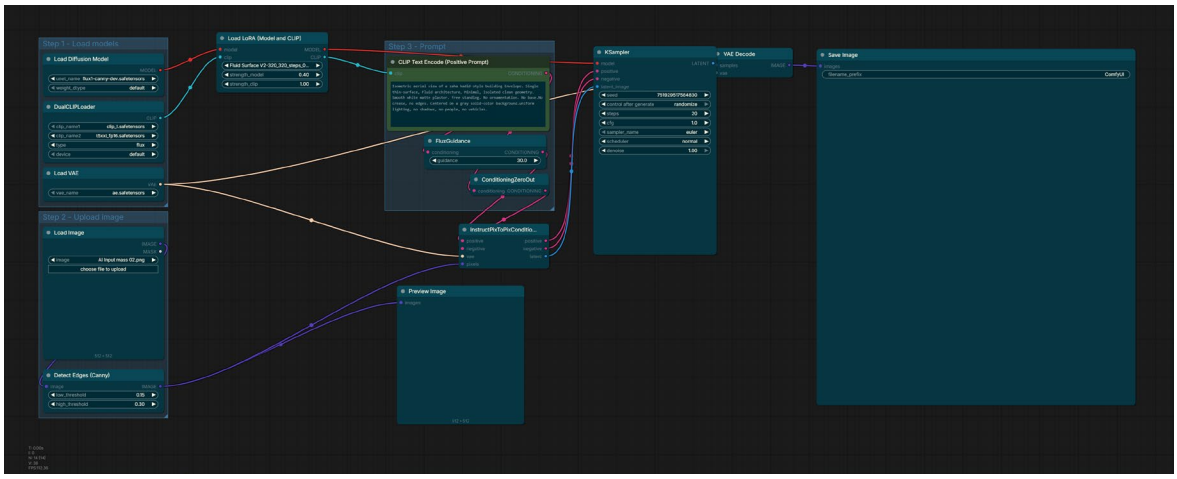


Figure 40: Flux 2 Canny ControlNet Workflow

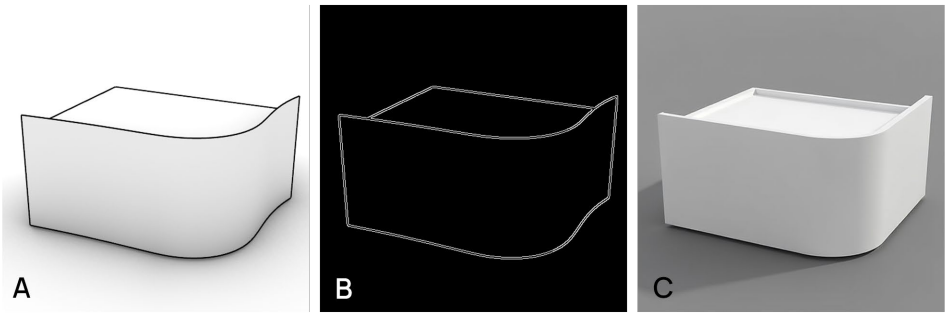


Figure 41: Flux 2 Canny ControlNet Outputs

Integrated with the Flux 2 image-to-image model, these conditioning mechanisms improve generation stability, enhance prompt adherence, and produce visually coherent outputs that closely follow both the reference image and the textual description. As a result, the approach offers greater control over the image synthesis process and is well suited for applications requiring high structural consistency alongside flexible semantic editing.

4.1.3. LORA image training

A collection of 20 reference images – shown in Figure 42 – was gathered from Pinterest and carefully selected according to a set of visual criteria to ensure suitability for LoRA training. The objective was to obtain images that closely matched the intended architectural and sculptural characteristics. The target description was:

"An isometric view of a dynamic curved sculptural open surface composed of a single thin shell without creases, featuring a smooth white matte plaster finish. The form is free-standing, without ornamentation or a base, with soft folds and no sharp edges. The object is cantered on a solid-colour background under uniform lighting, with no shadows, people, or vehicles."

Only images that best satisfied these requirements were included in the dataset. To maintain consistency, all images were resized and standardized to 1024 × 1024 pixels before being used for training.

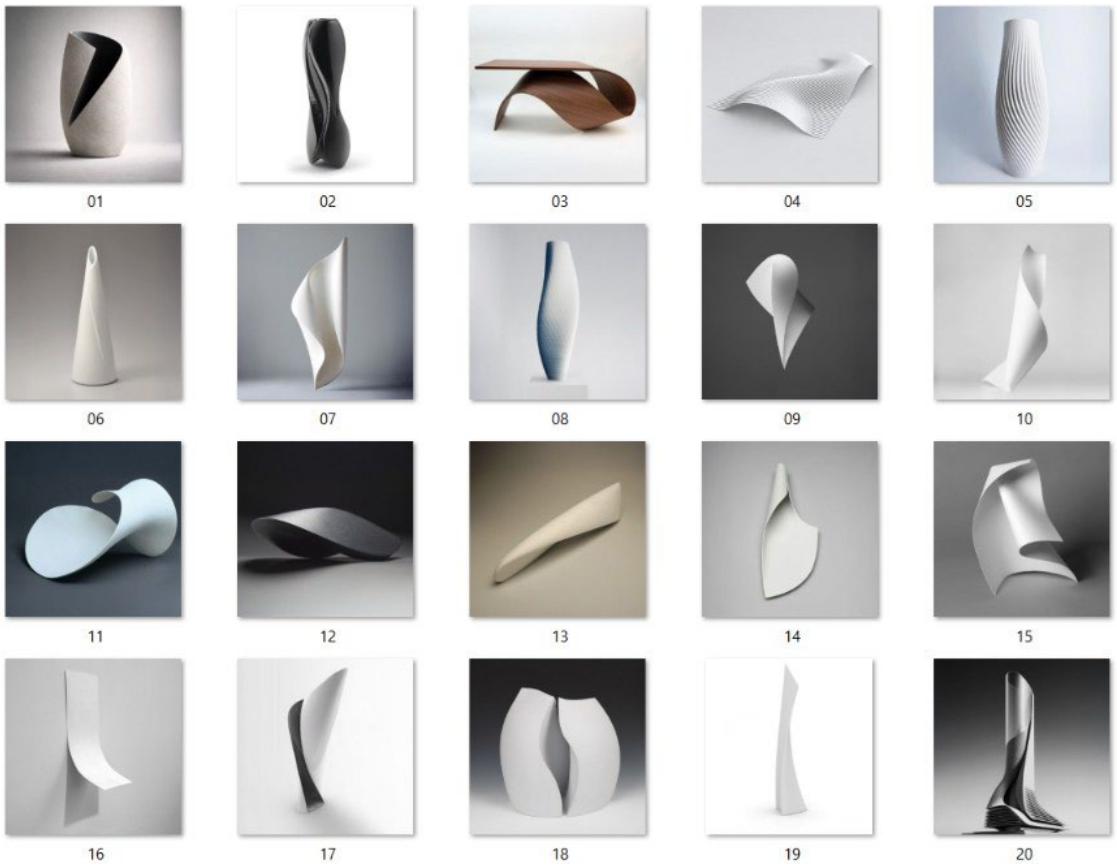


Figure 42: 20 reference images for LORA image training

The LoRA training process was implemented in **ComfyUI** using the workflow shown in **Figure 42**. The base model selected for training was **SDXL Base 1.0**. The reference images were encoded through the **VAE encoder,** while the text prompt provided the conditioning information describing the desired geometry. Training was performed with a batch size of 1, gradient accumulation of 4, and **1,000 training steps**. A learning rate of 0.0005 was used with the AdamW optimizer, while the loss function was set to Mean Squared Error (MSE). The **LoRA rank** was set to **16**, and training was carried out in **bfloat16 (bf16) precision** with gradient checkpointing enabled to reduce memory consumption. After training, the resulting LoRA weights were saved for later inference.

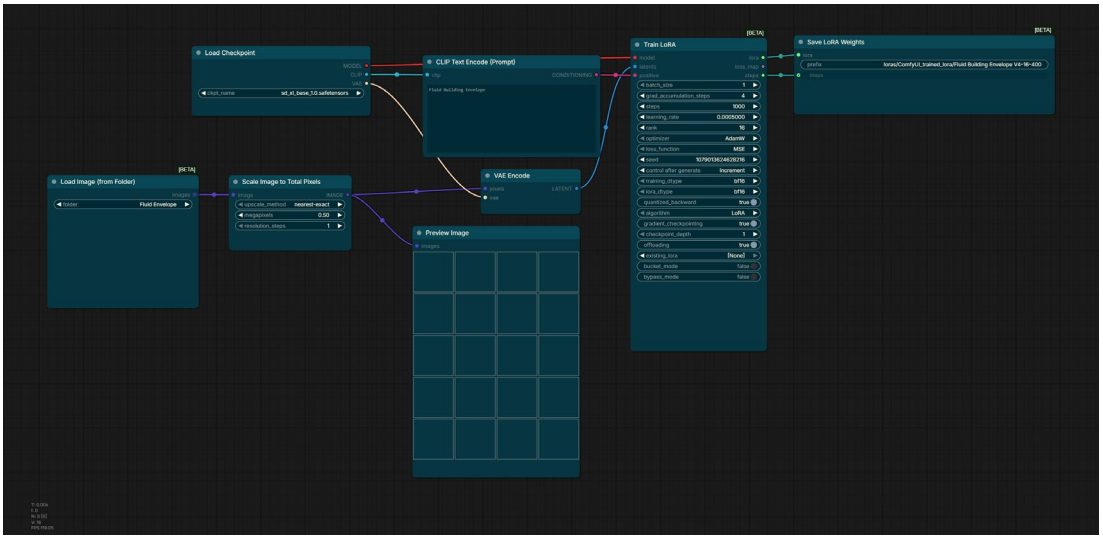


Figure 43: LORA image training Workflow in ComfyUI

A second workflow in ComfyUI was used for image generation **Figure 43**. The trained LoRA was loaded and combined with the **Realistic Vision V60 B1 VAE** model, which provided the most convincing results. Positive and negative prompts were used to guide the generation process, and images were sampled using the **Euler sampler with 20 steps**, a **CFG scale of 8**, and a resolution of **1024 × 1024 pixels**. The generated latent image was decoded through the VAE and saved as the final output.



Figure 44: Realistic Vision V60 B1 VAE Workflow with Trained LoRA

A selection of the **most successful outputs** generated using the trained LoRA is presented in **Figure 45**. These results demonstrate the model's ability to reproduce the intended architectural language while maintaining consistency with the characteristics defined during dataset preparation. The generated forms exhibit smooth and continuous curved surfaces, soft transitions without sharp edges or creases, and a clean white matte material appearance. The compositions remain centred and isolated against simple backgrounds, allowing the sculptural geometry to be clearly perceived.



Figure 45: Trained LoRA Most Successful Outputs

The generated forms exhibit a degree of consistency, with their geometric and morphological characteristics closely reflecting the architectural intent established during the training process. Although the majority of the outputs successfully capture the intended design language, they cannot yet be interpreted as viable conceptual building forms due to their arbitrary geometries. These forms require further contextualization within a specific building typology to develop into architecturally meaningful proposals. Nevertheless, the consistency demonstrated across the generated geometries provides a robust foundation for subsequent refinement and optimization. This establishes a reliable starting point for future research aimed at transforming these abstract forms into more coherent, architectural concept solutions.

Following the LoRA image training process, an additional experimental trial was conducted using the Flux-2-klein workflow in ComfyUI by integrating the trained LoRA model into the image generation pipeline.

Flux-2-klein-Base-9B-fp8 is a 9-billion-parameter open-weight generative image foundation model designed for high-quality text-to-image generation and image editing tasks. The model employs FP8 quantization to reduce memory requirements and improve inference efficiency while maintaining competitive image quality. As an undistilled base model, it offers greater flexibility for customization, fine-tuning, and adaptation to downstream applications. Its key features include high-fidelity image synthesis, strong prompt adherence, efficient resource utilization through FP8 precision, compatibility with parameter-efficient fine-tuning methods such as LoRA, and support for a wide range of image generation and editing tasks.

The experimental results demonstrated improved image quality compared with the previous workflow. The **best outputs** were obtained **when the text prompt included sufficient contextual information**, indicating that the model benefits from detailed and descriptive prompts. An example of the prompt used in this experiment is presented below.

*"Isometric aerial view of a **zaha hadid**-style **tower building**. Single thin-surface, **Fluid architecture**, **Minimal**, Isolated clean geometry. Smooth white matte plaster. free standing. No ornamentation. No model base.No crease, no edges. **Centered on a gray solid-color background**.uniform lighting, no shadows, no people, no vehicles."*

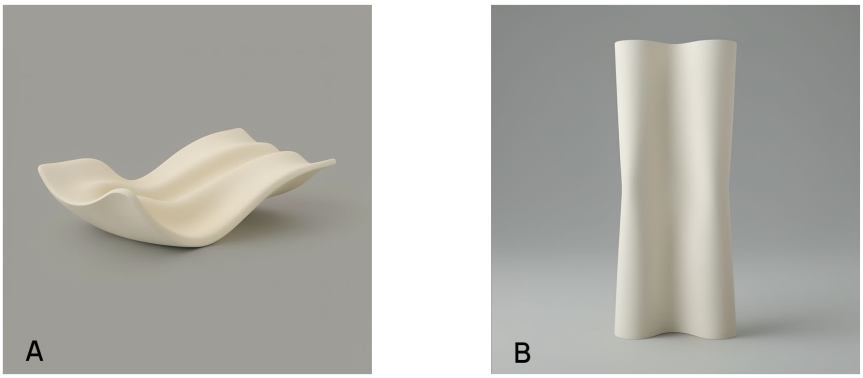


Figure 46: Flux-2-klein + Trained LoRA Surface outputs

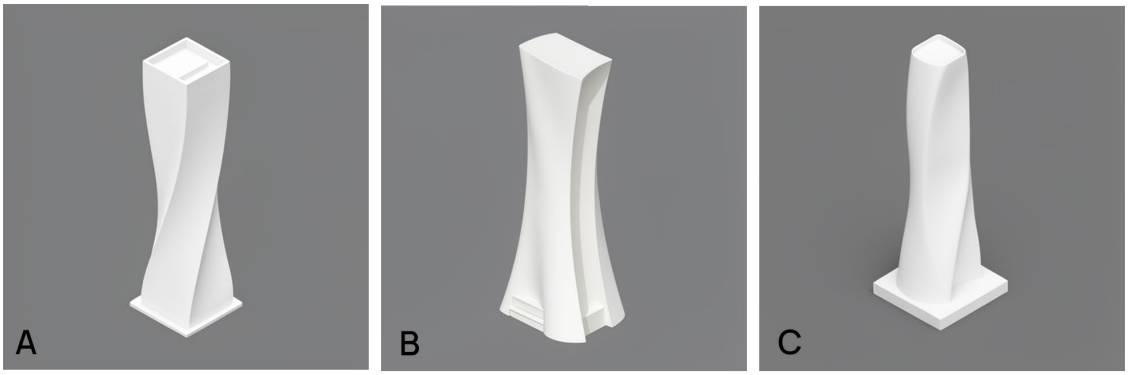


Figure 47: Flux-2-klein + Trained LoRA contextual Prompt outputs

4.1.4. Image to 3D mesh model

Hunyuan 3D 2.0 was selected for the transformation of the generated images into three-dimensional mesh models. As discussed in 2.1.2.AI image to 3D model At the time of this research, it represented one of the most capable open-source AI models for image-to-3D generation, producing detailed and relatively high-polygon meshes while preserving the overall geometry and visual characteristics of the input image. This made it particularly suitable for architectural concept exploration, where maintaining the continuity and curvature of the generated forms is essential.

The workflow implemented in **ComfyUI** is simple and straightforward, as illustrated in **Figure 48**. The process begins by providing a generated architectural image as input, which is then processed by the **Hunyuan 3D 2.0** model to estimate the object's three-dimensional geometry. The model automatically reconstructs a textured 3D mesh from a single image, generating both the geometric form and its associated surface information. The resulting mesh can then be exported in standard 3D formats for further processing and refinement.

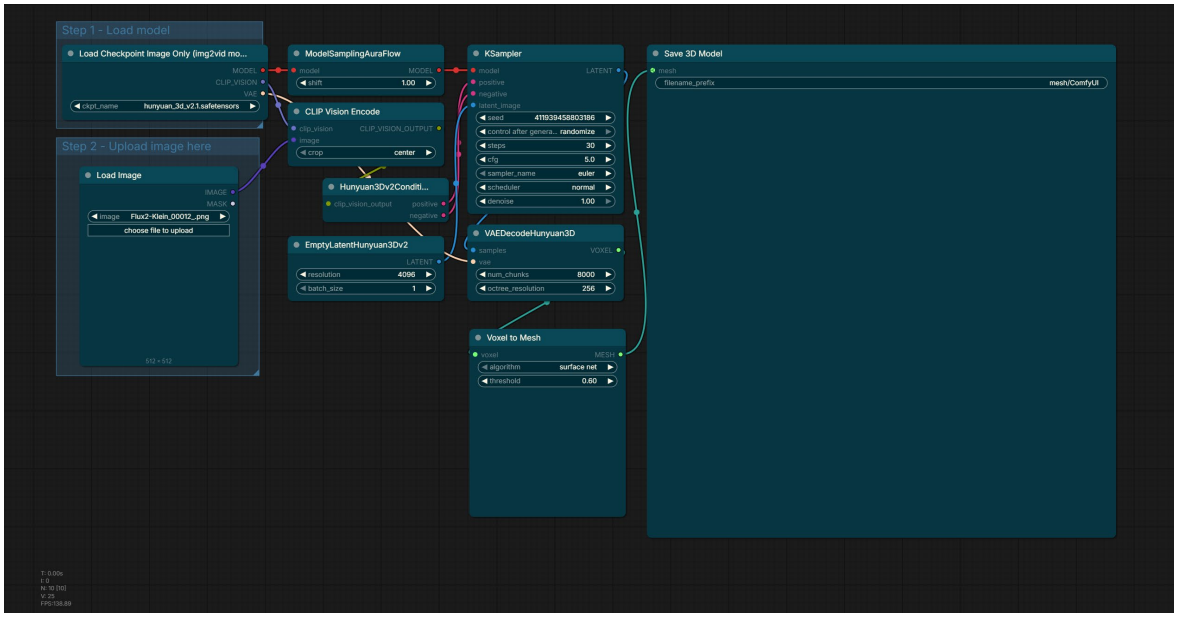


Figure 48: Hunyuan 3D 2.0 Image to 3D mesh model

The generated meshes successfully captured the main geometric features of the original AI-generated concepts, including their overall proportions, curvature, and spatial characteristics. Examples of the resulting 3D meshes are presented in **Figure 49**. While the generated models are not intended to be fabrication-ready, they provide a valuable intermediate representation that bridges the gap between two-dimensional AI image generation and computational geometric modelling.

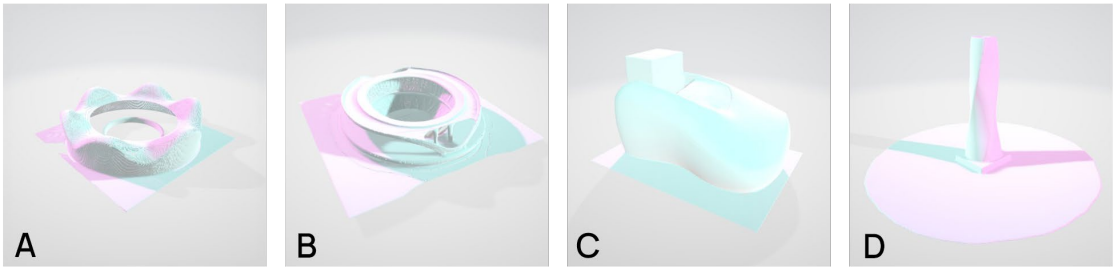


Figure 49: Produced 3D Meshes by Hunyuan 3D 2.0

At this stage, the AI-assisted design and generation process reaches its conclusion. The workflow then transitions to the computational design phase, where the generated meshes become the basis for geometric reconstruction. Rather than working directly with polygonal meshes, the objective is to convert the AI-generated forms into clean and continuous NURBS surfaces. This conversion enables greater geometric control, improved accuracy, and compatibility with digital fabrication workflows. Through this process, the conceptual forms generated by AI can be transformed into rationalized single-surface geometries suitable for further structural development, analysis, and fabrication.

4.2. Geometry Build-up

4.2.1. High Poly Mesh to single smooth surface

The Grasshopper workflow is designed to transform an unorganized, high-polygon open mesh into a geometrically smooth and topologically organized surface suitable for subsequent double-curved surface discretization into strip-based fabrication pipeline. The process bridges the gap between the given AI generated 3D meshes or irregular mesh data and a clean NURBS representation, that is more suitable for modelling and fabrication workflows.

Workflow Overview

Stages: (1) Optional mesh reduction, (2) Mesh reconstruction and subdivision, (3) Quad remeshing, (4) Boundary extraction and guide generation, (5) Generation of Structured Vertical Quad Mesh, (6) Kangaroo physics relaxation, and (7) NURBS surface reconstruction.

Stage 1 – Mesh Input and Reduction

The workflow begins with an open and unorganized polygon mesh. A user-controlled option determines whether mesh reduction is applied. Reduction decreases polygon count while preserving global morphology, improving computational efficiency for later operations. It's very important that the given mesh must be open with understandable edges that could be traced by the logic and if not this step of preparing the given mesh must be done manually.



Figure 50: Mesh Input and Reduction

Stage 2 – Mesh Reconstruction and Subdivision

The mesh is reconstructed using triangular topology to improve geometric consistency. Subsequent subdivision increases local resolution and smoothness while reducing irregularities.

Note:
This step is applicable only to meshes that do not contain duplicated faces or holes.

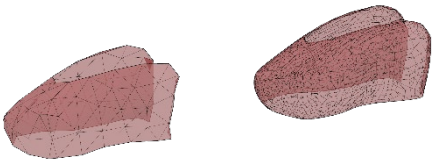


Figure 51: Mesh Reconstruction and Subdivision

Stage 3 – Quad remeshing

The subdivided mesh is subsequently converted into a quadrilateral topology. Quad-based representations are preferred in geometric processing due to their inherent directional continuity, which facilitates smoother parameterization and improves compatibility with NURBS surface reconstruction and strip-based discretization strategies. However, when the input geometry represents a complex double-curved surface—particularly those exhibiting significant variations in Gaussian curvature, including concave and convex regions—the resulting quad mesh may not maintain uniform cell distribution across its parametric directions. Instead, distortion manifests as localized stretching and compression of quad elements, leading to non-equivalent cell sizes and irregular parameter spacing. This non-uniformity can negatively affect vertex-based NURBS surface construction.

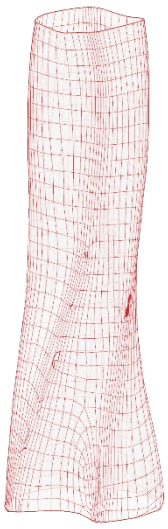


Figure 52: Quad remeshing

Stage 4 – Boundary Extraction and Guide Generation

In addition to non-uniform cell distribution, uneven number of cells in each row of the quadrilateral mesh, and irregular spacing, the given mesh may also contain holes. To address this issue, the boundary edges are first extracted and offset outward. These offset curves serve as guide geometry for generating a regular enclosing vertical mesh structure.

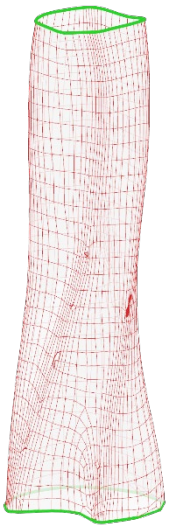


Figure 53: Boundary edge detection

Stage 5 – Generation of Structured Vertical Quad Mesh

A regular vertical quad grid is generated around the model. User-defined U and V parameters control row and column density. The resulting mesh serves as a topologically clean starting fabric.

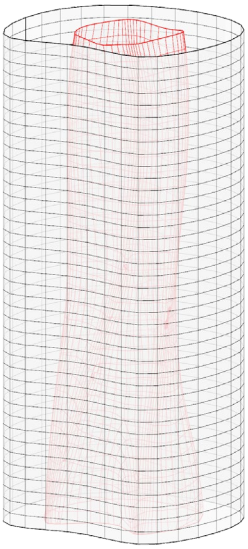


Figure 54: Generation of Structured Vertical Quad Mesh

Stage 6 – Kangaroo Physics Relaxation

The generated structured vertical quadrilateral mesh is refined using the Kangaroo Physics engine, which simulates the behaviour of an elastic fabric. The regular quad mesh is attracted toward the target geometry (the given mesh) while preserving its topological structure through a combination of physical constraints. The process is similar to draping a piece of elastic fabric over a rigid form, allowing the mesh to conform smoothly to the target geometry without losing its structured organization.

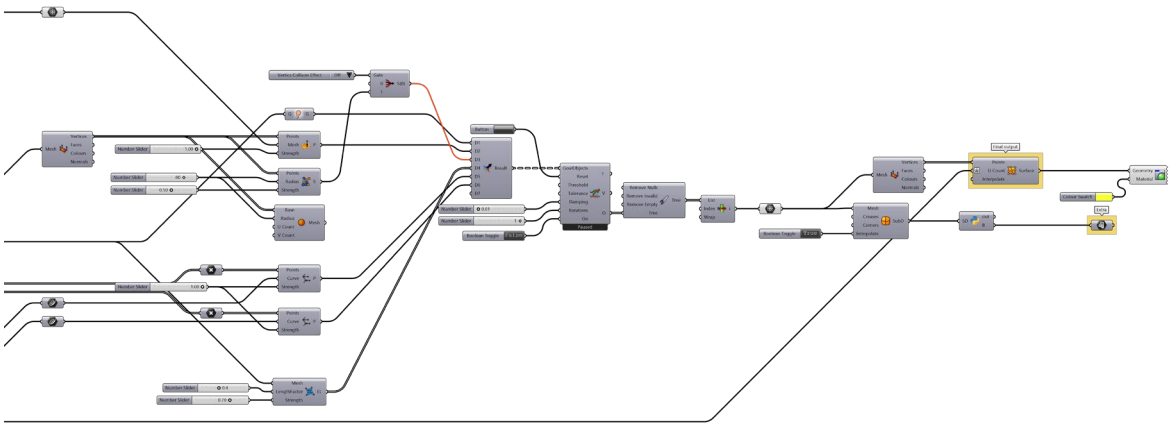


Figure 55: Kangaroo Physics Grasshopper Logic

To ensure geometric accuracy and stability during the simulation, several Kangaroo goals are applied:

- OnMesh Goal: Each vertex of the structured quad mesh is attracted toward the target mesh, ensuring that the relaxed mesh conforms closely to the original geometry.
- OnCurve Goal: The upper and lower boundary edges of the structured vertical quad mesh are constrained to remain on the corresponding top and bottom boundary curves of the given mesh. These anchor constraints preserve the overall height and boundary alignment throughout the relaxation process.
- SphereCollide Goal: A collision sphere is generated around each mesh vertex to prevent neighbouring vertices from intersecting or overlapping during deformation. This maintains a valid mesh configuration and improves the uniformity of the resulting cell distribution.
- EdgeLength Goal: Elastic behaviour is assigned to every mesh edge by specifying a target edge-length reduction factor. This goal acts as a network of springs that regulates edge stretching and compression, promoting a more uniform distribution of quadrilateral cells while allowing sufficient flexibility for the mesh to adapt to the target geometry.

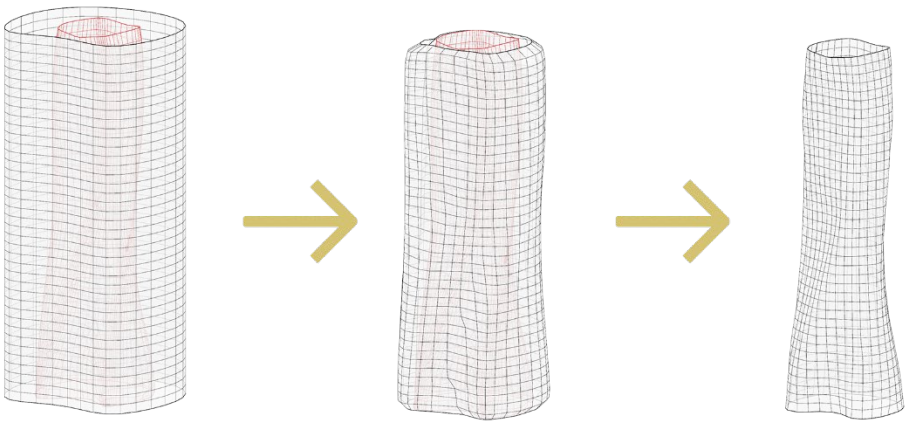


Figure 56: Kangaroo Physics Relaxation

These goals are solved simultaneously by the Kangaroo engine until an equilibrium state is reached, producing a structured quadrilateral mesh that closely approximates the original geometry while maintaining regular topology and improved cell distribution.

Stage 7 – NURBS Surface Reconstruction

After relaxation, the organized quad mesh possesses coherent edge flow and regular topology. This allows reliable conversion into a single smooth NURBS surface suitable for downstream rationalization and strip discretization.

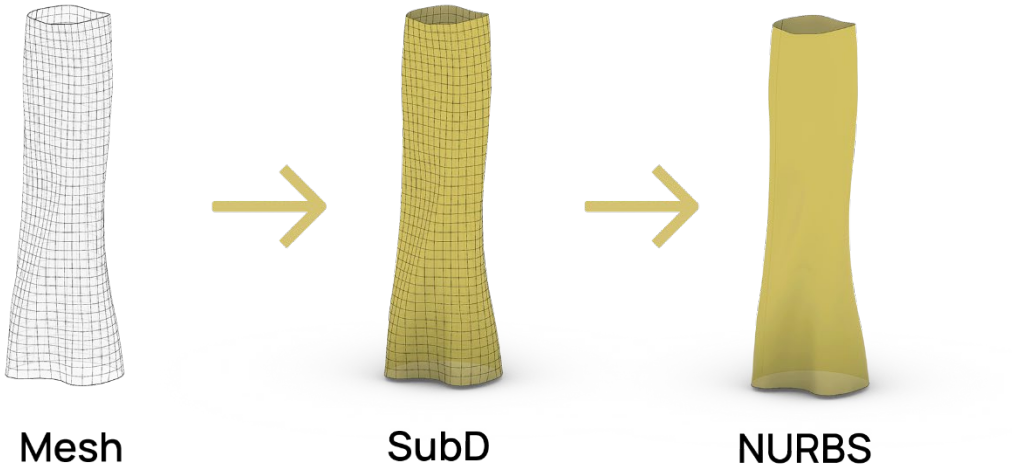


Figure 57: NURBS Surface Reconstruction

Following the physical simulation, the resulting mesh is post-processed to remove any null or invalid mesh elements generated during the iterative solution. The cleaned quadrilateral mesh is then converted into a Subdivision (SubD) surface to further smooth the geometry while preserving the underlying quadrilateral topology. Finally, a custom Python script converts the SubD representation into a NURBS surface, producing a smooth and editable surface suitable for downstream Further fabrication workflows.

Overall, this workflow creates a controlled transition from irregular AI generated 3D geometry to fabrication-friendly surfaces. The organized topology facilitates, strip discretization, and manufacturing workflows for double-curved architectural surfaces.

4.2.2. Gaussian curvature: principal understanding

As part of the stripe modelling exploration, and informed by the referenced academic studies, more than six modelling methods were tested and compared based on their resulting Gaussian curvature values. While all methods produced values extremely close to zero, slight variations were observed. In architectural metal applications, Gaussian curvature magnitudes below $\mathbf{10^{-4}}$ are commonly considered negligible. Since none of the tested methods exceeded this threshold, the selection of the final method was guided primarily by architectural and geometric coherence before proceeding with the surface discretization into stripes.

In the following some important definitions that have been used in the mentioned methods.

Definitions

Consider a special curve in space $(\boldsymbol{c(t)})$. This curve can be described using two frames that move along it according to a parameter $(\boldsymbol{t})$, which represents a point on the curve. In the following section, useful definitions employed to model the surface stripes.

Tangent Vector (Maekawa and Scholz 2024; Capone 2018) [45] [46]

The tangent vector $(\boldsymbol{t})$ represents the direction of a curve at a given point.

For a smooth space curve: If the curve is arc-length-parameterized $(\boldsymbol{s})$, the unit tangent vector is the first derivative with respect to arc length, $\boldsymbol{t(s) = c'(s)}$

For an arbitrarily parameterized curve $(\boldsymbol{c(t)})$, it is defined as the first derivative divided by its magnitude, $\boldsymbol{t(t) = \dot{c}\,(t) / \| \dot{c}\,(t) \|}$.

The symbol $\| \boldsymbol{x} \|$ represents the norm, magnitude, or length of a vector.

Principal Normal Vector (Maekawa and Scholz 2024) [45]

The principal normal vector $(\boldsymbol{n})$ is a unit vector that is orthogonal to the tangent vector and lies within the curve's osculating plane.

For a smooth space curve: In arc-length parameterization $(\boldsymbol{s})$, it is the unit vector in the direction of the second derivative: $\boldsymbol{n(s) = c''(s) / \| c''(s) \|}$.

For arbitrary parameterization, it is calculated as the cross product of the binormal and tangent vectors: $\boldsymbol{n(t) = b(t) \times t(t)}$

Key Properties: It points toward the center of the curve's osculating circle (the circle that most tightly fits the curve at that point), and its direction is associated with the curve's curvature $(\boldsymbol{k})$

Binormal Vector (Maekawa and Scholz 2024) [45]

The binormal vector ($\boldsymbol{b}$) is the third vector of the moving frame, serving as the normal to the osculating plane.

Mathematical Definition: It is the unit vector obtained by the cross product of the tangent and principal normal vectors: $\boldsymbol{b} = \boldsymbol{t} \times \boldsymbol{n}$

For arbitrary parameterization: It can be calculated directly using the first and second derivatives of the curve:

$$b(t) = \frac{(\dot{c}(t) \times \ddot{c}(t))}{\| \dot{c}(t) \times \ddot{c}(t) \|}$$

Frenet Frame (Maekawa and Scholz 2024) [45]

The Frenet frame (also known as the Frenet-Serret frame) is the moving orthonormal triplet ($\boldsymbol{t}, \boldsymbol{n}, \boldsymbol{b}$) that provides an intrinsic basis for analyzing a space curve's spatial behavior

Function: It describes how the curve bends and twists in space through the Frenet-Serret formulae, which are differential equations relating the derivatives of the frame vectors to the curve's curvature (k) and torsion (τ)

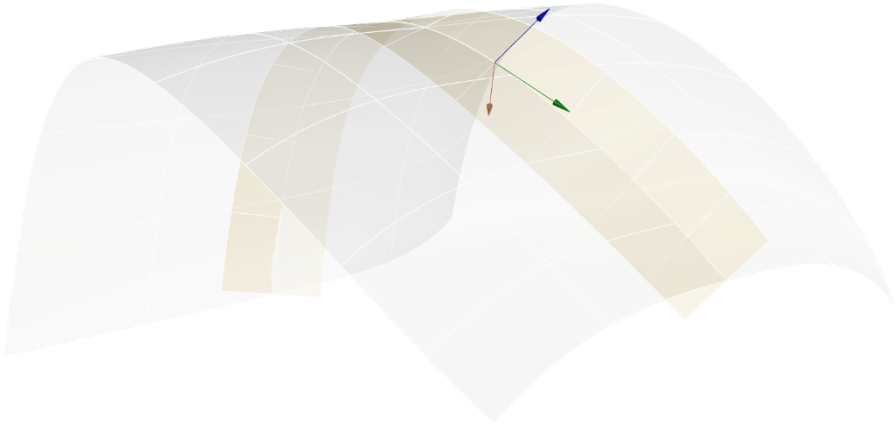


Figure 58: Surface Stripe-Frenet Frame

400mm thk. Stripe modelling using the binormal vector of the Frenet frame along a spatial curve on a doubly curved surface approx. 2 x 3 m. The stripe follows the curvature direction defined by the guiding curve, independent of the underlying

surface geometry. This indicates that the stripe is extrinsically governed by the spatial curve rather than by the surface orientation.

Curvature

The Curvature (k) is a geometric invariant that measures how sharply a curve bends or deviates from being a straight line.

Geometric Definition: It is defined as the reciprocal of the radius (r) of the osculating circle, which is the circle that most tightly fits the curve at a specific point ($k = 1/r$)

Arc-Length Parameterization (s): For a curve parameterized by its arc length, the curvature is the magnitude of the second derivative: $\kappa(s) = \| c''(s) \|$.

Arbitrary Parameterization (t): For a curve $c(t)$ with a general parameter, the formula is:

$$\kappa(t) = \frac{\| \dot{c}(t) \times \ddot{c}(t) \|}{\| \dot{c}(t) \|^3}$$

where $\dot{c}(t)$ and $\ddot{c}(t)$ are the first and second derivatives with respect to (t).

Torsion

Torsion (τ) measures the extent to which a curve twists out of its osculating plane, indicating its deviation from being a planar curve.

Arc-Length Parameterization (s): For a curve $c(t)$, it is defined using the scalar triple product of its first three derivatives:

$$\tau(s) = \frac{[c'(s), c''(s), c'''(s)]}{\| c''(s) \|^2}$$

Arbitrary Parameterization (t): For a curve $c(t)$, the calculation is:

$$\tau(t) = \frac{[\dot{c}(t), \ddot{c}(t), \dddot{c}(t)]}{\| \ddot{c}(t) \|^2}$$

Darboux Frame (Maekawa and Scholz 2024) [45]

The Darboux frame is a moving orthogonal coordinate system used to describe the geometry of a curve as it relates to a surface on which it lies. Unlike the Frenet frame, which is purely intrinsic to the curve's own path, the Darboux frame is extrinsic, meaning it depends on the orientation and curvature of the underlying surface.

Tangent Vector (t): This is the unit vector pointing in the direction of the curve at a specific point. It is the same tangent vector used in the Frenet frame.

Surface Normal Vector ($\mathbf{N}$*)*: This is the unit vector that is orthogonal to the tangent plane of the surface at the point where the curve is located.

Geodesic Normal Vector ($\mathbf{B}$*)*: Often referred to as the tangent normal, this unit vector lies within the tangent plane of the surface and is orthogonal to both the tangent vector and the surface normal. It is mathematically defined by the cross product of the surface normal and the tangent vector ($\boldsymbol{B} = \boldsymbol{N} \times \boldsymbol{t}$).

Gaussian curvature Comparison

The following diagrams illustrate the Stripe surface modelled using different reference frames, along with the degree to which each configuration aligns with the principal directions of the given doubly curved surface.

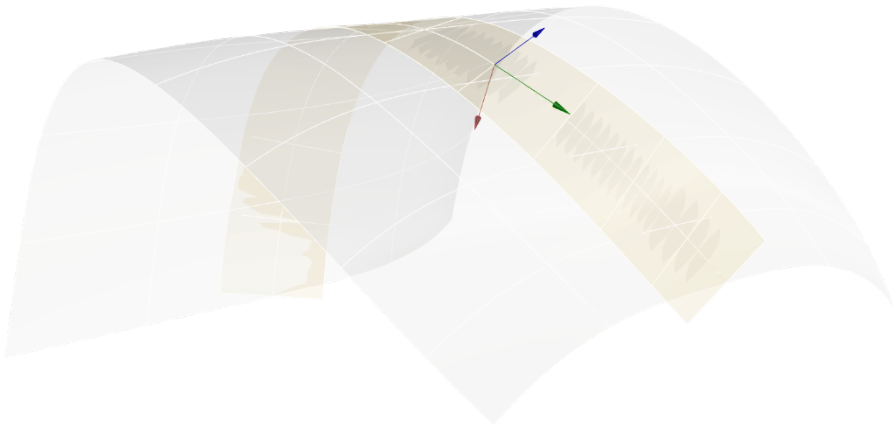


Figure 59: Surface Stripe-Darboux Frame

400mm thk. Stripe modelling using the binormal vector of the Darboux frame along a spatial curve on a doubly curved surface approx. 2 x 3 m. The stripe adapts to the surface orientation while following the guiding curve, showing that the geometry is intrinsically linked to both the curve and the surface and produces lower Gaussian curvature compared to the Frenet frame method.

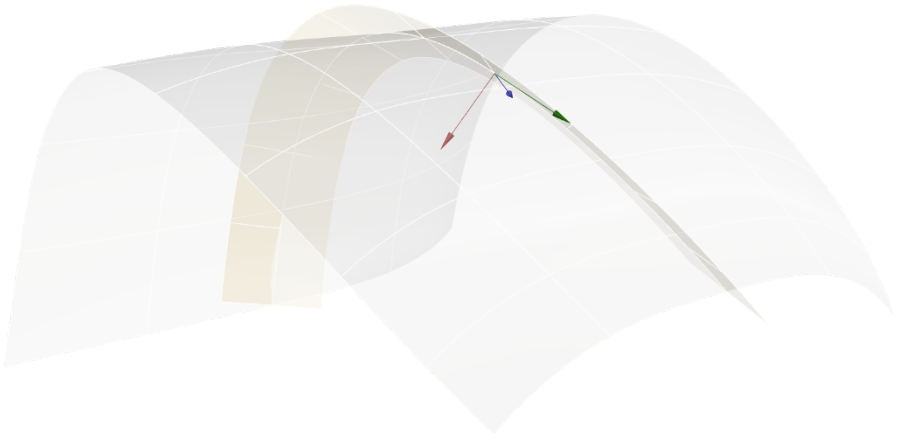


Figure 60: Surface Stripe- Perpendicular Frames

400mm thk. Stripe modelling using the Y-direction of perpendicular frames defined along a guiding curve. on a doubly curved surface approx. 2 x 3 m. At each point, the frame aligns its Z-axis with the tangent of the curve, while the remaining axes can freely rotate about the tangent direction. Unlike the Frenet and Darboux frames, which are constrained by the curve's principal normal and the surface normal respectively, the perpendicular frame allows rotational freedom around the tangent, resulting in near-zero Gaussian curvature values. (Most developable option)

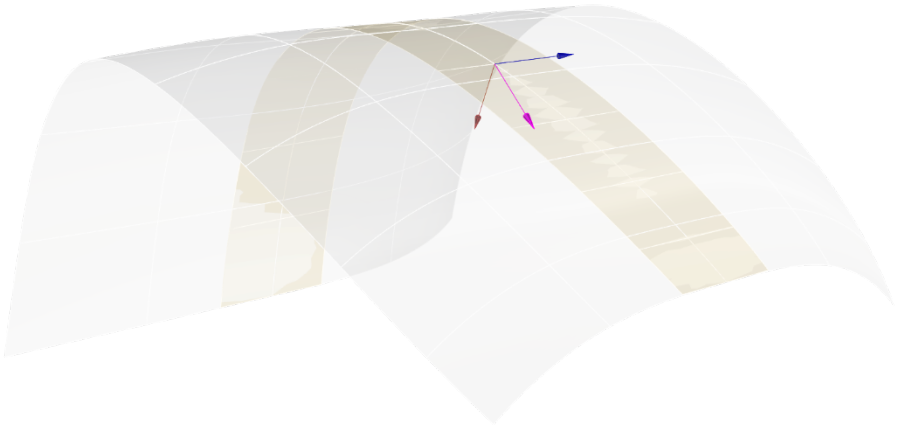
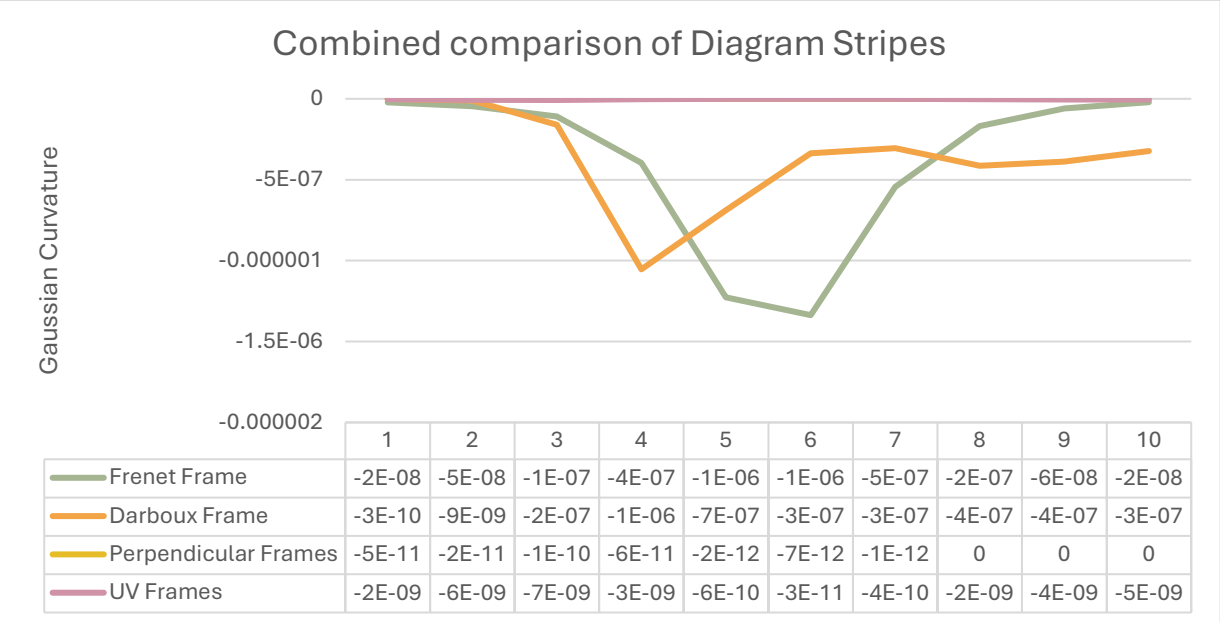


Figure 61: Surface Stripe- UV Frame

400mm thk. Stripe modelling using the U-direction of the surface UV frame along the guiding curve. on a doubly curved surface approx. 2 x 3 m. While the resulting stripe is not strictly developable, it exhibits very low Gaussian curvature. This allows the stripe to closely align with the surface orientation, including along its edges, producing the

most consistent geometric integration with the underlying surface. (most pleasing option)

The following graph shows the Gaussian curvature values of the 4 methods employed in the previous diagrams showing the negligible values and proving the extreme closeness of the UV frames values to Zero witch at the same time the most coherent solution to the original surface orientation.



	1	2	3	4	5	6	7	8	9	10
Frenet Frame	-2E-08	-5E-08	-1E-07	-4E-07	-1E-06	-1E-06	-5E-07	-2E-07	-6E-08	-2E-08
Darboux Frame	-3E-10	-9E-09	-2E-07	-1E-06	-7E-07	-3E-07	-3E-07	-4E-07	-4E-07	-3E-07
Perpendicular Frames	-5E-11	-2E-11	-1E-10	-6E-11	-2E-12	-7E-12	-1E-12	0	0	0
UV Frames	-2E-09	-6E-09	-7E-09	-3E-09	-6E-10	-3E-11	-4E-10	-2E-09	-4E-09	-5E-09

Graph 1: Combined comparison of Diagram Stripes using deferent frames

In the following graphs it is being compared deferent types of frame methods implemented to build the stripes according to two different guiding curves on the surface, 1st geodesic curves and 2nd designed pattern curves, unlike expectation the Gaussian curvature values of the stripes built by the predesigned pattern were less than using the geodesic curves. But in summary all the values are less than 10-4 a common threshold in unrolling for fabrication and that make sense since almost all architectural application does not contain a very high curvature in a small area with contribute to much higher Gaussian curvature value

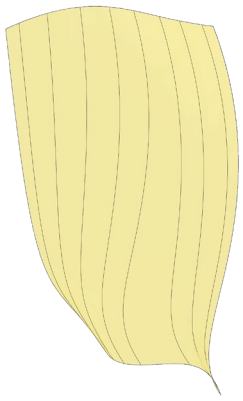


Figure 62: Designed Pattern curves on

The framing methods implemented are as follows

1. DP_Frenet Frame
2. DP_Darboux Frame
3. DP_Perpendicular Frames – A built-in component in grasshopper
4. DP_UV Frames – UV direction at each point on the original surface to assure the orientation of the stripes.
5. DP_Python Frames – In this option the main reference was this paper (Maekawa and Scholz 2024) where I used AI to build a python scripts using the equations and methods used in the reference, but the result were identical to the result by drobox frame!

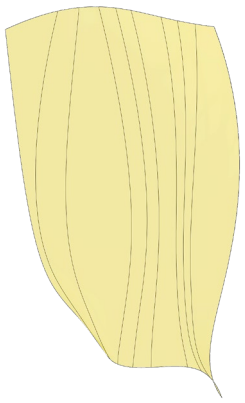
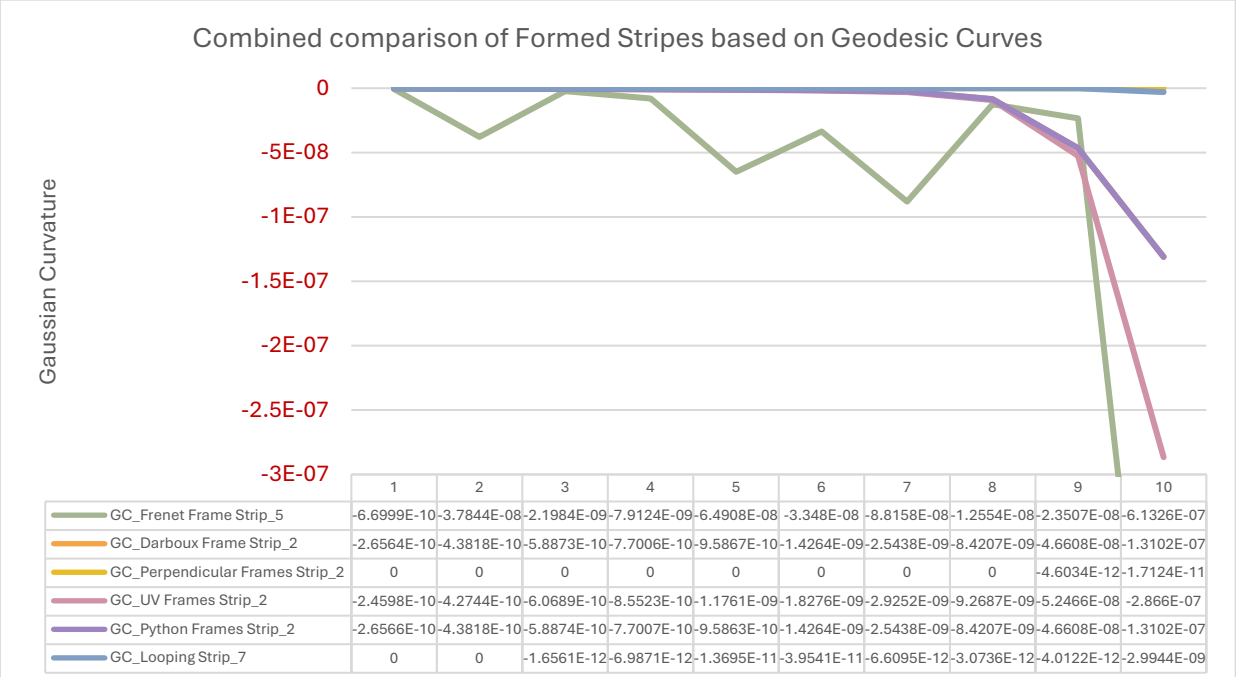


Figure 63: Geodesic Curves on the Proposed

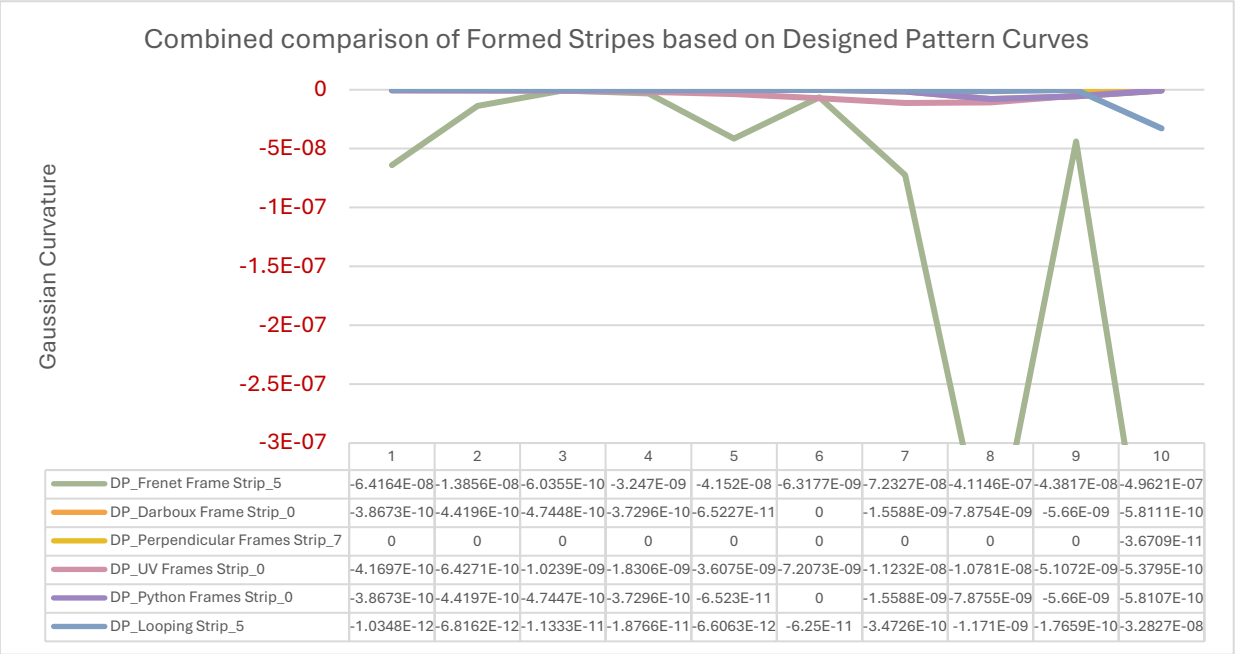
6. DP_Looping – In this option the angle between following tangents has been calculated to be used to rotate a generic direction - defined by designer – from point to point and by this methodology we avoided the unnecessary torsion that could be caused whin using the normal direction at each point on the original surface.

DP stand for Designed Pattern while GC stands for Geodesic Curves



	1	2	3	4	5	6	7	8	9	10
GC_Frenet Frame Strip_5	-6.6999E-10	-3.7844E-08	-2.1984E-09	-7.9124E-09	-6.4908E-08	-3.348E-08	-8.8158E-08	-1.2554E-08	-2.3507E-08	-6.1326E-07
GC_Darboux Frame Strip_2	-2.6564E-10	-4.3818E-10	-5.8873E-10	-7.7006E-10	-9.5867E-10	-1.4264E-09	-2.5438E-09	-8.4207E-09	-4.6608E-08	-1.3102E-07
GC_Perpendicular Frames Strip_2	0	0	0	0	0	0	0	0	-4.6034E-12	-1.7124E-11
GC_UV Frames Strip_2	-2.4598E-10	-4.2744E-10	-6.0689E-10	-8.5523E-10	-1.1761E-09	-1.8276E-09	-2.9252E-09	-9.2687E-09	-5.2466E-08	-2.866E-07
GC_Python Frames Strip_2	-2.6566E-10	-4.3818E-10	-5.8874E-10	-7.7007E-10	-9.5863E-10	-1.4264E-09	-2.5438E-09	-8.4207E-09	-4.6608E-08	-1.3102E-07
GC_Looping Strip_7	0	0	-1.6561E-12	-6.9871E-12	-1.3695E-11	-3.9541E-11	-6.6095E-12	-3.0736E-12	-4.0122E-12	-2.9944E-09

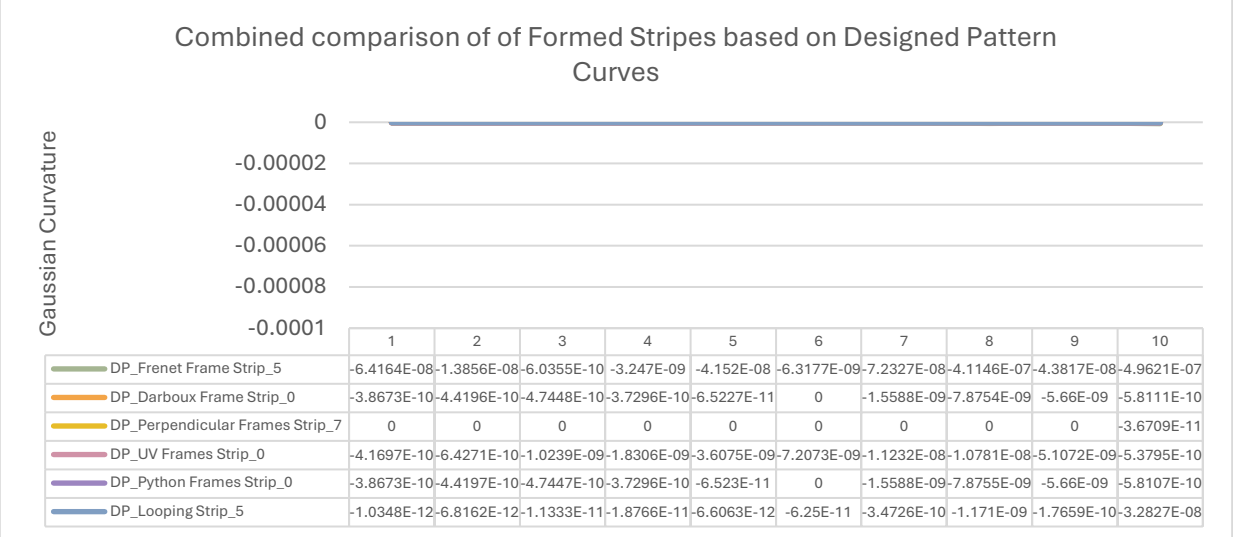
Graph 2: Combined comparison of Formed Stripes based on Geodesic Curves



	1	2	3	4	5	6	7	8	9	10
DP_Frenet Frame Strip_5	-6.4164E-08	-1.3856E-08	-6.0355E-10	-3.247E-09	-4.152E-08	-6.3177E-09	-7.2327E-08	-4.1146E-07	-4.3817E-08	-4.9621E-07
DP_Darboux Frame Strip_0	-3.8673E-10	-4.4196E-10	-4.7448E-10	-3.7296E-10	-6.5227E-11	0	-1.5588E-09	-7.8754E-09	-5.66E-09	-5.8111E-10
DP_Perpendicular Frames Strip_7	0	0	0	0	0	0	0	0	0	-3.6709E-11
DP_UV Frames Strip_0	-4.1697E-10	-6.4271E-10	-1.0239E-09	-1.8306E-09	-3.6075E-09	-7.2073E-09	-1.1232E-08	-1.0781E-08	-5.1072E-09	-5.3795E-10
DP_Python Frames Strip_0	-3.8673E-10	-4.4197E-10	-4.7447E-10	-3.7296E-10	-6.523E-11	0	-1.5588E-09	-7.8755E-09	-5.66E-09	-5.8107E-10
DP_Looping Strip_5	-1.0348E-12	-6.8162E-12	-1.1333E-11	-1.8766E-11	-6.6063E-12	-6.25E-11	-3.4726E-10	-1.171E-09	-1.7659E-10	-3.2827E-08

Graph 3: Combined comparison of Formed Stripes based on Designed Pattern Curves

In the following graph you can see the real scale to the reported values when compared to the common threshold value of the Gaussian curvature 10^{-4}.



	1	2	3	4	5	6	7	8	9	10
DP_Frenet Frame Strip_5	-6.4164E-08	-1.3856E-08	-6.0355E-10	-3.247E-09	-4.152E-08	-6.3177E-09	-7.2327E-08	-4.1146E-07	-4.3817E-08	-4.9621E-07
DP_Darboux Frame Strip_0	-3.8673E-10	-4.4196E-10	-4.7448E-10	-3.7296E-10	-6.5227E-11	0	-1.5588E-09	-7.8754E-09	-5.66E-09	-5.8111E-10
DP_Perpendicular Frames Strip_7	0	0	0	0	0	0	0	0	0	-3.6709E-11
DP_UV Frames Strip_0	-4.1697E-10	-6.4271E-10	-1.0239E-09	-1.8306E-09	-3.6075E-09	-7.2073E-09	-1.1232E-08	-1.0781E-08	-5.1072E-09	-5.3795E-10
DP_Python Frames Strip_0	-3.8673E-10	-4.4197E-10	-4.7447E-10	-3.7296E-10	-6.523E-11	0	-1.5588E-09	-7.8755E-09	-5.66E-09	-5.8107E-10
DP_Looping Strip_5	-1.0348E-12	-6.8162E-12	-1.1333E-11	-1.8766E-11	-6.6063E-12	-6.25E-11	-3.4726E-10	-1.171E-09	-1.7659E-10	-3.2827E-08

Graph 4: Combined comparison

Combined comparison of Formed Stripes based on Designed Pattern Curves Graph with Max. -0.0001.

Therefore, and as proven and discussed using the UV frames is feasible and fabricable in the architectural scale and more coherent with original surface orientation and design intent.

4.2.3. Surface Band Strips-discretisation (Main Grasshopper Logic)

This section and the following two sections describe a comprehensive parametric workflow implemented in **Grasshopper** for the automated conversion of freeform surface geometries into structurally integrated, assembly-ready components for **sheet-based digital fabrication**.

The workflow consists of three consecutive stages. First, **Surface Band Strips Discretisation** subdivides the continuous surface into a set of discrete, interconnected, near-developable strip elements. Second, **Geometry Detailing** augments these elements with structural substructures, connection interfaces, and fastening mechanisms necessary for assembly. Finally, **Unrolling and Nesting for Fabrication** transforms the three-dimensional components into optimized data organized two-dimensional cutting elements, enabling efficient material usage and fabrication planning.

All computational operations were implemented using **native Grasshopper components**, including surface analysis, curve manipulation, and data structure management. This approach enhances the transparency, reproducibility, and long-term maintainability of the workflow while minimizing reliance on external software dependencies. The only exception is Section **4.2.5 Unrolling and Nesting for Fabrication**, where the workflow employs the **OpenNest 2** plugin to perform two-dimensional nesting and sheet layout optimization for efficient material utilization during fabrication.

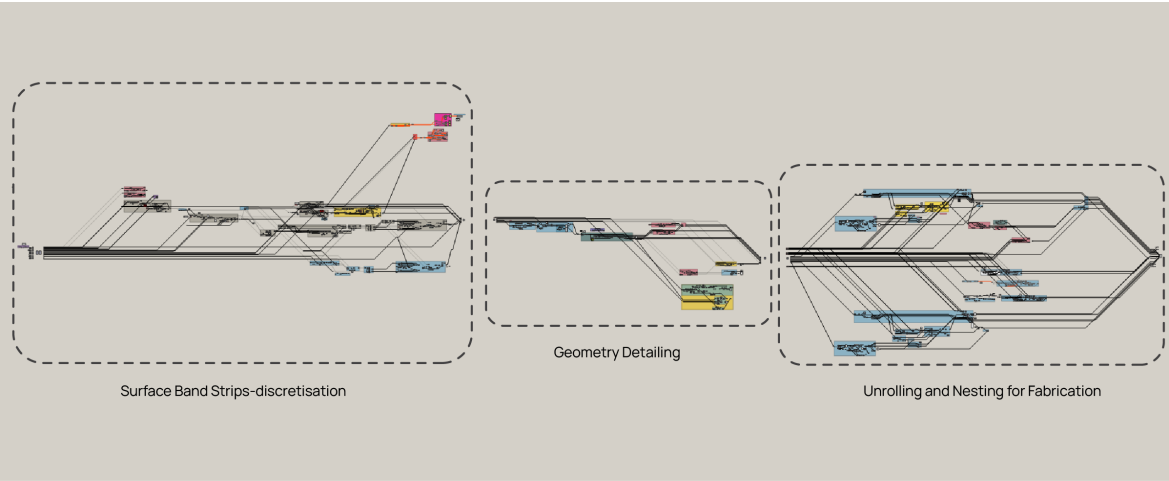


Figure 64: General view of the Main Grasshopper Logic Parts.

The discretization stage transforms continuous input surface geometry into a series of **band strip components** that preserve the formal expression of the original given surface while enabling digital fabrication. This process addresses the fundamental challenge of converting freeform surfaces into components compatible with sheet-based manufacturing constraints. The discretization strategy is predicated upon three core principles: (1) **preservation of surface curvature** continuity along the strips and

near the edges, (2) minimization of Gaussian curvature to maintain **near-developability**, and (3) structured hierarchy of **primary strips and secondary branching** connections that form the band according to a desired pattern input.

To illustrate the proposed geometric discretization method in a clear and systematic manner, the process is first demonstrated on a representative portion of the surface rather than the complete geometry. This simplified approach facilitates the explanation of the underlying principles while improving the readability and robustness of the methodology. The following sections describe the main computational logic of the workflow, which is developed entirely within the Grasshopper visual programming environment.

1. Input Geometry

The workflow begins with the import of a network of boundary curves defining the intended freeform geometry. These curves establish the principal U and V parametric directions of the surface and serve as the geometric framework for all subsequent operations. For example, a façade portion measuring approximately 9.5 m × 17 m (typical of a five-storey building bay).

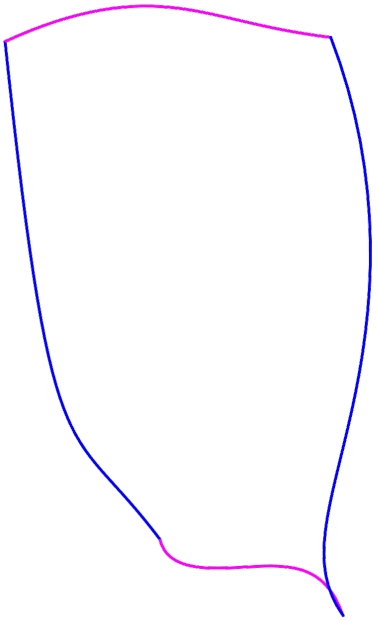


Figure 65: Input Geometry

2. Surface Creation

A reference surface is generated using the Surface from Network of Curves operation. This surface acts as the primary geometric host throughout the workflow, providing both the spatial reference and the parametric domain required for discretisation, structural detailing, and fabrication preparation.

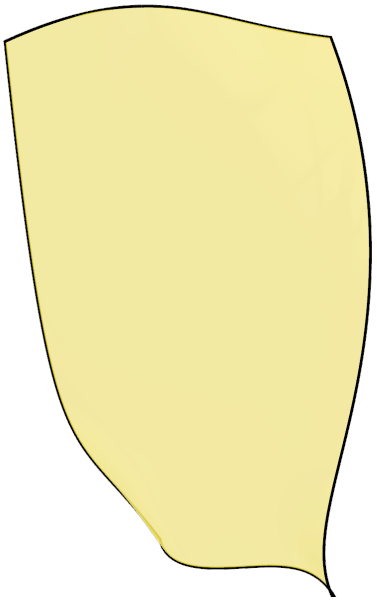


Figure 66: Surface Creation

3. Surface Curvature analysis

Prior to discretisation, a Gaussian curvature analysis is performed to evaluate the geometric behaviour and developability of the input surface. This analysis provides an understanding of regions exhibiting single and double curvature.

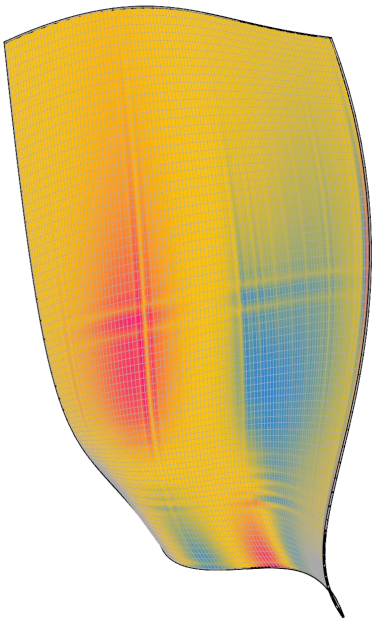


Figure 67: Surface Curvature Analysis

Alternative Developability Verification Using Normal Vector Convergence

In addition to conventional Gaussian curvature analysis, an alternative graphical method was employed to qualitatively assess the developability of the generated surfaces. This procedure follows the approach presented by Bollinger + Grohmann Ingenierie in TUTO#01 – Introduction to Curvature Analysis and Developability as part of the Poplar Workshop / ENSAV series.

The method begins by discretizing the surface into a regular UV grid, generating a network of surface points together with their associated normal vectors. For each grid point, the local normal direction is evaluated. Rather than preserving the original base point of each normal vector, all vectors are translated so that their tails coincide at a single arbitrary point in space, while maintaining their original orientation and length. Consequently, the original UV grid curves move with the translated normal vectors according to their original topological order.
Since the base points of all vectors have been unified, the resulting curves represent the directional evolution of the surface normals rather than the surface geometry itself.

For a developable surface, the normal vectors vary only along one principal direction. Therefore, when the vector tails are relocated to a common origin, the reconstructed grid curves tend to collapse onto a single curve or a narrow bundle of nearly coincident curves. This behaviour indicates that the normal field is effectively one-dimensional, which is characteristic of developable surfaces exhibiting zero Gaussian curvature.

Conversely, when the surface contains regions of double curvature, the orientation of the normal vectors that share the same grid curve changes in independent directions. After relocating the vectors to a common origin, the reconstructed grid curves no longer overlap and instead form a complex network of intersecting or diverging curves. Such behaviour reveals that the normal field occupies a two-dimensional region, indicating the presence of non-developable areas and positive or negative Gaussian curvature.

Reference: Bollinger + Grohmann Ingenieure, TUTO#01 – Introduction to Curvature Analysis and Developability, Poplar Workshop / ENSAV, Intelligent Fabrication YouTube Channel.
Link: TUTO#01 - Introduction to Curvature Analysis and Developability

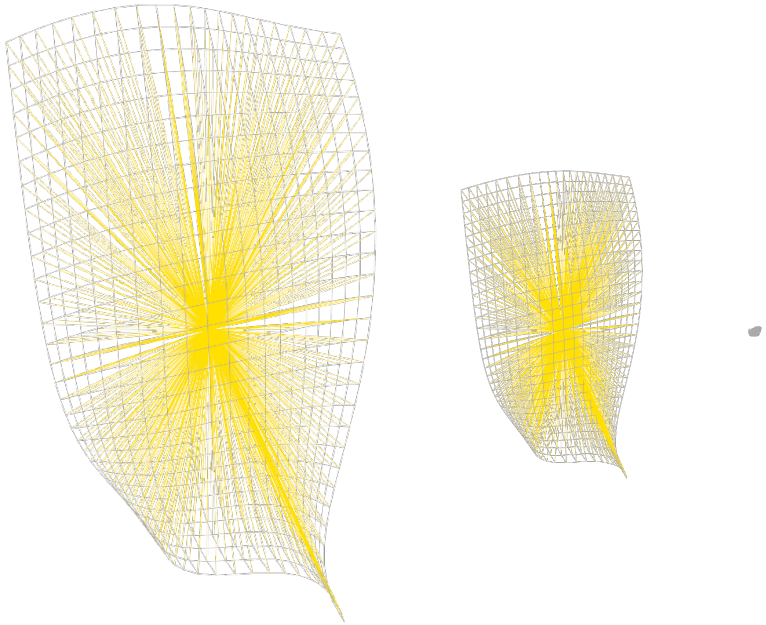


Figure 68: Vector Transport Surface Curvature Analysis

This graphical verification complements the numerical Gaussian curvature analysis and provides an intuitive visual assessment of surface developability. At the end of strips discretisation to this double curved surface, both methods will be employed to confirm that the generated strip surfaces exhibit negligible double curvature and are therefore compatible with sheet-based fabrication processes.

4. Surface Division (Strips Modelling)

After the reference surface is established, a set of longitudinal curves is generated excluding the side-edge boundaries and manipulated to define the main strip layout.

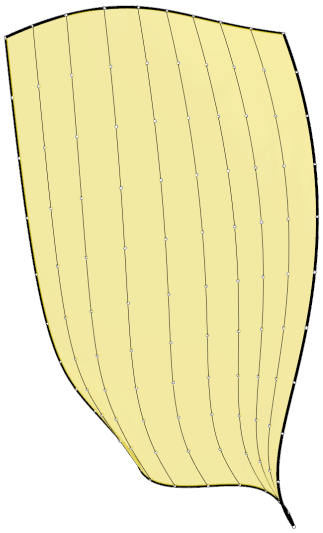


Figure 69: Surface Division

5. Wavy parametric Curve Manipulation and Strip Control

The definition includes a “Wavy Curves Manipulation” stage, where the strip curves are parametrically adjusted through applying sinusoidal deformation using a sine graph. This operation controls the number, spacing, and shape of the curves before they are used to generate the strip surfaces.

The logic reduces and reorganises the curve network, then introduces controlled deformation to create a more expressive strip pattern. This approach enables the strip system to accommodate a wider range of design parameters and geometric variations.

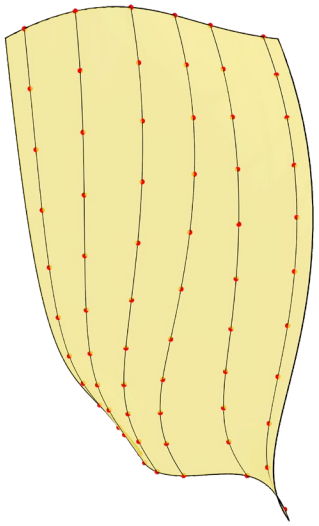


Figure 70: Curve Manipulation

6. Branch Strip Guide Curves Generation

The workflow incorporates a branching logic to establish connections between adjacent main strips. Branch curves are generated by selecting division points along consecutive strip curves according to a user-defined pattern, with the specific locations controlled through a random seed parameter. Corresponding points on neighbouring curves are then connected using interpolated curves that maintain tangential continuity with the main strip curves at their endpoints. The resulting branch curves are subsequently projected onto the base surface to preserve geometric coherence and ensure smooth transitions between the interconnected strip elements.

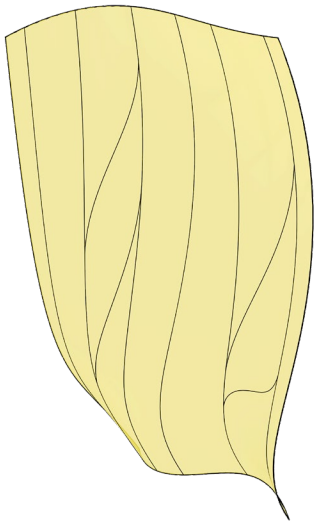


Figure 71: Branch Strip Guide Curves Generation

7. Main Strip Surface Generation

The main strip surfaces are generated from curve division points and cross-section lines in the U direction of each division point that is pulled to the surface. The definition contains a clearly labelled "Main Strips Surfaces" group, where cross-sections are constructed along the curves and lofted to create continuous rolling strip surfaces.

These strip surfaces are designed to follow the orientation of the original surface while reducing double curvature. The Gaussian curvature check confirms that the resulting strip surfaces have very low curvature values. The **observed values range between 10^{-8} and 10^{-11}, indicating negligible double curvature.** allowing them to be treated as near-developable. This makes them suitable for later unrolling and sheet fabrication.

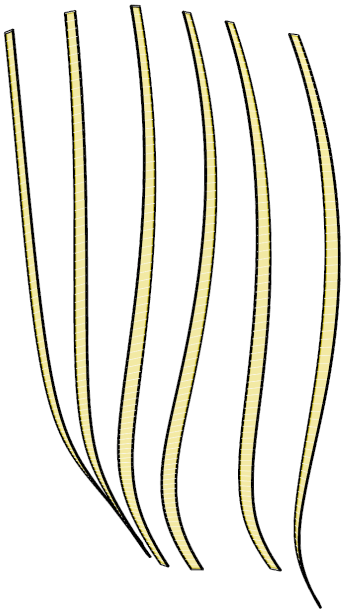


Figure 72: Main Strip Surface Generation

Side Note:

Alternative more developable surfaces were tested, where they have gaussian curvature equal to 0 for most of the strips but they resulted in undesirable orientation, exposing sub-structure externally and failing to follow the facing direction of the input surface. As described in section 4.2.2. Gaussian curvature - principal understanding.

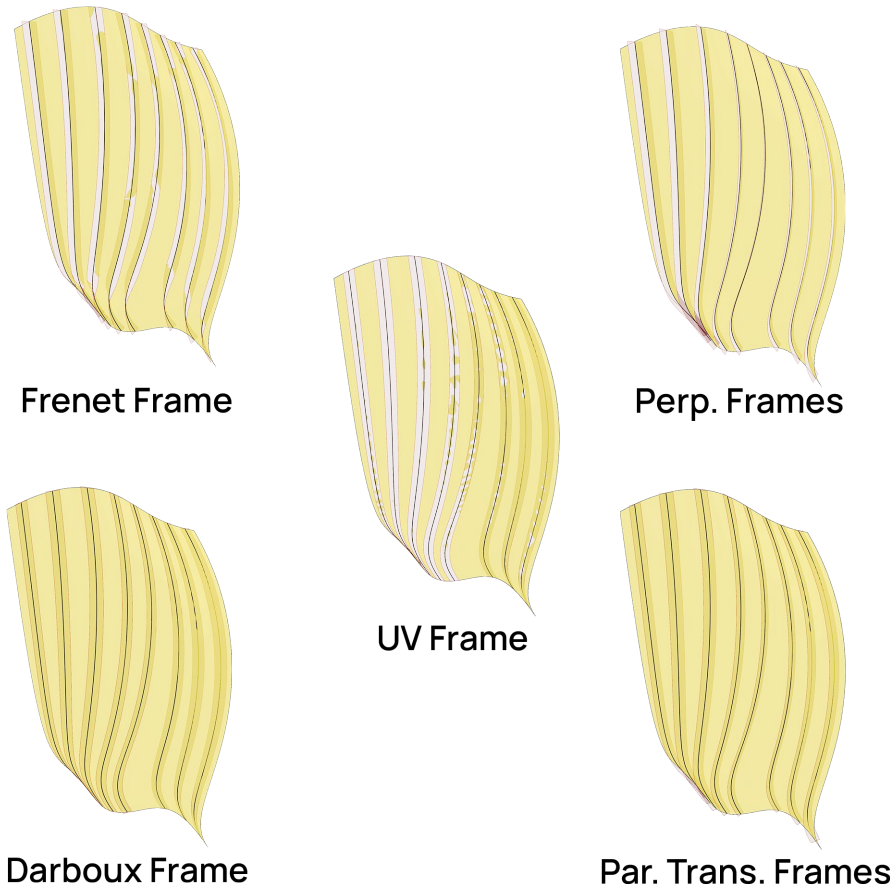


Figure 73: Strips construction comparison

The strip surfaces were generated using five different local frame definitions:

- Frenet Frame – The strip orientation is determined from the tangent, normal, and binormal vectors computed at each discretized point along the guiding curve. Although this moving frame is mathematically well-defined, it is highly sensitive to local curvature variations and becomes unstable at inflection points, resulting in excessive twisting and inconsistent strip orientation.
- Darboux Frame – The frame is derived from the differential properties of the underlying reference surface by combining the curve tangent with the surface normal. Because it is surface-dependent, it better represents the local geometry and generates strips that conform closely to the base surface. However, near trimmed or open boundaries, the frame becomes unstable, causing the strips to detach from the surface edges.
- Perpendicular Frame – Constructed using the Grasshopper *Perp Frame* component, this method generates a plane perpendicular to the curve tangent at each division point. While computationally efficient and straightforward to implement, it is independent of the surface geometry and therefore cannot

account for changes in surface curvature, leading to accumulated rotational drift and inconsistent strip alignment.

- UV Frame – The frame is defined using the local U and V parametric directions of the underlying surface. Since the frame is directly linked to the surface parameterization, the generated strips naturally follow the intended flow of the base surface while remaining aligned with its boundary edges. This method produced the most consistent orientation and best satisfied the architectural design intent.
- Parallel Transport Frame – A user-defined guiding vector is propagated along the curve using parallel transport, minimizing unnecessary rotation between consecutive frames. This rotation-minimizing frame provides excellent local stability and significantly reduces twisting. However, because its orientation is governed primarily by the initial guiding vector rather than the local surface geometry, the strips gradually deviate from the base surface as they extend farther from the starting point.

Although four methods successfully generated strip surfaces with a high degree of developability, none consistently satisfied both the geometric and architectural requirements except the UV Frame approach which provided less developable solution. But the UV-based method provided the most reliable strip orientation, maintained the intended facade facing direction, and preserved alignment with the surface boundaries, making it the most suitable solution for the proposed façade system.

8. Branch Strip Surface Generation

Using the guide curves generated in previous step, the workflow constructs secondary rolling branch surfaces through a lofting operation. The branch geometry is defined by two sets of cross-sections. Along the edges of the branch strips, partial cross-sections are derived from the side edges of the main strips, and the positions of the branch guide curves, the extension of these curves is controlled by a distance from the guide curve as anchor trajectory ensuring smooth geometric transitions between the primary and secondary surfaces. Intermediate cross-sections are then generated parametrically, with their orientation, size, and spacing controlled through user-defined parameters. This approach enables the branch surfaces to adapt continuously to the overall geometry while maintaining smooth connectivity and geometric coherence.

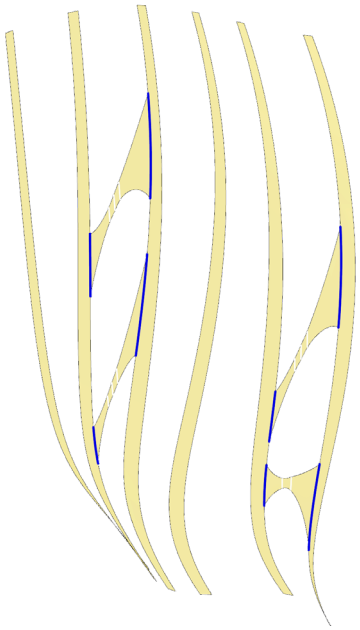


Figure 74: Branch Strip Surface Generation

The branch strips provide smooth transitions between neighbouring main bands. Their cross-sections are parametrically controlled so that the resulting geometry remains coherent with both the main strip surfaces and the underlying reference surface.

9. Main Stripes Sub-Structure Segmentation

The strip curves are divided into smaller segments to prepare for structural subdivision. This division is adjusted according to the intersection point with the branch curves. This means that the workflow does not rely only on uniform curve division. Instead, it inserts intersection points with branch curves into the segmentation system.

After intersection points are inserted, the curve segments are redivided using these points as fixed references. This creates a more controlled point structure along the main strip curves. These points become the anchors for the triangulated substructure.

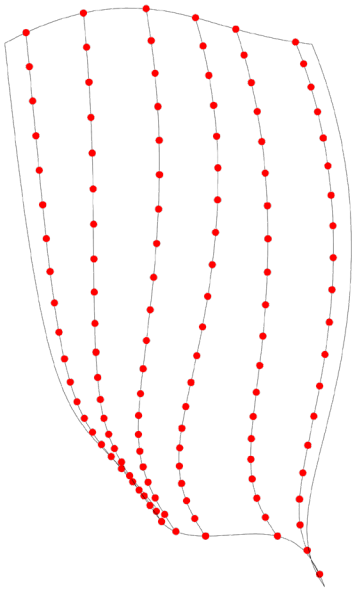


Figure 75: Main Stripes Sub-Structure Initial Division

The redivision step is important because it maintains geometric continuity between the strip pattern and the structural logic. It ensures that the substructure responds to both the main strip direction and the branching intersections.

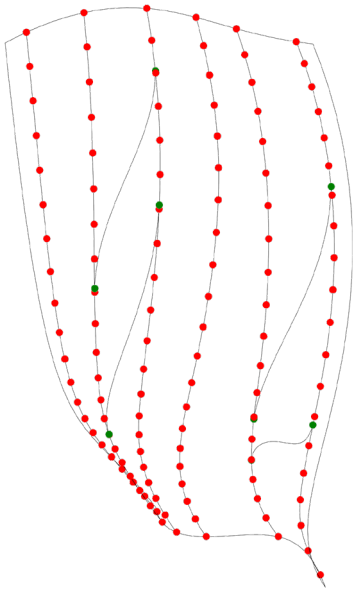


Figure 77: Main Stripes Sub-Structure
Branch intersections insertion

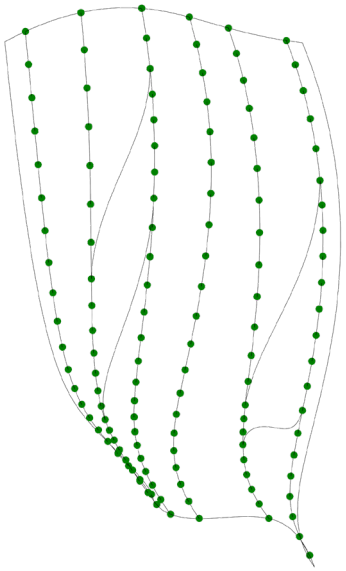


Figure 76: Main Stripes Sub-Structure
Final Redivision

10. Normal Construction (Main Strips Sub-Structure)

The substructure is generated from points evaluated on original surface. lines following the normal direction of the points construct the base of the backbone triangulation.

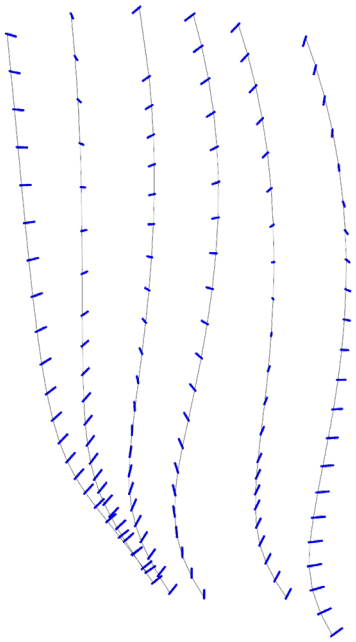


Figure 78: Normal Construction

11. Main Strips Sub-Structure Generation

The triangulated backbone sub-structure forms a secondary structural layer that supports the strip system and provides the welding interface for the outer strip surfaces. The triangulation is generated from normal vectors constructed at the segmentation points, with a user-defined depth controlling the extension of the structural elements. These normal-based members are intentionally extended beyond the original surface, after which they are intersected with the base surface to remove the excess portions. As a result, only the curved side of the triangulated backbone remains, conforming to the local geometry of the reference surface and serving as a guide onto which the stripe surfaces are subsequently welded.



Figure 79: Main Strips Sub-Structure Generation

This intersection depth provides an additional fabrication advantage. By controlling the offset between the triangulated backbone and the original surface, the common bending lines between adjacent triangular panels are positioned away from the normal members. Since these common lines are folded according to computationally determined bending angles, they generate the required curvature of the initially flat-cut strip components. Separating the bending lines from the normal members prevents geometric interference during the folding process and improves fabrication accuracy. Along the free edges of the triangulated backbone, the distance between the bending line and the edge is controlled parametrically by the user, allowing the bending tolerance to be adjusted according to material properties and fabrication requirements.

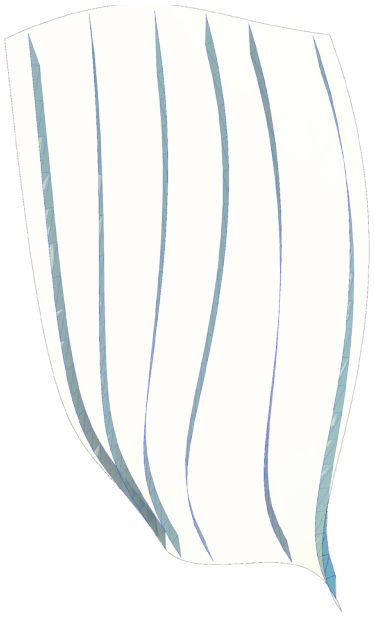


Figure 80: ain Strips Sub-Structure Split with Base Surface

Through this process, the secondary structure remains geometrically aligned with the local orientation of the base surface, while preserving a perpendicular relationship that depends on the used surface local normals and ensuring compatibility with the bending behaviour required to reproduce the intended curved geometry.

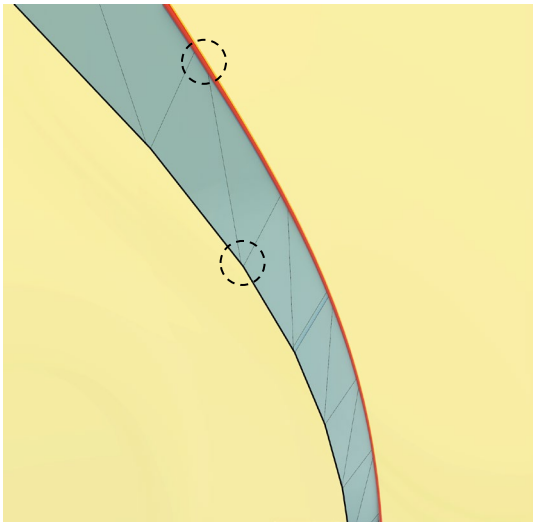


Figure 81: Bending Tolerances - Surface Side & Free Side

4.2.4. Geometry Detailing

This second section of the Main Grasshopper Logic enriches the discretised strip geometry with assembly and fabrication details. While the previous stage focuses on geometric rationalisation, this stage adds the physical information required for the installation.

12. Fabrication-Based Grouping

Group triangulated surfaces according to fabrication sheet constraints (Maximum sheet size: 3000 mm).

The workflow includes controlled grouping of the triangulated surfaces according to fabrication sheet constraints by a maximum part length parameter (currently se to maximum sheet size of 3 m). This allows long strip elements to be subdivided into smaller fabrication units.

This step is necessary because the full strip geometry may exceed available sheet dimensions. According to the given base surface.

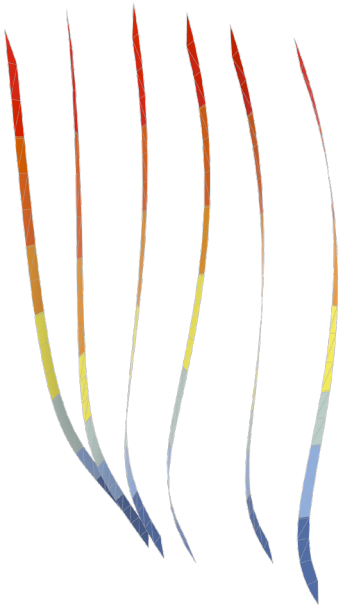


Figure 82: Fabrication-Based Grouping

13. Branches Sub-Structure Generation

The branch substructures are generated following the same methodology as the main strips. However, independent parameters are assigned to control both the backbone depth and the intersection depth with the base surface, allowing the branch elements to respond to their specific geometric and structural requirements.



Figure 83: Branches Sub-Structure Generation

14. Connection Plate Extension

After the strip components are divided, connection plates are generated. These extensions create additional geometric areas at the interfaces between parts.

The connection plates allow separated fabrication pieces to be mechanically joined after cutting and bending. They also provide space for bolts, slots, and alignment details.

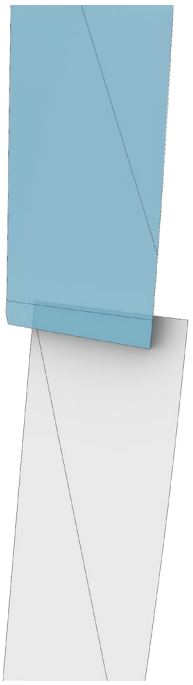


Figure 84: Connection Plate Extension

15. Connection Surface Selection

These operations identify the edges and surfaces that will receive connection details. Which is basically the extension plates and the first plate of the following part. This selection process ensures that bolts, holes, plates, and folding information are applied only where required.

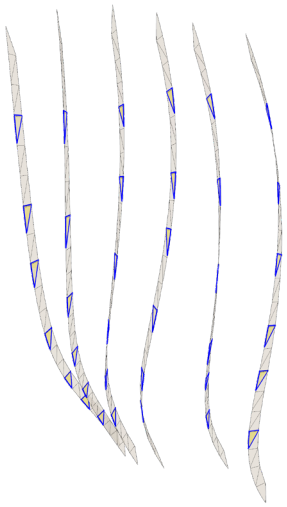


Figure 85: Connection Surface Selection

16. Slotted Holes and Bolt Locations

The fastening system is generated parametrically in 3D before unrolling. The bolt logic defines the position, size and shape of the holes. Slotted holes are used where tolerance or directional adjustment is required, where the extension plates meet the initial plate of the following part while circular bolt holes are used for the later part. This ensures adaptability and allow for some tolerances during the installation.

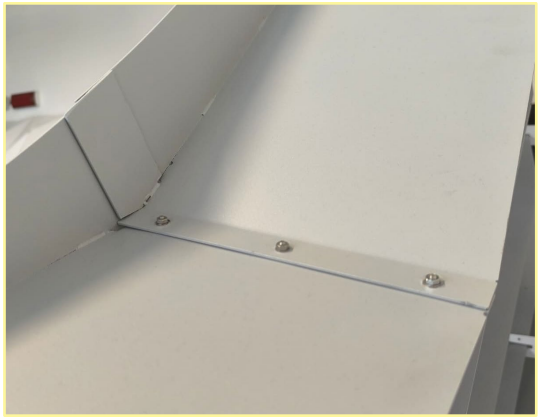


Figure 86: Parts Assembly - LOUVEN by INDEXLAB

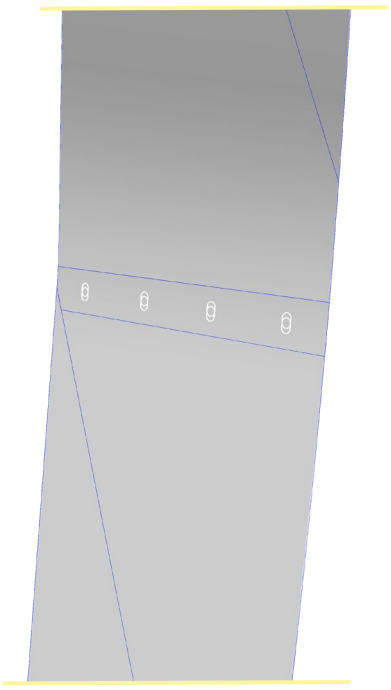


Figure 87: Slotted Holes and Bolt Locations

17. Main Stripes Parts

The main strip geometry is further separated according to connection logic and fabrication constraints, where the triangulated backbone splits.

This step converts the continuous strip system into discrete assembly parts. Each part retains its relationship to the original surface, while being controlled by the fabrication constrains and sheet size.

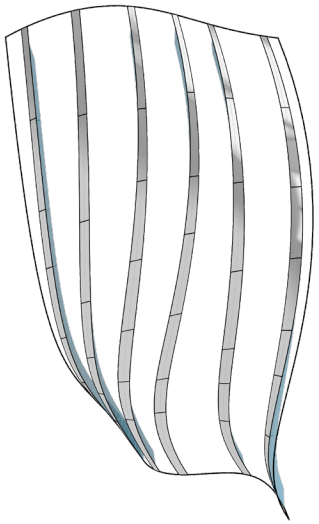


Figure 88: Main Stripes Parts

4.2.5. Unrolling and Nesting for Fabrication

In the third section the detailed 3D geometry is being converted into 2D fabrication drawings. This includes unrolling, transferring attributes, projecting holes, extracting bending marks, and nesting parts onto sheets.

18. Numbering and codding of components

This stage is critical for establishing a consistent correspondence between all fabrication components before and after the unrolling process. A significant computational challenge lies in maintaining accurate data-tree relationships throughout the workflow, ensuring that each generated element—including main strips, branch strips, extension plates, and their corresponding connection holes—retains its correct identity and association after successive geometric operations. In **Figure 89** First number represents the branch, while the second Number represents the place order in the branch.

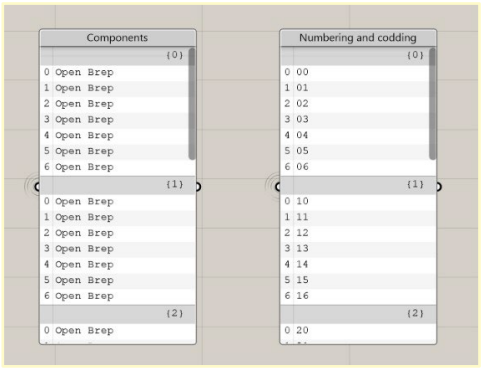


Figure 89: Numbering and codding of components

19. Bending Angles and Direction Extraction

In addition, the workflow computes the bending angles between adjacent triangulated surfaces and organizes these values within a matching data structure, where each fabrication part is linked to its corresponding shared edges. Since every shared edge of the triangulated backbone must be folded to a specific computed angle to reproduce the intended three-dimensional geometry, preserving these relationships is essential. This structured numbering and data management strategy ensures that all connection references, bending information, and fabrication attributes remain traceable and consistent between the original three-dimensional model and the flattened two-dimensional fabrication layouts, thereby supporting accurate manufacturing and assembly.

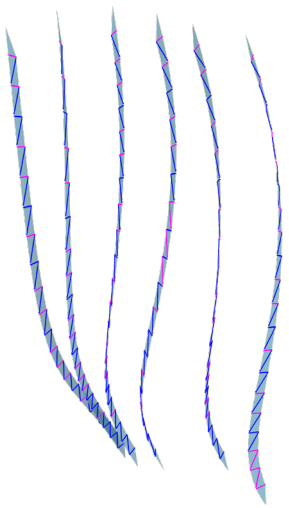


Figure 90: Bending Direction Coloring

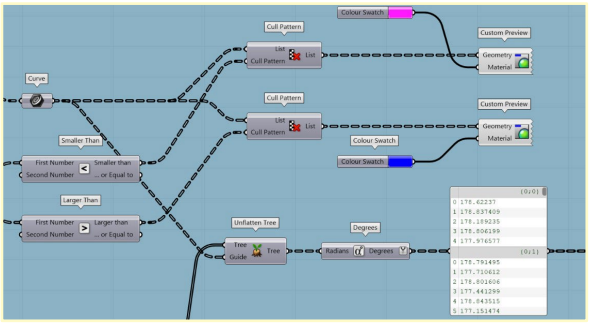


Figure 91: Bending Angles Extraction

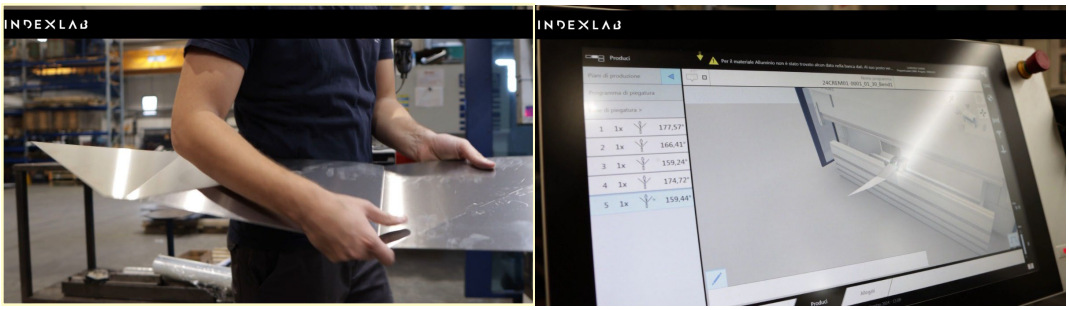


Figure 92: CNC Bending Process – LOUVEN by INDEXLAB

Figure 92 illustrating the **CNC bending** sequence according to the specified angles and bending directions.

20. Branch Stripes Discretization for Unrolling

Since Branch Strips is the pieces that connects between 2 main strips each is built with 2 curves on the connection sides with the main strips and a number of straight-line sections in the middle, this results in a double curved surface, that needs to be discretised in adjacent strips to be able to unfold.

For that the surface domain is divided into mesh strips with controlled U and V numbering depending on the precession desired and the size that the stripe allows, in this case it was seen that only 2 strips is enough to discretise each branch stripe.

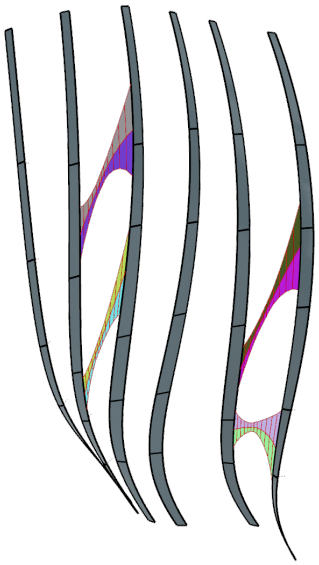


Figure 93: Branch Stripes Discretization for Unrolling

Then this strip meshes where deconstructed and loft has been made between its v direction lines, to build a smooth NURBS surface, that is unfordable in the coming stages.

21. Nesting Geometry with OpenNest2 with Attributes

The final stage of the workflow converts the detailed three-dimensional components into **fabrication-ready two-dimensional layouts** through a combination of **automated unrolling** and **sheet nesting**. During the unrolling process, the workflow transfers not only the component boundaries but also all associated fabrication attributes, including connection holes, bending lines, identification labels and its points, and assembly information. Preserving these attributes is essential to ensure that the flattened geometry retains all construction data required for accurate fabrication and assembly.

Following the unrolling process, the extracted fabrication information is organised into distinct curve categories, including boundary cut lines, hole cut lines, bending offset lines, mountain fold lines, and valley fold lines. This classification allows fabrication drawings to clearly distinguish between cutting, bending, marking, and drilling operations. Furthermore, mountain and valley fold lines are represented using different colours, enabling the fabrication drawings to communicate both the location and direction of each bend. Connection holes are transferred directly from the three-dimensional model to the corresponding unrolled components, ensuring their precise alignment with the flattened boundaries and bending lines while maintaining consistency with the assembly logic.

The fabrication layouts are subsequently processed using the **OpenNest2 plugin**, which automatically arranges the unrolled components onto user-defined sheet

dimensions. The workflow employs the OpenNest geometry attachment component to preserve all fabrication attributes during nesting, while the sheet component defines the dimensions of the fabrication material. The nesting algorithm optimises the placement of all components within the available sheet area to maximise material utilisation while preserving the geometric relationships and fabrication information associated with each part.

The output of this stage is a set of organised two-dimensional fabrication sheets containing all cutting boundaries, hole locations, bending data, identification data, and nesting information. These layouts can be exported directly for CNC cutting or other sheet-based digital fabrication processes without requiring additional manual processing.

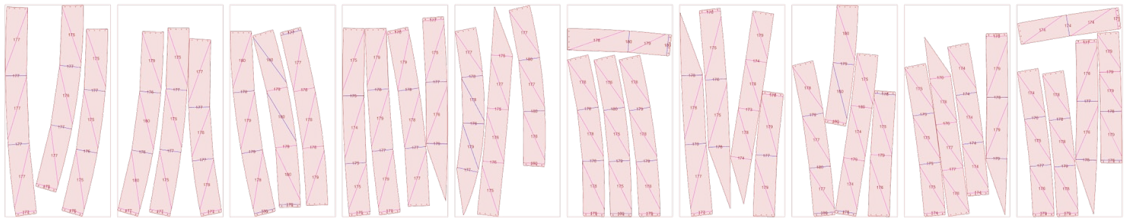


Figure 94: Main Strips Triangulated Geometry Nesting with Bending Angles

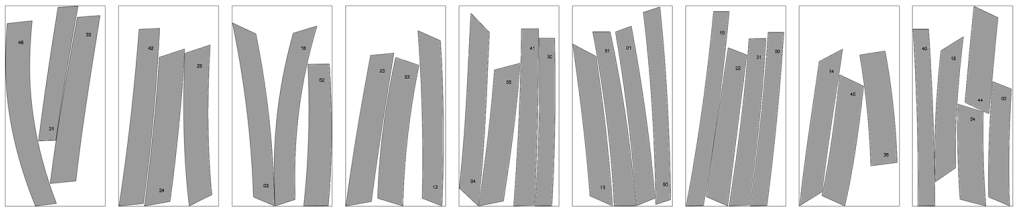


Figure 95: Main Strips Parts Nesting with Identification Coding



Figure 96: CNC Cutting - LOUVEN by INDEXLAB

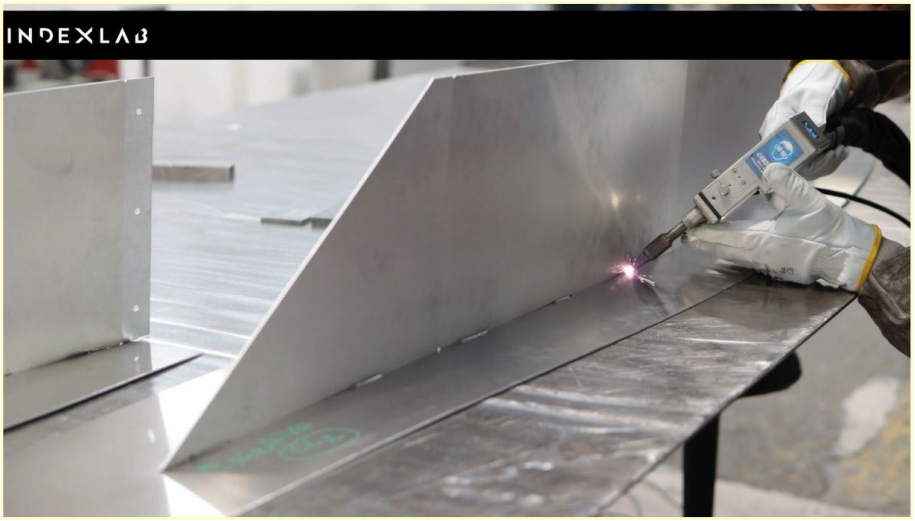


Figure 97: Laser Welding of components - LOUVEN by INDEXLAB

Figures 96 and 97 illustrate the CNC cutting and laser welding processes employed during the fabrication of the original LOUVEN project conducted by INDEXLAB. The successful fabrication and assembly of the full-scale prototype demonstrate the practical feasibility of the proposed façade system and validate its construction methodology at a real architectural scale.

Logic summary

Overall, the Grasshopper definition follows a clear computational sequence. First, a freeform reference surface is generated from input curves and analysed for curvature. Which can be replaced by the output surface of the previous stage 4.2.1. High Poly Mesh to single smooth surface. Second, the surface is discretised into main and branch strip surfaces through controlled curve manipulation, cross-section generation, and lofting. Third, these strips are segmented and equipped with a triangulated substructure based on surface points and normals. Fourth, the geometry is detailed with connection plates, bolt holes, slotted holes, bending tolerances, and assembly logic. Finally, all components are unrolled with their associated attributes and nested onto fabrication sheets using OpenNest2.

The workflow therefore transforms a continuous freeform surface into a fabrication-ready system of discrete, labelled, bendable, and connectable sheet components.

4.3. BIM Integration and Automated Documentation

To align the proposed workflow with Building Information Modelling (BIM) standards and the information requirements of the Architecture, Engineering, and Construction (AEC) industry, an additional interoperability workflow was developed to bridge the computational model generated in Grasshopper with a BIM environment. While the Grasshopper definition efficiently generates and optimizes the geometric and fabrication data, these outputs must ultimately be represented within a BIM platform to support project documentation, information management, coordination, and downstream construction workflows.

To achieve this, an automated data-transfer workflow was implemented using Rhino.Inside.Revit. Each fabrication component of the LOUVEN+ facade system, together with its unique identification code generated within the computational model, is automatically converted into an individual Revit family. During this process, the corresponding geometric representation and associated metadata are transferred directly from the Grasshopper model into the BIM environment. Consequently, every fabrication component is represented as a native Revit family containing the information required for scheduling, documentation, and model coordination. Figures 98.99 illustrates the resulting BIM representation, where each fabrication element is instantiated as an independent parametric family within the Revit model.

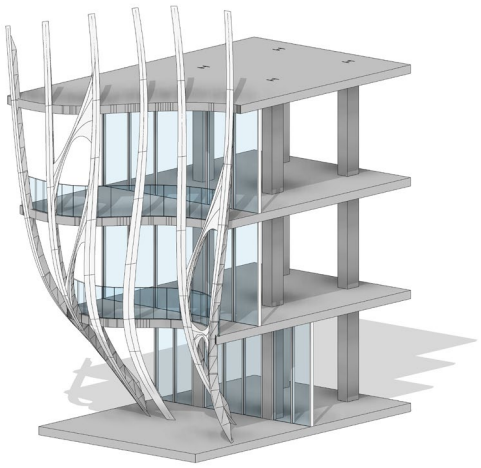


Figure 98: Geometry Placement in Revit Model

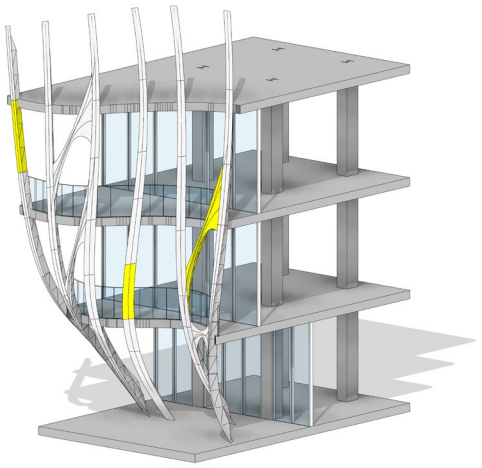


Figure 99: Coded native Families in Revit Model

Beyond geometric interoperability, the workflow also addresses the **documentation of fabrication information**. Since the fabrication process relies on the unrolling and nesting of sheet-metal components, these layouts must be documented for manufacturing. However, Autodesk Revit provides limited capabilities for generating parametric unrolled views of custom folded sheet components. To overcome this limitation, an automated documentation workflow was developed in which drafting views are generated directly from the computational model.

For each nested fabrication element, a corresponding Revit drafting view is automatically created. The nested component boundaries, bending lines, hole locations, connection details, and identification labels are transferred into these views while maintaining their association with the original computational model. This establishes a **continuous digital relationship** between the parametric Grasshopper definition, the generated Revit families, and the fabrication documentation.

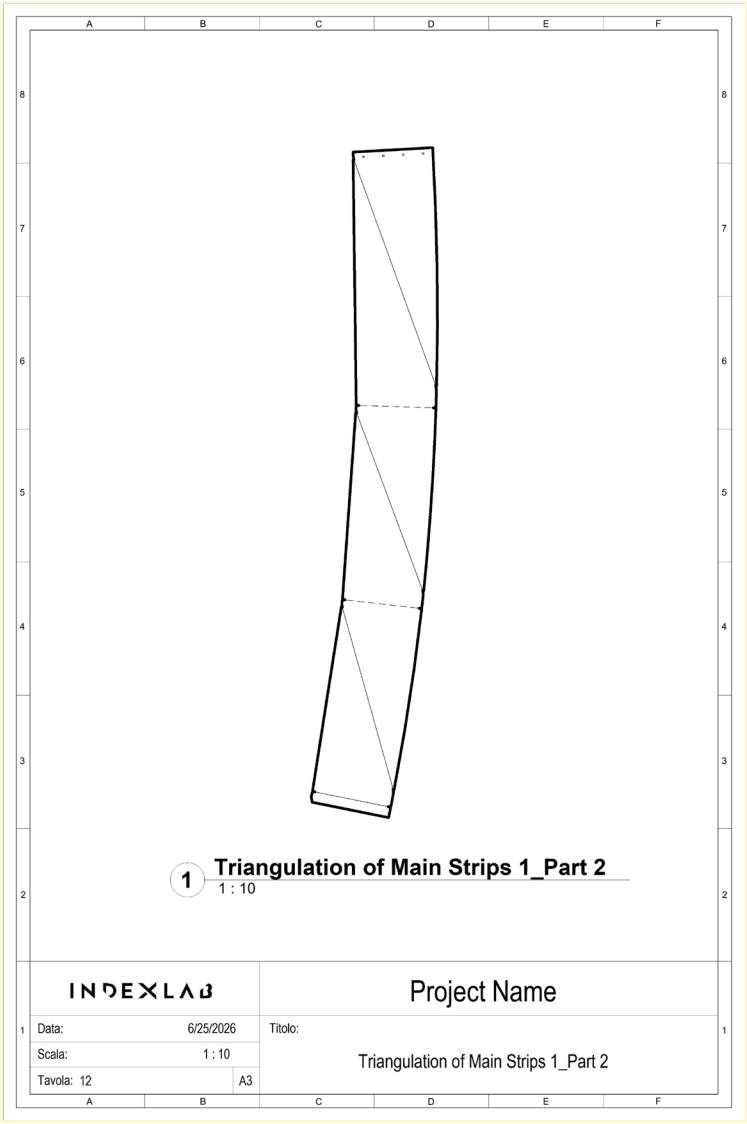


Figure 100: Automated Views Creation & Sheet Documentation

A significant advantage of this workflow is its associative behaviour. Any modification to the original design surface, discretisation parameters, structural detailing, or fabrication logic automatically propagates through the computational model and updates the corresponding Revit families, drafting views, and fabrication layouts. This maintains consistency between the design model, BIM documentation, and fabrication information throughout the iterative design process, thereby reducing manual rework, minimizing coordination errors, and supporting a fully integrated digital workflow from computational design to fabrication documentation.

4.4. Physical Prototype Validation

To validate the proposed computational workflow, a 1:20 scale physical prototype was fabricated, verifying the complete design-to-fabrication pipeline developed in this research. The prototype evaluates the workflow from Surface Band Strips Discretisation through Geometry Detailing to Unrolling and Nesting for Fabrication, demonstrating that the computationally generated geometry can be accurately translated into a physical assembly.

The fabrication components generated by the Grasshopper workflow were exported directly as two-dimensional cutting layouts and transferred to Silhouette Studio, the proprietary software used to operate the Silhouette Cameo 4 desktop CNC cutting machine. The Cameo 4 is a precision blade-based CNC plotter capable of accurately cutting lightweight sheet materials such as cardstock, vinyl, paper, and thin cardboard. Since the computational workflow produces fabrication-ready nested sheets, no manual redrawing or geometric modification was required prior to fabrication, providing a direct verification of the digital workflow.

The facade components were fabricated from silver cardstock to represent the LOUVEN+ strip system, while the supporting building structure was constructed from heavy cardboard and finished with a black coating to maximise the visual contrast between the structural substrate and the fabricated façade components. The resulting prototype accurately reproduces the digital geometry, demonstrating the geometric consistency of the discretisation, detailing, and fabrication workflow. Figure 101 presents the digital model while Figures 102,103 presents the completed physical prototype.

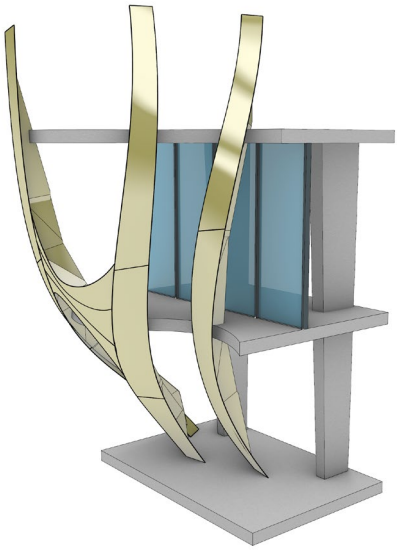


Figure 101: Prototype Digital Model

To facilitate accurate assembly, reference marks generated by the CNC cutting process were marked into the floor slabs. These marks define the exact locations where the triangulated substructure connects to the building model. In addition, precisely cut openings were incorporated into the slabs, allowing the strip components to be inserted into their intended positions while maintaining the correct spatial orientation throughout the assembly process.



Figure 102: Prototype Physical Model - Shot 01

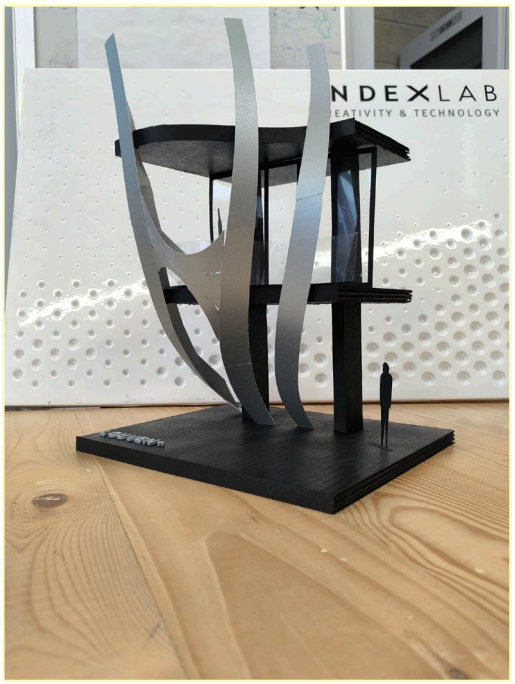


Figure 103: Prototype Physical Model - Shot 02

The assembly sequence also verified the construction logic embedded within the computational workflow. The primary strip components were installed first, establishing the overall facade geometry and structural alignment. The branch strip components were installed last, as their geometry depends on the precise positioning of the adjacent main strips. This assembly sequence ensured continuity of the facade curvature and maintained the geometric coherence between neighbouring strip surfaces. In a full-scale implementation, these branch strips would be positioned simultaneously with the supporting slab, undergo controlled bending to achieve the required curvature, and subsequently be welded to the rear surfaces of the adjacent main strips.

The fabrication and assembly process also served as an important verification stage for the computational model. During the initial prototype assembly, inconsistencies were identified in the numbering and coding of some fabrication components. These errors were traced to mismatches in the Grasshopper data-tree structure responsible for assigning identification codes to individual parts. Following the identification of this issue, the data management logic within the Grasshopper definition was refined, resulting in consistent numbering and correct correspondence between the digital model, fabrication drawings, and physical components. This iterative refinement highlights one of the principal advantages of computational workflows: errors discovered during physical prototyping can be traced directly to their algorithmic source, corrected once within the parametric definition, and automatically implemented throughout the entire fabrication dataset.

Overall, the successful fabrication and assembly of the physical prototype demonstrate the validity of the proposed workflow. The prototype confirms not only the geometric accuracy of the discretisation and near-developable strip generation, but also the effectiveness of the automated detailing, fabrication preparation, and assembly logic. As such, the physical model provides practical validation of the proposed digital workflow and demonstrates its applicability as a foundation for full-scale fabrication and construction.

Unlike purely digital validation, the fabrication of the physical prototype exposed practical issues related to assembly sequencing, part identification, and data management that would not have been evident through computational simulation alone. The resulting refinements improved the robustness and reliability of the proposed workflow, demonstrating the importance of iterative feedback between digital modelling and physical prototyping.

5.
Conclusion & Future Developments

5. Conclusion & Future Developments

5.1. Discussion

The significance of this research extends beyond the development of the proposed LOUVEN+ facade system. Its primary contribution lies in establishing an integrated computational workflow that connects conceptual design, geometric modelling, fabrication planning, and Building Information Modelling (BIM) within a continuous digital process. Rather than treating architectural design, engineering, fabrication, and documentation as isolated stages, the proposed methodology maintains a common parametric model throughout the workflow, reducing information loss between project phases while improving design adaptability and fabrication consistency.

A key contribution of the research is the integration of artificial intelligence into the early design process. AI-generated façade concepts provide a rapid means of exploring architectural alternatives, which are subsequently translated into editable parametric models within Grasshopper. Instead of replacing the designer, AI functions as a design assistant that accelerates conceptual exploration while preserving designer control during computational modelling and optimisation.

The computational workflow further demonstrates how freeform architectural surfaces can be systematically transformed into fabrication-ready sheet-metal assemblies. Through surface discretisation, near-developable strip generation, structural backbone creation, connection detailing, unrolling, and material nesting, fabrication requirements are embedded directly into the design process. Consequently, changes to the input geometry automatically propagate throughout the structural, fabrication, and documentation stages, significantly reducing repetitive modelling tasks and improving design consistency.

An important finding of this research is that near-developable strip surfaces provide a more practical solution than enforcing perfectly developable geometry. Although fully developable surfaces are theoretically advantageous for fabrication, they were found to compromise the intended architectural form and expose structural elements in undesirable ways. The adopted near-developable strategy successfully balances geometric fidelity with fabrication feasibility, while Gaussian curvature analysis and graphical developability verification confirm that the generated strip surfaces remain suitable for sheet-metal bending processes.

The integration of Rhino.Inside.Revit extends the workflow beyond computational modelling by automatically transferring fabrication components into native BIM families together with their associated metadata. Fabrication drawings, drafting views, and nested sheet layouts are likewise generated directly from the computational model, establishing a continuous relationship between design geometry, fabrication information, and project documentation. This interoperability demonstrates the potential of computational design to support both digital fabrication and BIM-based project delivery within a unified workflow.

Despite these contributions, several limitations remain. The AI-assisted design stage is currently limited to conceptual image generation and does not incorporate engineering constraints, fabrication knowledge, or environmental performance during the generation process. Likewise, although the workflow automates the generation of fabrication-ready geometry and documentation, the manufacturing process itself still relies on conventional CNC machining, bending, welding, and manual assembly. Finally, the current methodology focuses primarily on geometric rationalisation and fabrication feasibility, without directly integrating environmental performance criteria such as daylight optimisation, solar control, or energy efficiency.

Overall, this research should therefore be regarded as a computational framework rather than a fully autonomous design-to-production system. Its principal contribution is the demonstration that artificial intelligence, computational geometry, fabrication-aware modelling, digital fabrication, and BIM interoperability can be successfully integrated into a single parametric workflow, providing a robust foundation for future developments in digitally integrated architectural design.

5.2. Future Research and Development

The proposed workflow establishes a strong foundation for an integrated digital design-to-production process; however, several opportunities remain for future development.

The first area concerns the integration of artificial intelligence into the early design process. The current workflow employs AI primarily as a conceptual image-generation tool, requiring manual interpretation before computational modelling. Future developments could introduce interactive AI-assisted design generation that incorporates architectural constraints directly during the generation process. Rather than relying solely on text prompts, designers could provide sketches, dimensional constraints, façade proportions, opening ratios, and contextual information to guide the AI towards solutions that are both aesthetically coherent and geometrically feasible.

Such an approach would significantly reduce the number of design iterations required to reach a satisfactory façade concept.

A second development concerns software interoperability. The present research demonstrates the successful integration of ComfyUI, Rhino, Grasshopper, and Revit through individual workflows. However, future implementations could consolidate these applications into a unified computational platform operating through their respective APIs. Such an interface would enable designers to interact with a single environment in which AI-assisted façade generation, computational modelling, structural discretisation, fabrication detailing, BIM integration, and documentation are executed automatically in the background. This would establish a seamless digital workflow from conceptual design to production-ready documentation while reducing user intervention and improving workflow efficiency.

Another important direction involves the automation of the physical fabrication process. Although the proposed workflow generates fabrication-ready components, the manufacturing and assembly stages remain largely dependent on conventional CNC equipment and manual operations. Future work could investigate the integration of robotic fabrication systems capable of performing CNC cutting, automated bending, robotic welding, and automated assembly. Particular attention should be given to developing robotic positioning and fixturing strategies that accurately support and align components during welding without requiring manual intervention. Such developments would extend the digital workflow beyond design and documentation towards a fully automated manufacturing pipeline.

Finally, future research should strengthen the environmental performance of the proposed façade system. In the current workflow, the orientation and distribution of the louver elements are primarily determined by geometric and architectural considerations. Integrating environmental simulation tools within the parametric workflow would enable the louver configuration to respond directly to climatic conditions such as solar radiation, daylight availability, glare control, thermal performance, and energy demand. Multi-objective optimisation techniques could then be employed to balance architectural expression, structural efficiency, fabrication constraints, and environmental performance, resulting in façade systems that are simultaneously aesthetically expressive, constructible, and environmentally responsive.

Collectively, these developments would transform the proposed workflow into a comprehensive digital design-to-production framework that integrates artificial

intelligence, computational geometry, Building Information Modelling (BIM), digital fabrication, robotic manufacturing, and environmental optimisation within a unified computational ecosystem.

5.3. Conclusion

This research demonstrates that computational design can extend beyond geometric modelling to become an integrated methodology connecting artificial intelligence, parametric modelling, fabrication-aware design, digital fabrication, and Building Information Modelling within a continuous digital workflow. The proposed framework successfully transforms AI-assisted conceptual façade designs into fabrication-ready components while maintaining geometric consistency, fabrication feasibility, and BIM interoperability throughout the process.

Rather than presenting a single computational technique, the research contributes a comprehensive methodology that establishes a continuous digital thread from conceptual design to fabrication documentation. Although further developments are required to incorporate intelligent AI constraints, environmental optimisation, and robotic manufacturing, the proposed framework provides a robust foundation for future research into fully integrated, AI-assisted, and fabrication-aware architectural design workflows.

6. Bibliography

6. Bibliography

[1] F. Bie *et al.*, "RenAIssance: A Survey into AI Text-to-Image Generation in the Era of Large Model," Sep. 2023, [Online]. Available: http://arxiv.org/abs/2309.00810

[2] I. J. Goodfellow *et al.*, "Generative Adversarial Networks," Jun. 2014, [Online]. Available: http://arxiv.org/abs/1406.2661

[3] A. Vaswani *et al.*, "Attention Is All You Need," 2017, [Online]. Available: http://arxiv.org/abs/1706.03762

[4] A. Radford *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," 2021, [Online]. Available: http://arxiv.org/abs/2103.00020

[5] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," 2020, [Online]. Available: http://arxiv.org/abs/2006.11239

[6] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical Text-Conditional Image Generation with CLIP Latents," Apr. 2022, [Online]. Available: http://arxiv.org/abs/2204.06125

[7] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," 2022, [Online]. Available: http://arxiv.org/abs/2112.10752

[8] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," 2022, [Online]. Available: http://arxiv.org/abs/1312.6114

[9] S. Nuzhnyy, "The Best AI Image Generators in 2026: 12 Models Tested." Accessed: Jun. 03, 2026. [Online]. Available: https://aimlapi.com/blog/the-best-ai-image-generators

[10] PXZ.ai., "Best AI Image Generators 2025: 12 Tools Tested & Ranked." Accessed: Jun. 03, 2026. [Online]. Available: https://pxz.ai/blog/best-ai-image-generators-2025-tested-ranked

[11] PromptItIn, "Best AI Image Generators in 2026: Midjourney vs DALL-E 3 vs Stable Diffusion Compared." Accessed: Jun. 03, 2026. [Online]. Available: https://promptitin.com/blog/best-ai-image-generators-2026

[12] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," 2020, [Online]. Available: http://arxiv.org/abs/2003.08934

[13] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images," 2018, [Online]. Available: http://arxiv.org/abs/1804.01654

[14] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy Networks: Learning 3D Reconstruction in Function Space," 2019, [Online]. Available: http://arxiv.org/abs/1812.03828

[15] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation," Jan. 2019, [Online]. Available: http://arxiv.org/abs/1901.05103

[16] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Trans. Graph.*, vol. 42, no. 4, Aug. 2023, doi: 10.1145/3592433.

[17] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "DreamFusion: Text-to-3D using 2D Diffusion," Sep. 2022, [Online]. Available: http://arxiv.org/abs/2209.14988

[18] Y. Hong *et al.*, "LRM: Large Reconstruction Model for Single Image to 3D," 2024, [Online]. Available: http://arxiv.org/abs/2311.04400

[19] D. Tochilkin *et al.*, "TripoSR: Fast 3D Object Reconstruction from a Single Image," Mar. 2024, [Online]. Available: http://arxiv.org/abs/2403.02151

[20] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot One Image to 3D Object," Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.11328

[21] M. Liu *et al.*, "One-2-3-45: Any Single Image to 3D Mesh in 45 Seconds without Per-Shape Optimization," Jun. 2023, [Online]. Available: http://arxiv.org/abs/2306.16928

[22] Modelfy, "2025 Image to 3D Tools Comparison." Accessed: Jun. 12, 2026. [Online]. Available: https://modelfy.art/blog/2025-image-to-3d-tools-comparison

[23] Tripo AI, "The Best AI Sculpting Tools of 2025." Accessed: Jun. 12, 2026. [Online]. Available: https://www.tripo3d.ai/content/en/use-case/the-best-ai-sculpting-tool

[24] 3D AI Studio, "9 Best Image-to-3D Tools for Beginners." Accessed: Jun. 12, 2026. [Online]. Available: https://www.3daistudio.com/blog/best-image-to-3d-tools-beginners-2026

[25] Tridimensi, "The Best AI 3D Model Generator Platforms for Creators in 2025." Accessed: Jun. 12, 2026. [Online]. Available: https://tridimensi.pro/blog/ai-3d-model-generator/

[26] SIGGRAPH, "3D Generative AI Transforms How We Create, Design, Interact with Digital Content." Accessed: Jun. 12, 2026. [Online]. Available: https://s2025.siggraph.org/3d-generative-ai-transforms-how-we-create-design-interact-with-digital-content/

[27] WaveSpeed AI, "Introducing Tripo3D V2.5 Image To 3D." Accessed: Jun. 12, 2026. [Online]. Available: https://wavespeed.ai/blog/posts/introducing-tripo3d-v2-5-image-to-3d-on-wavespeedai/

[28] ComfyUI Team, "ComfyUI." Accessed: Jun. 14, 2026. [Online]. Available: https://github.com/Comfy-Org/ComfyUI

[29] Black Forest Labs, "FLUX.1 Technical Report." Accessed: Jun. 14, 2026. [Online]. Available: https://bfl.ai/

[30] Emergent Mind, "ComfyUI — Modular AI Workflow Engine." Accessed: Jun. 14, 2026. [Online]. Available: https://www.emergentmind.com/topics/comfyui

[31] ComfyUI.org., "Flux.1-dev Workflows — FLUX T5-XXL + CLIP-L + VAE pipeline." Accessed: Jun. 14, 2026. [Online]. Available: https://comfyui.org

[32] MrForExample, "ComfyUI-3D-Pack: Extensive Node Suite for 3D Generation in ComfyUI." Accessed: Jul. 02, 2026. [Online]. Available: https://github.com/MrForExample/ComfyUI-3D-Pack

[33] APATERO, "TRELLIS.2 in ComfyUI: Image to 3D in Seconds." Accessed: Jun. 14, 2026. [Online]. Available: https://www.apatero.com/blog/trellis-2-comfyui-image-to-3d-complete-guide-2025

[34] K. Crane, "DISCRETE DIFFERENTIAL GEOMETRY: AN APPLIED INTRODUCTION," 2025. Accessed: Apr. 08, 2026. [Online]. Available: https://www.cs.cmu.edu/~kmcrane/Projects/DDG/paper.pdf

[35] P. Ruttico, *X-boundaries*. Florence, Italy: EDIFIR Edizioni Firenze srl, 2025.

[36] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno L´evy, *Polygon Mesh Processing*, 2nd ed. Natick, Massachusetts: A K Peters, Ltd, 2010.

[37] J. Warren and H. Weimer, *Subdivision Methods for Geometric Design: A Constructive Approach*. San Francisco, CA, USA: Morgan Kaufmann, 2001.

[38] L. Piegl and W. Tiller, *The NURBS Book*, 2nd ed. Berlin, Germany: Springer, 1997.

[39] Branko Kolarevic, *Architecture in the Digital Age_Design and Manufacturing*. New York: Spon Press, 2003.

[40] H. Pottmann, M. Eigensatz, A. Vaxman, and J. Wallner, "Architectural Geometry," 2014. [Online]. Available: www.smartgeometry.com

[41] “OpenNest 2.0.” Accessed: Jun. 28, 2026. [Online]. Available: https://www.food4rhino.com/en/app/opennest

[42] T. El-Khouly and O. Abdelhalim, “Preserving conceptual design integrity: strategies for enhancing interoperability in architectural digital design workflows,” *Sci. Rep.*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-78640-8.

[43] “Rhino.Inside Revit.” Accessed: Jun. 28, 2026. [Online]. Available: https://www.rhino3d.com/inside/revit/1.0/guides/revit-revit

[44] “INDEXLAB Politecnico di Milano - LOUVEN Project.” Accessed: May 03, 2026. [Online]. Available: https://www.indexlab.it/louven

[45] T. Maekawa and F. Scholz, “All you need is rotation: Construction of developable strips,” *ACM Trans. Graph.*, vol. 43, no. 6, Dec. 2024, doi: 10.1145/3687947.

[46] M. Capone, “Theories and methods for development of developable ruled surfaces and approximate flattening of non-developable surfaces,” *Disegno*, vol. 2018, no. 3, pp. 53–68, 2018, doi: 10.26375/disegno.3.2018.7.

List of Figures

Figure 1: Text Prompt to Image Generation 12
Figure 2: Midjourney Logo 20
Figure 3: Chat GPT Logo 21
Figure 4: Adobe Firefly Logo 21
Figure 5: Ideogram Logo 22
Figure 6: Google Gemini Logo 22
Figure 7: Image to 3D Model Generation 23
Figure 8: Tripo 3D Logo 28
Figure 9: Meshy AI Logo 28
Figure 10: Luma AI Logo 29
Figure 11: Hyper 3D Rodin AI Logo 29
Figure 12: Kaedim Logo 29
Figure 13: Polycam Logo 30
Figure 14: Tencent Hunyuan 3D Logo 30
Figure 15: ComfyUI Logo 32
Figure 16: Image To Image Generation 35
Figure 17: Double curved surface 39
Figure 18: Surface Lofting 40
Figure 19: Rotating a Frame Around an Axis 41
Figure 20: Sweeping a Line Along a Curve 42
Figure 21: Warped (or Twisted) Surface 42
Figure 22: Hyperbolic Paraboloid Surface 43
Figure 23: Cylindrical Developable Surfce 44
Figure 24: Conical Developable Surface 45
Figure 25: Tangential Developable Surface 46
Figure 26: Curvature Aligned Stripes 47
Figure 27: Band Strips 48
Figure 28: Cross Strips 48
Figure 29: MeshGraph-based Strips Network 49
Figure 30: Polygonal Mesh 50
Figure 31: Subdivision (SubD) Surface 52
Figure 32: NURBS Surface 54
Figure 33: Grasshopper Logo 56
Figure 34: Aluminium Physical model - LOUVEN by INDEXLAB 63
Figure 35: Methodology Roadmap 65
Figure 36: Text to Image ComfyUI Workflow 66
Figure 37: Text to Image Selected Outputs 67
Figure 38: Flux 2 Depth ControNet workflow + Upscaling and render addition in ComfyUI 69
Figure 39: Flux 2 Depth ControNet outputs 69
Figure 40: Flux 2 Canny ControlNet Workflow 69
Figure 41: Flux 2 Canny ControlNet Outputs 70
Figure 42: 20 reference images for LORA image training 71
Figure 43: LORA image training Workflow in ComfyUI 72
Figure 44: Realistic Vision V60 B1 VAE Workflow with Trained LoRA 72
Figure 45: Trained LoRA Most Successful Outputs 73
Figure 46: Flux-2-klein + Trained LoRA Surface outputs 74
Figure 47: Flux-2-klein + Trained LoRA contextual Prompt outputs 74
Figure 48: Hunyuan 3D 2.0 Image to 3D mesh model 75

Figure 49: Produced 3D Meshes by Hunyuan 3D 2.0 76
Figure 50: Mesh Input and Reduction 77
Figure 51: Mesh Reconstruction and Subdivision 78
Figure 52: Quad remeshing 78
Figure 53: Boundary edge detection 79
Figure 54: Generation of Structured Vertical Quad Mesh 79
Figure 55: Kangaroo Physics Grasshopper Logic 80
Figure 56: Kangaroo Physics Relaxation 81
Figure 57: NURBS Surface Reconstruction 81
Figure 58: Surface Stripe-Frenet Frame 84
Figure 59: Surface Stripe-Darboux Frame 86
Figure 60: Surface Stripe- Perpendicular Frames 87
Figure 61: Surface Stripe- UV Frame 87
Figure 62: Designed Pattern curves on 88
Figure 63: Geodesic Curves on the Proposed 89
Figure 64: General view of the Main Grasshopper Logic Parts. 91
Figure 65: Input Geometry 92
Figure 66: Surface Creation 92
Figure 67: Surface Curvature Analysis 93
Figure 68: Vector Transport Surface Curvature Analysis 94
Figure 69: Surface Division 95
Figure 70: Curve Manipulation 95
Figure 71: Branch Strip Guide Curves Generation 96
Figure 72: Main Strip Surface Generation 96
Figure 73: Strips construction comparison 97
Figure 74: Branch Strip Surface Generation 99
Figure 75: Main Stripes Sub-Structure Initial Division 99
Figure 76: Main Stripes Sub-Structure Final Redivision 100
Figure 77: Main Stripes Sub-Structure Branch intersections insertion 100
Figure 78: Normal Construction 100
Figure 79: Main Strips Sub-Structure Generation 101
Figure 80: ain Strips Sub-Structure Split with Base Surface 101
Figure 81: Bending Tolerances - Surface Side & Free Side 102
Figure 82: Fabrication-Based Grouping 102
Figure 83: Branches Sub-Structure Generation 103
Figure 84: Connection Plate Extension 103
Figure 85: Connection Surface Selection 104
Figure 86: Parts Assembly - LOUVEN by INDEXLAB 104
Figure 87: Slotted Holes and Bolt Locations 104
Figure 88: Main Stripes Parts 105
Figure 89: Numbering and codding of components 105
Figure 90: Bending Direction Coloring 106
Figure 91: Bending Angles Extraction 106
Figure 92: CNC Bending Process – LOUVEN by INDEXLAB 106
Figure 93: Branch Stripes Discretization for Unrolling 107
Figure 94: Main Strips Triangulated Geometry Nesting with Bending Angles 108
Figure 95: Main Strips Parts Nesting with Identification Coding 108
Figure 96: CNC Cutting - LOUVEN by INDEXLAB 109
Figure 97: Laser Welding of components - LOUVEN by INDEXLAB 109
Figure 98: Geometry Placement in Revit Model 111
Figure 99: Coded native Families in Revit Model 111
Figure 100: Automated Views Creation & Sheet Documentation 112
Figure 101: Prototype Digital Model 114

Figure 102: Prototype Physical Model - Shot 01 ..... *114*
Figure 103: Prototype Physical Model - Shot 02 ..... *114*

List of Graphs

Graph 1: Combined comparison of Diagram Stripes using deferent frames *88*
Graph 2: Combined comparison of Formed Stripes based on Geodesic Curves *89*
Graph 3: Combined comparison of Formed Stripes based on Designed Pattern Curves *90*
Graph 4: Combined comparison *90*

List of Tables

Table 1: Brief Text-to-Image Platforms Comparison *22*
Table 2: Brief image-to-3D Platforms Comparison *31*

Acknowledgment

To Allah, the Most Merciful and the Most Just, my Lord, my Creator, and my first companion throughout this journey. By His grace, I have reached the end of this work, and I sincerely hope that I have given it my very best.

My sincere gratitude goes to INDEXLAB and to Professor Pierpaolo Ruttico for the opportunities he opened for me, and to Eng. Imane El Bakkali for her continuous support and guidance.

To my mom and dad, thank you for every sacrifice you made and for enduring the distance, believing that it would lead your son toward a better future. I owe you more than words can express.

I am deeply thankful to all the professors, teachers, and professionals from whom I have gained knowledge and who have contributed to shape the person I am today.

To my family and friends, thank you for your unwavering support, your prayers, and your kindness. You have never hesitated to stand by me whenever I needed you.

Finally, I would like to thank Politecnico di Milano and Italy for this remarkable experience, which has broadened my horizons, enriched my perspective, and opened doors to opportunities I will always be grateful for.

Ali Elsayed

إلى الله، ربي ومؤنسي الأول في غربتي، الذي ما خاب من توكل عليه، او استعان به.

إلى أمي وأبي، اللذين لم يدّخرا جهدًا ولا تضحيةً في سبيل أن أصل إلى ما أنا عليه اليوم، ساعيًا إلى تطوير ذاتي، وباحثًا عن آفاقٍ أرحب وفرصٍ أوسع.

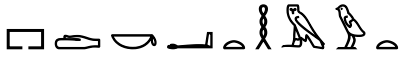
إلى أساتذتي ومعلمي من جامعتي الأزهر و بوليتكنيكوا دي ميلانو، الذين لم يبخلوا عليّ بعلمٍ أو توجيه، وكان لهم بالغ الأثر في رحلتي.

إلى أهلي وأصدقائي، الذين أحاطوني بدعمهم، ولم تنقطع دعواتهم الصادقة لي بالخير والتوفيق.

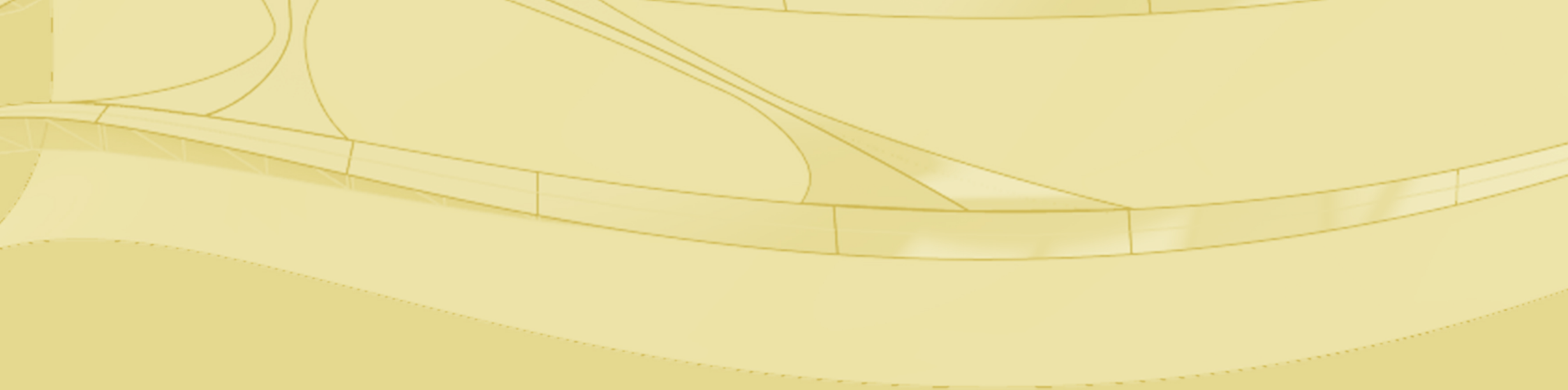
إلى مصر، وطني الأول، ومصدر فخري واعتزازي، التي أحملها في قلبي أينما كنت.

كنتم جميعًا السند والعون، والدافع الذي منحني القوة لأواصل هذا الطريق، الذي ما كان ليسهل لولا فضل الله، ثم بكم.

على السيد



House-building and skilled craft
In Hieroglyphics



LOUVEN+