

Free-form Modeling with Physical Constraint

Optimization: Exploration of a Design Space

Qiqin Le

May 24, 2021

Abstract

For decades, researchers are devoted to creating computer-aided design(CAD) tools that help users gain a more precise and intuitive control over the free-form virtual models. Many efforts have been done in the industry and academics to integrate physical simulation methods with various existing CAD systems to open new possibilities of creation to the users. Amongst these methods, constraint-based simulation is gaining importance in recent years because it is robust and fast and easy to implement. This thesis introduces a novel design pipeline that allows the users to interfere the process of constraint optimization which is used to be a black box to the users, thus widening the scope of explorable design space of the traditional pipeline.

Contents

1	Introduction	4
2	Background	10
2.1	The General Study of Sketching and Conceptual Design	10
2.2	The Comparison between Digital Tools and Conventional tools for Conceptual Design	12
2.3	Relative Techniques	14
2.3.1	Digital Sculpture	14
2.3.2	Interactive Physical-based Modeling	16
3	Statement of Issues and Research Goal	27
3.1	Design Pipeline and Design Space of Constraint-based For- ward Design	30

3.2	Design Pipeline and Design Space of Constraint-based Inverse	
	Design	33
3.3	Discussion of Current Pipeline and the Targeted Design Pipeline	35
4	Statement of Method	39
4.1	The Algorithms to Enforce the Constraints	40
4.1.1	1D - Fast projection	40
4.1.2	2D - Position-based dynamics	41
4.2	Software Platform	42
4.2.1	1D Platform	43
4.2.2	2D Platform	44
4.2.3	Experiment Platform	46
5	Analysis and Experiments	48
5.1	1D Analysis	49
5.1.1	Interaction with Freely Moving vertices	51
5.1.2	Interaction with smoothing algorithm	52
5.2	2D Analysis	53
5.2.1	Manually Assigned Rest values without user interference	53
5.2.2	Interaction with Freely Moving vertices	55

5.2.3	Cross interaction between different constraints	58
5.3	User Experiment	60
5.3.1	Tester Information and tutorial	60
5.3.2	Experiment and observation	62
6	Discussion	68
6.1	Contribution	68
6.2	Limitation	70
6.2.1	Technical limitation	70
6.2.2	Design limitation	71
6.3	Future Direction	72

Chapter 1

Introduction

Since Sutherland proposed the first sketch-based computer interface in the 1960s' (Sutherland, 1963), computer-aided-design (CAD) tools have gradually replaced paper and pens as the mainstream tools in most design fields. CAD plays a critical role in modern industry, ranging from designing microchip processors to creating virtual objects in game industry. The development of CAD system provides unprecedented precision to design objects and efficiency to design processes.

There is nonetheless one domain where the CAD system needs to be improved - conceptual ideation and creative exploration in the early stage of

creative design. The brainstorming phase of a design process features fast iteration, vague explanation, and frequent reinterpretation, while various researches, as I discuss in greater detail in the background chapter, have shown CAD's inefficiency in stimulating creativity and producing and maintaining interaction between the system and the designers.

For decades, researchers have been trying to increase CAD's ability to support aesthetic work. Séquin proposed "a dream system," which states the ideal form of a CAD system (Séquin, 2005).

"Such a system should combine the best of both the virtual CAD environment and of the real physical world. As virtual elements, any construction parts possess infinite strength; they can be glued together easily, and just as easily be disassembled again...However, some hands-on interaction seems rather desirable. Sweeping a hand through space to define a curve or a profile is often the most natural way to express one's intent. On other occasions, one would like to use a piece of physical material, such as an elastic steel blade, or a piece of heavy velvet cloth to define a shape that is then governed by the intrinsic properties of the chosen material...Beams might bend like steel

wires (MEC); surfaces may stretch like soap films (MES), or they could be subjected to some optimization process under the influence of one of several artificial 'beauty' functionals."

The industry and academics have applied techniques from different fields – sketch-based-interface, digital sculpture, augmented/virtual reality, and tangible interaction to improve CAD platform’s ability of supporting ideation. Amongst all techniques, physical based approaches that seize the material properties of the design object always play an important role in exploring the suitable interaction between the machine and the user. Researchers tried to preserve the deformation constraints of virtual objects like length, area, or volume since they represent geometric properties of realistic materials induced from physics. Many of these explorations are proved to be suitable for interactive design. Typically the users interact with physics-based animation triggered by their manipulation on element positions of the model. Then the movements of the model induced by conservative forces and external forces given by the user will lead the model to a new equilibrium. The conservative forces are purely simulated by enforcing the constraints via optimization methods that happens implicitly at every frame of the animation.

Not only the interactivity is largely enhanced by these constraint-driven design tools, the design space accessible to the users are largely widen by these techniques. Throughout the long history of design, designers are trying to produce creative combination of different materials in parallel or sequential or hierarchical way, but most of them are produced by ad-hoc or case-by-case methods. The recent development of constraint-based shape editing tools potentially induce a systematic way of exploring the design space in the early stage of design. Most often, these tools play a similar role as generative design tools.

On the other hand, sometimes physical constraints exist for a more practical reason that the created form is going to be fabricated in real life. The fabrication materials in the real life often bring some constraints on valid deformation. Mostly these constraints are handled by post-rationalization which means optimizing the model given by the user to produce the nearest form that is producible.

In both situations, the optimization processes applied to reach the constraint conditions are rarely viewed as independent tools or separate oper-

ation accessible to the users by themselves, thus making the optimization process alone a black-box to its users. For post-rationalization, the creative portion of modeling ends before the optimization begins. For constraint-based simulation, the optimization is done implicitly in every updating step. In most of implementations, the optimization steps is purely automatic and does not even have interfaces to the users. There will be more details referring to these issues included in "Statement of Research Goal" chapter.

To explore these questions, the thesis introduces a new design pipeline that makes the step processes of optimization explicit and accessible to the users. There are two main purposes for this. First, users could gain a better control over the whole optimization process instead of waiting for the black box finishing its jobs. Second, users could combine different constraints and insert external manipulations spatially and temporally to explore the design space enabled by various DIY combinations.

Compared with traditional implicit method used in forward design, this new pipeline let users explicitly manually project the constraints when they think it's needed instead of letting the machine handle the projection at every frame.

Compared with traditional optimization method for inverse design, this pipeline allows the user to participate in the process of optimization and edit the object freely in the intermediate stage. Therefore, users might gain better control with the new pipeline.

Chapter 2

Background

2.1 The General Study of Sketching and Conceptual Design

Sketching is an inseparable part of the design process. At the early stage of design, sketching as a tool helps designers analyze, synthesize, and enrich their initial immature ideas and gradually convert them into practical solutions to specific problems.

Purcell summarizes sketching properties as "Ambiguity, reinterpretation, knowledge generation, cyclic process, and expertise related"(Purcell & Gero,

1998) , which are essential for sketching being necessary to promote creativity and efficient communication. These five properties are interrelated. Being ambiguous is why the reinterpretation for the same sketched object is possible, and reinterpretation is the basis of generating knowledge, which happens gradually in a cyclic process.

Sketching is an interaction between the subject and object of a design process. Reinterpretation can be conducted by a different interpreter, but it also happens when a single designer faces his/her work at different phases. Sketching not only mediates communication between collaborators, but it is a tool for a designer to communicate with him/herself as well. Different multiple researchers have proposed cognitive patterns of sketching. Goldschmidt recognizes sketching as a process of rapid switching between two phases known as "seeing that" and "seeing as"(Goldschmidt, 1991), a different approach from scientific and technological reasoning. Schon promote a similar model that regards such a process as "seeing - moving - seeing", a conversation between the designer and design material(Schon & Wiggins, 1992).

The most important shared opinion among these studies is that sketching

is both a demonstrative and a creative tool that externalizes the imagination of designers and motivates, stimulates, and inspires them. This observation is essential to analyze further the deficiency and advantage of digital media – a potential stage of future sketching.

2.2 The Comparison between Digital Tools and Conventional tools for Conceptual Design

Based on the property of sketching and conceptual design, the majority of the works at the early stage of design is still completed through conventional paper-based methods.

Goel and Bilda (Goel, 1995) (Bilda & Demirkan, 2003) have explored the possibility of conducting conceptual design with digital media, and most of them conclude that the current CAD tools are not suitable for fast iteration and stimulating creativity. Compared with paper-based conventional meth-

ods, CAD is less efficient in producing unique creative works. Alcaide-marzal and Bilda shew that the group of designers using digital media as their tools produces smaller number of total works with lower variety than the group using pen and paper (Alcaide-Marzal et al., 2013)(Bilda & Demirkan, 2003). While most of studies claim that conventional paper-based has advantage in providing inspiration for alternate solution and fast experiment, some researchers (Won, 2001)(Séquin, 2005)(Jonson, 2005) claim that CAD may stimulate more complex cognitive behavior that is useful for generating new shape and patterns, which indicates that there is still potential for CAD to become a powerful tool for conceptual design so long as its current deficiencies can be overcome.

Lee (Lee & Yan, 2016) suggested the monotonicity of the mathematical expression confines the design space of designers. They claim that the unavoidable discretization of digital data also discretizes the design space, which converts creative works into making choices. They suggest the future direction for developing CAD tools could be adding more variety to the mathematical expression of digital models.

2.3 Relative Techniques

2.3.1 Digital Sculpture

Digital sculpture was defined as direct 3D meshes deformation (Alcaide-Marzal et al., 2013). "Digital sculpting software provides an interaction with 'digital clay' (in the form of polygonal meshes). Through strokes performed in a graphic tablet, the mesh is deformed and material is added or removed almost like real clay, and shapes are constructed this way." A more precise definition was given by (Cheutet et al., 2005) as a system with free-form deformation features, i.e. "being the shapes obtained by deforming parts of a free-form surface according to adequate constraints, which are the parameters of the feature." In general, digital sculpture enables users to interact with virtual material with the natural material property.

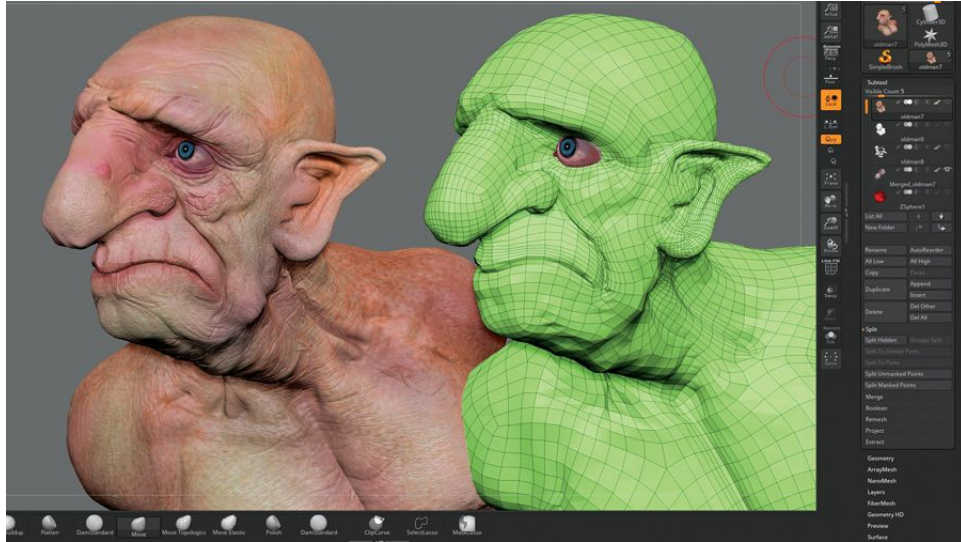


Figure 2.1: The interface of Zbrush: user freely edit the monster model

The digital sculpture is developed well in the industry; there are multiple software features or supporting mesh deformation, e.g., ZBrush, Mudbox, or 3D Coat. The tangibility of a CAD system can be significantly improved through a coupled system of tablet I/O and digital clay.

Traditionally, users are usually not able to interact directly with force related to material property. Recently, researchers also started to explore the possibility of basing deformation rules on accurate continuum mechanics. Kelvinlets (de Goes & James, 2019)(De Goes & James, 2017)(De Goes & James, 2018) is a typical example among these techniques. It applied the

analytical solution of elasticity to derive the suitable deformation of digital clay under the manipulation of the users.

Alcaide-marzal has explored the possibility of applying digital sculpture to increase the tangibility of CAD tools. The result of their investigation proved Their hypothesis that a coupled system with tangible I/O, gesture recognition and interactive incremental geometric deformation can mend the deficiency of conventional CAD.

2.3.2 Interactive Physical-based Modeling

Physical simulation has been long combined with a CAD system. It is usually for testing or demonstration purpose. Based on the outcome of simulation, inverse design and optimization-based methods are always possible to further modify the design outcome.

Material properties can be simulated in different perspectives, both in optics and dynamics. For optics properties, the most common example is the physical-based rendering of real objects. The visual material properties can be represented in a bidirectional reflecting distribution function which is

directed applied in rendering algorithms. Since the proposing of the Monte Carlo ray-tracing algorithm, there has been a wide range of its applications in various fields(Pharr et al., 2016), including the entertainment industry, architecture, industrial design, Etc. In industrial design and architecture, heat simulation(Asawa et al., 2008)(de Béjar et al., 1993) and structure analysis are widely used to estimate the realistic situation after the product/building is produced.

Recent studies explored possibility of combining material-properties based simulation with user interface. (Zhu et al., 2017) explored how fluid simulation can be applied to the interface of medical CAD software. The CADenary project (Galbraith, n.d.) applied mass-spring system as a design interface in his course project at Yale University. Another series of works explored how to interactively simulate curved-crease-folding and developable surface based on their geometry properties(Rabinovich et al., 2018). Kelvinlets mentioned in the last subsection is of course another typical series of examples. In the industry, The newest version of (Zbrush, n.d.), a well-known digital sculpture software, is equipped with a dynamic module that can simulate clothes and soft material interactively. When the user drags the cloth, it will drift as real

until the user stops.

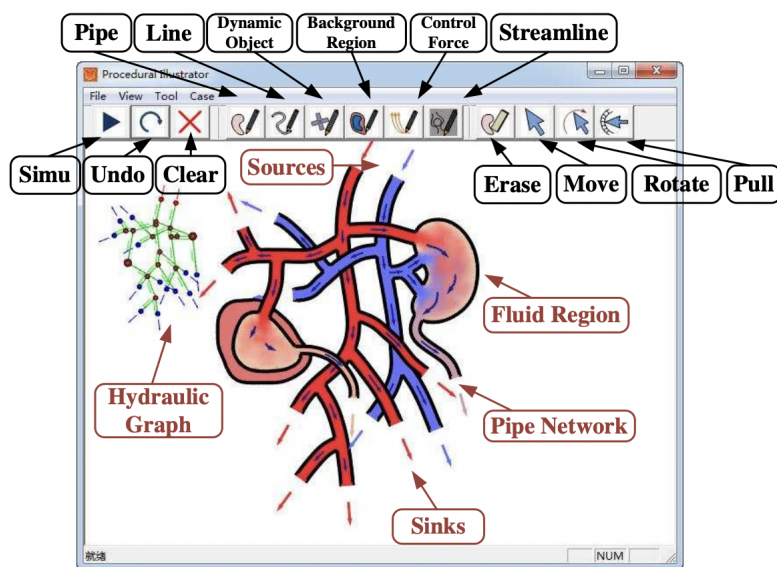


Figure 2.2: Fluid simulation for medical purposes

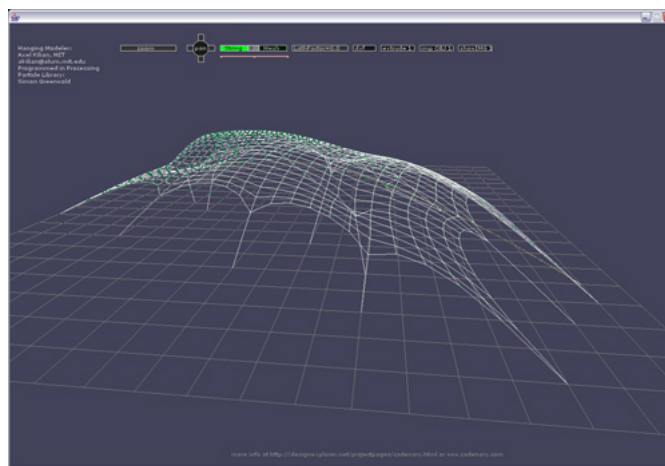


Figure 2.3: The interface of CADenary: An platform for modeling based on mass-spring system

Next we shall introduce a series of common techniques applied in physical-based modeling.

Following Classical Mechanics

The traditional way of simulating deformable objects is always based on the strict mathematical depiction of physical laws. Throughout history, there have been many mathematical abstractions of physical laws with different levels of generality and limitations. One may choose to talk about force, displacement, velocity, and acceleration, or talk about momentum, impulse, and energy, or talk about Hamiltonian and Lagrangian. The common point

of these methods is that they all can accurately depict natural laws within a specific scope.

To simulate the models' proper deformation, one needs to follow the equation of motion derived from the above abstractions. Sometimes the force can be easily computed, and then the simulation might be relatively straightforward. One would only need to discretize the equation of motion based on various time integrators. However, under most of the circumstances, the force has to be derived under the complicated situation. One may need to derive the force via calculating the gradient of potential energy or averaging the impulse. Sometimes one can also choose to skip computing the force via deriving the Lagrangian and directly obtain the discrete equation of motion.

The traditional methods used to simulate deformable objects include different ways of discretizing the physical system spatially or temporally. The temporal discretization includes various types of time integrators. These integrators can be derived via the finite difference method or the discrete version of the principle of the least action if the conservation of some specific physical quantities such as momentum or angular momentum is required. Spatial

discretization often applies finite element methods, finite volume methods, or finite difference methods. The partial differential equations that depict the realistic physical phenomena are always too complicated to solve analytically, so people use the above methods to approximate the solution. First, the model's domain shall be discretized into any finite data structure such as nodes and edges in 1D, triangular surface in 2D, and tetrahedrons in 3D. Then based on these discretizations, finite difference uses difference quotients to approximate differential quotients. Finite volume calculates the values on the faces of the elements based on the divergence theorem. The finite element method defines a set of basis functions, typically these are different kernel functions or polynomials, for every spatial element and uses Ritz-Galerkin projection to find the most suitable solution in the space spanned by these bases. The common ground of these methods is their respects towards the detailed mathematical formula defined in classical mechanics.

However, the mathematical depiction used in the simulation is not always the same as the one in classical physics. Nonetheless, they are often derived following the same methodology. For example, Bergou derived potential energy based on discrete differential geometry for elastic rods (Bergou et al.,

2008) while the traditional methods often involve strain and stress. The geometry-based approach can avoid lots of complex computation compared with traditional methods. However, it still follows the same methodology of classical mechanics at the high level, i.e., using quadratic energy to derive the force and use an integrator to discrete the equation of motion.

Constraints in Simulation

Constraints exist everywhere in a real physical system. For a rigid body, its shape would be invariant throughout the simulation before it is broken. Paper subjects to bending but not sketching, which means the lengths of any curves and the areas of any closed shapes on the paper shall never change during any possible deformations on the paper. Two solid objects usually cannot penetrate each other, which means the shortest distance between any portions of these objects should always be larger than zero.

In simulations, constraints are used for different purposes and handled in very different manners.

Most often, constraints are used as projection methods and methods men-

tioned in the last section to increase stability. Because of numerical error, some constraints cannot be preserved well by traditional methods. For example, a pendulum always keeps the length of the rod while it is swinging. If a simulator applied explicit Euler methods to integrate the phases, the rod's length is likely to grow during the simulation. So it is often necessary to handle the constraints explicitly by projecting the current configuration onto the valid submanifold in the configuration space.

Sometimes, constraints need not to be handled explicitly. One could enforce the constraint by modifying the traditional methods. For example, if a simulation is based on Lagrangian mechanics, to enforce the conservation of linear momentum or angular momentum, one would only need to design a proper Lagrangian, which is generalized kinetic energy minus generalized potential energy, to be symmetric to translation or rotation, and the conservation is automatically satisfied due to Noether's Theorem. Sometimes, if the physical system is relatively easy to be understood, one could use reduced coordinates. For example, a rigid body's motion can always be represented by a translation and a rotation, which only needs three degrees of freedom for translation and another three for rotation. With reduced coordinates,

the configuration of the system always stays on the valid manifold without external Interference.

Also, constraints can be handled by traditional force-based methods. Taking the pendulum simulator as an example again, one can imagine there is always an force that resists the change of the rod's length during swinging.

Constraint-based Inverse Design and Forward design

Constraint-based methods are widely used both in inverse design (post-rationalization) and forward design (pre-rationalization).

The basic procedure of post-rationalization involves first providing a user-specified free-form shape and some user-defined physical properties and constraints, and then the computer will optimize the input shape to make it satisfy these conditions meanwhile ensure the modification is slight as possible. For example, to make a shape valid to be a paper folding structure, one would need to enforce the paper's constraint condition mentioned in the last subsection. Geometrically, it is equivalent to enforcing the developability of

the surface. Stein introduces a method to convert any triangular meshes into a developable surface in 2018(Stein et al., 2018).

Sometimes the input shape can be omitted, and the computer will search all possible shapes that meet the condition. Topology optimization is a good instance for this method(Matusik & Schulz, 2019). The user would only need to assign the total amount of available materials and desired mechanical properties. The computer will optimize the distribution of the material to reach the best suitable status. Also, some recent developments of topology optimization make it possible to support optimization of global functions such as target deformation under certain pressure(Zhu et al., 2017).

Pre-rationalization takes a different approach. It does not enforce the constraint conditions after the creative part of the design is done. Instead, it is enforced on every deformation step manipulated by the user. Either reduced coordinates handle this so the user can never leave the constraint design space, either the optimization (projection) is performed after each deformation step. (Rabinovich et al., 2018) is an example for pre-rationalization, where every manipulation will be converted into isometric transformation,

so the developability of the deformed surface is always satisfied.

There are multiple commercial CAD modeling software or plugins that support constraint-based forward design. Kangaroo 2, a plugin for Grasshopper, is a typical example among them. It contains lots of algorithms mentioned above, including inverse design, and its core part is constraint-based pre-rationalization. Users can specify the desired constraints, and the plugin will send the current configuration to the nearest point in the constraint space. The user's manipulation will never cause the configuration to leave the constraint space afterwards unless the user releases the constraint condition.

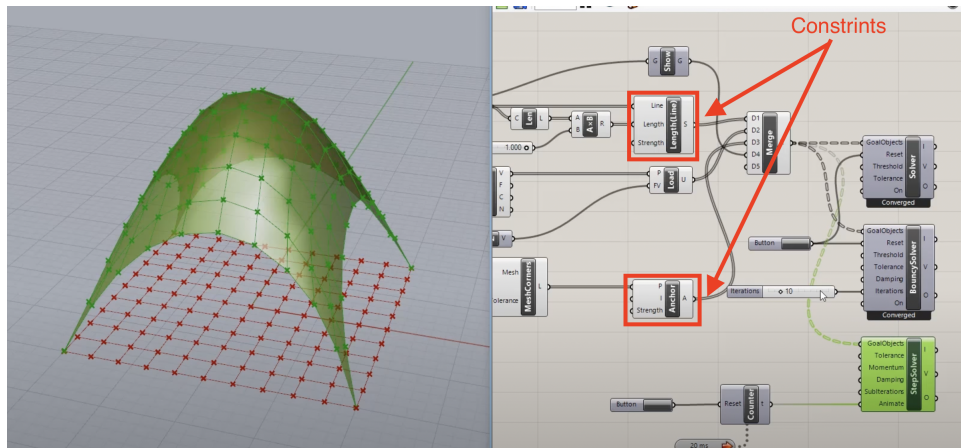


Figure 2.4: The interface of CADenary: An platform for modeling based on mass-spring system

Chapter 3

Statement of Issues and Research Goal

Constraints-based methods have been widely used in forward and inverse design and deeply relate to design ideation. In forward design, it functions as a tool to aid creative work. In inverse design, it is a tool that brings an arbitrary creative work to a manufacturable item.

However, the current pipeline for inverse and forward design does not support feedback loop between the user and the constraint-based simulation tool. In forward design, the conventional pipeline typically mimic the real-

istic situation where the constraints exist implicitly everywhere. Therefore, the methods that enforce the constraints normally do their duties without explicitly notify the user. For inverse design, the optimization process happens after the user input an arbitrary shape. It is handing over and taking over job in an assembly line, and there is no communication between the user and the machine.

As a result, the user of the conventional constraint-based pipeline can only observe the outcome at high level. Especially for inverse design, the user can only see the input shape and the final outcome and decide to accept it or not. Sometime the final result diverges a lot from the given shape, and the user has to redesign the input shape, which turns out to be a great obstacle to design ideation.

There is also a lack of flexibility in using constraint-based tools to explore the design space for ideation. For example, in the conventional forward design pipeline that applies projection-based method, the configuration is projected multiple times until convergence every frame. The user can choose if any constraints exist, but if they exist, they will always be handled in every

frame in the same way. The user cannot choose to detach the projection from the frame-based procedure of simulation. For example, the user cannot ask the computer to project the

To explore and solve these questions, the thesis will try to develop a new design pipeline that detach the constraint from frame-based automatic operation or implicit operation. The pipeline is still based on projection as forward design but user can take projection as a geometry processing tool that needs active triggering.

3.1 Design Pipeline and Design Space of Constraint-based Forward Design

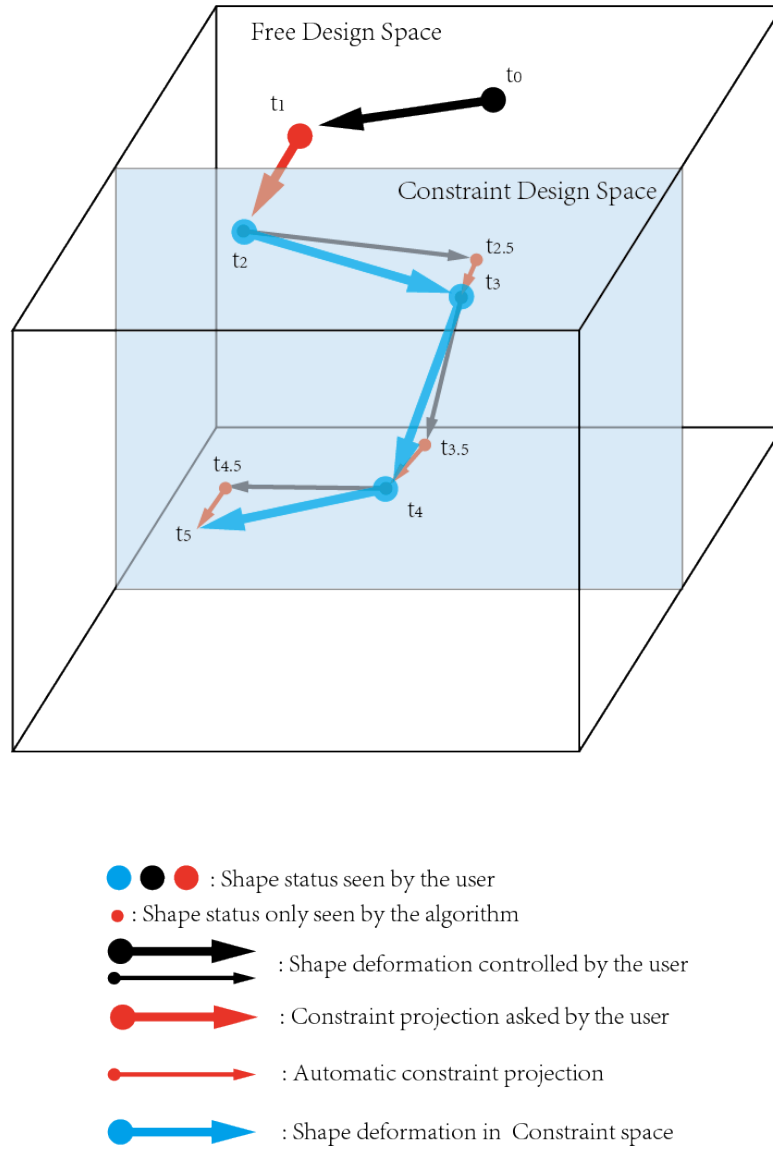


Figure 3.1: Design pipeline of constraint-based forward design

Figure 3.1 has shown the design pipeline of constraint-based forward design and its design space. The initial shape is given at t_0 . User freely edits the shape and reached t_1 . Then the user adds the constraint, the computer projects the shape to t_2 . Then the user keep freely editing the shape, if reduced coordinates is applied, t_2 will directly change to t_3 . If projection methods is applied, the shape will first change to $t_{2.5}$ as if there is no constraints, then projects it to t_3 . The whole process is invisible to the user. Whether by reduced coordinates or projection, the user can only see the processes marked by bold blue arrows.

Note: Sometime people can see fluctuating happens when elastic object reaching an equilibrium status. This is fundamentally different from the optimization (projection) processes introduced above. It can be viewed as a sequence of blue arrows approaching an stable configuration.

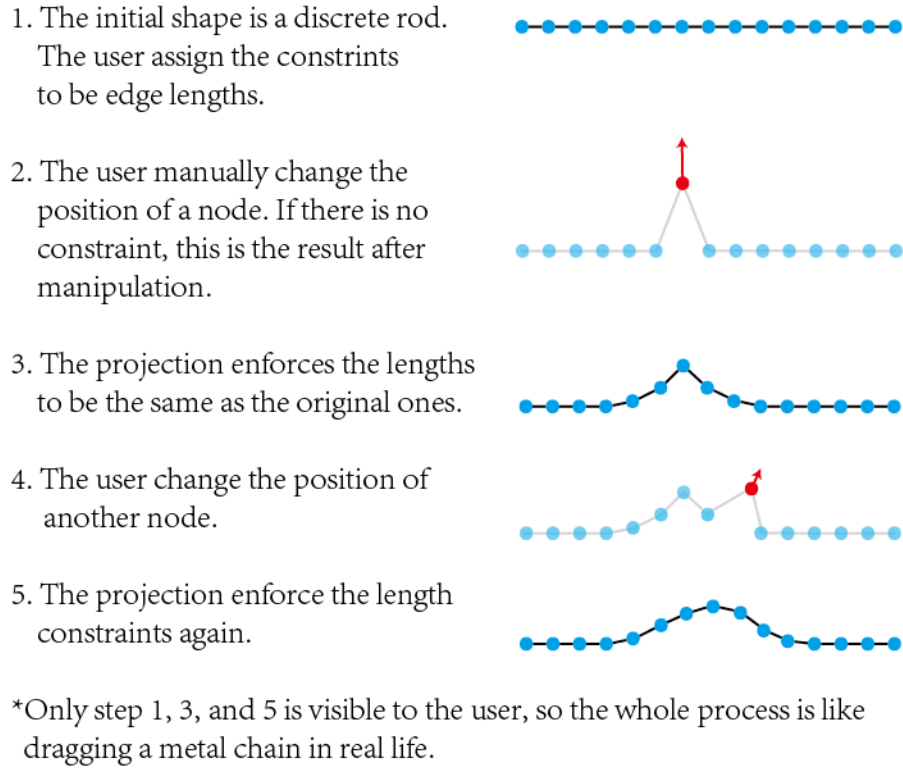


Figure 3.2: An example of forward design pipeline

Figure 3.2 shows an example of forward design pipeline using projection. Step 2 and step 4 is completely invisible to the users. The users are completely confined in constraint space during the whole process of design ideation. The projection is like a tacit servant who does not communicate with the master at all.

3.2 Design Pipeline and Design Space of Constraint-based Inverse Design

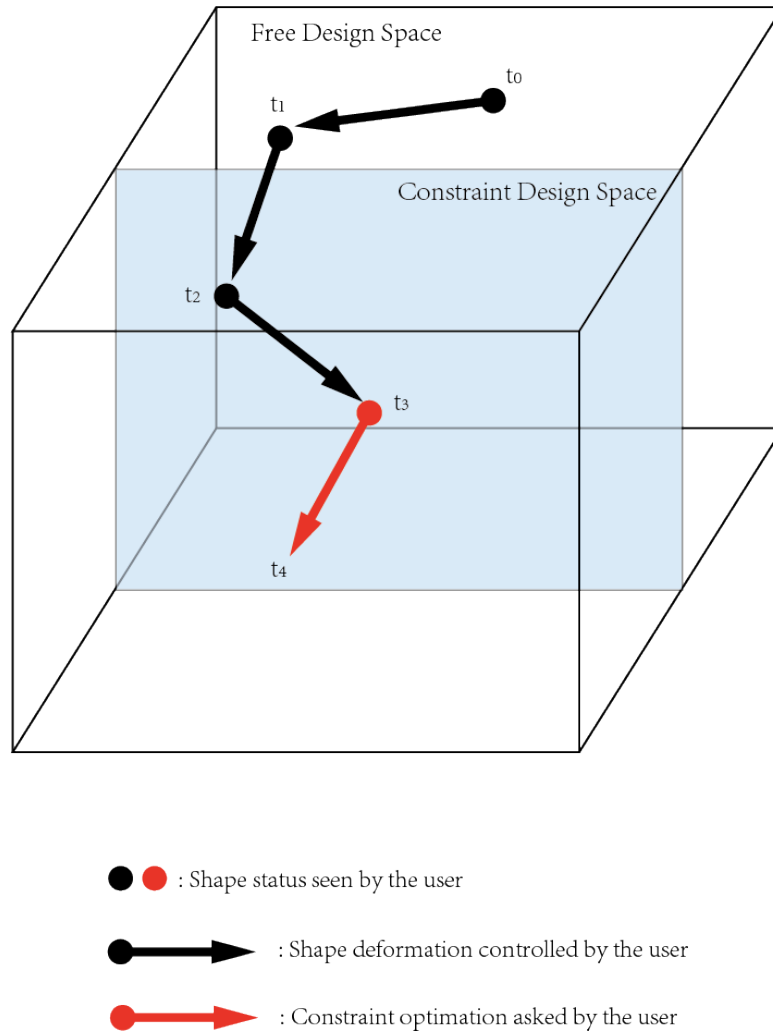


Figure 3.3: Design pipeline of constraint-based inverse design

Figure 3.3 has shown the design pipeline of constraint-based forward design and its design space. The initial shape is given at t_0 . User freely edits the shape and reached t_1 to t_3 . Then the algorithm handles the optimization to give the final shape at t_4 .

Note: Simply judged from the pattern, the process seems to be much easier than forward design. That is due to two reasons. 1. The optimization is not as simple as the projection in the forward case. It often involves very tricky techniques that costs a lot of computational resources. For example, a single round of topology optimization at high resolution might takes hours or even days to complete. 2. The process is often designed as automatic as possible, which means it usually do not require user's manipulation at constraint design space. That is usually taken as an advantage of these methods.

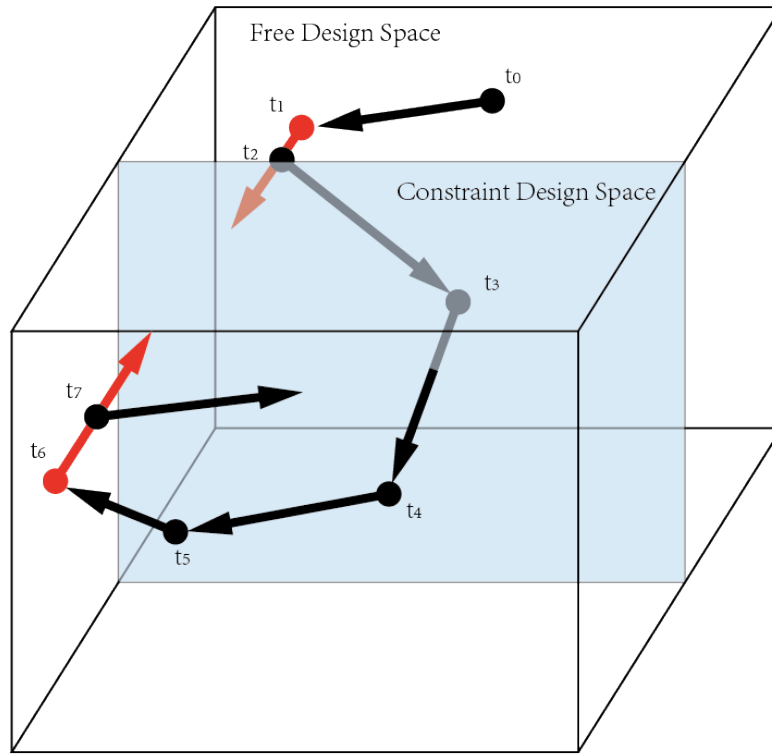
3.3 Discussion of Current Pipeline and the Targeted Design Pipeline

Though constraints have shown great power in both forward and inverse design. There are still some limitations of the current design pipeline.

For forward design, users are always trapped in the constraint space if the constraint condition is enforced, and that means the exploration of design space is confined. Any trial leaving the constraint space will instantly and implicitly projected back.

For inverse design, usually users cannot interfere the optimization process in the middle. That often leads to a largely deformed final outcome that is not desired by the user.

For both methods, the process of optimization/projection is not helpful for exploring the design space and supporting the creative ideation. They are purely viewed as functional units of the pipeline.



- ● : Shape status seen by the user
- → : Shape deformation controlled by the user
- → : Controlable constraint projection

Figure 3.4: Targeted design pipeline

Figure 3.4 has shown the design pipeline of constraint-based design that this thesis is trying to introduce.

The core idea is simple as introducing optimization processes that can be interrupted or interfered by users (t_2 and t_7) and making each step of optimization visible to the user. The user will manipulate the optimization process manually. The user shall actively call the projector to deform the shape. This pipeline is different from both inverse design pipeline and forward design pipeline because projection is no longer a automatic process, instead it becomes a design tool.

There are two purposes for this introduction. First, users are able to grasp the whole process of the optimization to gain a more precise control. Second, users can take optimization as a generative tool to explore the design space instead of purely functional unit. In this case, constraints are not set as goals that the shape must achieve. Instead, the process that walks towards these goals opens possibility of creating new shapes. One could choose to enforce one constraint condition for several times and then enforce another, and different combinations may lead to different result. Even if the constraint conditions must be satisfied for practical reason, one would only need to satisfy it in the last iteration.

Compared with traditional implicit method used in forward design, this new pipeline let users explicitly manually project the constraints when they think it's needed instead of letting the machine handle the projection at every frame.

Compared with traditional optimization method for inverse design, this pipeline allows the user to participate in the process of optimization and edit the object freely in the intermediate stage. Therefore, users might gain better control with the new pipeline.

Chapter 4

Statement of Method

This chapter focus on the technical methods and platform built for achieving the research goal. The outline of the platform and its function is simple. The platform supports basic mesh operation like editing the position of vertices. Also, it supports manually enforcing the constraint. To be specific, there is button for projection. Also, the user has the option to specify the parameter of the constraints.

There are two types of platform built, one is for 1D discrete curve, the other is for 2D triangular mesh. 1D platform only supports length constraints. And there are two different types of constraints supported by the

2D platform. The first one is edge length, the second one is dihedral angle. And the corresponding constraint parameters are the value of lengths and dihedral angle.

4.1 The Algorithms to Enforce the Constraints

4.1.1 1D - Fast projection

1D curves are discretized as a set of nodes and edges. The essential constraint is the lengths of edges and the angles between them. The way to enforce these constraints condition is fast projection, which is essentially a method to solve Lagrangian multiplier (Goldenthal et al., 2007).

The origin positions of nodes are denoted as x_0 . The movement introduced by the constraints is δx . $x = x_0 + \delta x$. The constraint is

$$C(x) = 0. \tag{4.1}$$

The objective function to be optimize is

$$f(x) = (\delta x)^T M(\delta x) + C(x)\lambda \tag{4.2}$$

where λ is the Lagrangian multiplier. To solve this optimization problem, one would need to first derive the values of λ which can be solved by a linear equation:

$$\nabla C(x_0)M^{-1}\nabla C(x_0)^T\lambda = C'(x) \quad (4.3)$$

where M is the mass matrix. Then δx can be computed as :

$$\delta x = -M^{-1}\nabla C(x)^T\lambda \quad (4.4)$$

and x can be computed easily.

Usually, fast projection is a iterative process that does not stop until its convergence. But due to the purpose of this thesis. A single step processing is available to users.

4.1.2 2D - Position-based dynamics

In 2D case, the surface is discretized as a triangular mesh and the constraint enforcing algorithm in position-based dynamics is applied(Müller et al., 2007).

We denote the positions of vertices before projection as a $x = (x_0, \dots, x_n)$

and the movement of every vertices as $\delta x = (\delta x_0, \dots, \delta x_n)$. For every vertex, we associate a mass m_i and a corresponding weight $w_i = 1/m_i$. And the constraint equation is still denoted as:

$$c(x) = 0 \tag{4.5}$$

The movement can be computed as:

$$\delta x_i = -s w_i \nabla_{x_i} C(x) \tag{4.6}$$

where

$$s = \frac{C(x)}{\sum_j |\nabla_{p_j} C(p)|^2} \tag{4.7}$$

There are two constraints enforced. The first one is constraining the length of every edge. The second one is constraining the dihedral angles between every two faces. Still, for the same reason, though a typical implementation requires convergence, we allow one step optimization.

4.2 Software Platform

To achieve the research goal, two software platforms are built to test the idea mentioned in the last chapter.

Both of them are based on the geometric processing library Geometry Central and the visualization library Polyscope(Sharp, Crane, et al., 2019)(Sharp et al., 2019).

4.2.1 1D Platform

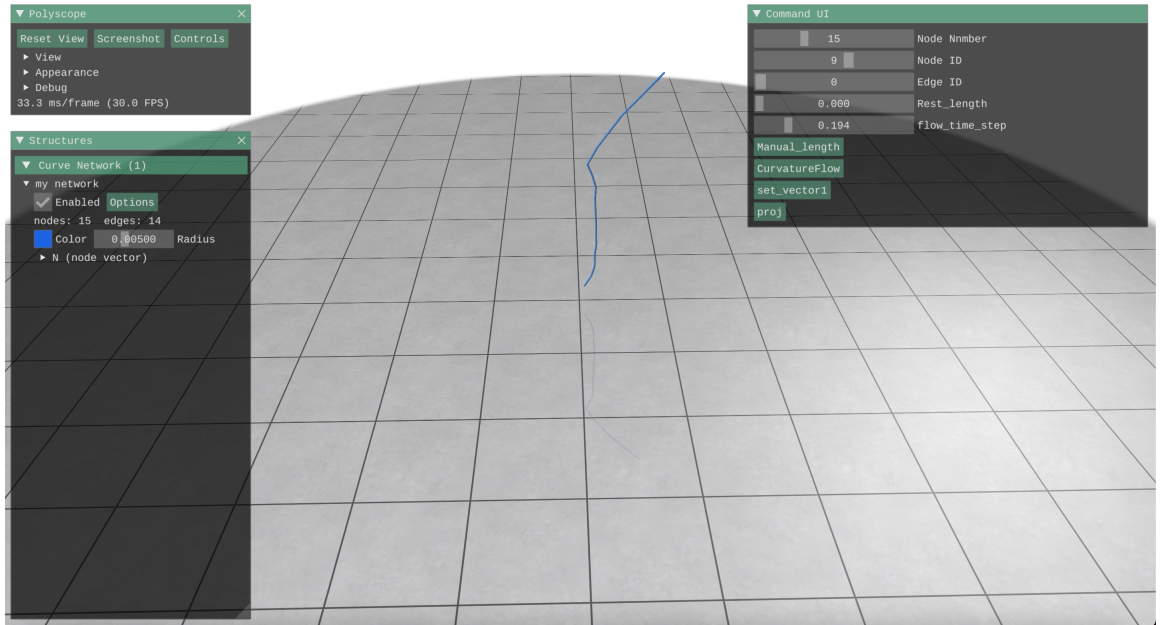


Figure 4.1: Platform for 1D

1D platform allows free editing of any nodes. Users can first specify the node to move then manipulate its position by keyboard. Users can also specify the number of nodes on the discrete curve by dragging the slider in the upper left.

The constraint of lengths can be enforced by clicking "proj" button. The fast projection will be executed by only once. If the user is not intend to manually specify the length, the default rest length would be the initial value.

Besides the projection of length, the platform also implemented mean curvature flow to provide more interaction with constraints. Also this is for fixing some uncontrollable affects caused by enforcing the length constraints.

4.2.2 2D Platform

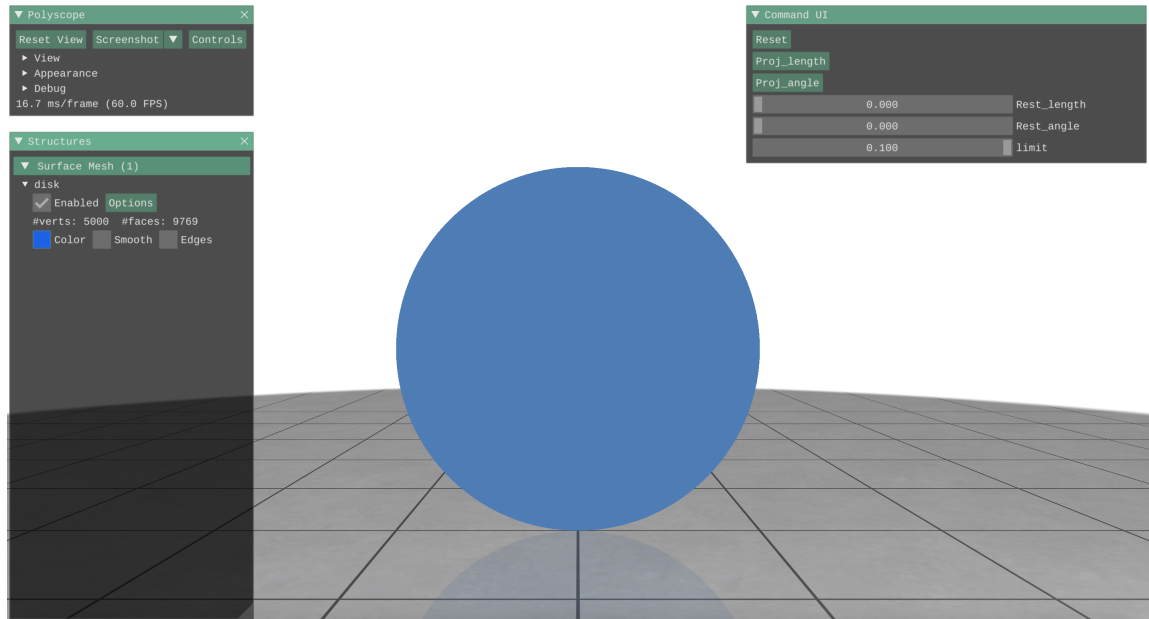


Figure 4.2: Platform for 2D

The 2D platform supports almost the same functionality as 1D platform. Users can pick the a subset of vertices and freely move them by clicking the keyboard.

The platform supports both constraints of lengths of edges and of dihdral angle between every two faces. Users can specify the rest lengths or dihdral angles by dragging the slider in the upper right. If the rest lengths are zeros or the dihdral angles are zeros, then the default values which are the original value of the input mesh will be taken as rest values.

Users can also specify the limitation of the largest movement. If a movement is larger than this value, it will be normalized.

If users click reset button. The picked subset will be cleared and the user can select a new subset or vertices.

4.2.3 Experiment Platform

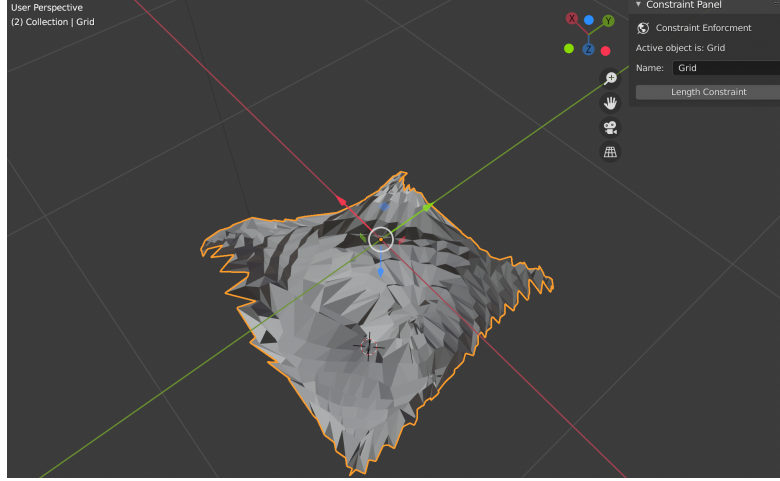


Figure 4.3: Platform for Experiment

The experiment platform is built to support user tests. The previous platforms does not support complicated free form editing, but this platform is built based on open-source software Blender(Community, 2018), therefore, users can apply all mesh editing techniques supported by blender and explore the design space enabled by different conventional tools and constraints.

This platform only support 2D case with edge length constraint. To activate the algorithm, users would need to click the panel on the far-right of 3D-view.

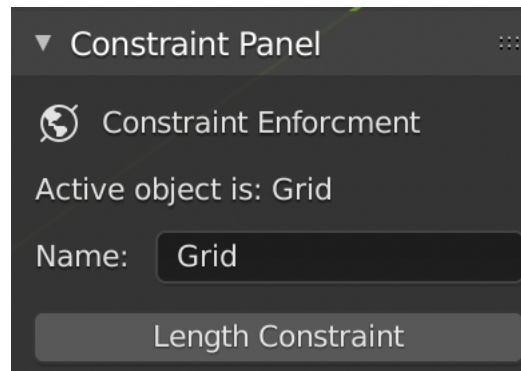


Figure 4.4: Platform for Experiment

Another window will prompt out if the user click the button on the first window. User would need to specify the length to be constrained and click "OK" to start the process.

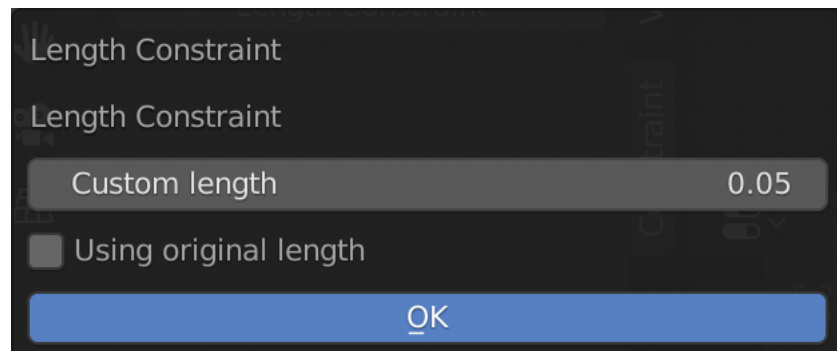


Figure 4.5: Platform for Experiment

Chapter 5

Analysis and Experiments

The first part of this chapter is analyzing the potential ability provided by these platforms. Even if the geometric intuitions of constraining dihedral angles or edge lengths are clear, it remains unknown what global effects they might produce. Therefore, it is important to analyze the actual functions of enforcing constraints before user testing.

In the first chapter, instead of emphasizing the feedback loop between the user the machine, we will temporarily only focus on the design space spanned by the algorithm.

The analysis is performed in both 1D and 2D cases.

Figure 5.1 shows the input shape for testing. For 1D platform, the shape is a simple discrete rod made with contiguous edges and nodes. For 2D platform, the testing shape is a triangular disk.

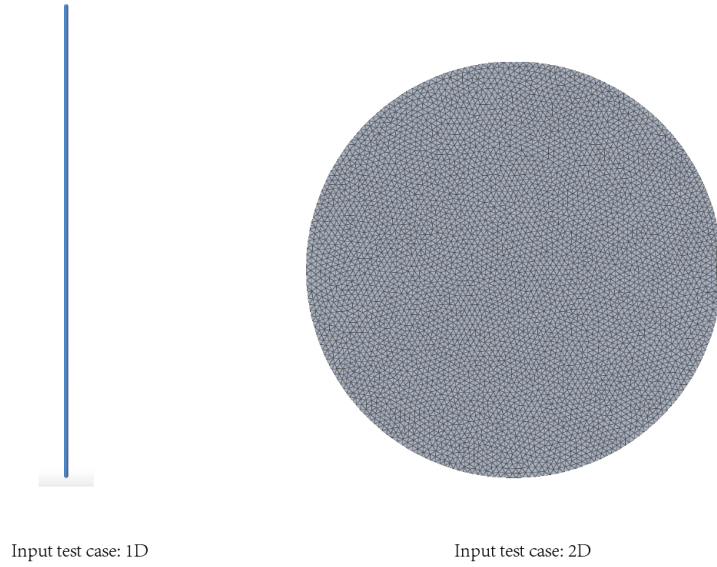


Figure 5.1: Input shapes for testing

5.1 1D Analysis

In 1D case, the possible deformation enabled by constraint enforcement is limited. User can freely move any subset of the nodes on the discrete curve.

By clicking the projection button and assigning the constraint parameter, the user can make the nodes move towards the direction that satisfy the constraint condition.

5.1.1 Interaction with Freely Moving vertices

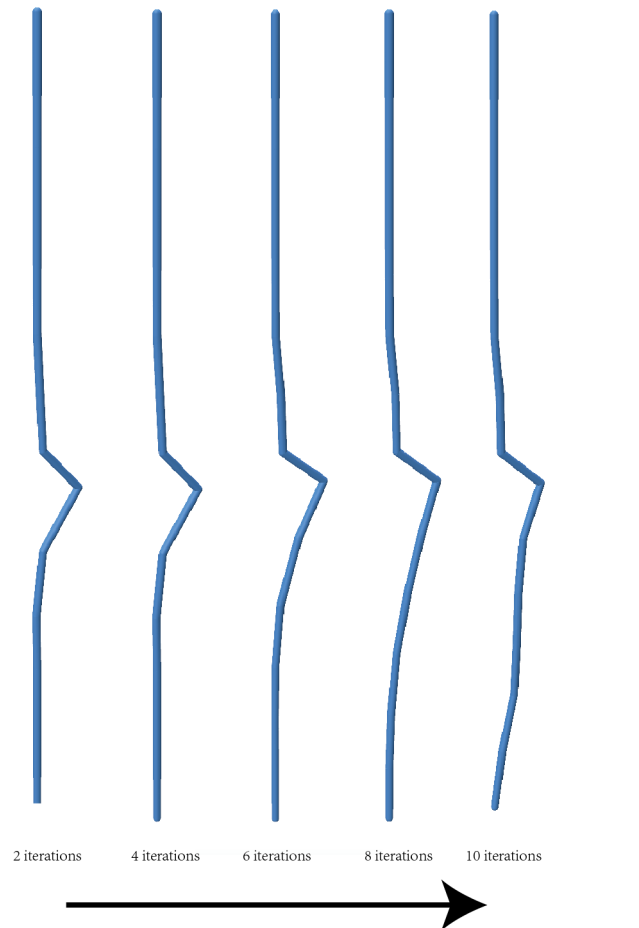


Figure 5.2: Experiment result

In figure 5.2, user firstly change the position of one node on the discrete curve. Then keeping clicking the projection button results to shapes that more and more close to the constraint enforced shape.

5.1.2 Interaction with smoothing algorithm

The combination of smoothing algorithm and constraints enforcement might produce more aesthetic results under certain circumstances.

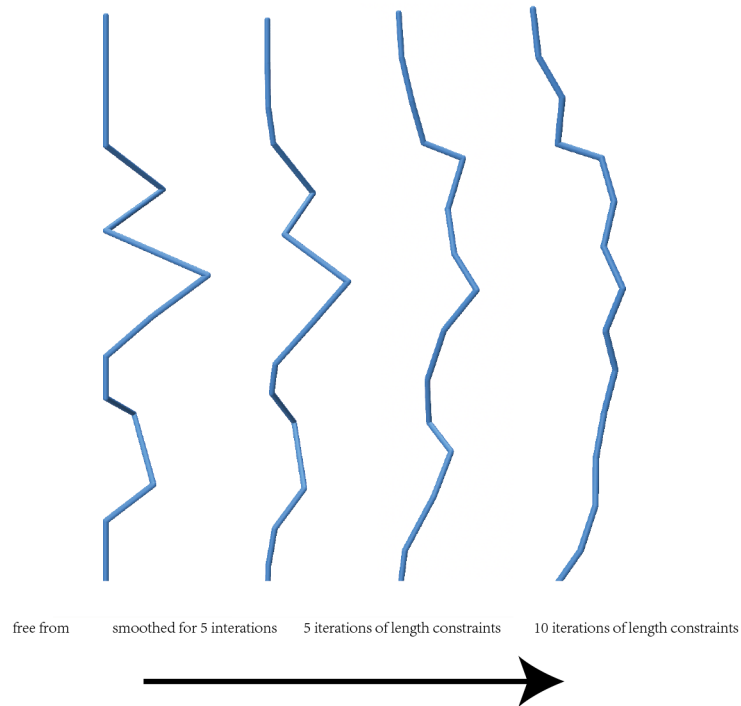


Figure 5.3: Experiment result

5.2 2D Analysis

However, 2D algorithm shows abundant properties and possibilities of deformation under different circumstances.

5.2.1 Manually Assigned Rest values without user interference

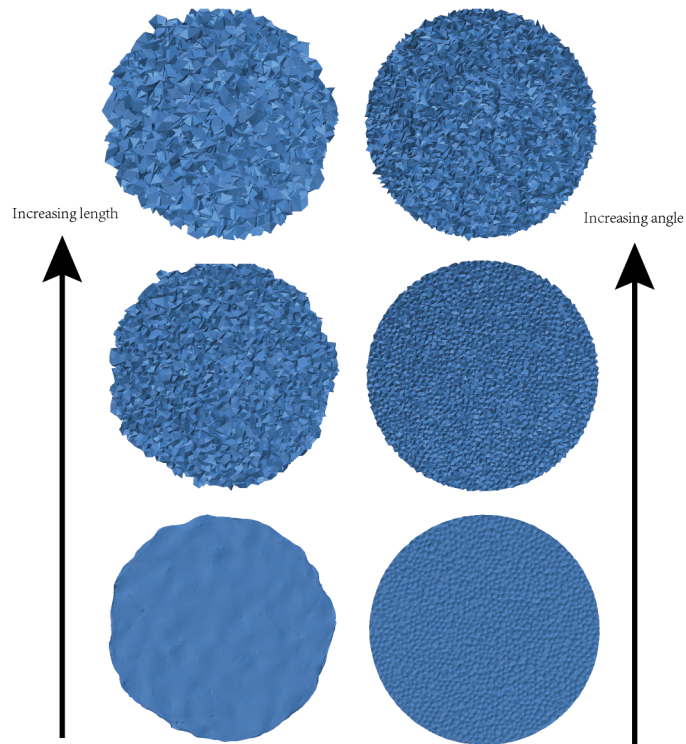


Figure 5.4: Experiment result

First of all, even without user interference, enforcing the algorithm with different types of constraints and constraints parameter brings different results.

The left column is the results of length constraints and the right are the results of angle constraints. If the user increases the target constraint length, the surface will become coarse because longer edges will squeeze with each other. If the user decreases the length, the surface is going to be smoothed.

If the user increases the target constraint dihedral angle, the surface is also going to be smoothed. If the target angle is small, the surface is also going to be coarse, but its generated texture is different from length constraints. If the target angle is too small, the result will explode soon because it does not satisfy the requirements of a valid geometry.

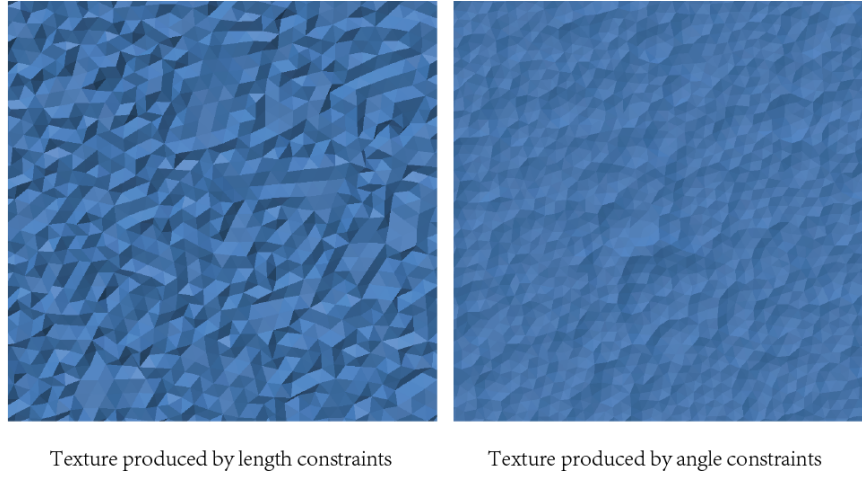


Figure 5.5: Experiment result

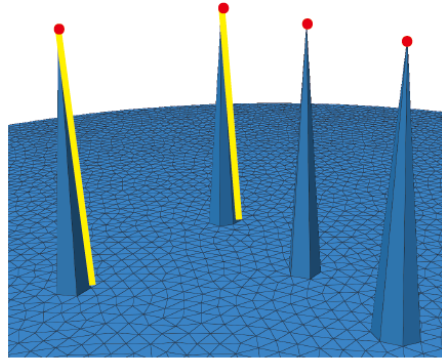
5.2.2 Interaction with Freely Moving vertices

The next subsection will focus on how constraints enforcement interacts with user manipulation. That is to say the user will interleave the free editing of the mesh and constraint projection.

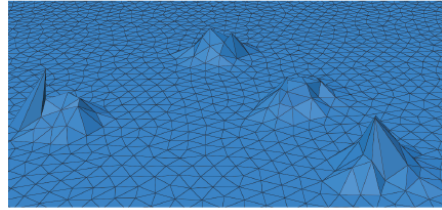
The first part of the testing is for length constraints. The result is similar to 1D case. The user firstly manually manipulate the mesh, then they press the projection button to constrain the length. The constraints after manipulation often function as a smoother. One difference between conventional geometric smoother and length constraints is that the intermediate processes

create more organic form than conventional methods.

Step 1: The user freely lifts four vertices on the mesh. We observe that this operation causes several long edges (some examples are marked by yellow).



Step 2: The user clicks the button to project the length constraint. This operation draws the long edges back. It also changes the positions of nearby vertices to further meet the needs. The overall effect is like smoothing.



Step 1: Further projection leads to further smoothing.

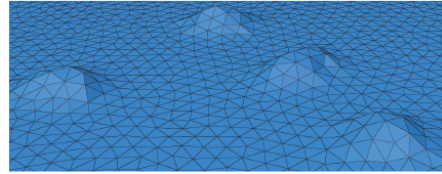
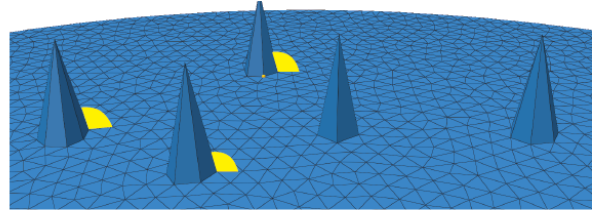


Figure 5.6: Experiment result: Length Constraints

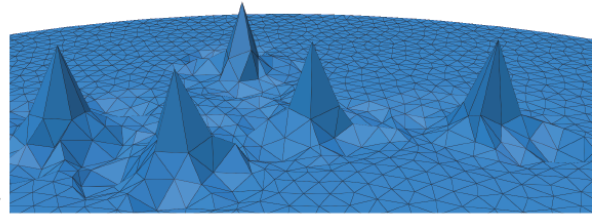
The second part of the testing is for angle constraints. The basic procedure is the same as length constraints. The user firstly freely lifts arbitrary subset of the nodes then keep clicking the projection button. The target angle here is flat angle. The algorithm will try to recover the deformed an-

gles produced by the user by bending the surface around these angles. Such operation may produce some organic forms in the end.

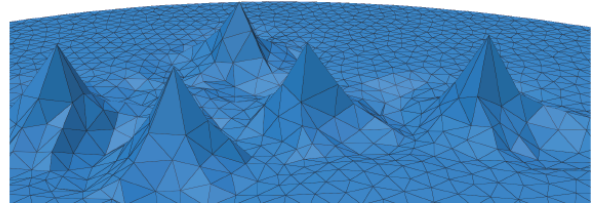
Step 1: The user freely lifts five vertices on the mesh. We observe that this operation causes several small dihedral angles (some examples are marked by yellow).



Step 2: The user clicks the button to project the angle constraint (with target angle as 0, a.k.a. flat surface). This operation try to bend the surface around effected angle in step 1. So it turns out to be digging the ground around these cones.



Step 3: Further projection leads to the same effects.



Step 4: After the final step, the series of projections leads to an outcome looks like a range of mountain.

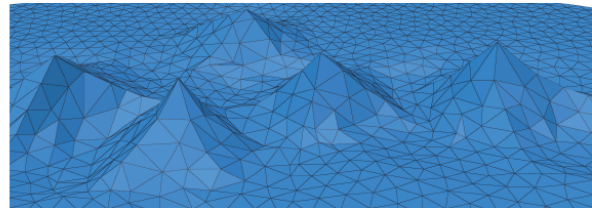


Figure 5.7: Experiment result: Angle Constraints

5.2.3 Cross interaction between different constraints

This subsection focuses on the potential of combining different kinds of constraints. Theoretically there should be many possibilities to conduct the experiment. We only raise the most interesting example here.

The most interesting example is made by first enforcing little-angle constraints followed by enforcing small length constraints. As has been mentioned above, little angle constraints make the projection highly unstable, so the surface grows in an arbitrary and unpredictable way. After the surface "exploded", enforcing small length constraints brings back the shape to its original state. But it turns out to be such operation would not results to exactly the same result before the explosion, some deformation would still be introduced. The partially recovered outcome is shown as another organic form.

Because the explosion is random, every trial always leads to different results. The user could also introduce texture by methods mentioned above.

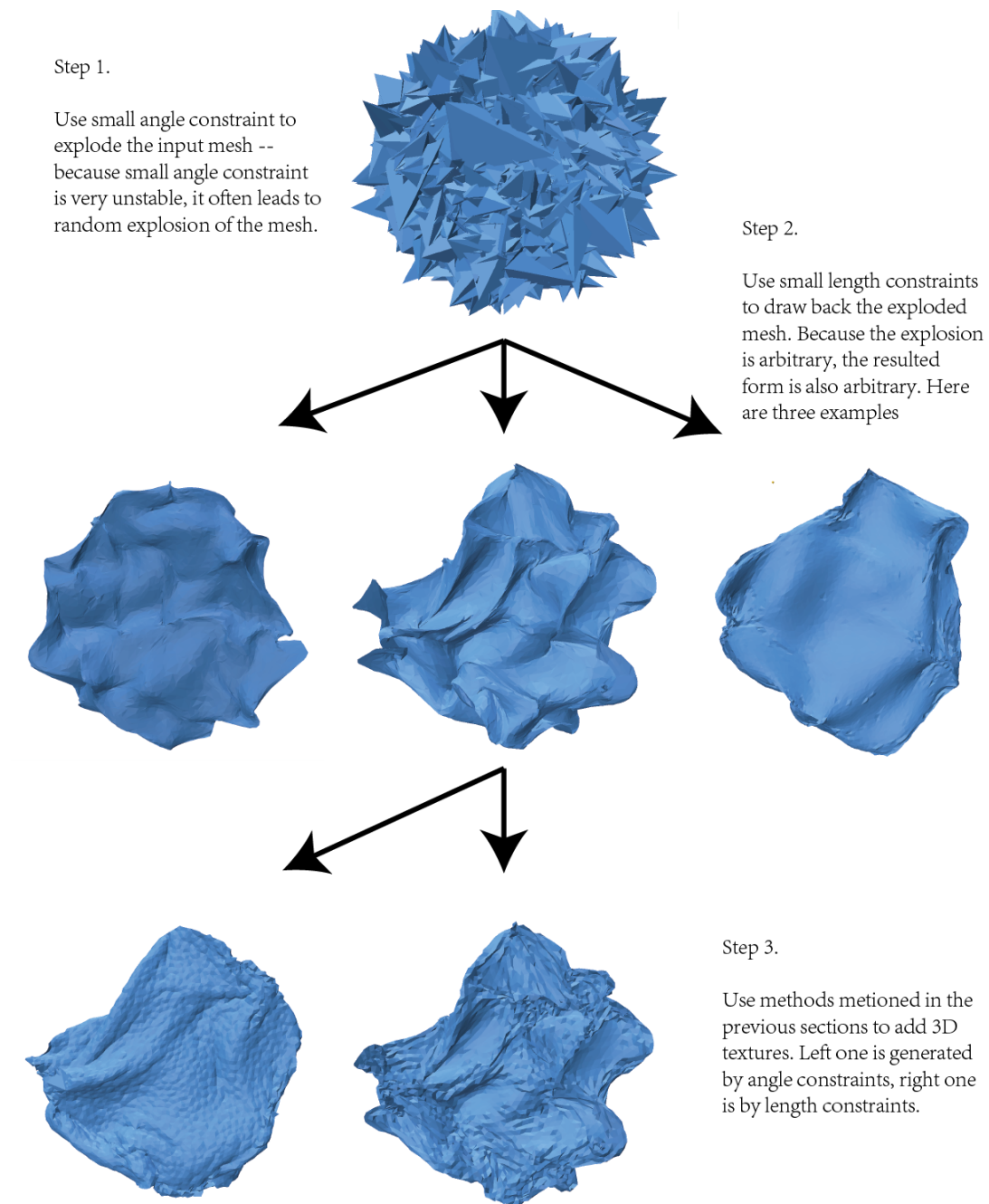


Figure 5.8: Experiment result 1

The above cases reveal the possibility of free form design enabled by the novel design pipeline proposed in this thesis.

5.3 User Experiment

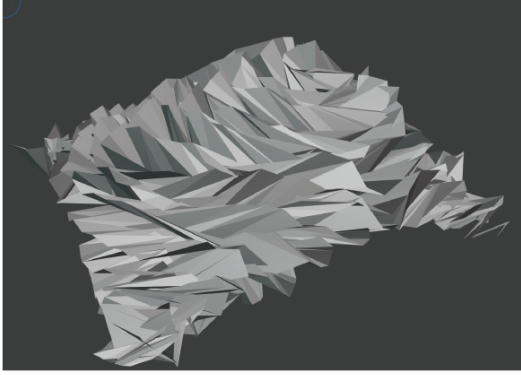
The purpose of doing this user experiment is to show that the new pipeline allows users to communicate and interact with constraints, that is to say, there is an active feedback loop between users and constraints. The outline of the experiment is having subjects handle a design task using the constraint-based design platform mentioned in the previous chapter. There is only one subject who is a graduate student at the school of Architecture. She can use the platform to design any free forms she want in limited time.

5.3.1 Tester Information and tutorial

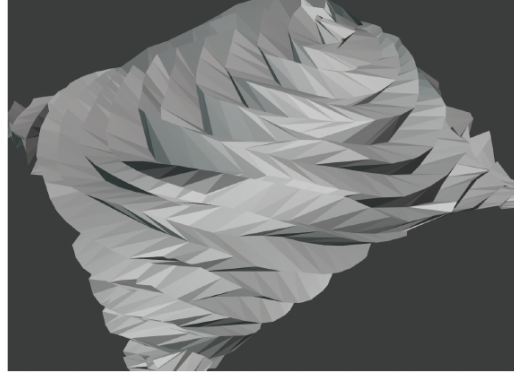
The first part of user experiment is giving limited information about constraint projection. Because in practical situation, designers typically do not possess professional knowledge about constraints. Therefore the tutorial is

empirical. The subject was taught how to use the interface to activate the algorithm and superficial information about constraints. The observer demonstrated a loop of projection without free manipulation. In the demonstration, there was a square in the beginning. The observer "explode" the surface first using large length constraints and assigning a much smaller angle parameter and keep iterating with different target length to reach the final outcome.

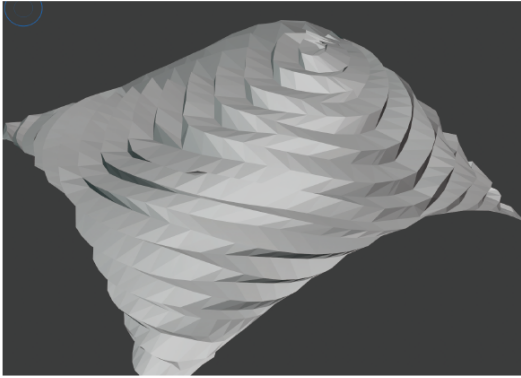
1. Using large length constraints to perturb the shape



2. Using smaller length constraints to smooth the result



3. Fix the target length to preserve some texture



3. One more iteration with the same constraint.

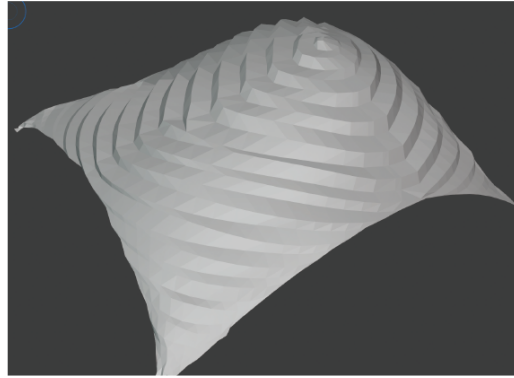


Figure 5.9: Demonstration of constraint projection to the subject

5.3.2 Experiment and observation

There are no constraints for the subject during the experiment. The subject can freely use any mesh editing tools provided in blender. The time limit is 20 minutes. The subject was also given a square at the beginning. The

shape need not consider any practical usage. The subject could take any design situation she would like, for example, facade design.

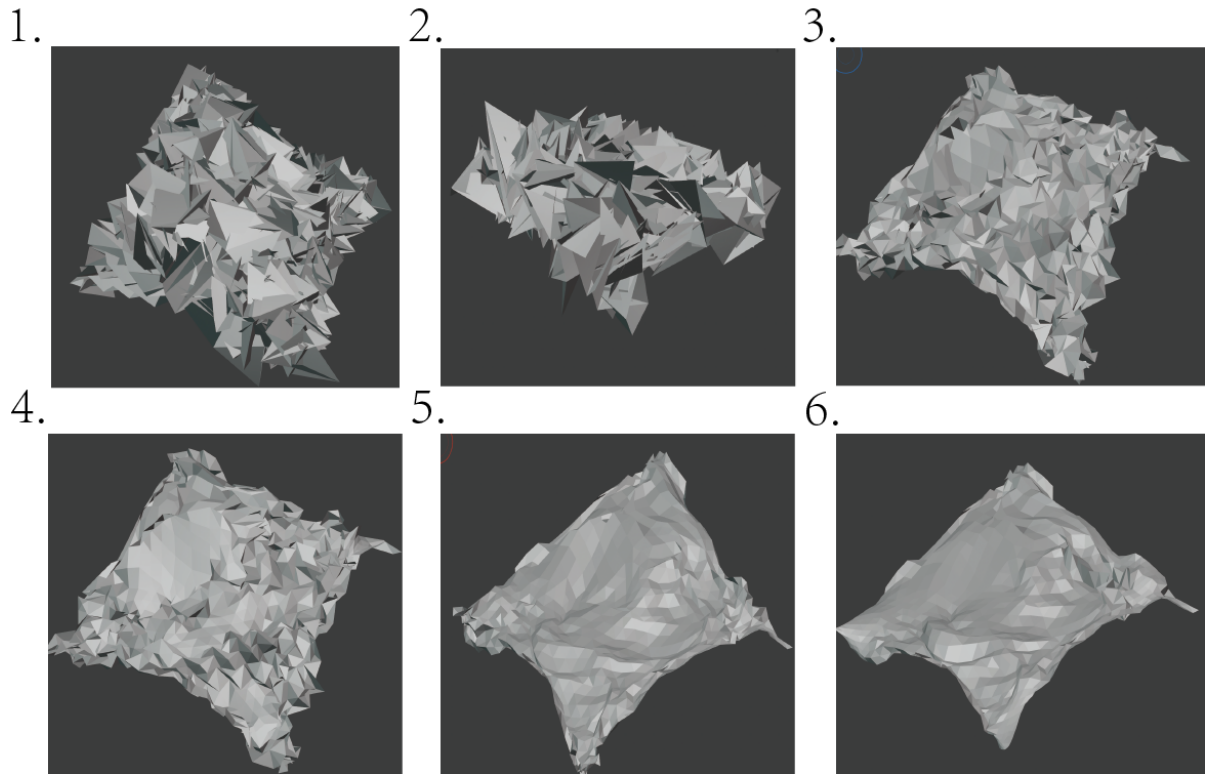
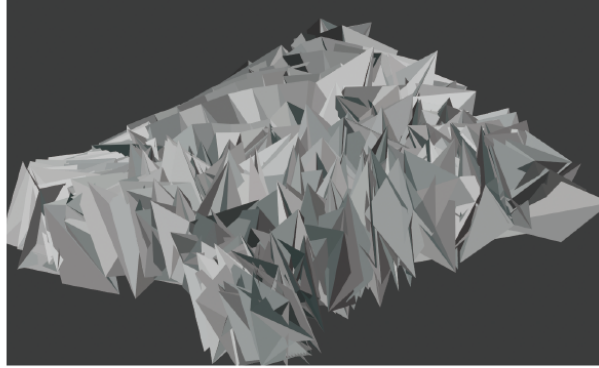


Figure 5.10: The first round of trial

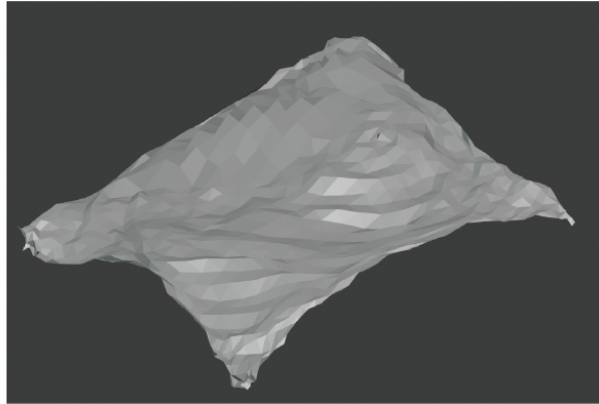
In the first loop, the subject was mostly learning the properties of the algorithm. At stage(1), she exploded the beginning shape using large length constraints as the observer. At stage(2), she manually flattened the shape. At stage(3), she kept smoothing the shape using the algorithm. And at

stage(4), she tried to manually modify and strengthen some unpredictably generated shape details by existing tools in blender. But in stage(5) and stage(6), she realized the following iterations are likely to erase her manual work. Therefore, she decided to end this loop and exploded the shape again using large length constraints.

1.



2.



3.



Figure 5.11: The second round of trial

In the second loop, the subject began again with an exploded shape. In this round she was more fluent and aware of several factors that may effect the final result. At stage(2), she decided to immensely change the shape to reach stage(3). Based on stage(3), she tried several target length to reach the final outcome.

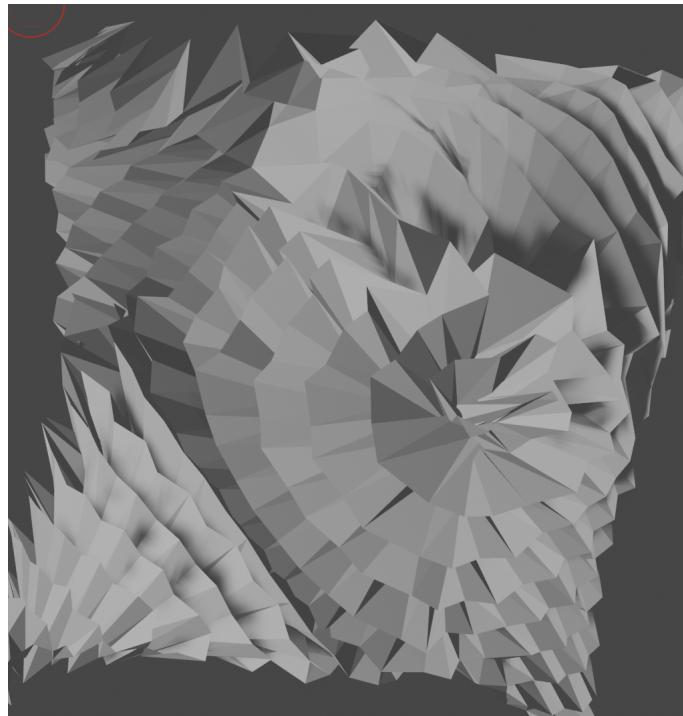


Figure 5.12: The final outcome

We could clearly observe that there is a salient feedback loop between the subject and constraints. As a general generative tool, it's impossible to

predict the outcome with high precision. But that is also the key that leads to design exploration. The subject learned from the feedback and changed her own plan. In this new pipeline, constraints are no longer tacit. There is an efficient communication.

Chapter 6

Discussion

6.1 Contribution

The thesis introduces a design pipeline of free-form editing where physical constraints are explicitly and actively enforced step-by-step. The pipeline provides abundant form-diversity and supports communication between users and constraints at lower level.

In one aspect, a single step of constraint projection/optimization may serve as a multi-functional geometry processing tool. Based only on the exploration within this thesis, it can be applied to generating 3-D texture

and organic forms on an existing mesh. From the perspective of functionality, The thesis aims not at showing that the current pipeline possesses more form-diversity than conventional constraint-based generative tools, but rather at showing its own ability independently.

The advantage of this pipeline lies more at supporting communication between users and constraint-based design tool. It shows the possibility that geometry constraints need not necessarily constrain the users.

Compared with traditional implicit method used in forward design, this new pipeline let users explicitly manually project the constraints when they think it's needed instead of letting the machine handle the projection at every frame.

Compared with traditional optimization method for inverse design, this pipeline allows the user to participate in the process of optimization and edit the object freely in the intermediate stage. Therefore, users might gain better control with the new pipeline.

6.2 Limitation

6.2.1 Technical limitation

Current technical methods cannot explore complex and advanced constraint projections. Because the projection methods applied in this thesis is based on position-based dynamics and fast projection which are known for efficiency on handling simple constraints like angle, area, and length. But these methods are also limited in handle these situations, for complex constraints like developability, often more mature and deep techniques are need. It remains unknown if the argument can be generalize to these situation.

For forward design, this should not be a serious issue because people typically only work with simple constraints. But for inverse design, people usually apply much more complicated optimization methods than what has been explored in the thesis. Therefore, the comparison between conventional inverse design and introduced pipeline based on the current experiments is somewhat trivial. More test cases are necessarily needed if this pipeline shall be practically applied to inverse design.

6.2.2 Design limitation

Although the user can directly see the constraint feedback given by the machine rather than being ignorant of or even not aware of it, it does not guarantee that the interaction commutes information in an effective way.

Users cannot predict the outcome globally even if locally the geometric constraints are intuitive. In the experiment, the user have to use enumeration to find the parameter and suitable operation to reach the result she wanted.

There should be another layer of interaction interface between the user and current pipeline that informs the users about its functionality. The raw pipeline cannot be directly placed as a completed tool but rather a bottom mechanism.

6.3 Future Direction

User interface

How to use the pipeline to truly benefit the users? A carefully designed user interface that wraps the pipeline and provides more effective and intuitive information is needed.

More user research

To be able to design a suitable interface. More user researches should be done to precisely locate the pain-point of the current interaction provided by the pipeline.

Other Projectors

Studying the difference of applying different projectors might be interesting. There are other types of projectors. For example, it remains unknown what if fast projection is applied to the 2D case.

Also, complicated optimization can be considered. As stated earlier, in-

verse design often requires maturer techniques than what was used in the thesis. It remains unknown if these techniques could also produce interesting results.

The deformation mechanism

The mechanism that relates these organic forms and constraint projection remains unknown. It might be hard to classify and explore, but it will be useful to help design the interface.

Bibliography

- Sutherland, I. E. (1963). Sketchpad: A man-machine graphical communication system. *Proceedings of the May 21-23, 1963, spring joint computer conference*, 329–346. <https://doi.org/10.1145/1461551.1461591>
- Goldschmidt, G. (1991). The dialectics of sketching [Publisher: Routledge _eprint: <https://doi.org/10.1080/10400419109534381>]. *Creativity Research Journal*, 4(2), 123–143. <https://doi.org/10.1080/10400419109534381>
- Schon, D. A., & Wiggins, G. (1992). Kinds of seeing and their functions in designing. *Design Studies*, 13(2), 135–156. [https://doi.org/10.1016/0142-694X\(92\)90268-F](https://doi.org/10.1016/0142-694X(92)90268-F)
- de Béjar, L., Robinson, P., & Avent, R. (1993). Integrating mathematical CAD and artificial simulation of plastic rotation in heat-straightening processes. *Applied Mathematical Modelling*, 17(1), 34–40. [https://doi.org/10.1016/0307-904X\(93\)90125-Z](https://doi.org/10.1016/0307-904X(93)90125-Z)

- Goel, V. (1995). *Sketches of Thought* [Google-Books-ID: TzM9ltY8wWIC]. MIT Press.
- Purcell, A. T., & Gero, J. S. (1998). Drawings and the design process: A review of protocol studies in design and other disciplines and related research in cognitive psychology. *Design Studies*, 19(4), 389–430. [https://doi.org/10.1016/S0142-694X\(98\)00015-5](https://doi.org/10.1016/S0142-694X(98)00015-5)
- Won, P. .-H. (2001). The comparison between visual thinking using computer and conventional media in the concept generation stages of design. *Automation in Construction*, 10(3), 319–325. [https://doi.org/10.1016/S0926-5805\(00\)00048-0](https://doi.org/10.1016/S0926-5805(00)00048-0)
- Bilda, Z., & Demirkan, H. (2003). An insight on designers' sketching activities in traditional versus digital media. *Design Studies*, 24(1), 27–50. [https://doi.org/10.1016/S0142-694X\(02\)00032-7](https://doi.org/10.1016/S0142-694X(02)00032-7)
- Cheutet, V., Catalano, C. E., Pernot, J. P., Falcidieno, B., Giannini, F., & Leon, J. C. (2005). 3D sketching for aesthetic design using fully free-form deformation features. *Computers & Graphics*, 29(6), 916–930. <https://doi.org/10.1016/j.cag.2005.09.009>

- Jonson, B. (2005). Design ideation: The conceptual sketch in the digital age. *Design Studies*, 26(6), 613–624. <https://doi.org/10.1016/j.destud.2005.03.001>
- Séquin, C. H. (2005). CAD tools for aesthetic engineering. *Computer-Aided Design*, 37(7), 737–750. <https://doi.org/10.1016/j.cad.2004.08.011>
- Goldenthal, R., Harmon, D., Fattal, R., Bercovier, M., & Grinspun, E. (2007). Efficient simulation of inextensible cloth. *ACM SIGGRAPH 2007 Papers*, 49–es. <https://doi.org/10.1145/1275808.1276438>
- Müller, M., Heidelberger, B., Hennix, M., & Ratcliff, J. (2007). Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2), 109–118. <https://doi.org/https://doi.org/10.1016/j.jvcir.2007.01.005>
- Asawa, T., Hoyano, A., & Nakaohkubo, K. (2008). Thermal design tool for outdoor spaces based on heat balance simulation using a 3D-CAD system. *Building and Environment*, 43(12), 2112–2123. <https://doi.org/10.1016/j.buildenv.2007.12.007>
- Bergou, M., Wardetzky, M., Robinson, S., Audoly, B., & Grinspun, E. (2008). Discrete elastic rods. *ACM Trans. Graph.*, 27(3), 1–12. <https://doi.org/10.1145/1360612.1360662>

- Alcaide-Marzal, J., Diego-Más, J. A., Asensio-Cuesta, S., & Piqueras-Fiszman, B. (2013). An exploratory study on the use of digital sculpting in conceptual product design. *Design Studies*, 34(2), 264–284. <https://doi.org/10.1016/j.destud.2012.09.001>
- Lee, S., & Yan, J. (2016). The impact of 3D CAD interfaces on user ideation: A comparative analysis using SketchUp and Silhouette Modeler. *Design Studies*, 44, 52–73. <https://doi.org/10.1016/j.destud.2016.02.001>
- Pharr, M., Jakob, W., & Humphreys, G. (2016). *Physically Based Rendering: From Theory to Implementation* (3rd edition). Morgan Kaufmann.
- De Goes, F., & James, D. L. (2017). Regularized kelinlets: Sculpting brushes based on fundamental solutions of elasticity. *ACM Trans. Graph.*, 36(4). <https://doi.org/10.1145/3072959.3073595>
- Zhu, B., Skouras, M., Chen, D., & Matusik, W. (2017). Two-scale topology optimization with microstructures. *ACM transactions on graphics*, 36(5), 1–16.
- Community, B. O. (2018). *Blender - a 3d modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam. <http://www.blender.org>

- De Goes, F., & James, D. L. (2018). Dynamic kelinlets: Secondary motions based on fundamental solutions of elastodynamics. *ACM Trans. Graph.*, 37(4). <https://doi.org/10.1145/3197517.3201280>
- Rabinovich, M., Hoffmann, T., & Sorkine-Hornung, O. (2018). Discrete geodesic nets for modeling developable surfaces. *ACM Trans. Graph.*, 37(2). <https://doi.org/10.1145/3180494>
- Stein, O., Grinspun, E., & Crane, K. (2018). Developability of triangle meshes. *ACM Trans. Graph.*, 37(4).
- de Goes, F., & James, D. L. (2019). Sharp kelinlets: Elastic deformations with cusps and localized falloffs. *Proceedings of the 2019 Digital Production Symposium*. <https://doi.org/10.1145/3329715.3338884>
- Matusik, W., & Schulz, A. (2019). Computational fabrication. *ACM SIGGRAPH 2019 Courses*. <https://doi.org/10.1145/3305366.3328064>
- Sharp, N. et al. (2019). Polyscope [www.polyscope.run].
- Sharp, N., Crane, K. et al. (2019). Geometry-central [www.geometry-central.net].
- Galbraith, M. (n.d.). Cadenary string modeller. Retrieved December 1, 2020, from <http://www.axelkilian.com/projectpages/cadenary.html>
- Zbrush. (n.d.). ZBrush 2021. Retrieved December 1, 2020, from <http://www.pixologic.com/features/>