Computational Laser Forming Origami of Convex Surfaces

Yue Hao  
George Mason University  
Fairfax, VA  
yhao3@gmu.edu

Jyh-Ming Lien  
George Mason University  
Fairfax, VA  
jmlien@cs.gmu.edu

ABSTRACT
Laser forming is a fabrication method that uses laser to fold sheets into 3D structures. To overcome the limitations in the traditional practice that relies on tedious manual design, this paper advances laser forming by developing computational methods that procedurally convert a polyhedron $P$ into laser formable 2D patterns and folding instructions or report that $P$ is not laser formable. Due to the limitation of the low-cost laser cutter considered in this paper, we will focus our discussion on laser forming convex surfaces. A 3D surface is called convex if the entire surface lies on the boundary of its convex hull. Our theoretical analysis shows that, even for convex surfaces, the laser formability can be expensive to determine. We then present a framework that efficiently computes patterns and motion instructions for laser forming convex surfaces. An end-to-end laser forming pipeline is presented with several fabrication results to demonstrate the capability and current limitations of the software and hardware framework.

CCS CONCEPTS
- Computing methodologies → Simulation evaluation;  
- Applied computing → Computer-aided manufacturing;  
- Theory of computation → Computational geometry;

KEYWORDS
Polyhedral nets, laser forming, laser origami, optimization, motion planning, personal fabrication

Figure 1: Laser forming 3D sinuous antenna with 70 faces. The substrate is 75 um thick stainless steel.
et al. 2013]. The current practice for designing these foldable structures often involves two decoupled steps: generating an unfolding from a 3D shape, and then finding collision-free folding motion that brings the 2D unfolding back to the 3D shape. These decoupled steps are iterated until a foldable unfolding is found. This raises a foldability problem, namely that there is no guarantee that continuous motion can be found in the latter step, given a net generated in the former step. The requirement to maintain visibility in laser forming exacerbates this design issue, as not only collisions within the structure should be avoided, collisions between the structure and the laser ray must also be avoided during folding, except at the crease lines. These constraints make the design process extremely challenging for engineers.

1.1 Challenges
In this paper we explore the computational aspects of laser forming: given a target 3D shape, how should one design a 2D pattern that can be rigidly folded into the desired 3D shape without stretching or self-collision by following a folding motion actuated by the laser? We face a challenge that, similar to most self-foldable structures, laser forming is only capable of folding simple structures with few crease lines. Thus, forming shapes with tens or even hundreds of crease lines, such as the antenna shown in Figure 1, has been prohibitively tedious. This is particularly true for low-cost laser cutters that we will consider in this work.

Theoretically, it remains unclear what type of structures can be laser formed and whether or not there are efficient algorithms that can produce laser-formable structures. In addition to the aforementioned geometric constraints from the structure itself and the line-of-sight from laser to the structure, there are additional mechanical and thermal constraints that we will have to consider computationally. One of the mechanical constraints is the degree of freedom of the laser head(s) and that of the cutting bed. Obviously, when the laser heads have higher degrees of freedom or there exist multiple heads, more complex structures can be laser formed. Unfortunately, the cost of these dexterous systems is prohibitively high for personal fabrication.

1.2 Our Contributions
The work presented in this paper takes the first step to procedurally generate laser-formable patterns as well as folding instructions that guide a low-cost laser cutter to cut and fold a sheet into a complex 3D shape.

The proposed method is universal as it takes an arbitrary convex polyhedral surface \( P \) as input and reports whether or not \( P \) is laser formable. A polyhedral surface \( P \) is convex if \( P \) lies on the boundary of \( P \)'s convex hull [Chazelle et al. 1995]. The polyhedral surface of the antenna in Figure 1 is convex. In the case that the input is laser-formable, the 2D pattern and folding instructions will be produced. For example, Figures 2 and 3 show two simulated folding sequences produced by our method and the former is used to fabricate the antenna in Figure 1. It is important to note that the instructions created by the proposed method enable the user to laser form structures that could not be formed in the past due to the complexity of the structure and the complexity of the constraints during the folding process.

Our theoretical analysis shows that, even for convex surfaces, the laser formability can be expensive to determine. We then present a framework that efficiently computes patterns and motion instructions for laser forming convex surfaces. An end-to-end laser-forming pipeline is presented with several fabrication results to demonstrate the capability and current limitations of the software and hardware framework. We hope that, as g-code, slicers and pathing software played their essential roles in 3D printing, the computational methods and software tools developed in this work can eventually transform laser forming and laser origami to practical methods for low-cost customizable fabrication.

2 BACKGROUND
This section provides a brief literature review in laser forming, computational design and structure optimization of foldable structures. We also provide necessary definitions of terms and concepts that will be used throughout the rest of the paper.

2.1 Laser Origami and Laser Forming
Laser origami is a general term for using a laser source to fold 2D sheets into 3D structures, and includes several different folding mechanisms including polymer softening [Mueller et al. 2013], release of pre-stressed films [Pique et al. 2012], as well as laser forming, the technique used here.

Laser forming is an established metal working technology that evolved in the 1980s [Shen and Vollertsen 2009] from "flame bending", a technique widely used in construction and shipbuilding where a metal sheet is heated with an acetylene torch for straightening [Cheng et al. 2005]. In laser forming, a laser is used to selectively heat the workpiece to create plastic thermal stresses for controlled deformation or folding [Paramasivan et al. 2018].

Several distinct modes of laser forming are possible, with the most important for origami known as the temperature gradient mechanism (TGM) and buckling mechanism (BM). In TGM bending, the laser is rapidly scanned across the surface, creating a vertical temperature gradient through the metal thickness. Initially the top expands relative to the bottom ("counterbending") [Madsen and Sondergaard 2014]; however, the cooler surroundings push back against this expansion, resulting in compressive plastic strains. When the workpiece is subsequently allowed to cool, the initial expansion disappears but the compressive plastic strains remain in the material, resulting in bending back upward toward the laser source [Shen and Vollertsen 2009].

For the buckling mechanism, the laser is scanned more slowly to allow heat to reach all the way through the thickness of the workpiece. The heated region again expands, generating compressive plastic stresses as before, but this time the compressive stresses are against the full thickness of the heated plate. Compressive stresses in a thin membrane result in buckling of the heated region [Edwardson 2004], and as the laser is scanned the buckling results in the folding of the workpiece. Unlike the TGM mode, the buckling mechanism can cause both upward or downward folding, and it is necessary to control the direction through an additional force such as a pre-strain in the material [Lazarus and Smith 2017].

Recently, it was demonstrated that combining laser cutting and laser forming can be used to create complex structures with both up
and down folds from a simple sheet of unpatterned metal [Lazarus and Smith 2017]. The TGM mode was used for upward folding, while the buckling mechanism is used for downward folding. In theory, very complicated structures are possible using this technology or other related laser origami; however, in practice most folding in laser origami has been limited to simple designs and manual placement of folds.

2.2 Computational Methods for Origami and Foldable Structure Design

The main purpose of this paper is to bridge the gap between the laser forming technique and the computational design of fabricable origami patterns. The critical issue for laser forming users is that the foldable 2D pattern must be cut out from a single layer of 2D sheet, which means there must be no overlapping components in the design and the entire shape must remain in one piece. The non-overlapping pattern is extremely difficult for a human to conceptualize and design manually, especially for non-convex 3D sheets, as it is not obvious to resolve the overlapping components without breaking the pattern apart. Fortunately, the 3D to 2D unfolding problem has been extensively studied in the area of computational origami. [Tachi 2010] proposed an algorithm to unfolding any 3D shape to a crease pattern. Such design can reliably approximate complex 3D shapes provided large enough 2D spaces. However, this method requires adding tuck faces that do not exist for the geometry of the original shape, often resulting in much larger surface area than necessary, especially for non-convex shapes to accommodate the non-overlapping constraint.

The kirigami design, often studies in the form of polyhedral nets, effectively solved this problem. Kirigami allows cutting in the 2D pattern, making the unfolding much easier. Heuristic methods [Prautzsch and Straub 2005; Schlickerrieder 1997] have been developed to unfold convex polyhedra by designing edge weights so that the resulting minimum spanning tree is likely to be a net. Finding nets of non-convex shapes is significantly more challenging. Often segmentation is needed to avoid overlapping [Prautzsch and Straub 2005]. To avoid over-segmentation, genetic-based optimization methods [Takahashi et al. 2011; Xi et al. 2016] evolve the unfoldings by mutating the cut edges until a net with zero overlapping is found.

2.3 Foldable Structure Optimization

Planning the continuous folding motion of a given net to bring it back to the 3D shape is nontrivial. Early research focused mostly on folding origami patterns [Demaine et al. 2004; Tachi 2009; Xi and Lien 2014, 2015b] that usually have many faces but low Degrees of Freedom (DoF) due to the closure constraints. On the other hand, polyhedral nets have high DoF. The motion of these nets is typically found by a motion planner that models the net as a tree-like articulated robot. For example, Song and Amato [Song and Amato 2004] proposed a PRM-based planner for nets generated from non-convex polyhedra. Recently, another motion planner has been developed by Xi and Lien [2015a] that focuses on finding implicit folding order of a net. This approach is shown to be more effective, especially for non-convex models. However, depending on the polyhedral nets, the planning may require long computation time even for simple shapes. For large nets, it can take hours to find a path and ends up with complex motions that are not desirable for physical realization. Until recently, Hao et al. [2018b] discovered that the geometry of the net has a substantial impact on the foldability of nets, the design of the net and the optimization of the continuous foldability should not be decoupled. They developed an optimization framework to optimized the foldability by generating more foldable nets.

3 PRELIMINARIES

This section defines the computational models, concepts and constraints used in this paper.

3.1 Laser Models

Let $L$ be a laser cutter with a single laser head that can translate on a 2D plane $z = 0$ and can only emit laser vertically downwards and focus on the $z = 0$ plane. The translational motion of $L$ as well as the size of the foldable sheet are bounded by the working
The problem of laser forming asks, given a (convex) polyhedral net, to fold it into a three-dimensional object. It has the two degrees of freedom (DOF) for translation and another DOF determines the strength of the laser. Furthermore, the motion of the laser head can only move along the edges of the polyhedral net and the strength of the laser is also predetermined by the types of edges: mountain or valley creases or cut lines, and the material of the sheet. Because mountain fold is known to be more difficult to produce reliably than valley folds [Lazarus and Smith 2017], we will also consider another model \( L^- \) that can only fold valley creases, i.e., fold the faces upward. In addition, our discussion on laser formability will also focus on convex surfaces due to the limitations of \( L \) and \( L^- \), even though the proposed method is general enough that it can be used to create laser formable nets of non-convex surfaces as shown in Figure 16 that can only be fabricated with two laser heads, one above and one below.

### 3.2 Representing the Foldable Structures

There are two main types of foldable structures: origami crease patterns or polyhedral net. Figure 5 compares these two representations using a simple 3D model with a single interior vertex. The origami crease pattern can be generated by adding tuck faces [Peraza Hernandez et al. 2017; Tachi 2010] between the triangle faces in the original mesh or so that the 3D mesh can be flattened while remaining a single connected component. This representation has two major issues in laser forming. First, the tuck neighboring faces need to be folded 180 degrees. This is not achievable due to visibility constraints. In addition, during folding, the rigid origami is subject to closure constraints, meaning that all creases in our simple example must be folded together. For models like the single vertex, the laser needs to focus on slopes with \( z > 0 \) as the creases folded in 3D. For polyhedral nets, creases can be folded independently and in a sequence. An example of the polyhedral net generated from a given 3D mesh is shown in Figure 6. Each pattern must have a base face that will remain attached to the sheet during the cutting and folding process.

### 3.3 Laser Forming

The problem of laser forming asks, given a (convex) polyhedral mesh \( P \) and a laser cutter \( L \) (or \( L^- \)), how \( P \) can be unfolded into a polyhedral net \( N \) so that \( L \) can cut and fold \( N \) from a sheet placed at \( z = 0 \) into \( P \).

The folding process must satisfy the following geometric constraints: (1) the structure cannot overlap with itself and (2) the crease lines that are not yet folded must maintain vertical visibility with the laser head. The first constraint is well studied, as mentioned in the previous section, motion planning methods can be used to find collision-free folding motion of a given polyhedral net. Moreover, optimization methods [Hao et al. 2018a,b] have also been proposed to create structures that can be continuously folded without collision, in particular for convex polyhedra [Demaine et al. 2011]. The second constraint, however, has not yet been studied in the literature. The picture on the right of Figure 4 provides an example of how the existing folded structure might block the visibility between a crease line (the dashed line) and the laser. Addressing this constraint will be one of the main focuses of this paper and will be discussed in Section 4. Note that one can easily circumvent the visibility constraints by using a multi-DoF robot arm to hold the laser head and shoot the laser beam in any direction. However, building such a robotic system with high dexterity and accuracy, while, in some cases, still small enough to operate inside the folded objects is non-trivial, and far from commercially viable.

In addition to the geometric constraints described above, because laser forming uses thermal plastic stresses to bend a crease, how the heat is distributed across the sheet is critical. In most cases, during the cut and fold process, our algorithm must consider thermal paths for sufficient heat transfer from the cut and fold edges to the rest of the sheet. Alternatively, we should also allow time delay between cuts and folds to ensure that the folded structure and the sheet are free of undesired deformation (e.g., bulging). For example, it is likely that if we cut the entire boundary of the crease pattern...
in Figure 6, except the gaps on the base face, the structure might under or over deformed because the heat build up near the creases has few paths to escape. One of the solutions used in this work, as shown in Figure 7 is to cut only the needed portion of the boundary before the next fold. Details of the thermal aspect of the fabrication will be discussed in Section 5.1.

Figure 7: Instead of cutting the entire boundary before folding, we iteratively cut and fold to ensure the stability of the structure and, more importantly, to ensure that heat can escape from the structure.

4 LASER FORMABLE STRUCTURES

There are two main steps in designing laser formable structures: (1) unfolding a polyhedron $P$ into a polyhedral net $N$ and (2) motion planning of the laser head to cut and fold $N$. These two steps are interdependent and cannot be solved separately. Therefore, an essential part of computational laser forming is to simultaneously solve mesh unfolding problem and path planning problem, i.e. finding a path for the laser to follow. In the first part of this section, we will provide theoretical analysis and a practical algorithm to determine the formability of convex surfaces using either $L$ or $L^\perp$ laser. The second part of this section presents optimization methods that ensure collision-free and more efficient folding motion.

4.1 Visibility Constraints

To incorporate the visibility constraints in our folding motion planning, we need to check the intersection between the laser beam and the folding pattern. Shown in Figure 8, we model the laser beam as a surface instead of a line segment. This is because we want our computational designing algorithms and the formality analysis to be universal, and independent of the specifications of the hardware and material properties. At this point, we do not need to know the exact path that the laser will follow and how fast the laser passes are. We assume the laser speed is infinitely fast, each laser pass will sweep a triangular surface. The laser head is infinitely far away and shoots straight down to the work plane. In practice, we also add thickness to the triangular surface to accommodate the potential calibration and model errors. While cutting a portion of the boundary, the folding pauses until the portion is entirely cut. Shown in Figure 8a, the laser sweeping forms a pyramid (the apex is infinitely far away) standing on the portion of the boundary of the folding pattern, and any other parts of the the pattern must not intersect with the pyramid.

Figure 8: Swept surfaces used for line-of-sight check between the laser the crease and cut lines.

4.2 $L^\perp$-formability

Before we present a more general motion-planning-based folding method, we present a more limited method but with theorectic guarantee. Given a net $N$, there can be many tree assignments, in fact $n!$ of them where $n$ is the number of faces in $N$. In each tree assignment, a root is selected and the child-parent relationship of all connected nodes is uniquely defined. Let $T$ be one of such tree assignment.

Because of the limitation of the laser cutter $L$ defined in Section 3, we have the constraint that a node of $T$ can only be folded if all of its descendants are already folded. (When the laser is folding an edge, we call the face that is rotating about the edge the folding face and the other face remains on the plane $z = 0$ the fixed face.) One of the main reason for this constraint is the focal point of the laser must be on the flat surface of the substrate in $R^2$. However, even if the laser can focus on any point in $R^2$ and fold the crease line in any direction, the folding of the material becomes much more challenging to characterize (due to the uncertainty of the structural configuration and the orientation of the incoming laser on the edge). If the node is folded before its child, the child is no longer on the plane and becomes more difficult to fold. There is a possibility that we can find an ancestor of the node in $T$ that can be folded and then bring the child node back of the plane $z = 0$. This can be achieved via linearization of a DAG that encodes the dependency of all the faces so that the source of an edge must be folded before the target of the edge. The linearization defines the folding order of the faces in $N$. Given a tree $T$, there can be many linearizations of $T$. It is clear that if one linearization can be laser formable, then all other linearizations can too.

Let us first consider an even more restricted model of $L$ called $L^\perp$ that can only fold the crease lines upward. Clearly, $L^\perp$ cannot fold any crease with the dihedral angle less than $\pi/2$ because the face itself will block the visibility, such as a regular tetrahedron. Therefore, if a net has none of such crease, then the net might be laser formable. If the net has one of such crease, then the faces adjacent to this crease must be folded last. If the net has more than one sharp creases, then the net is not $L^\perp$ formable. It should be noted that, even if a polyhedron has more than one sharp edges,
the polyhedron may still be $L^{-}$ formable if the sharp edges are cut edges in its net.

However, this is a necessary condition, not a sufficient condition. For each crease line $e$, we must check if the laser visibility of $e$ is occluded by folding all of its descendants in $T$. If the visibility is blocked, the net is not $L^{-}$ formable. This check can take $O(n^2)$ time for a net with $n$ faces. This means that, a brute-force method that enumerates all possible tree assignments and checks their formability will take $O(n^3)$ time.

One way to improve the time efficiency of this method is to avoid checking all possible tree assignments. For example, it is reasonable to pick the topological center of the net $N$ as the root of $T$. In this case, the root is the face that has its longest path to a leaf minimized. Consequently, this tree is the shallowest among all trees and this is a good candidate for laser forming because the number of descendants of a given face is minimized and the chance of possible occlusion by its folded descendants are also minimized. Based on this observation, let us now define a simple folding strategy, called “Outside-In folding” sketched in Algorithm 1.

```plaintext
function TreeAssignment (N)
    Data: A net N
    Result: A tree assignment T

    Let Q be all leaves of N;
    Let R = ∅ be a set of potential roots of N;
    while Q is not empty do
        Let f = Q.pop();
        Let p be the only face connected to f;
        if folding f blocks p then
            f = R ∪ f ;
        else
            Q.push(p);
        end
        remove f from N;
    end
    if R has only one face then
        report T with the face in R as the root;
    else
        if R contains two connected faces then
            report T with the edge incident to the faces in R as the root;
        else
            R has three or more faces or two non-adjacent faces;
            Report that N is not $L^{-}$ foldable.
        end
    end
```

Algorithm 1: Outside-In Folding

4.2.1 Outside-In Folding strategy. We propose a simple folding strategy that folds the net from creases that are near the perimeter of the net gradually into the center of the net. More specifically, because polyhedra nets are tree structures, we can always assign a face $f_r$ as a root of the tree and assign the children-parent relationships using BFS or DFS from the face $f_r$. Then the outside-in folding strategy simply requires the folding of child face into 3D before the parent face can be folded. Consequently the leaves of the net will be folded first and the root face $f_r$ will be folded last (or simply fixed on $z = 0$).

More specifically, let the leaves of $N$ be the faces in $N$ that have only one other face connect to it. Then let $Q$ be a set of all leaves in $N$. Clearly that all faces in $Q$ must be folded before other face in $N$. Therefore, starting from $Q$, the definition of the parent-child relationship can be defined recursively in $N$. Then we propose the algorithm shown in Algorithm 1 that has $O(n^2)$ time complexity.

Note that the structure $Q$ in Algorithm 1 can be either a queue or a stack as it does not matter which face in $Q$ is folded first. This observation also admits parallel laser forming discussed later in this paper. The second part of Algorithm 1 determines the final structure of the tree that is $L^{-}$ formable or reports that such tree does not exist.

With this simple folding strategy, we are interested in knowing the type of structures this strategy can fold, e.g., can all nets of convex polyhedra be laser formed via outside-in folding? Do nets created by the unfolding methods, such as source unfolding [Miller and Pak 2009], star unfolding [Aronov and O’Rourke 1992] or various edge unfolding methods [Schlickrieder 1997] of the convex polyhedra, have different laser formability? To answer these questions, let us first discuss a necessary condition for laser formable structure.

![Figure 9: $L^{-}$ formability of convex surfaces.](image)

4.2.2 Nets that are $L^{-}$ formable. Since $L$ can only reach the sheet and focus on $z = 0$ from directly above. A direct consequence is that, if a net has more than one crease line whose dihedral angle is less than $\pi/2$, then the net is not laser formable by $L$. This is because that one of the face incident to such a crease will occlude.
Thus, the only way to fold such a crease will require both incident faces to fold but make sure that they fold less than $\pi/2$. Since $L$ can only focus on $z = 0$, such a crease must only be folded at the end of the folding process. Therefore, we can only afford to have one crease line with an acute dihedral angle.

Figure 9a shows a cube with its front and back faces removed and its only net on the right. Algorithm 1 will create a tree rooted at an edge (labeled by a thick red line segment) incident to the shaded faces at the center of the net. The figure below the 3D mesh illustrates how the mesh is folded with root on the plane $z = 0$. On the other hand, a simple change to the mesh, by adding a gap to the top face, make the mesh unfoldable by $L^-$ laser. Algorithm 1 also correctly predicts this outcome as the potential root faces (shown as the shaded faces on the right) are disjoint. Nevertheless, as shown in Figure 9b, another similar variation of the cubes in Figure 9a, has multiple nets and three of them are shown. Only the rightmost net among the three nets is $L^-$ formable. This observation indicates the possibility that we might need to exhaust all nets before we can conclude that a structure is not laser formable.

In addition, all the nets of a regular cube (or any box) can be $L^-$ formed. In terms of platonic solids, other than tetrahedron and cube, we also know that octahedron is laser $L^-$ formable.

### 4.3 L-formability

Unlike $L^-$ laser, $L$ laser can fold a crease either up or down. Therefore, even though $L$ is subject to the same visibility and laser-focus constraints, $L$ can fold shapes that are unfoldable by $L^-$, such as a regular tetrahedron and the mesh on the right of Figure 9a. In fact, all the nets shown in Figure 9 are $L$ formable. In addition, it is easy to show that if a net of a convex polytope is continuous blooming [Demaine et al. 2011] (i.e., folding without collision), however, source unfolding introduces many additional edges to ensure continuous blooming to the nets. Comparing to edge unfolding in which no new edges are created, source unfolding, as well as star unfolding, are not suitable for laser forming or many other designs of self-foldable machines and materials.

One significant challenge in edge unfolding is that, not all edge unfoldings of a polyhedron have the same foldability. Some nets can be folded as continuous blooming but other nets require complicated instructions to orchestrate the sequence of folding motion. Recently, Hao et al. [2018b] proposed a learning-based framework to evolve polyhedral nets based on foldability. The main idea is to approximate the foldability of a net using geometric and topological features of the net. Only the nets that have potentially high foldability are tested by a motion planner. A straightforward application in laser forming is to use [Hao et al. 2018b] to create a net with blooming motion and apply Algorithm 1 to check if it violates the visibility constraint.

In addition to utilizing the nets created by [Hao et al. 2018b], the visibility constraint considered in laser formability can be easily incorporated into this framework. This results in a more general way to create laser formable nets than Algorithm 1. Note that this approach is general enough that it can be used to create laser formable nets of non-convex surfaces as shown in Figure 16.

### 4.4 Net Optimization for Laser Formability

The algorithm discussed above ensures that the crease lines to be folded are visible from the laser head, but it does not guarantee that the nets are free from collisions during folding. It is known that nets of convex polyhedron created via source unfolding can have continuous blooming [Demaine et al. 2011] (i.e., folding without collision). However, source unfolding introduces many additional edges to ensure continuous blooming to the nets. Comparing to edge unfolding in which no new edges are created, source unfolding, as well as star unfolding, are not suitable for laser forming or many other designs of self-foldable machines and materials.

In this section, we discuss how to decide the laser path based on the folding motion we have obtained. The laser forming technology fabricates 3D models by ultimately moving a control point to follow a path on a 2D plane. One critical characteristic that distinguishes laser forming from traditional 2D fabrication, such as laser cutting and 2D printing, is that the 2D path is not just a geometry, but a time series motion that is extremely sensitive to timing and scheduling.

**Figure 10:** (a-b) Laser forming 3D sinuous antenna
Shown in Figure 11, the PRM based motion planner [2004; 2015a] minimizes the total distance in the configuration space, folding from the flat state to the folded state, which means multiple hinges will fold at the same time if possible. We only have one laser head, in order to fold multiple creases at the same time we need a scheduling strategy to make it act as if we have multiple laser heads, we call this strategy fake parallelization.

It is not hard to tell from our outside-in folding strategy that any two sibling faces from different branches in the tree structure of polyhedral net can be folded independently at any time, provided they do not cause self-collisions or violate the visibility constraints. According to existing experiments [Lazarus et al. 2018], the success of laser forming is extremely sensitive to thermal conditions, we must allow a certain time for the substrate to dissipate heat before applying another laser pass on the same area. Although with a single laser head, we can plan a sequential folding motion that only folds one crease at a time, it would be much more efficient to fold as many creases in parallel as possible, due to the need of time delay required for sufficient heat transfer. A schedule can be designed so that when a crease is cooling down, the laser head can fold another crease that can be folded independently and does not affect the heat dissipation.

\[
\text{function FakeParallelization}(Q) \\
\text{Data: A folding motion } Q \\
\text{Result: A laser path } P \\
\text{Let } K \text{ be the set of timers for the creases;}
\]

\[
\text{while pattern not folded do} \hspace{1cm} \\
\hspace{2cm}\text{for each crease } i \text{ in the pattern do} \hspace{1cm} \\
\hspace{3cm}\text{if } k^i \leq 0 \text{ then} \hspace{1cm} \\
\hspace{4cm}\text{Let } \delta = f(q^i_{t-1}) - f(q^i_t) \text{ be the number of passes needed to fold the crease from angle } q^i_{t-1} \text{ to } q^i_t \text{ in } Q; \hspace{1cm} \\
\hspace{4cm}\text{Add } \delta \text{ laser passes on crease } i \text{ into } P; \hspace{1cm} \\
\hspace{4cm}\text{Decrease all } k^j \in K \text{ with } j \neq i \text{ by } \delta; \hspace{1cm} \\
\hspace{4cm}k^i = k^i + \delta \tau; \hspace{1cm} \\
\hspace{3cm}\text{end} \hspace{1cm} \\
\hspace{2cm}\text{end} \hspace{1cm} \\
\hspace{1cm}\text{if min}(K) > 0 \text{ then} \hspace{1cm} \\
\hspace{2cm}\text{Decrease all } k \text{ by min}(K); \hspace{1cm} \\
\hspace{1cm}\text{end} \hspace{1cm} \\
\text{end}
\]

Algorithm 2: The Fake Parallelization algorithm. \(f(q)\) is the folding rate, i.e. the number of laser passes needed to fold an angle \(q\), provided by the user.

The laser forming process is considered quasi-static and folding is a cumulative effect of multiple laser passes applied on the same crease. On each laser pass, the crease only folds a small degree. Using the working principle of modern computer CPU as an analogy, for a single core to perform multiple tasks, the CPU executes a small set of operations for one task and quickly switch to another task. In the long run, it acts as if doing multitasking. In our fake parallelization algorithm as shown in Algorithm 2, we give each crease a timer \(k\), starting with the initial value 0. Every time the laser passes through the crease, the timer increases by \(\tau > 0\) based on the laser strength, material and cooling conditions etc. Every time the laser passes through other creases or boundaries, the timer decreases by 1. At any given time \(t\) of the folding motion, we schedule the single-head laser path in a sequence based on the timers, the crease with lowest timer value less or equal to 0 goes first and so on so forth. If all the timers are greater than 0, we pause the laser motion or mark some random lines outside the shape to wait for the heat to dissipate. Using the fake parallelization scheduling strategy, we can fabricate the same model much more efficiently than sequential folding.

5 FABRICATION AND RESULTS

5.1 Hardware and Software Setup

A low-cost (≈$10K) Full Spectrum Laser with 20 W, 1064 nm fiber laser and the substrate of 75 um thick stainless steel are used in all fabrication reported in this paper. We implemented the proposed method in single-threaded C++ and all the designs and optimization are done on a MacBook Pro with a 2.9 GHz Intel Core i7 CPU with 16GB Memory.
### 5.2 Fabrication Process

Two main steps in laser forming process, namely polyhedral unfolding and the laser path planning, are both affected by factors additional to visibility constraints, including material properties, the setup of the laser machine and the geometry of the object to be fabricated. To maximize the reusability of computed structures on different materials and laser cutters, we separate the laser forming problem into two steps. In the first step, we consider only the geometry of the 3D object, i.e., unfolding and planning folding motion for the sheet with only geometric constraints. This planning algorithm is universal and can be used for any laser machines and substrates of various materials properties. In the second step, we modify the path for the laser head based on the hardware specifications and physical conditions.

This design principle leads to an end-to-end laser forming pipeline shown in Figure 12. Steps (a)-(c) are the design phase and steps (d)-(f) are the production phase. In our design process, the input is a convex polyhedral surface and the output is a polyhedral net and a feasible folding motion that satisfies all the constraints discussed in Section 3. The mesh is unfolded and optimized using our proposed algorithms and strategies discussed in Section 4.

In the production phase, the input is the design from the previous phase and hardware related specifications, including the number of laser passes needed for a crease to fold up to a certain angle. We use the parameters reported in [Lazarus and Smith 2017] in our experiment as shown in Figure 13. There are many mature computer-aided design (CAD) software available. However, they are mainly developed for the purpose of cutting and engraving. In order to make the laser motion useful for these software, we need to speak the language that these software speaks, meaning we need to encode all the design information in a 2D format. Shown in step (d) in Figure 12, we write the laser path in SVG format. Each blue crease is a set of overlapping laser passes written in the time sequences. We also use a laser path outside the boundary of the net for the delay time. The laser simply marks a long path if a long delay is needed. We also developed the Laser Origami CAD (LoCaD) to serve as a middleware for our low-cost laser marking machine, based on the API of the software EzCaD developed by the manufacture of the machine as an interactive design software and driver. Shown in Figure 12 (e), LoCaD gives the freedom to the end user to adjust the settings, such as laser power and delay time, based on the substrate properties and the hardware environment without redesigning the nets or replanning the folding motion in the design steps.

### 5.3 Experiments with Various Structures

This section reports the fabrication results obtained from the proposed computation framework. In particular, we study a sinuous antenna with various mesh resolutions and a curved surface that requires precise folding to close up the mesh.

Figure 14 shows the results of fabricating the sinuous antenna using the proposed methods. The top row shows the design of the unfolding pattern using the 3D meshes of sinuous antenna tessellated in various resolutions. The bottom row shows the fabricated model of the corresponding design. This experiment reveals some interesting characteristics of the laser forming. Firstly, it all costs about five minutes to fabricate each model regardless the resolution of tessellation. This is because the total number of laser passes needed are nearly linearly related to the sum of all folding angles from all the creases. For models with the same geometry, low resolution tessellations will have less number of creases, however, each crease will have larger angles to be folded. The number of laser passes to form the sinuous antennas are nearly the same regardless the tessellation. Secondly, high resolution models fold even better, i.e. with higher fidelity to the input mesh, than low resolution models. This is quite the opposite of the many traditional origami inspired fabrication methods that smooth surfaces are often harder to achieve. One of the advantage with laser fabrication is that the motion can be controlled with high accuracy using the fine movements of the laser controller, and the laser beam can be focused on extremely thin and densely aligned creases. Although the laser power is carefully controlled to minimize the damage on the workpiece, the metallic material still loses bit by bit every time.

![Figure 12: An end-to-end fabrication process developed in this work. (a) 3D model of the target structure; (b) mesh unfolding and optimization; (c) simulation and verification; (d) generated laser passes in SVG format; (e) customizing the laser power and time delay settings using our software called LoCaD and deploying the settings to hardware; (f) laser fabrication;](image-url)

![Figure 13: Measurements of laser passes vs folding angle using laser forming setup in [Lazarus and Smith 2017].](image-url)
Figure 14: Laser forming sinuous antennas with various resolutions. The number indicates the number of faces in the mesh. The fabrication time ranges between 4 and 5 minutes for all models. As the number of crease lines increase, the folding angle and number of laser passes decrease for each crease line. Thus the folding time is not significantly affected by the resolution. Substrate: 75 um thick stainless steel. Laser cut settings: 100 mm/s speed, 100% power, 3x loop Laser fold up settings: 30 mm/s speed, 25% power. Delay settings: 10 mm/s speed, 0% power.

Figure 15: Laser forming a convex surface patch. Figure (c) shows that the cuts along the net boundary are in fact rasterized overlapping polygons instead of line segments. Figure (d) shows failed fabrication result without using overlapping rasterization. Figures (e) and (f) show fabrication result with overlapping cut rasterization.

The laser passing through it. This may cause destructive loss of material for large folding angles as more passes are applied on the same creases, especially for short creases connecting fragile parts of the geometries. A solution is to leave long delays between laser passes to reduce the damage for cooling, which sacrifices efficiency to fold the same model. For smooth shapes, larger folding angles are subdivided to smaller folds, which disperses the damage done to the same creases. In summary, instead of challenged by high resolution tessellations and smooth curvatures as many traditional fabrications do, laser forming embraces it.

Our second experiment involves a convex surface shown in Figure 15. The results show that the laser formed surface using our iterative cut and folding (Section 3.3). In the folding process, the resulted angles are always consistent with the laser passes applied, as the thermal plastic stress can generate enough torque even on a short crease to compensate the gravitational “drag” force. In the cutting process, the lasers are set to the highest power, rasterize and destroy a thick hatch area around the boundary of the shape. In certain cases, the end of the hatch area near the folding crease is not fully destroyed and the folding creases get “pinned” on the panel, which leads to the failure of folding as showing in Figure 15d. To solve this issue, we extend the boundary of the cut rasterization area, making a small overlapping between the neighboring hatches. As a result, we can successfully fold the pattern to the desired 3D shape as shown in Figures 15e and 15f.

6 CONCLUSION AND DISCUSSION

This paper presents a computational method that can produce laser formable structures of convex surfaces. The main contribution of this work includes theoretical analysis of laser formable structures for a low-cost laser with limited capability. Our method increases the complexity of laser forming far more than the manually designed structures that typically contain only a few folds. An end-to-end laser-forming pipeline is presented to demonstrate the potential of laser forming complex shapes and, as g-code and slicers in 3D printing, this framework can pave the ground work and eventually transform laser forming and laser origami to a practical methods for low-cost customizable fabrication.
ACKNOWLEDGMENTS

This work is in part funded by ARO W911NF-19-2-0121. The authors would like to thank Dr. Nathan Lazarus and Mr. Gabriel Smith at the Adelphi Lab Center who helped us fabricating the antennas and the curved surface reported in this paper.

REFERENCES


