Plotting with Thread: Fabricating Delicate Punch Needle Embroidery with X-Y Plotters

Shiqing He
University of Michigan
Ann Arbor, U.S.A.
heslicia@umich.edu

Eytan Adar
University of Michigan
Ann Arbor, U.S.A.
eadar@umich.edu

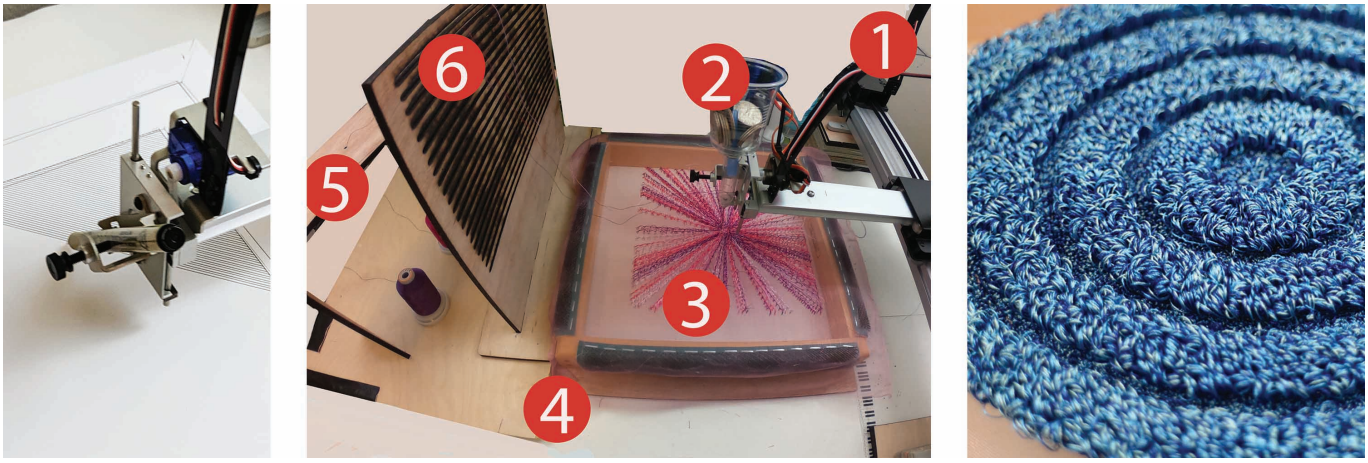

Figure 1: **Left:** A standard X-Y plotter that we repurposed to fabricate punch needle embroidery. **Middle:** Our modified plotter consisting of 1) an *Axidraw* plotter, 2) a customized punch needle tool, 3) a gripper frame, 4) a frame holder, 5) a threading station, and 6) a thread separator; **Right:** One example of the many styles that we can produce (3D embroidery).

ABSTRACT

Punch needle embroidery is a unique type of embroidery that uses loops of threads to create designs. Technology for punch needle embroidery ranges from popular handheld manual tools to high-cost industrial tufting machines. Computer-controlled punch needle fabrication tools remain out-of-reach for most practitioners. In this work, we describe how a low-cost X-Y plotter can be repurposed to support punch needle embroidery fabrication. By adding easy-to-make physical accessories coupled with a novel software toolkit, we support the production of delicate and precise punch needle embroideries with minimal manual labor. After examining and evaluating the potential and challenges of converting X-Y plotters into punch needle embroidery fabricators, we propose design and fabrication guidelines that are specific to plotter-based punch needle embroideries. We demonstrate how this novel fabrication approach enables the production of a wide range of artifacts and textures.

DIS '20, July 6–10, 2020, Eindhoven, Netherlands.

ACM ISBN 978-1-4503-6974-9/20/07 ...$15.00.
http://dx.doi.org/10.1145/3357236.3395540

Author Keywords

Embroidery Fabrication; Punch Needle Embroidery; Plotter; Craft Fabrication; Fiber Art; Craft Design; X-Y Plotter;

CCS Concepts

•**Applied computing** → **Fine arts;** •**Human-centered computing** → *Systems and tools for interaction design;*

INTRODUCTION

Punch needle embroidery is a traditional embroidery method where loops of threads are punched into backing fabrics using a tubular needle (Figure 2). As with other embroideries, punch needle can be used for decoration and textile art. However, punch needle embroidery is also commonly seen in rug production because of the unique textures it can create [16].

There are a variety of punch needle embroidery tools. Manual punch needles are the most popular and accessible options (Figure 3). While the designs of these tools vary, they mostly feature two main parts: a tubular, sharp-head needle with a threading eye, and a handle to attach the needle. For rug making and large-scale pieces, thick yarns are paired with large-size needles. In contrast, *delicate punch needle embroideries* are made with thin embroidery flosses (see Figure 3

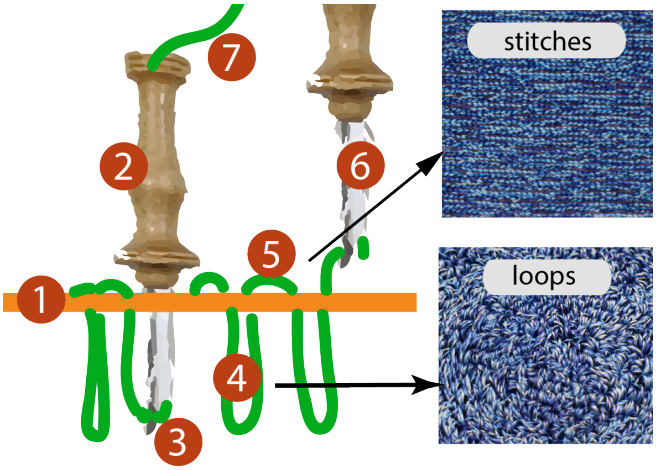

Figure 2: Materials and mechanics of punch needle embroidery: 1) the backing fabric, 2) a punch needle tool with two parts: handle and needle (also called "head"), 3) a punch needle tool punches through the backing fabric to make a thread loop, 4) loops created by the punch movement stay on the other side of the fabric, which is typically considered as the front side of the embroidery, 5) the thread connecting adjacent loops forms stitches, 6) a punch needle tool normally punches away from the previous loop to avoid damaging threads.

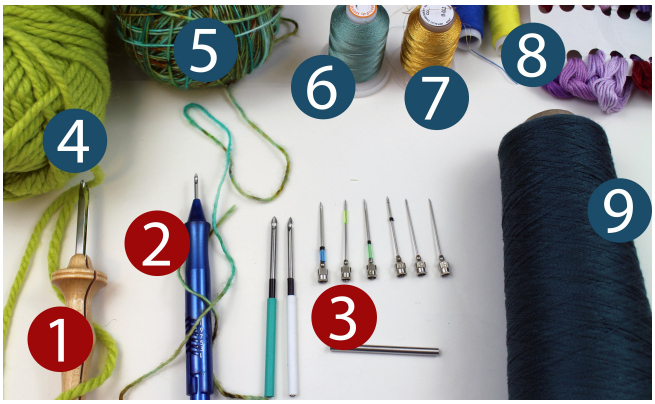

Figure 3: Commercially available manual punch needle tools and commonly used threads: 1-2) Oxford Punch Needle® and Ultra Punch® Needle, 3) a variety of punch needle heads and their interchangeable handles, 4-9) examples of threads with diverse thicknesses and materials.

for commonly used threads). These delicate embroideries are called "miniature punch needle embroidery," regardless of the dimensions of the finished pieces [20]. Mechanically, punch needle embroidery only requires three types of movement (Figure 2): a threaded needle punches through the backing fabric; the needle is pulled out, leaving a loop of "unpulled" thread underneath the backing fabric; and finally, the needle is moved to the next position [16, 20]. Despite its simplicity, punch needle embroidery is "fragile" as it depends on the friction between fabric and thread to hold the material in place. Thus, delicate punch needle embroidery is a labor-intensive and time-consuming craft that may not be as easily automated as other styles. Automated solutions tend to be either industrial tools or solutions that only partially automate this embroidery practice. The former include heavy-duty rug-making systems that work with thick materials. The latter include tufting guns – handheld electrical tools that accelerate punch needle embroidery production by executing the punching and pulling actions with motors. While a tufting gun dramatically increases the embroidery speed, practitioners still need to "drive" the power-tool-like tufting gun manually to create desired patterns. The cost of these machines and the limited materials they work with (heavy yarn) make them impractical for more delicate work. Delicate punch needle embroidery is present across a diverse set of cultures but always done manually. Examples include the fine handheld needles such as the Russian Igolochkoy™ punch needle and the Japanese Bunka needle [20].

Because of the considerable manual labor involved, and the lack of economical digital tools, *delicate* punch needle embroideries are often *small* in size. Large embroidery pieces require significant labor investment in addition to the material costs. Because they are often hard to produce, delicate punch needle embroideries are not necessarily attractive as design or rapid prototyping mediums, which is unfortunate, as they are texturally interesting and potentially adaptable for wearable (e.g., [12]) and embroidery-related research (e.g., [23]).

To address this challenge, we demonstrate how a low-cost X-Y plotter can be repurposed into a delicate punch needle embroidery fabricator. By adding easy-to-make physical accessories and a software toolkit, we support the production of delicate punch needle embroideries in a precise and efficient fashion. We demonstrate how this novel and accessible fabrication approach enables the production of a wide range of artifacts and textures.

We contribute analysis (and solution) to the challenge of converting a plotter into a delicate punch needle embroidery fabricator. Additionally, we identify the specific constraints of automated punch needle fabrication where hardware, software, and materials all interact in unique ways. We describe an open-source toolkit, *ThreadPlotter*, that supports the designing, editing, and printing of images as punch needle embroidery.[1] Finally, we reflect on how crafting experience and practice with 1) the manual form of punch needle and 2) the automated form of plotting translate (or not) to new fabrication technologies.

RELATED WORK

The Fabrication of Punch Needle Embroidery

Although the origin of punch needle embroidery is unknown [20], this versatile craft technique is widely used across many different applications. Rug making is likely the most well known. In the late 1800s, a variety of punch needle tools designed for the making of "New England Style" rug hookings became available in the United States [16]. However, punch needle embroidery is neither fixed geographically

[1] *ThreadPlotter* is available at http://eyesofpanda.com/projects/thread_plotter.

nor in how it is applied. For example, Igolochkoy embroidery (also called as Russian punch needle embroidery) and Japanese Bunka embroidery utilize thin embroidery floss to create much more delicate patterns [20]. Both approaches share a similar type of fine needle but use different types of thread. The Igolochkoy embroidery is found as decorations for the Russian Old Believers' traditional costumes, whereas Bunka embroidery utilizes special curly rayon thread to create textured wall hangings [20].

Punch needle embroidery excels at creating textures because of several defining characteristics:

1. In contrast to embroidery techniques that tie the thread to the fabric, punch needle *does not secure threads to the fabric* – allowing for faster motion. The tension within the fabric holds the threads in place (Figure 2). The simple punch-pull movement makes punch needle embroidery an easy technique for beginners and professionals alike.
2. Punch needle embroidery is a "backward" embroidery. Instead of working from the front side, a horizontally reversed image is *punched from the backside of the fabric*. Two sides of a punch needle embroidery piece have drastically different textures (Figure 2). Typically, the side with thread loops is the front side of the embroidery. The flat stitches are the back side of the embroidery (though some techniques reverse this).
3. Punch needle embroidery is a *3D embroidery technique* where practitioners can easily incorporate depth into the design. Punch needles typically come with gauges (also called stoppers) that fix the loop length by limiting the depth that the needle can be punched into the fabric. Practitioners change the gauge location to adjust loop length, thereby creating thick or thin embroideries. The use of thread trimming can create additional 3D forms.

Fully automatic industrial tufting machines make it possible to produce punch needle embroidery rugs at a speed and precision that handheld tools cannot compete with. However, automatic tufting machines are generally inaccessible to most punch needle practitioners because they are often large and expensive. Examples range from the AutoTuft to the Mtuft machines (with prices ranging from $15k to $1.5M USD). Moreover, these automatic machines and more affordable handheld tufting guns are primarily designed for rug production. Therefore, they tend to only support thick rug yarn. Our goal is to produce a solution that is approachable, automated, and still maintains the advantages and uniqueness of delicate punch needle embroidery.

Repurposing Fabricators

With the rise of the maker culture [22] and advances in fabrication research, we came across various projects that develop custom CNC machines for specific fabrication projects. Laser cutters, 3D printer, and CNC mills have become increasingly available to hobbyists and individuals [24]. As the material and technology for CNC machines mature, designers and researchers have also developed specialized CNC machines to support the fabrication of craft objects. For example, Hudson [7] demonstrates how a customized 3-axis machine can print needle felting sculpture. Others have found ways of adapting existing fabricators (e.g., salt and coffee-based 3D printers [17]). Given the ability to customize computerized fabricators, unconventional materials such as food can now be digitally enhanced [19].

Developing and assembling 3-axis CNC machines has become significantly easier as many open-source projects become available (e.g., the Maslow CNC). Nevertheless, producing a functioning and precise CNC machine remains to be a technically challenging and labor-intensive task. Instead of designing a new machine from scratch, we aim to convert an existing machine. Doing so allowed us to focus on finding the fundamental fabrication requirements rather than solving machine-specific design issues. We demonstrate how a broad type of machine intended for one task can fabricate delicate punch needle embroidery. More practically, repurposing existing machines can relieve the technical challenges of developing and calibrating physical and software components. If existing machines can be repurposed to produce delicate punch needle embroidery at a desirable quality and efficiency, it would likely make the fabricator more accessible, as users might already be familiar with the hardware and the software that come with the machine. In this project, we aim to impose minimal physical changes to the existing machines and ensure that the machine can still operate for its original functions.

Applications of Punch Needle Embroidery

A specific motivation for our project was to enable novel applications of punch needle embroidery. In addition to rug making and fabric decoration, we have also seen punch needle embroidery being used as a way to produce customized fabrics. These fabrics can be further processed into decorative and functioning artifacts. There are examples of using this unique texture in customized plush toys, mini 3D floral sculptures, furniture covers, and bedding covers [16, 20, 21].

With recent developments in personal fabrication (e.g., [14]) wearable technology (e.g., [12]), and algorithmic craft (e.g., [9]), there are many craft-based research projects that explore the design and application of traditional fiber crafts [2]. An entry-level sewing and embroidery machine makes it possible for designers and researchers to develop and test the possibility of using traditional embroidery to embedded electronics [5, 15]. Accessible knitting machines enable a vibrant group of studies that specializes in the design, simulation, and execution of knitting patterns (e.g., [8, 10, 26, 25]). In addition to technologies that are related to the fabrication process, the existence of efficient knitting fabricators makes it possible to utilize knitted artifacts as mediums for wearable and sensing researches such as [4] and [18].

In comparison to other thread-related craft such as traditional embroidery and knitting, punch needle embroidery pieces are less likely to be used as design and prototyping material despite their unique textures. We believe that an accessible automated punch needle embroidery fabricator can enable a greater variety of punch needle embroidery applications.

PHYSICAL SETUP

A standard punch needle embroidery set-up requires fabric, fabric stretcher, thread, and needles. As any practitioner would

attest to, any specific choice of one element will restrict (and inform) the choices of the others. To these elements, we must factor in the constraints of the mechanical fabricator–the plotter.

Selecting the Right X-Y Plotter

X-Y plotters are computer numerical control (CNC) machines that guide plotting tools (such as pens and markers) along vector paths. Although the name "X-Y plotter" might suggest a 2-axis machine, they are often movable along a third axis to allow the pen to move off the drawing surface. This lift provides for z-axis movement. Today, X-Y pen plotters are accessible machines. Models range from consumer-friendly self-assembling kits such as mini processor-powered Makeblock® robot kit to heavy-duty HP® vintage plotters.

3-axis movement is a minimum criterion for a punch needle fabricator. Other basic criteria include:

1. *The distance between the plotting tool and the plotting surface is adjustable and sufficient for holding a fabric stretching frame.* Different types of plotters control the x– and y-axis movement with different mechanisms. For example, a movable arm can control one or both axes. Here, the plotting surface is fixed while the arm travels. Alternatively, the plotting tool itself might be fixed while the plotting surface travels. While both approaches are viable for punch needle embroidery production, the later design might have less flexibility in the distance between the plotting tool and the plotting surface. For example, the HP7550 plotter, which utilizes a paper-feeder to control the y–axis movement, cannot hold a fabric stretcher without significant modification.
2. *The z–axis movement is large enough to create a minimal stitch.* The distance traveled in the z direction controls the size of the thread loop. Furthermore, if z–axis movement is controllable, we can fabricate punch needle embroideries with various loop sizes.
3. *Sufficient downward force can be applied in the z–axis to punch through the tightened fabric.* The force required to punch through the backing fabric varies due to fabric thickness, needle size, and stitch density. Some X-Y plotters do not provide any downward force along the z–axis at all. Instead, they rely on gravity to lower the plotting tool and only sufficient upward force to counteract light weights. While we can add weight to the drawing tool, the mechanism to raise it may no longer work. It is also worth noting that some plotters are designed to plot on a non-horizontal surface. Therefore the weight-adding approach might not be applicable to these machines.

Based on the criteria above, we chose to convert the commercially available *AxiDraw* Pen Plotter. The AxiDraw is a high-precision 1-arm X-Y plotter designed for plotting on flat surfaces. It utilizes step motors to control the movement along the x– and y–axis. It does not provide a downward force along the z–axis but utilizes gravity to lower the pen (or needle). In addition to meeting our technical requirements, the AxiDraw is also economical and has accessible operating software. It provides two control interfaces: an Inkscape plugin where end-users can import and plot vector graph, and a Python API to control the machine programmatically. Various AxiDraw models support different plotting sizes. For our work, we used the V3/A3 model that comes with a plotting area of 11×17 inches.

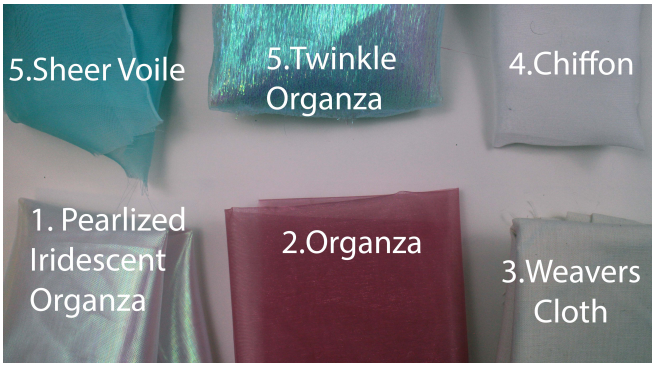

Figure 4: We tested six different types of fabrics. Pearlized Iridescent Organza (1) and Organza (2) can produce stable results. Weavers Cloth (3) is too thick for our machine to pierce. Chiffon (4) and Twinkle Organza (5) are too fragile for the gripper frame we used. Sheer Voile (6) can produce reasonable results but tends to have missing loops.

Fabric and Fabric Stretcher

The backing fabric is an essential part of the punch needle embroidery because the tension between weaves of the fabric is the only thing that holds the thread loops in place. The fabric and thread need to be matched. For example, a loose-weave fabric will not provide enough force to hold thin thread but might work with thicker thread. Conversely, punching through a tight-weave fabric with a large needle (for a thick thread) will require significant force and likely damage the weave.

Thick yarn punch needle embroidery is mostly done on monk's cloth, an even-weave cotton fabric that contains tiny holes formed by the warp and weft threads [16]. When making delicate punch needle embroidery, the most popular fabric is weavers cloth, a polyester-cotton blend fabric. In addition to weavers cloth, a variety of fabrics such as muslin, cotton chambray, wool flannel, silk noil, and linen might also work with particular combinations of needle and thread [20].

The choice of our plotter constrains our choice of fabric. For example, the AxiDraw does not provide enough downward force to punch through most fabrics. This issue can be partially solved by increasing the weight of the punch needle. However, in our experiments, we discovered that increasing this weight too much would quickly wear out the servo motor controlling the pen lifter. In experimenting with various fabrics, we found that thin fabrics, such as organza and voile, were pierceable without much change to the stock AxiDraw. Additional modifications (e.g., a heavy-duty servo on the pen lifter) would allow for thicker fabrics.

In addition to fabric choice, we found that the choice of fabric stretching mechanisms was critical for smooth operations. Unlike traditional embroidery, where the stretching of the fabric might be optional, punch needle embroidery requires

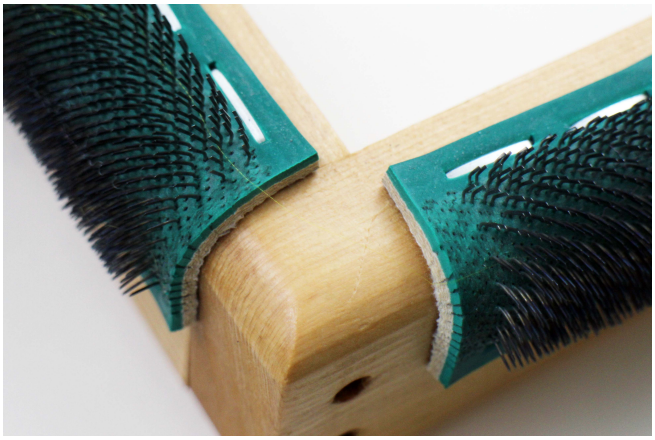

Figure 5: We recommend using gripper frames for plotter embroidery. They secure and stretch fabrics with curved metal needles that grasp the fabric.

tightly stretched fabric. It is crucial to stretch the fabric "drum-tight" because loosely stretched fabric requires considerable piercing force [16]. Stretching the fabric also reduces the damage to the fabric during the punching process.

A variety of embroidery hoops and gripper frames can be used for manual punch needle embroidery. Embroidery hoops are circular stretchers that secure and tighten the fabric by clasping the fabric between the inner hoop and the outer hoop. They are economical, lightweight, and adjustable stretchers that would work for manual punch needle embroideries. However, the embroidery hoops we tested could not stretch the fabric to be tight enough for machine-based embroidery. A loose stretch that may work for manual crafting will fail when we use the more delicate servo motors of the plotter. Additionally, the unevenly stretched fabric will also cause uneven piercing, impacting both loop height and density.

Given this, we found gripper frames to be ideal for plotter-based punch needle embroideries. Gripper frames are non-adjustable solid frames covered with metal gripper strips that are made with bent metal needles (called "teeth") (Figure 5). Gripper frames are more secure than general embroidery hoops because these sharp needles prevent any slipping. It is important to note, however, that it is possible to overstretch some fabrics. Large teeth might also tear the delicate fabric. It is crucial to pair the right fabric with the right teeth size and avoid overstretching. In our experiments, we used a 10×10 inch wooden frame, covered with EH4 gripper strips manufactured by Howard Brush.

Finally, we found that it is critical to secure the frame to a stable surface to prevent unintentional movement. Even a heavy frame may move enough during the fabrication process to ruin a piece. To prevent this, We designed a simple wooden holder to secure the frame. The holder also works as a registration tool that ensures we place the frame at the same location every time (Figure 1).

Figure 4 displays a variety of materials that we have tested–with organza being our primary choice. Voile, cotton, and chiffon might also be viable options if paired with the right thread and gripper teeth size. All examples shown in this paper were fabricated with organza.

Thread and Thread Feeder

Theoretically, any thread that, "flows easily through the needle and leaves even, consistent loops in the fabric will work [20]." For delicate punch needle embroidery, the most frequently used thread is cotton embroidery floss. Embroidery flosses are widely-available threads manufactured for embroidery-making that come in a variety of colors and fibers. Six-strand pre-cut cotton embroidery floss is one of the most popular embroidery threads [20]. However, pre-cut embroidery thread requires practitioners to re-thread the needle quite frequently. Therefore, continuous spools are preferable.

We tested wool, cotton, polyester embroidery thread, and polyester metallic thread (Figure 3). We identified the following three factors to be considered when selecting threads:

1. **Thread thickness**: Pairing fine thread with loosely-weaved fabric cannot work because the tension within the fabric weave cannot hold the thread loops. Pairing thick thread with fine needles cannot work either because the thread cannot pass through the needle freely. Additionally, a larger needle requires more piercing force, and the piercing force provided by our plotter is limited by the maximum weight the plotter pen lifter can hold.
2. **Thread Smoothness**: In order to form thread loops, the thread needs to flow freely through the needle. Natural fiber such as cotton and wool thread might come with tiny strands of fiber that increase the friction. Additionally, it is common to use multiple strands of threads to increase thread thickness and to blend color. In these cases, natural fibers can tangle if they have fuzzy finishes. Similarly, metallic threads that tangle easily might not work with plotter-based punch needle embroidery.
3. **Thread strength**: When making manual punch needle embroidery, the punch needle is angled so that it always punches away from the thread, ensuring that the sharp head of the needle does not damage the thread. X-Y plotters are generally not equipped with a pen rotation mechanism. Varying this dynamically is rarely needed, so most plotters fix the pen angle. The fixed angle might lead to cuts in weaker threads. To avoid this problem, we choose to use strong threads that are less likely to break or become fuzzy even when pressed by the punch needle.

Among the threads tested, polyester embroidery threads produced the most stable results. Polyester embroidery threads are strong and smooth synthetic threads that are manufactured for machine embroidery. A variety of sizes and colors are available. We used 120 deniers, two-ply 100% polyester embroidery threads that are widely available on the market. We tested punching one, two, and three strands of this embroidery thread through organza. Three strands of thread produce the most stable result.

A thread feeder is necessary because we rely on continuous thread spools. Standard embroidery machines usually have a thread tension adjustment system to ensure the thread is

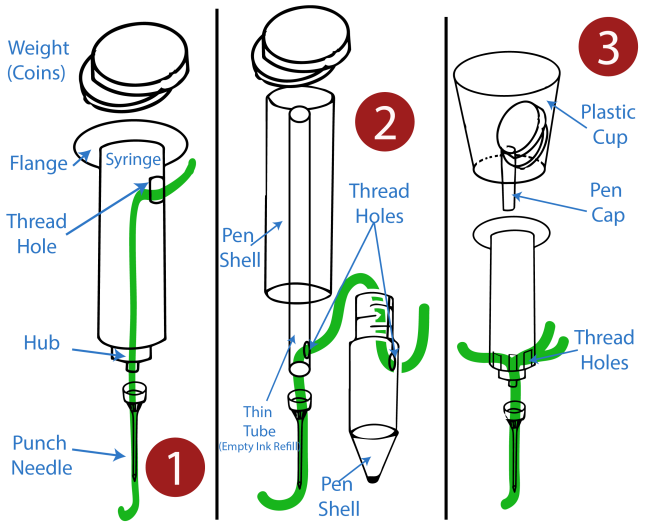

Figure 6: Three iterations of the punch needle handle design using pre-manufactured plastic materials. Through these prototyping iterations, we locate the optimal material (i.e., syringe), location for the threading holes (i.e., close to the syringe hub), and mechanism to add adjustable weight (i.e., use the syringe flange to hold a plastic cup).

tightened. In contrast, punch needle embroidery requires that there exist no tension on the thread. Even the slightest tension would cause the thread loops to pull out of the fabric along with the punch needle. Fortunately, most embroidery spools are designed to unwind from the top. As long as the force to unwind the thread is directly above the thread, it is effortless to unwind. To ensure that the force unwinding the thread is directly above the spool, we designed a simple thread unwind station where individual spools are pulled and passed to the punch needle.

Punch Needle

The last physical component we designed is the punch needle. For plotters that provide piercing force, it is theoretically possible to use any commercially available punch needles. However, the handles on these needles are often difficult to modify (e.g., for adding weight). All punch needles we examined accept thread at the end of the handle, making it impossible to add weight directly above the needle (see the threaded tools in Figure 2 and Figure 3). Adding weight to the side of the handle or the pen lifter is less effective.

In addition to the weight issue, some fine punch needles have very short handles that are not long enough for AxiDraw to hold. Because of these constraints, we designed our own punch needle handle that can: 1) holds commercially-available punch needle heads, 2) holds an adjustable amount of weight directly above the needle, and 3) allows for thread feeding without tangles.

To make our design as accessible as possible, we searched for economical materials that can be easily converted into the handle. We selected pre-manufactured plastic and tubular materials (e.g., syringes and pen casings) as the primary materials for rapid prototyping. These were cheap, easily available, and could be modified without specialized equipment. We designed and tested three different handles using syringes, plastic pens, coins (as the weight), and plastic cups. In our progress towards our final design, we identified several key constraints.

First, we need to make the threading process as simple as possible. In existing handle designs, threads are normally fed into the needle using metal wire threaders. However, straight threaders are less effective when the threading pass is not straight. In the first design iteration, we made a curve threader to pass the thread from the side. Nevertheless, operating a curved threader requires end-users to aim for the needle, which complicates the threading process significantly. In the second iteration, we place a tubular insert made from an empty gel pen lead inside of a 2-part twist pen. Threading the gel pen lead is easier than using the curved threader. However, because the diameter of the insert is small, threading multiple strands is still a time-consuming activity that also requires a fine crochet needle to pull the thread.

As a result, we separated the threading into two parts in the third iteration. We placed the threading holes close to the hub, which makes it possible to pass the thread through the hub without a threader. After passing the thread through the hub, we use a straight wire threader to thread the needle. Then, we attach the needle to the syringe. Besides adjusting the location of the threading hole, we also noticed that multiple thread holes are necessary if multiple strands of threads are used. Feeding individual strands of thread to dedicated thread holes reduces the chance of thread tangling dramatically. Consequently, we designed three threading holes on the barrel of the syringe.

Second, we needed to design a platform to hold adjustable weight. We chose to use coins as weight in our prototypes because they are accessible "heavy" metals. In the first iteration, we taped coins to the syringe flange. This worked well with the caveat that it was difficult to adjust the number of coins. In the second iteration where we re-purposed a pen, we laser cut a flat wooden square and taped coins to it.

Comparing the first two iterations, we noticed that the syringe flange and hub are handy structures for punch needle handle. The syringe hub fits most of the fine needles we examined. Therefore it automatically holds the needles vertically. The flange of the syringe is flat, making it easy to add weight on top. To make a platform that can hold an adjustable number of coins, we combined a cylinder-shaped pen cap and a small plastic cup. The glued-together platform sits directly on top of the flange. End-users can place any number of coins inside the cup. The cylinder-shaped pen cap ensures that the platform stays on top of the syringe without the need for adhesives.

The final punch needle design is economical and straightforward. In comparison to the first variants, it features easy threading processes and an effortless weight adjustment system. An optional component to add is a gauge/stopper that indicates the location where the needle should be held. End-

users can use the existing volume markings on the syringe as guidelines for placing the punch needle tool. End-users can also use tapes and markers to indicate the desired punch needle location. Figure 6 illustrates the design of our customized punch needles.

SOFTWARE CONTROL AND THREADPLOTTER

The AxiDraw offers two control interfaces: an Inkscape plugin where end-users can plot vector image (in the SVG format), and a Python scripting interface. The latter provides low-level controls such as *pen_up* and *pen_up_speed*. The graphical SVG interface can be used for many punch needle embroidery fabrications. However, we found that having access to low-level controls opens up more fabrication possibilities. Some design patterns that involve different loop and stitch sizes have complex setting changes. This can be "hacked" by creating various layers and fabricating them one at a time. However, because not all controls can be adjusted by inputting the images, the end-user must pause the machine to adjust the plotter settings. Access to low-level functions significantly reduces the manual labor involved.

To control our physical setup, we developed *ThreadPlotter*, a Python-based API that supports the design of X-Y plotter compatible embroidery patterns. The API primarily helps end-users address the problem: Given a vector path to be fabricated into punch needle embroidery, *where* and *how* should the machine punch? *ThreadPlotter* answers these questions by processing path information and translating continuous paths into a set of punch point locations. Additionally, *ThreadPlotter* will determine effective settings for the depth and speed of the punch. While translating vector information into vector-based punch needle embroidery patterns is the primary function of *ThreadPlotter*, the system also provides utility functions (e.g., for converting raster images to acceptable vector formats).

ThreadPlotter can produce two kinds of outputs. First, *ThreadPlotter* produces SVG files that can be fabricated through the graphical Inkscape environment. Our focus on vector formats is due to the observation that these are the most widely accepted by X-Y plotters. Second, *ThreadPlotter* produces Python scripts that can be directly executed by Axidraw. While these scripts may not directly execute on other plotters, we utilize 'simple' calls that are likely to be supported by most devices. Thus, we believe *ThreadPlotter*-output scripts can be readily adapted.

Determining Punch Points Locations

ThreadPlotter converts vector images into punch needle embroidery patterns in three steps. To start, *ThreadPlotter* extracts vector elements such as <line> and <polygon> (in SVG syntax). For each vector element, *ThreadPlotter* identifies and processes the location information of the element. It also converts curves into straight lines in preparation for later path segmentation (Figure 7). For example, *ThreadPlotter* will approximate cubic Bézier curves with lines. At the end of this step, we have a list of polylines: continuous paths that consist of only straight line segments.

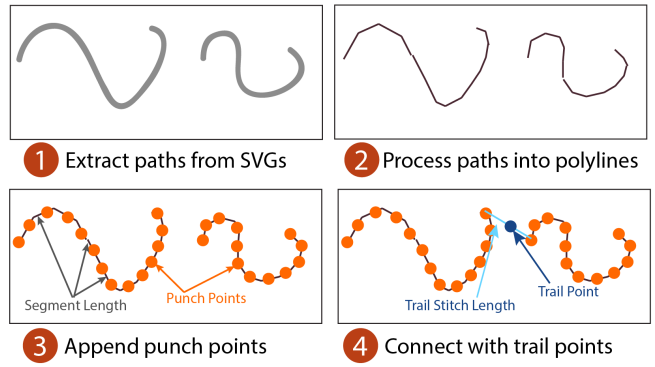

Figure 7: Pipeline for converting vector paths into plotter-compatible punch needle patterns.

ThreadPlotter divides line segments within the polylines into equal-length sections. The unit length used in the dividing algorithm controls the density of the thread loop and the size of the stitches. We refer to this length as *segment length*. Setting a suitable segment length is important for a good weave. For example, the finished piece will not have a firm, rug-like texture if the segment length is too long. If the segment length is too short, the needle will pierce a section of the fabric repetitively and damage the fabric. Different combinations of fabric and thread require different segment lengths. For our combination of fabric and thread, we found 1.05 mm to be optimal.

At the end of each line segment, *ThreadPlotter* generates a punch point. Each punch point represents a location where the needle will pierce the fabric. After annotating the end of each line segment, we obtain polylines consisting of a series of punch points. We refer to punch points within one polyline as a *punch point group*.

Finally, *ThreadPlotter* connects different punch point groups with *trail points*. Because punch needle embroidery uses a continuous, untied thread, pulling the thread can revert the previous stitches. In other words, a threaded punch needle cannot travel a long distance without pulling the previous loops out. Due to this characteristic, a punch needle cannot directly move to the start of the next punch group after finishing one group of punch points. *ThreadPlotter* adds piercing points as supporting trails to avoid pulling previous loops.

The additional loops created on trail points ensure that the loops in the previous punch points stay in place. The segment length and loop height in the trails should be different from that of the polyline for a couple of reasons. First, as long as the punch needle touch the fabric and forms a minimal loop, the previous loop will not be pulled. Therefore, it is not necessary to punch long loops in trail points. Second, trail lines are supporting structures that need to be removed after the fabrication process because they are not part of the original pattern. Making dense trail stitches wastes materials, and increases the difficulty during the removing process. When punch points of the same color are scattered around the image, there might be many trail points. However, as long as the loop

length of the trail points is less than that of the punch point, longer loops can cover up the shorter loops. Thus, not all trail stitches need to be removed (as they are covered up).

If the end-user opts to output an SVG file (rather than code), they can use the Inkscape interface to print. There is a tradeoff in doing so. On the one hand, the end-user can intervene to better inspect and control the output. On the other hand, end-users must pause the plotting process in order to adjust the loop length (i.e., the lowest position that the needle can travel along the z–axis). An alternative to manual pausing is to separate punch point groups into different layers so that Axidraw can stop before the transition between punch point and trail point. However, the time and labor involved in the setting adjusting process outweigh the potential benefits. To balance this, trail points in the SVG mode share the same loop length as punch points, but use a longer segment length (2.64mm). In the script mode, the needle position can be adjusted automatically. Hence, trail loops have a minimal loop length of 2.2 mm. When operating using the SVG interface, the plotter will automatically return the arm to its origin. The returning path also needs to be covered with trail points to avoid pulling previous loops.

If a design has multiple colors, *ThreadPlotter* preprocesses vector paths by grouping them according to their colors. Then, *ThreadPlotter* processes individual color groups separately, generating files for each color. In both the SVG and scripted forms, some manual work is required. When users finish plotting one color, they can remove trails, re-thread the needle, and proceed to the next color.

Stitches, Loops, and Pen Speed

With more sophisticated embroideries, such as those with a 3D effect or the inverted Bunka stitching (where the stitch side becomes the front), the scripted output of *ThreadPlotter* shines. Without it, end-users must constantly adjust loop length and other elements. However, even the scripted interface does not offer infinite flexibility. The properties of the materials and their interaction constrain what is feasible.

To understand the relationship between loop length and stitch size, we experimented with a number of combinations to find workable settings. We started by measuring the maximum and minimal loop length that AxiDraw can create, then conducted several tests using different loop sizes within the range.

We found that we can approximate the relationship between loop depth ($loop\_depth$), punch depth ($needle\_depth$ – the depth of the needle below the fabric), and stitch length ($stitch\_length$) using a simple formula:

$$loop\_depth = \frac{needle\_depth - stitch\_length}{2}$$

Additionally, we observed that if the needle moves too fast, the thread might follow the needle and get pulled out of the fabric because of inertia. Short loops are especially vulnerable as they can be pulled out easily. As a result, when punching short loops, the needle needs to have a slower lifting speed in comparison to the speed used for long loops. With this observation, we measured the optimal needle lifting speed associated with the minimal and maximum loop length. *ThreadPlotter* can suggest the optimal needle lifting speed given the loop size using a simple linear mapping calculation.

Figure 8: *ThreadPlotter* processes raster images into plotter-compatible embroidery patterns. Original artwork by Shiqing He [6].

Given the desired stitch size, we can also calculate the loop length and lifting speed that ensures the creation of a minimal loop. We used this calculation when processing trail points in the scripted mode so that we can use the maximum stitch size and the minimal loop length. Moreover, this insight enables the pattern generation for Bunka-style punch needle embroideries, where stitches of different sizes are treated as the front side of the embroidery.

Raster versus Vector Images

Thus far, we have focused our attention on converting a vector image into a plotter-compatible embroidery pattern. Using drafting tools such as Inkscape and Adobe Illustrator™, designers can produce precise and scalable vector images. Many tools also offer the ability to convert raster images into vector formats. *ThreadPlotter* offers direct creation and manipulation of SVG files through scripting. It also implements an alternative raster-to-vector conversion module that takes into account the intended target (the punch needle embroidery) in the conversion. *ThreadPlotter* uses the following conversion pipeline.

1. *ThreadPlotter* loads an image in the format of JPG or PNG, then adjusts the number of colors in the image. We experimented with multiple color-reducing algorithms, including quantization[3], k-means clustering [11], and grouping colors that have short Euclidean distances. All three methods produce usable results.
2. *ThreadPlotter* groups pixels into squares whose width and height equal to the segment length. This process adjusts the unit pixel size to be equal to the segment length. At the end of this step, we obtain a copy of the image that has a limited palette and a specific Pixels Per Inch (PPI) controlled by the segment length. At the center of each adjusted pixel, *ThreadPlotter* produces a punch point using a random color selected from this group of original pixels.
3. *ThreadPlotter* then links punch points that share the same color using trails. The trail generating process is identical to that within the SVG mode.

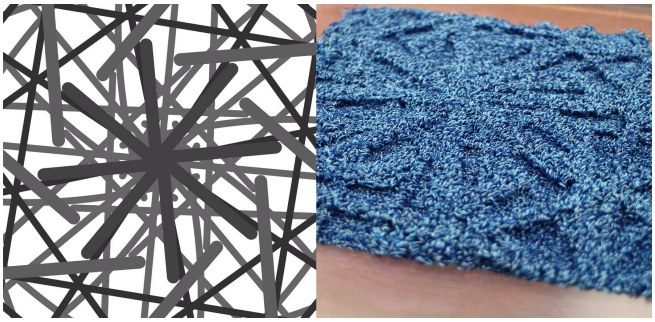

Figure 9: By assigning specific loop lengths to different colors, users can create patterns for 3D punch needle embroidery.

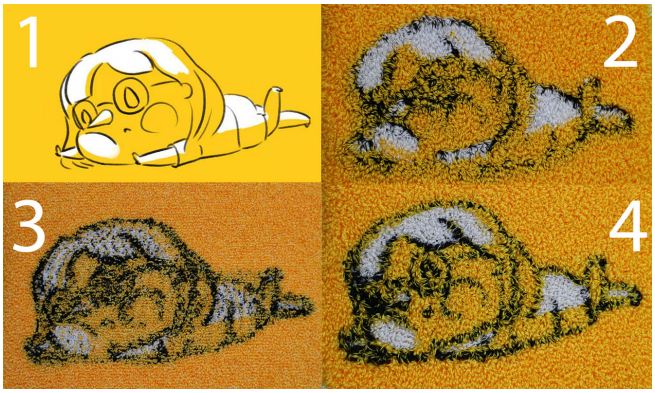

Figure 10: Features within images might be blurry if the embroidery size is too small. The effect is especially observable when using a long loop length (in 2, loop length = 6mm). Besides enlarging the embroidery size, users can reduce this effect by using a shorter loop length (in 3, loop length = 2.4mm), or assigning different loop lengths to specific colors (4). Original artwork courtesy of P Mei [13].

Using this conversion module, users can generate punch needle embroidery patterns directly from raster images (e.g., paintings or photos). Figure 8 exhibits a finished punch needle embroidery piece designed and fabricated using this approach.

In addition to the basic pipeline described above, we built a few additional extensions to the vector production system. For example, *ThreadPlotter* can treat colors within the images as indicators for loop size. For example, it will map punch points with darker colors into longer loops. By adjusting the loop size dynamically, users can produce 3D embroideries with one color of thread. Figure 9 provides a 3D example fabricated with one color.

While we can convert fine details in the raster image (e.g., a thin line) to a vector, this detail may not correctly render when embroidered. Punch needle-produced loops can intertwine, making colors appear blended, and the image appears blurry. In manual punch needle embroidery, practitioners address this using sharp tools to separate these intertwine loops after all stitches are made [16]. End-users can certainly do this additional step after the machine fabrication process.

In building *ThreadPlotter*, we have identified three alternative approaches to support printing detailed images without manual intervention (Figure 10). The most straightforward and effective approach is to scale up the embroidery size. Scaling up will increase the "resolution" of the finished piece, therefore making the finished piece more recognizable. Users can also consider using a short loop length. When the length of the loop is short, loops are less likely to blend. However, the shortest loops should still be longer than the trail loops. Otherwise, punch loops cannot hide trail points completely. Finally, users can assign individual colors with different loop sizes. Figure 10 shows an example fabricated with this approach. The piece with mixed loop length better preserves the original feature of the image in comparison to the untreated design.

Thread Color Matching

Selecting the closest thread color might be a tedious task when plotting a multicolor embroidery, especially if there are multiple similar colors within the palette. The last function of *ThreadPlotter* is to provide a simple thread color matching tool. In our experiments, we gathered embroidery threads in more than sixty colors. We collected the RGB value of each thread from the official color chart provided by the manufacturer (other threads can be added). *ThreadPlotter* finds the closest color match to the color in the embroidery pattern by calculating the euclidean distance between two colors.

Because the number of shades we can gather is limited, there are cases where our closest match is still drastically different from the desired color. In these cases, we provide an experimental feature that suggests potential threads that "blend" into the desired color. The blending is possible because we can use three strands of embroidery thread in different colors for our punch needle. When looking from a distance, the distinct colors of these three strands of threads will appear as if they blend into one color. Because physical color blending is very different from digital color blending, the blending suggestions we generated might not always function correctly in the physical world. Nevertheless, users can use these thread color suggestions as starting points for finding the ideal thread color.

We briefly summarize the overall experience of using *ThreadPlotter*. When end-users input an image (vector or raster), *ThreadPlotter* processes the image into plotter-compatible punch points and trail points. For every color within the image, *ThreadPlotter* generates an SVG file and a Python script. The SVG file contains location information of punch points and trail points. The Python script contains both the location information and machine settings, such as needle raise speed and needle position along the z–axis. Additionally, *ThreadPlotter* produces another SVG file that displays the suggested color of thread to use. Using this toolkit, we can create a wide range of artifacts.

DISCUSSION

Adapting an X-Y plotter – a tool intended for one type of "fabrication" – to an entirely new form, presented many challenges. While we could address a number of these challenges in our hardware and software implementation, our default settings might not entirely account for the complex interactions

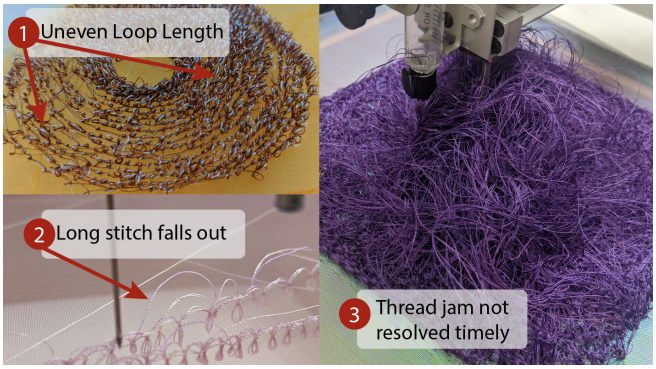

Figure 11: Several failed examples that demonstrate common issues.

between materials. A mistake can unravel the entire image or break the materials (see Figure 11 for some examples). We reflect on these not only because they may be useful to those using our approach, but also to describe the challenges of adapting manual techniques and expertise–specifically around troubleshooting–to the automated infrastructure.

Many specific challenges are due to the lack of human monitoring. Automation invites one to "set-it-and-forget-it," walking away from the machine as it works. As a specific example, we found that it is necessary to be cautious when fabricating long stitches with short loop length because these loops are more likely to fall out of the fabric even when they are produced at a low speed. When loops fall out of the fabric, the extra thread hanging on the stitch side can also trigger additional tangling, or even seize the moving needle.

Practitioners of manual punch needle embroidery can react, troubleshoot, and correct problems dynamically, which is only possible due to their engagement with the process and their extensive experiences with the craft. What was notable to us in building this platform were the places where these experiences could or could not help in the automated scenario.

A specific example of this relates to difficulties for the needle to pierce the fabric–a rare issue in the manual form. Reasons for this may include the obvious: a dull needle (which can be resolved with sharpening). The needle piercing problem also includes subtle differences in how the fabric is stretched. While insufficient or uneven stretching does not present a problem for a person embroidering, this is critical for a successful automated plot. Trouble piercing the fabric can also be resolved with more "elbow grease" – simply applying more force (or equivalently) weight. However, simply adding weight to the needle will not work if the weight exceeds the AxiDraw's lifting limit.

Another issue relates to the specific problem of converting a device that fabricates one material to another. There was not always a direct translation from our understanding and experience with the X-Y plotter to the X-Y needle punch machine. For example, the AxiDraw is sufficiently heavy and robust to work with light pens. Pens tend not to "drift" up or down, and the metal of the platform does not tend to break. While needle-punch is in some ways delicate, it invariably pushes the limits of the plotter and required adaptation. For example, we found that securing the AxiDraw down using clamps was necessary to avoid drift or shake. We also found that some plotters' arms bend when they are extended, especially when they are holding a heavy-weight needle. In this case, it is useful to angle the gripper frame slightly to create a slanted surface that is parallel to the plotter arm.

In some cases, we also made trade-offs in what we supported in printing. For example, in manual embroidery, the needle should be as close to the fabric as possible. However, if a needle is caught under the fabric, and it is not resolved immediately, the needle will likely tear the fabric because the plotter cannot detect this issue. Therefore, we lift our needle 3mm above the fabric. The extra distance reduces the chance of needle getting caught, but also reduces the maximum loop length that we can fabricate.

Future Direction

We believe having an accessible punch needle embroidery fabricator can encourage innovative applications of this unique and versatile craft. The ability to fabricate customized textile at a low cost gives practitioners the freedom to explore and experiment. Besides producing aesthetically pleasing textile art, this technique and the produced artifacts have the potential to bring a unique touch to wearable technologies and the fabrication of soft IoT devices. For example, we can imagine custom-made bathroom rugs that embed health-measuring sensors. We are also excited about the potential of using this textile as a material for building soft computing devices (e.g., [1]). In the future, we hope to expand this study by collecting additional user feedback and analyzing the performance of the *ThreadPlotter* on other platforms such as DIY X-Y plotters. Last but not least, we hope this work spurs conversations in designing accessible craft fabricators.

CONCLUSION

In this work, we demonstrated how a repurposed low-cost X-Y plotter could produce delicate punch needle embroideries in a precise and efficient fashion. We examined the opportunities and challenges within this novel fabrication method. Hoping to make this fabricator economical and accessible, we used easy-to-source materials for building physical accessories and imposed minimal change to the plotter (ensuring that it can still be used for its original purpose). We presented *ThreadPlotter*, a toolkit that contains all physical and digital tools needed for the fabrication process. It is publicly available at `http://eyesofpanda.com/projects/thread_plotter`. We hope this work could support unconventional applications of this versatile fiber-based craft and spurs discussion on designing accessible craft fabricators.

REFERENCES

[1] Joanna Berzowska and Marguerite Bromley. 2007. Soft computation through conductive textiles. In *Proceedings of the International Foundation of Fashion Technology Institutes Conference*. 12–15.

[2] Raune Frankjær and Peter Dalsgaard. 2018. Understanding craft-based inquiry in HCI. In

Proceedings of the 2018 Designing Interactive Systems Conference. 473–484.

[3] Robert M. Gray and David L. Neuhoff. 1998. Quantization. *IEEE transactions on information theory* 44, 6 (1998), 2325–2383.

[4] Li Guo, Joel Peterson, Waqas Qureshi, Adib Kalantar Mehrjerdi, Mikael Skrifvars, and Lena Berglin. 2011. Knitted wearable stretch sensor for breathing monitoring application. In *Ambience'11, Borås, Sweden, 2011*.

[5] Nur Al-huda Hamdan, Simon Voelker, and Jan Borchers. 2018. Sketch&Stitch: Interactive Embroidery for E-textiles. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[6] Shiqing He. 2015. Fruit Study. (2015). Watercolor painting. Artist's personal collection.

[7] Scott E Hudson. 2014. Printing teddy bears: a technique for 3D printing of soft interactive objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 459–468.

[8] Yuki Igarashi, Takeo Igarashi, and Hiromasa Suzuki. 2008. Knitting a 3D model. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1737–1743.

[9] Jennifer Jennifer Mary Jacobs. 2013. *Algorithmic craft: the synthesis of computational design, digital fabrication, and hand craft*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[10] Jonathan Leaf, Rundong Wu, Eston Schweickart, Doug L James, and Steve Marschner. 2018. Interactive design of periodic yarn-level cloth patterns. In *SIGGRAPH Asia 2018 Technical Papers*. ACM, 202.

[11] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. 2003. The global k-means clustering algorithm. *Pattern recognition* 36, 2 (2003), 451–461.

[12] Jane McCann and David Bryson. 2009. *Smart clothes and wearable technology*. Elsevier.

[13] P Mei. 2019. Daily Life. (2019). `https://pmei.art/` Digital painting. Artist's personal collection.

[14] Catarina Mota. 2011. The rise of personal fabrication. In *Proceedings of the 8th ACM conference on Creativity and cognition*. 279–288.

[15] Sara Nabil, Jan Kučera, Nikoletta Karastathi, David S Kirk, and Peter Wright. 2019. Seamless Seams: Crafting Techniques for Embedding Fabrics with Interactive Actuation. In *Proceedings of the 2019 on Designing Interactive Systems Conference*. 987–999.

[16] Amy Oxford. 2016. *Punch needle rug hooking: techniques and designs*. Schiffer Publishing.

[17] Ronald Rael and Virginia San Fratello. 2018. *Printing architecture: Innovative recipes for 3D printing*. Chronicle Books.

[18] Rafiu K Raji, Xuhong Miao, Shu Zhang, Yutian Li, Ailan Wan, and Andrews Boakye. 2019. Knitted piezoresistive strain sensor performance, impact of conductive area and profile design. *Journal of Industrial Textiles* (2019), 1528083719837732.

[19] Johannes Schoning, Yvonne Rogers, and Antonio Kruger. 2012. Digitally enhanced food. *IEEE Pervasive Computing* 11, 3 (2012), 4–6.

[20] Marinda Stewart. 2009. *Punchneedle The Complete Guide*. Penguin.

[21] Marinda Stewart. 2013. *Easy, elegant punchneedle*. Stackpole Books.

[22] Joshua G Tanenbaum, Amanda M Williams, Audrey Desjardins, and Karen Tanenbaum. 2013. Democratizing technology: pleasure, utility and expressiveness in DIY and maker practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2603–2612.

[23] Aris Tsolis, William G Whittow, Antonis A Alexandridis, and JC Vardaxoglou. 2014. Embroidery and related manufacturing techniques for wearable antennas: challenges and opportunities. *Electronics* 3, 2 (2014), 314–338.

[24] Scooter Willis. 2018. The maker revolution. *Computer* 51, 3 (2018), 62–65.

[25] Kui Wu, Xifeng Gao, Zachary Ferguson, Daniele Panozzo, and Cem Yuksel. 2018. Stitch meshing. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 130.

[26] Kui Wu, Hannah Swan, and Cem Yuksel. 2019. Knittable stitch meshes. *ACM Transactions on Graphics (TOG)* 38, 1 (2019), 1–13.