Core Memory: Encoded

Ahree Lee interview transcript. Running time 10 min, 27 sec.
Conducted via Zoom on November 10, 2021, by Curator Laura Blereau. Written interview transcription by Curatorial Assistant Alex Landry. Interview text and audio have been edited for clarity by Blereau and Landry.

I'm Ahree Lee. I'm based in Los Angeles, and I grew up in Philadelphia.

I have a somewhat uneasy relationship with my cultural heritage. My family emigrated from Korea to the US when I was a baby. And we were among the first wave of families to benefit from the Immigration and Nationality Act, which reversed a decades-long restriction on the number of Asians permitted to emigrate to the US. So my parents were concerned that if they spoke to me in Korean, I would have trouble learning English, so they only spoke to me in English. And consequently, I never learned Korean. They also weren't big on continuing Korean traditions. I think for them, Korea represented the past, and they were intent on creating a better future in America. As a consequence, many of my cultural traditions are the same ones other Americans have, like opening presents on Christmas Eve, or making Easter baskets or things like that. Then we had other traditions--they weren't really Korean traditions, or even exactly American traditions--kind of a hybrid cultural tradition that we made up as we went along. I feel like this hybridization is at the heart of my art practice. I merge forms that might not seem to go together, like weaving and computers, into something that's unique to my vision.

Before I turned to my art practice full-time, I had a career in the technology sector. I designed and hand-coded HTML websites in the 90s. And then later, I moved to Silicon Valley, and I worked in user experience design. So my experience in the technology industry runs from working at a small startup to an experienced design consultancy and to Apple. How that relates my art practice is that in my art practice, algorithms are at the heart of my method. An algorithm is a set of often repetitive steps to solve a problem or accomplish a task. Most often they're associated with computer programs, but an algorithm can also be expressed as a recipe or a knitting pattern. Sometimes I use code to create things like computational videos, but other times I follow rule-based manual processes that I set for myself. In my body of work, "Pattern : Code", which examines the connections between looms and computers, craft and code, and the value of visible female labor, I use my hands rather than computers to execute the algorithmic process, weaving data into visual patterns.

I do consider myself a weaver in that I make weavings. I think that, a lot of times, we have a hesitancy to claim an identity because we feel like we don't meet a certain threshold. And in the same spirit, I consider myself a coder even though I'm not an expert. And that reminds me of this anecdote: When I did my artist residency at the Feminist Center for Creative Work in Los Angeles, I brought together a group of women collaborators to help me teach a workshop in weaving, coding, and physical computing. And while we were planning the workshop, the question came up--Am I a coder? And I kind of hesitated and said, "Well, I know how to do some coding, but I'm not very good at it. And I can't do anything really complex." And then one of them said, "Hold on, I can tell by the way you're talking about it, that you actually CAN code, but you're just not owning it." And then I realized she was right. I'm a coder, because I do coding--not because I'm an expert at all, but because I'm a practitioner. And a lot of people, women especially, hold themselves back and don't own their identity as coders because of some misplaced modesty. I feel like there's less of that dynamic in craft, because it's perceived as a feminine activity to begin with, but it's still there to a degree. And now I'm trying to stop holding myself back and owning my various identities as a weaver, as a coder, competently.

I think that I'm always trying to take concepts that are often perceived as oppositional or binary, and show the nuances, and how they're not as binary as we think they are. For example, we tend to think of craft and technology as two opposing concepts. Craft is made by hand, and technology involves machines. But the etymology of the two words reveals something different. The root of the word technology comes from the Greek 'tekhne', which means art or craft, and then craft derives from the Old English--I don't know how this is pronounced, but I'm gonna say 'cræft' maybe--meaning strength or skill. And technology is craft, and not its opponent. My current work reclaims weaving as a computational activity and reframes computing as a craft. By reestablishing the fluidity between technology and craft, I want to bring a human, tactile quality back to technology and then use weaving and craft as a gateway to coding, demystifying and reclaiming technology for all who feel left out by it.

I may have alluded to this before, but for many years, I was exclusively a video and new media artist. But then a few years ago, I heard that the technology on which the first computers was based was derived from weaving looms, and that the first computer program was written by a woman, Ada Lovelace, as you mentioned. And though I had been working in Silicon Valley and the technology industry for a decade, I had never heard this before. So I wanted to learn more, and I found a weaving teacher, and I started researching the history of interconnections between weaving and computing--and through it, the interrelationships between craft, women's labor, and technology itself.

And one of the things that I learned was that in the early 1800s, Joseph Murray Jacquard invented a way to speed up the process of weaving complex patterns by using punched-cards to determine which threads would be lifted or lowered in each row of a weaving. Then the inventors of the earliest computers saw that the binary nature of weaving, in which a thread is either lifted above or lowered below the surface of the cloth, was analogous to the binary nature of computing, which is based on ones and zeros, and used punched-cards as input devices for their machines. As recently as the 1960s or 70s, computers were still operated with punched-cards. So in my piece "Ada", there are two rectangular shapes that are roughly the size and proportion of these 1960s and 70s era IBM punched-cards. So the black yarn marks spots that correspond to punched holes. And it's named for Ada Lovelace, who, as I mentioned before, was the 19th-century woman who wrote the first computer program. She observed that the computer weaves algebraic patterns, just as the Jacquard Loom weaves flowers and leaves. And so this is a message that I wove into this punched-card weaving, using the same language that computers read in the punched-cards themselves. So if the computer were to read a punched-card that had punches in those same corresponding positions that the black weft in my weaving occupy, then it would read that message, Ada Lovelace's quote.

Women dominated the computer programming industry in the early days, which is not something a lot of people remember. Not only did a woman write the first computer program, Ada Lovelace, but the teams who programmed the earliest computers like ENIAC were actually mostly women. And the ENIAC team was, in fact, all women. How that happened is that increasing numbers of women with math degrees, who graduated from college in the 1950s, they graduated amidst a boom of workplaces who needed computer programmers to operate these machines that they had just bought for their companies. So women's perceived patience, perseverance, and fastidiousness were seen as making them ideal computer programmers. And recruiters at that time looked for skills like knitting or being able to cook precisely from a cookbook.

So my piece, "Disrupting the Industry", depicts the rise and fall of the percentage of computer science bachelor's degrees that were earned by women from 1966 to 2010. And there's a peak at 1984, which is marked in my weaving by a band of copper wire, and then the curve drops precipitously to a level close to that of 1966. So the question is, what happened in 1984? There are a number of things: Apple Macintosh was released, so there was this boom of personal computers that people could buy and then have in their homes. And it turned out that a lot of times, it was fathers who bought these computers for their young sons, thinking that they would have a father-son project to work on together. Also, around that time, movies like "WarGames" and "Weird Science" were out, and they were popularizing this image of a nerdy teenage boy-hacker. Maybe that was starting to turn girls away from computing. Also, at that time, computer science was becoming an increasingly popular major in colleges around the country, but there weren't enough people to teach this. And so computer science departments had to put enrollment caps on their majors, and they limited enrollment, often to people who already had computer experience--in many cases, boys who grew up with personal computers that they worked on with their fathers. And as recently as 2017, only 20% of technical roles at Google were held by women. As the work became associated with men, the value and pay of computer programming went up.

So my title "Disrupting the Industry" comes from this often repeated boast by tech entrepreneurs, that their app or new technology will disrupt the industry. For example, how Uber disrupted the taxi industry, or how Amazon disrupted first book selling and now all kinds of selling. And in any other context, like a classroom, disruption is discouraged. But, in Silicon Valley, this behavior is rewarded with venture capital money. The qualities valued in a tech worker and any idea of who belongs in technology have done a complete 180, and not many people remember that it wasn't always this way.