

Embedded Control and Image Processing Approach for Automated String Art

Ambili A.R.¹, Vinitha V², Job Sajan John³, Kiran Jayaram⁴, Manu Sankar⁵, Sharath K⁶

¹Assistant Professor, Dept. Of ECE, FISAT, Angamaly, Kerala

^{2,3,4,5,6}B.Tech Students (2022 Passout), ECE, FISAT, Angamaly, India

Abstract—An automated string art machine that creates tangible artwork from digital photos is presented in this study. Nail locations are generated using Bresenham's technique, saved as text files, and then successively sent to an Arduino Mega microcontroller. To carry out accurate weaving tasks, we created a hardware-based system that is automatically operated by servo motors and an Arduino Mega, along with a CNC Shield V3. For precise, effective, and imaginative string art creation, the system integrates embedded electronics with image processing.

Index Terms— String Art, Embedded Systems, Servo Motors, Arduino Uno

I. INTRODUCTION

String art is an artistic technique used to create visually appealing images by tensioning straight lines of thread or string between pins arranged on a frame. As the strings intersect and overlap, they collectively form abstract or realistic patterns that fuse into a perceptible image. Traditionally, string art has been handcrafted by artists through a labor-intensive and highly sophisticated design process. To overcome these limitations, computational approaches have been introduced, enabling the automatic generation of string art from ordinary pictures. Such methods employ discrete optimization algorithms that transform an input image into a connected graph of string paths, aiming to reconstruct the visual features of the original image as closely as possible. While artists have explored proprietary algorithmic solutions in combination with manual crafting or automated fabrication, our approach proposes a formal computational pipeline that generates string art automatically, producing grayscale images that resemble the input. The process computes the optimal path of a single unbroken string around pins

distributed on a circular frame enclosing the canvas, thereby eliminating the need for cutting and ensuring a continuous fabrication process, which can be executed by a robotic system.

II. LITERATURE REVIEW

Research in computational fabrication and image-based design has explored a variety of approaches to algorithmically encode visual information into physical forms. In the study given in [1], an FPGA-based approach was utilized to generate string art from MATLAB-processed images. The method applied Bresenham's algorithm to enhance computational efficiency, with the images initially converted to grayscale for simplification. Darker regions were analyzed to identify pin locations, and the FPGA was subsequently used to virtually render the string art on a monitor. Alexa and Matusik [2] introduced methods for generating surface-relief geometries that reproduce target images under specific lighting conditions, formulating the problem as an image-to-surface mapping that optimizes depth for shading. Their follow-up work [3] extended this concept by exploiting self-occlusion, embedding multiple images into a single object such that different viewpoints reveal distinct visuals. Building on these ideas, Bermano et al. [4] proposed *ShadowPix*, which designs geometries that project multiple shadow images under different lighting directions, while Baran et al. [5] focused on the fabrication aspects of layered attenuators to ensure such designs are physically realizable. Together, these works underscore the importance of optimization and manufacturability in transforming abstract image mappings into tangible outputs.

At the algorithmic level, Bresenham’s classic work [6] on efficient line rasterization provides a foundational method for mapping continuous line segments to discrete points, making it directly applicable to automated string art where pin-to-pin connections must be calculated with precision. Similarly, Chu et al. [7] demonstrated how optimization can balance visual fidelity with functional constraints in their halftone QR codes, a principle analogous to maintaining both recognizability and fabrication feasibility in string art. Finally, Chang and Zhou [8] developed the MILES toolkit for solving mixed-integer least squares problems, offering computational tools well-suited for discrete optimization challenges such as pin selection, path ordering, and string iteration in automated art generation. Collectively, these studies provide both theoretical and practical foundations for our work, which applies discrete optimization and algorithmic design within the constraints of physical fabrication to realize automated string art.

III. PROPOSED SYSTEM WORKING

In order to map the locations of the nails, the system first processes an input image using Bresenham's technique. A text file contains the output of this procedure. Through input iterations in the Python code, users can change parameters like the maximum line count and pin locations. Another Python script reads the text file and transforms the stored pin positions into integer values that the Arduino Mega microcontroller can use to interface with the hardware. All linked hardware is controlled by the Arduino Mega's programming. Through the serial monitor, the Arduino receives the nail positions that were extracted from the text file, enabling the automated weaving of threads over the nails. The integration and control of multiple stepper and servo motors are facilitated by mounting the Arduino on a CNC Shield V3. Micro-stepping, less port consumption, and support for more stepper motor drivers are some benefits of the CNC Shield. Thread weaving, stepper motors, servo motors, and sensors are all controlled by a special program that is uploaded to the Arduino Mega, which acts as the main microcontroller. With a 1.8° step angle and 200 steps per revolution, the NEMA 17 stepper motor is a

hybrid motor that can be used in CNC machines and automation applications because it offers precise motion control and 3.2 kg-cm of holding torque. Using drivers like the A4988 or DRV8825, the CNC Shield V3.0 is utilized as an extension board that is compatible with Arduino and supports up to four stepper motors with micro-stepping settings in order to effectively handle these motors. The Futaba S3003, a dependable conventional servo motor with 180° rotational capability and the ability to be adjusted for continuous rotation as needed, is used for servo-based movements. Furthermore, for finer mechanical operations, the Tower Pro SG90 servo motor—a lightweight 9g servo with 180° rotation capability—is utilized. It runs at 5V and is controlled by PWM using microcontrollers such as Arduino. The NEMA 17 stepper motor that rotates the system's base is controlled by an A4988 stepper motor driver. Through a rack-and-pinion system, a Futaba S3003 servo motor propels the binding arm lever's translational movement. Lastly, utilizing the same COM port and baud rate, the Python script successively transmits the pin positions to the Arduino, allowing software and hardware to work in unison for accurate thread weaving

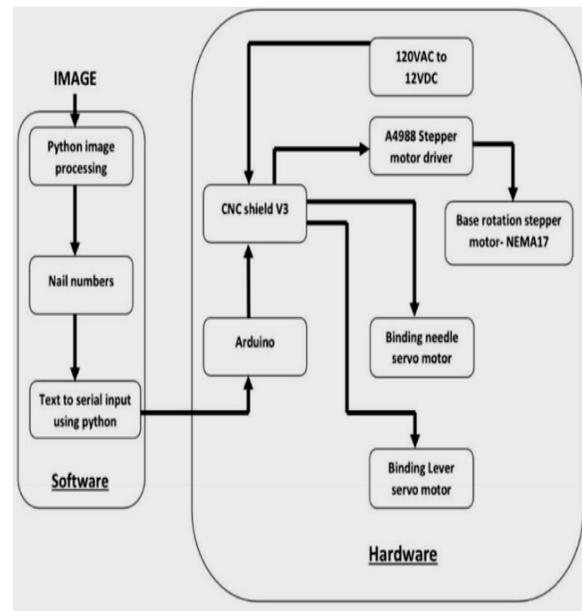


Figure 1 shows the working of the proposed system

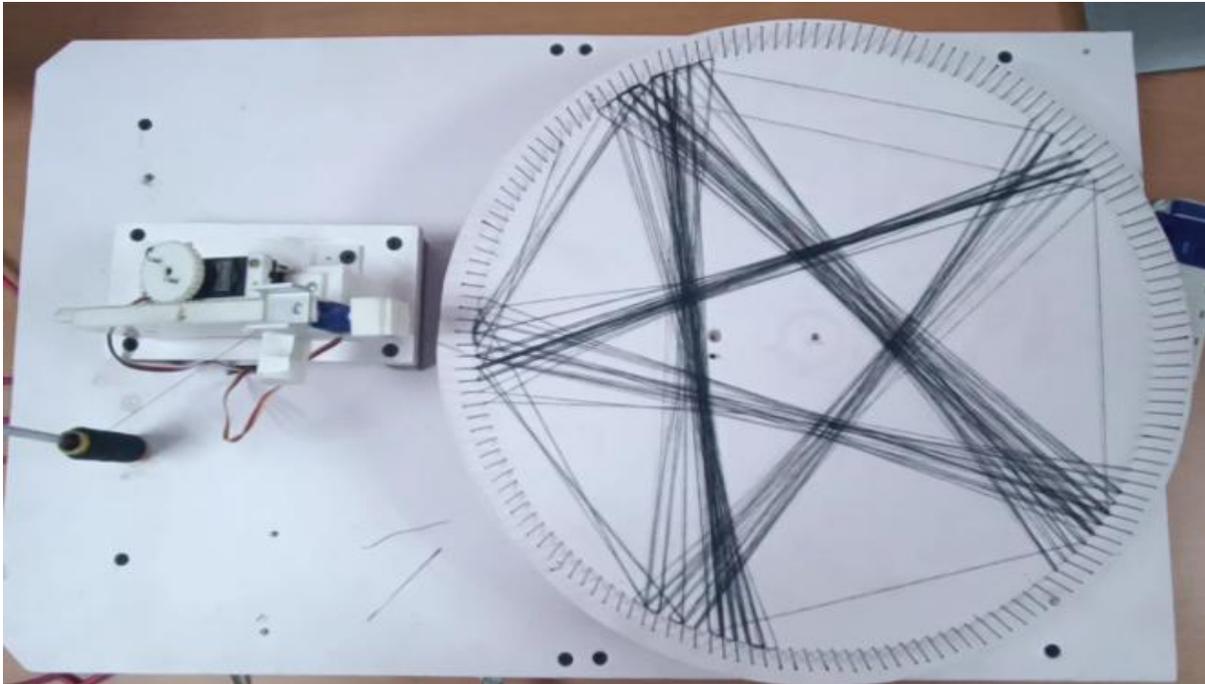


Figure 2. Prototype of the developed system

IV.RESULTS AND DISCUSSION

Figure 2. Prototype of the developed system. The process begins with the selection of a suitable image for string art. The image is chosen based on characteristics such as noise levels and distortions, without consideration for color or shape. It is resized to ensure that no part of the image is cropped during processing, and images with excessive detail are avoided to preserve the quality of the final artwork. Once selected and resized, the image is provided as input for the string art generation. The image is then processed using a Python program, which can be executed with various attributes such as canvas width, number of pins, total line count, string transparency, and image pixel width. Among these, the number of pins, canvas diameter, and total lines directly affect the final output, while the remaining attributes primarily influence the visualization of a simulated preview. Preprocessing steps include conversion to grayscale, circular cropping, and adjustment of saturation and contrast. Pin positions are calculated using

Bresenham's algorithm, a computationally efficient method for determining the optimal raster points to form straight lines. Using Matplotlib, the pins are virtually connected with lines to generate a preview of the final artwork. The calculated pin positions are stored in a text file, which serves as input for the automated hardware system. The hardware setup consists of an Arduino Mega interfaced with a CNC Shield, stepper motors, and servo motors. The Arduino code reads the pin positions from the text file and controls the movement of the hardware components. The stepper motor rotates the canvas to align each nail accurately, while one servo motor manages the longitudinal movement of the needle, and another controls the knotting of threads around the nails. Servo motion parameters, including delays and speed, can be adjusted in the Arduino code to achieve precise control. The process loops automatically, continuously threading around the nails until the final string art is complete. Communication between the computer and Arduino is maintained through the

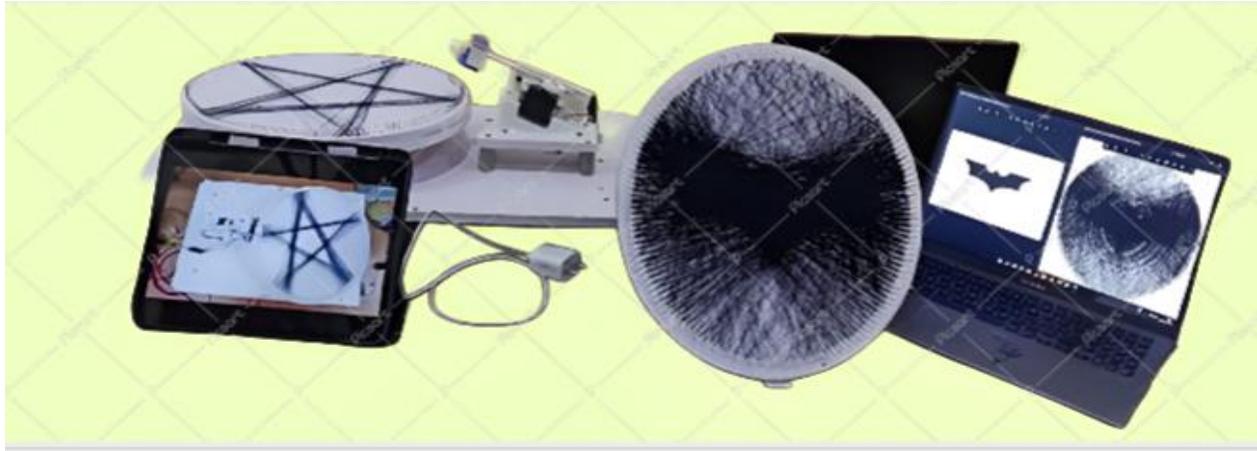


Figure 3. Screenshot of the bat pattern printed using the developed hardware

serial monitor, with the baud rate determining the speed of data transfer, ensuring reliable operation of the system throughout the process. To see the finished design, a virtual simulation of the string art is first created. Automatic threading pin locations are identified and set up for operation. These determined pin placements are used to thread the canvas to create the finished artwork. By using different photos and modifying the canvas appropriately, multiple designs can be created. To further expedite the procedure and reduce human mistakes, an automated nail-punching machine can be included. Figure 3. Screenshot of the bat pattern printed using the developed hardware.

V.CONCLUSION

To sum up, the project effectively illustrated how to design and build an automated string art machine that connects integrated hardware control with digital picture processing. Nail positions were effectively created and sent to the Arduino Mega for execution using Python and Bresenham's technique. By combining servo motors with a CNC Shield V3, weaving operations could be done precisely and automatically without the need for human interaction. The technology demonstrated accuracy and efficiency by fusing hardware automation with image processing techniques, providing a dependable and imaginative method of turning digital images into physical string art.

REFERENCES

[1] P. Wu, K. Shi, and Z. Peng, "String Art on FPGA," ECE 5760 Final Project, Cornell

University, May 2019. [Online]. Available: https://people.ece.cornell.edu/land/courses/ece5760/FinalProjects/s2019/pw444_kss223_zp83/pw444_kss223_zp83/pw444_kss223_zp83/index.html

- [2] ALEXA M., MATUSIK W.: Reliefs as images. *ACM Transactions on Graphics* 29, 4 (jul 2010)
- [3] ALEXA M., MATUSIK W.: Images from self-occlusion. In *Proceedings of the International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging - CAe '11* (New York, New York, USA, 2011), ACM Press, p. 17. 2
- [4] BERMANO A., BARAN I., ALEXA M., MATUSIK W.: ShadowPix: Multiple Images from Self Shadowing. *Computer Graphics Forum* 31, 2pt3 (may 2012), 593–602.2
- [5] BARAN I., KELLER P., BRADLEY D., COROS S., JAROSZ W. NOWROUZEZHAI D., GROSS M.: Manufacturing Layered Attenuators for Multiple Prescribed Shadow Images. *Computer Graphics Forum* 31, 2pt3 (may 2012), 603–610. 2
- [6] BRESENHAM J. E.: Algorithm for computer control of a digital plotter. *IBM Systems Journal* 4, 1 (1965), 25–303
- [7] CHU H.-K., CHANG C.-S., LEE R.-R., MITRAN. J.: Halftone QR codes. *ACM Transactions on Graphics* 32, 6 (nov 2013), 1–8. 2
- [8] CHANG X.-W., ZHOU T.: MILES: MATLAB package for solving Mixed Integer LEast Squares problems. *GPS Solutions* 11, 4 (nov 2007), 289–294. 5