Flexible and Printed Electronics

Machine vision methodology for inkjet printing drop sequence generation and validation

To cite this article before publication: Fahmida Pervin Brishty *et al* 2021 *Flex. Print. Electron.* in press https://doi.org/10.1088/2058-8585/ac1c5c

IOP Publishing Flexible and Printed Electronics

Machine Vision Methodology for Inkjet Printing Drop Sequence Generation and Validation

Fahmida Pervin Brishty and Gerd Grau

Department of Electrical Engineering and Computer Science, York University, Toronto, Canada

Email: grau@eecs.yorku.ca

***Abstract*—** Inkjet printing technology for printed microelectronics suffers from a number of non-idealities due to unwanted ink flow on the substrate. This can be mitigated and pattern fidelity can be improved by using an optimized drop placement sequence in contrast to the standard raster-scanning approach. However, it is challenging to auto-generate such printing sequences for complex printed patterns. Here, the generation and evaluation of the printing sequence are turned into a computer-vision problem. The desired printed pattern is taken as an input image and converted into a printing sequence using contour, symmetric, and matrix sequencing and corner compensation. After printing, pattern defects are detected by automated image processing to evaluate the printed pattern against the designed ground truth image and to determine the best possible algorithm for printing sequence generation for different pattern types. The machine vision-based experimental approach identifies the best solutions for solving the printing and defect optimization problem in terms of precision, recall, and accuracy. This methodology will enable the automated design of electronic circuits for applications such as wearable sensors, low-cost RFID tags, or flexible displays.

***Keywords*—**Computer Vision, Printed Electronics, Inkjet Printing, Raster Printing, Vector Printing, Printing Defect, Pattern Segmentation.

1 Introduction

Printed electronics is an emerging manufacturing technology complementary to traditional silicon microelectronics due to its advantageous properties such as low-temperature processing, absence of vacuum processes, compatibility with numerous cheap, flexible substrates, and large-scale roll-to-roll processability [1]. Multilayered inkjet patterning provides interesting possibilities for on-demand fabrication of passive

electronics such as inductors, capacitors, and resistors [2,3]. Radio frequency antennas, interconnects, active, and passive components can be printed using conductive inks on flexible substrates for low-cost consumer package tracking radio-frequency identification (RFID) tags [4–6]. The liquid crystal display (LCD) industry has started adopting inkjet-printed active-matrix color filter layers [7]. Hybrid inkjet-printed structures can also benefit large-area, low-cost, multicolor organic light emitting diode (OLED) manufacturing [8]. There have been several demonstrations of printed transistors using inkjet patterning on flexible substrates such as plastic or paper, avoiding complex traditional fabrication steps such as metal evaporation and photolithography [9–11]. Inkjet printing, as illustrated in Figure 1, allows patterns to be customized on-the-fly.

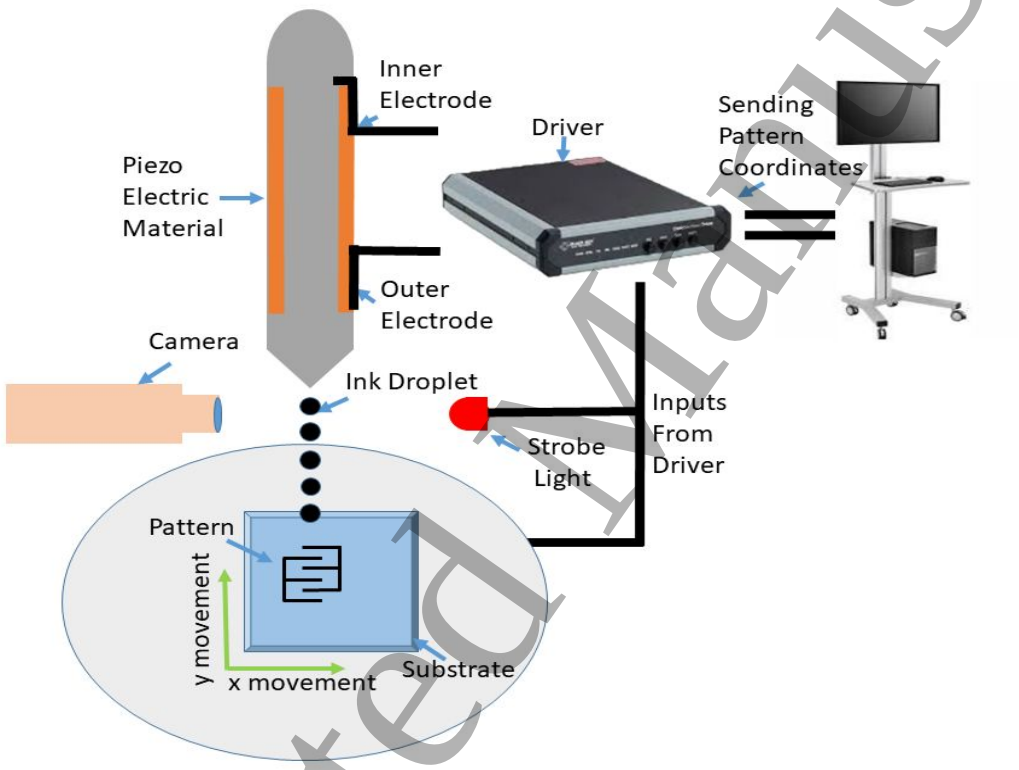

Figure 1: Schematic of the inkjet printer setup. The substrate moves relative to the drop-on-demand nozzle to create the final pattern, which is made up of individual drops that coalesce. The sequence in which drops are printed is instrumental in ensuring excellent pattern fidelity.

Patterning of electronic materials (e.g., conductors, semiconductors) at micrometer scales can suffer from several non-idealities due to unwanted fluid flow on the substrate, especially when control is limited over the surface to be printed on as in complex electronic devices. These two-dimensional deviations arise in the form of bulging, scalloping, and bead segregation at the beginning, junctions, and over the intended pattern boundaries [12]. The printed dimensions may deviate significantly from the designed dimensions due to substrate wettability and ink properties. This has a severe influence on the optimization, consistency, and

accuracy of printed electronics manufacturing. Moreover, electrical resistance depends on pattern length and width after printing. It changes significantly with the orientation of the printing direction relative to the pattern orientation and the edge roughness of the pattern [13,14]. It is, therefore, essential to print patterns without defects and make the printing process readily adjustable with respect to the underlying large-area flexible substrates. There has been significant work to understand the fluid mechanics of these non-idealities and avoid them for simple designs such as lines or squares by changing the location and order in which drops are printed. Inkjet pattern control has been thoroughly investigated for lines [12,15–21] and 2D shapes [22–24] by characterizing ink and substrate properties and manipulating drop and line-to-line spacing [25,26], which can enhance printed morphologies. However, automated implementation of optimized drop and line-to-line spacing for arbitrary complex circuit patterns and industrial inkjet-manufacturing is still a challenge. Line and film edges with consistent corners can be attained by managing coalescence speed, drop spacing, and the ratio between viscosity and surface tension [15,17,25]. Additional co-solvents [22], contact angle hysteresis [15,19], and substrate roughness [23,24] can further improve the definition of rectangle patterns that are raster printed, although these methods cannot be applied in all systems. The morphology of micro-scale liquid tracks, especially bulging, has been investigated before using fluid dynamics methods [15,19]. Pattern segmentation and symmetric printing of segments can inhibit bulging in line patterns compared to traditional raster printing [20]. Preprinting the contour of any feature improves corner morphology through enhanced pinning and additional anchoring in front of each segment can significantly subdue the bulging effect [21,26]. Other works have demonstrated compensation techniques in all-inkjet-printed processes to improve multi-line intersection thickness irregularities and minimum line-to-line separation [18,27]. Most of these reports describe pattern correction by adding or removing pixels from particular areas to improve downscaled print resolution without the implementation of drop ordering. A practical problem is that compensation is not applicable for shapes consisting of fewer than five droplets width and length. The pixel compensation method is not automatic and intelligent for arbitrary patterns. However, practical applications demand intricate patterns, for example, with multiple corners. Manually optimized drop sequence design using the above-described sophisticated methods for such complex patterns is challenging. Thus, in practice, most printers use simple raster printing where the nozzle traces successive rows without consideration for the desired pattern.

Here, we propose to deploy machine vision algorithms for optimizing computerized printed electronics manufacturing. The automated printing process is divided into two major parts both using computer vision algorithms: automated generation of printed drop sequences and evaluation of the printed pattern quality. The goal is to automatically generate improved patterning steps based on automated detection and elimination of

printing inaccuracies. We demonstrate several algorithms to convert different design patterns into drop sequences. By changing the order of deposition of droplets and changing the coalescence forces between droplets, the ink, underlying substrate, and solvent do not need to be changed, which is often undesirable when printing functional materials for printed electronics. The optimization is versatile for any shapes and patterns as design rules are implemented by the machine vision algorithms and no theoretical modelling of the fluid mechanics is needed. These algorithms are based on the above-described methods that have mostly been demonstrated for simple shapes and are extended here to be automated for arbitrary patterns. We validate and compare the various methods by micro inkjet printing with a conducting ink. Our results show significant improvements in print quality over simple raster printing. The quality of the printed patterns is evaluated using image-based pattern recognition and visual inspection techniques. Methods from the inspection of printed circuits board (PCB) manufacturing [28–30] are adopted to the inspection of inkjet-printed patterns. Printed patterns are automatically denoised and detected to quantitatively compare them with ground truth designs. We also use skeletonization to detect defects that would be catastrophic for circuit operation, i.e., open and short circuits. In summary, the automated printing process is divided into two major parts: computer vision informed optimized printing pattern generation through pixel sequencing and evaluation of the printed pattern. The vision-based electronics printing workflow is shown as a flowchart in Figure 2.

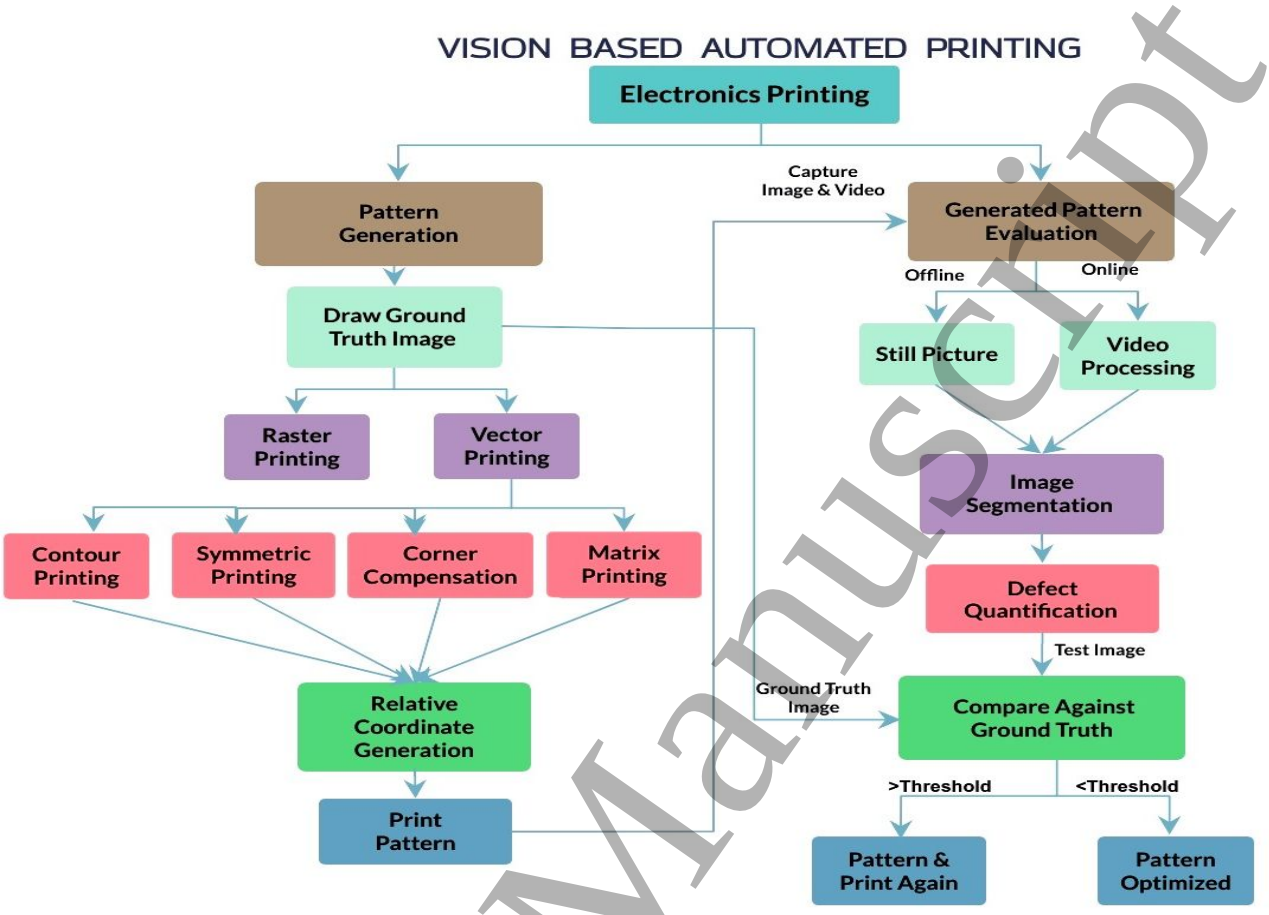

Figure 2: Workflow for automated electronics printing using machine vision. Drop sequences are generated with different vector printing methodologies and used for printing. Different methodologies are evaluated by processing images of the printed patterns and comparing with the ground truth.

The first step in the pattern generation process is to design the desired patterns, i.e., the ground truth patterns using conventional drawing tools (Adobe Illustrator, Electrical Computer-Aided Design (ECAD)). There are two ways to describe the dimensions of a circuit for printing: micrometers or number of drops. These scales are convertible from one to the other for a given drop spacing (DSp) whose optimal value depends on the ink material and substrate interaction. Specifying print dimensions in the pixel scale where each pixel represents one drop can be helpful to reduce fractional calculations. To convert back to micrometers only requires runtime multiplication with any desired drop spacing. The generated images are saved as black and white digital images for the next sequencing step. The goal is to determine the order in which the printer prints the drops (i.e. pixels) in the pattern. As shown in Figure 2, in contrast to traditional raster scanning, four different vector sequencing models were implemented using computer vision processes. The final output of the algorithm are relative coordinates between subsequent drops, which defines the drop order or sequence for the practical printing on the substrate. The success of these strategies is evaluated after printing using image segmentation, and the number of printing defects is quantified using precision, recall, and accuracy against the designed ground truth pattern. This evaluation scheme could be employed in manufacturing to determine if a print has acceptable quality below a certain threshold or if there are too many defects. We study shapes of different types (filled patterns and non-filled line patterns) and of different size.

2 Methods

2.1 Detection of Important Input Features

From a fluid mechanics perspective during printing, the most important features of a printed shape are corners, orientation relative to the print direction, edges, and contours [13,31]. For instance, patterns with a closed contour exhibit better printing results when the outline is printed before the drops on the inside. The contour creates a boundary for the internal fluid flow. Printing it first makes the edges smoother. Pattern optimization strategies are explored here to reduce printing defects by engineering the printing sequence based on these features. The features need to be identified from the drawn ground truth image in terms of image processing descriptors. The patterns are designed as binary images where zeros denote the printed pattern and ones the background.

Neighborhood-based operations are performed on each input pixel and its surrounding area to calculate and combine edge, contour, and corner features during vector sequencing. This includes different filtering operations with specific kernel sizes and weights. Edges are continuous or discontinuous pixels defining the physical extent of an object representing the maxima of intensity gradient obtained from edge detection techniques. Contours are a fit through the adjacent pixels of the detected edge map to locate the coordinates

of the meaningful boundary of an object [32]. The performance of contour detection is solely dependent on the previously detected edge map. For a discontinuous edge map, detected contours will be interrupted as well. The perfect edge detector was found after running several different edge detectors on test patterns to ensure that the detected contour is perfect with continuous x, y coordinates forming a closed pattern. A detector pipeline was established to execute each of the distinct algorithms. Several detectors were implemented to identify the best-suited edge detector, as presented in Figure 3, for filled shapes (lines or rectangles that are more than one drop wide) and line patterns (lines that are one drop wide).

Many edge detectors have been developed in the past, some based on the first derivative of the image (Sobel, Canny), some on the second derivative (Laplacian of Gaussian (LOG)) [32,33]. We studied filled and line patterns that are small-scale (only a few drops) and large-scale. For all pattern types, the Canny edge detector performs poorly, as shown in Figure 3 (ii). Following calculation of the derivative, it detects edges by applying non-maximum suppression (discard pixels with gradient less than its neighbors), thresholding, and filtering out weak edges that are not related to strong edges by hysteresis. For improved Canny edge outputs, the thresholding requires intense optimization. For all three pattern types, this results in irregular edges, and the intersections are lost. The Canny edge map loses corners, junctions, and blobs. For the large-scale spiral-shaped case, it misses numerous points on the edge map. Sobel edge detection uses the first derivative of the image with linear filtering. It combines horizontal and vertical filtering to locate the edges at the maximum and minimum values of the first derivative [32]. It behaves similarly to Canny except it does not apply non-maximum suppression to differentiate between thick and thin edges. The obtained edge map is not reduced to a single pixel boundary and not capable of identifying a distinct contour. Instead, it results in multiple segregated contours, as shown in Figure 3 (iii). LOG detects edges from zero crossings of the second derivative of the image as shown in equations (1) and (2). Gaussian blurring removes noise from the image and creates a defined transition from low to high pixels around the edges. Then, these edges are efficiently recognized by the Laplacian of Gaussian operation in equation (2). Figure 3 (iv) shows that LOG localizes edges, including corners, and blobs well for all image scales, for both filled and line patterns. Therefore, LOG is chosen for contour fitting going forward.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

$$LoG(x,y) = \frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

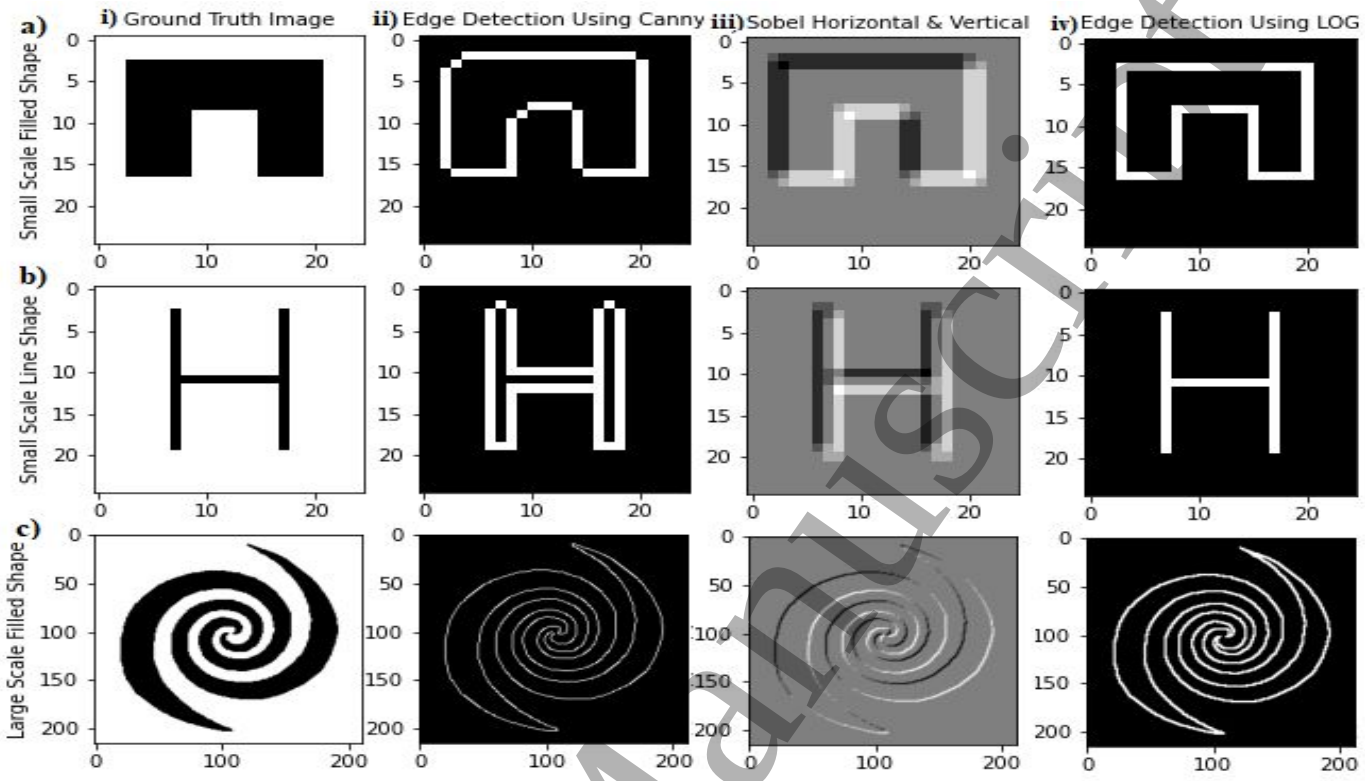

Figure 3: (i) Ground truth images for a variety of possible patterns (a) small-scale filled shape, (b) small-scale single-line pattern, (c) large-scale filled structure. As a basis for contour fitting, edge detection results are shown employing several edge detectors (ii) Canny, (iii) Sobel, (iv) Laplacian of Gaussian.

Next, to extract corner information from the whole pattern, Harris corner detection is found to work best. Two widely used corner detection schemes, contour-based and intensity-based, have proven consistent for detecting corner coordinates precisely. Our application requires reliable corner detectors with low parameter dependency and without a priori knowledge of the exact number of corners present in the ground truth image. Despite having the contour information from our feature detection steps, corner extraction from the contour fitting is not ideal as it requires knowledge of the exact number of candidate corners along with prior knowledge of contour type (closed or open) to localize them correctly. The intensity-based Harris corner detector does not require knowledge of the number of corners and contour pattern, and is also computationally much faster. It is independent of local features and noise, and it can be improved by thresholding. The pattern type (filled, non-filled), foreground and background information (edges and corners) are detected through this image pre-processing pipeline. All of the printing sequence generation algorithms are developed based on this feature information set.

2.2 Drop Sequence Generation

Different vision algorithms are developed for each unique drop sequencing method with the above discussed pre-processing pipeline to extract and sequence the recognized features. The designed pattern is represented in the form of a pixel matrix where every pixel represents the location of a drop. For generating a printable drop sequence, the original pixel representation is manipulated using point and neighborhood operations. The drop sequence is a set of relative vector coordinates x, y between successive drops. The most conventional technique for inkjet printing is raster-scan-based printing where the pattern is printed line by line (see Figure 4 (b)); however, it can lead to defects when printing in different orientations (rotated patterns). The raster approach to electronics printing does not work well for intricate patterns with rapid edge and corner transitions. Raster sequencing is a point transformation as the output pixel coordinate sequence is determined solely as a function of the input pixel value (0 or 1) corresponding to that location. Conversely, vector sequencing incorporates both point and neighborhood operations for identifying edge, corner, and contour features to generate the sequencing output, as shown in Figure 4 (c-f). Four types of vectorizing were studied (contour (parallel and radial filling), symmetric, corner compensation, and matrix) and are discussed in the next sections. Some identical pre-processing steps are used in all four vector sequencing models. After thresholding and smoothing, edges are detected through LOG edge detection followed by border contour extraction, and contour coordinate storing.

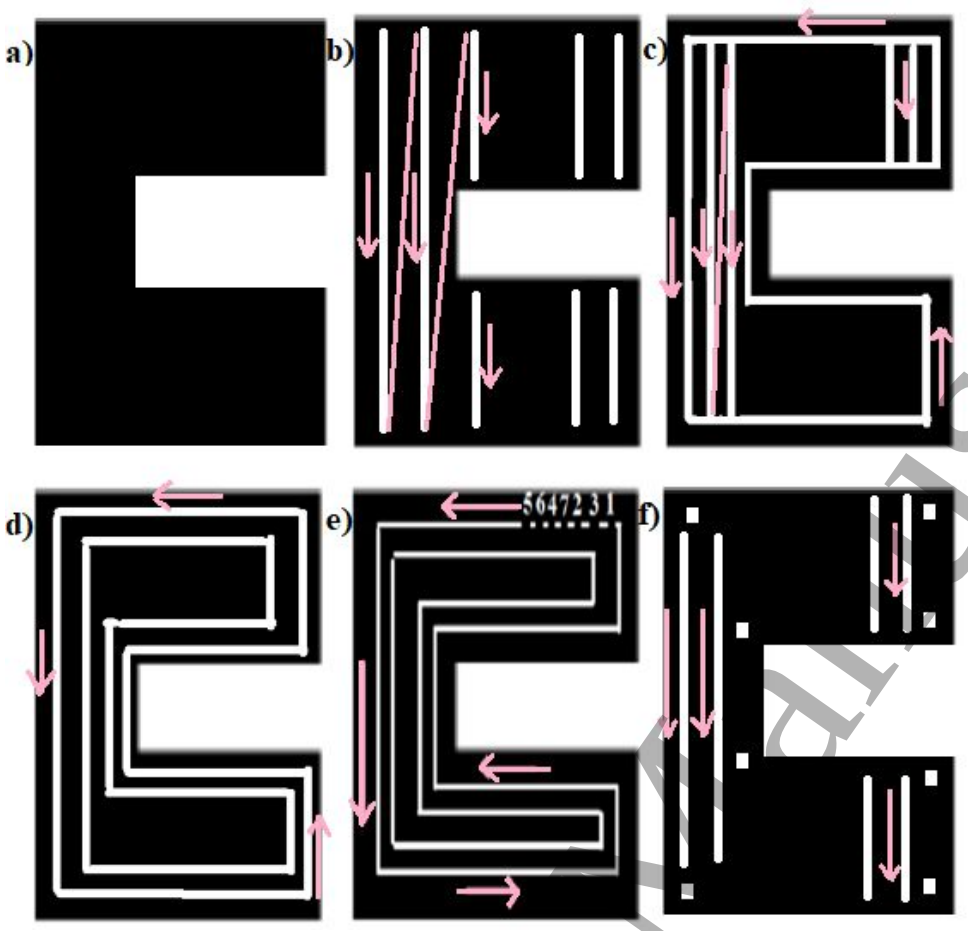

Figure 4: (a) Input image that is printed with different drop sequencing methods: (b) Raster pixel sequencing. The printer always prints one entire column before moving on to the next column. (c) Contour vectorization with parallel filling. The printer first prints a boundary and then fills in the center with a raster pattern. (d) Contour vectorization with radial filling. The printer first prints a boundary and then fills in the center in a spiral path. (e) Vectorizing using radial filling but with symmetric drop sequence. (f) Corner compensation deleting corner pixels and filling with raster sequencing.

2.2.1 Contour Vector Sequence Generation

With the contour vector method, the border of the pattern is printed first before filling the inside to reduce bulging and irregularities in the outline of the printed pattern [26]. The pattern outline (in computer vision terms contour) is significant since it defines the shape and encloses the boundary for the ink fluid flow. The pre-processed image edges need to be detected as the first step to implement boundary filling. As described in Figure 3, LOG captures the complete edge information well and is used as a preliminary step for contour vector sequencing. The detected edges are fitted to form complete contours. Contour coordinates are extracted in an clockwise sequence and saved in a text file, as shown in Figure 5 (a-d). The border contour is detached

from the original image, and the remaining pixel coordinates of the subtracted image need to be printed. We studied two methods to fill in this region on the inside of the contour: parallel and radial filling. In parallel filling, the remaining drops are simply collected with the raster method (see Figure 5 (e)). Finally, the boundary and inside coordinates are merged and converted to relative coordinates and stored as a text-based command file for the printer. For radial filling, the border contour is detected and removed continuously from the original image in a loop until a blank image is formed. Each time, contour coordinates are merged with the previous ones (see Figure 5 (f)). In the end, relative coordinates are calculated and saved as a printer input file. For filled shapes (i.e., more than one drop wide patterns), radial and parallel sequencing result in different printing sequences and the print result can be dissimilar; however, non-filled forms (i.e., one drop-thick line patterns) do not have multiple contours, so the generated sequence is the same for both algorithms.

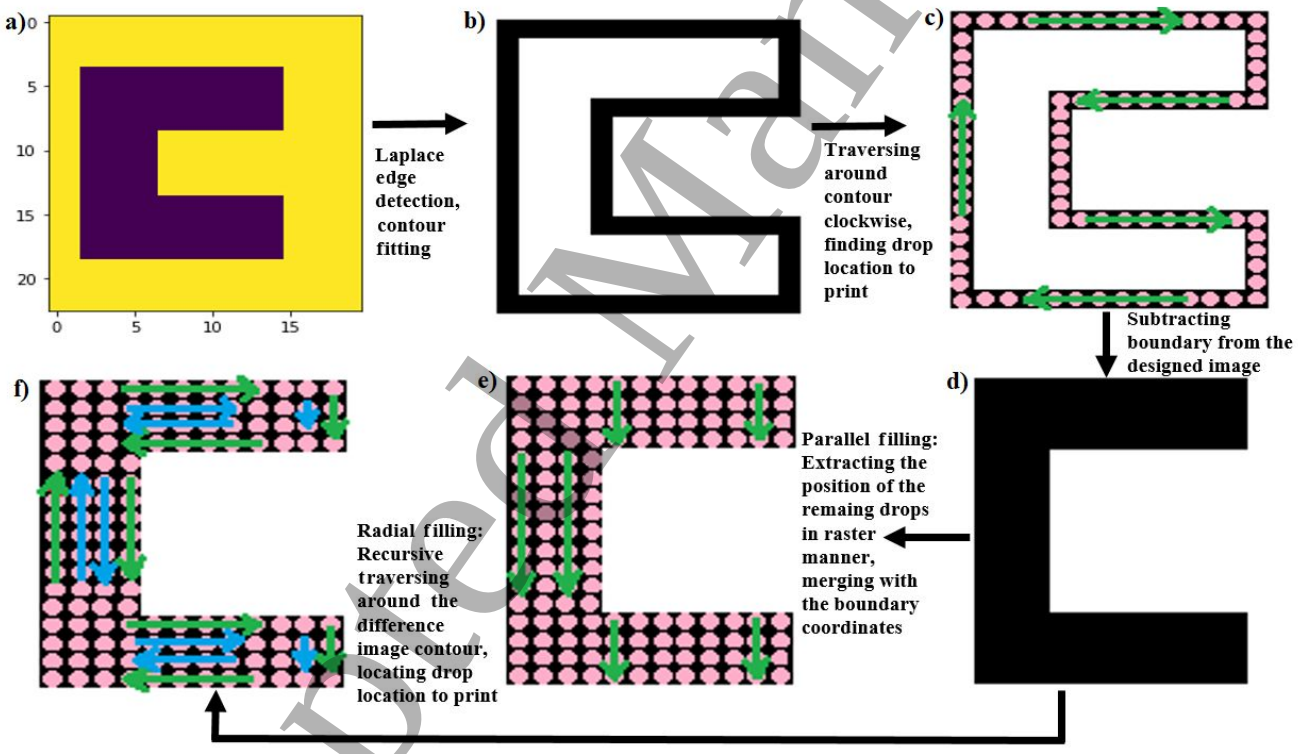

Figure 5: Pixel sequencing steps for contour printing with parallel and radial filling. (a) Designed pattern. (b) Detecting edge and fitting contour through the border pixels. (c) Extracting the boundary pixel coordinates x, y, collecting them clockwise and calculating the relative coordinates using drop spacing. (d) Subtracting the edge pixels from the ground truth image to get the difference image. (e) For parallel vectorization, the difference image pixel coordinates are extracted through column-major order based raster sequencing. (f) For radial vectorization, border pixel coordinates of the difference image are removed recursively and merged together to generate the final relative coordinates.

2.2.2 Symmetric Vector Sequence Generation

Inkjet-printed patterns often bulge at the beginning of lines and intersections between lines. This arises from the Laplace pressure difference between the already printed track and the newly added droplets. Consequently, linewidth can be very dissimilar in those regions. Bulging can be avoided with a segmented and symmetric printing methodology. In conventional raster printing, drops are positioned one after another along the pattern line without controlling for pressure imbalances, which can distort the printed pattern with irregular bulging. The imbalance can be considerably improved by fragmenting single-drop wide line patterns into three-drop long segments. Following the printing of the outer two drops of each segment, the central drop is printed and does not experience a pressure gradient due to the symmetry of the segment. Subsequently, the three-drop segments are joined with a connecting drop while maintaining pressure equilibrium on either side of the pattern [20].

Figure 6 (a) compares the raster and symmetric drop ordering methodologies. Raster uses a constant DSp in all directions, while symmetric has two distinct parameters (DSp and connecting drop spacing (CDSp)) and a different droplet print order, as shown in Figure 6 (a). DSp is used within each three-drop segment and CDSp is used between two consecutive segments. The second drop is placed two DSp away from the first drop and the third drop is placed in the middle between the first two. Then, the next segment starts with the fourth drop placed at a distance of (DSp+2CDSp) relative to the center of the first segment (the third drop), leaving 2CDSp of vacant space between two segments for a connecting drop to be filled in after all the segments have been printed. Previous work has experimentally shown that CDSp should be smaller than DSP, and the optimum CDSp varies between 0.6 and 0.95 times DSp for different substrate and ink combinations [20]. Here, the symmetric printing process is implemented for arbitrary patterns using a vision pipeline shown in Figure 6 (b). The pattern in binary format is passed through the symmetric sequencing algorithm. The symmetric pixel sequence implementation is different for filled and non-filled cases. A limitation of symmetric printing is that pattern dimensions are constrained to multiples of segments of three drops plus connecting drops. Filled forms are composed of multiple neighboring lines in the same or alternate directions. Each line in the same direction is extracted separately and passed through the symmetric sequencing steps. The contour of non-filled patterns is obtained using the parallel vectorizing sequence shown in Figure 5. The whole contour is then processed with the symmetric pixel vectorization routine.

A network graph is constructed as shown in Figure 6 (c&d) for filled and unfilled patterns with the generated symmetric drop order as node data, and the distance between two consecutive pixels as the edge attribute. The two phases of symmetric vectorization described in Figure 6 (b) are implemented. In the first step, the constant

DSp pixel map (Figure 6 (c&d)(ii)) is transformed to DSp and CDSp mapping in micrometer space as shown in Figure 6 (c&d)(iii). Every fourth pixel of the pattern array (filled pattern: raster sequence, non-filled pattern: contour sequence) is considered as a connecting drop. The distances of the connecting drops from the previous and next drops are changed to CDSp, and the rest are set to DSp. CDSp is smaller than DSp; therefore, there is a shift in the mapped pixel coordinates. This should be taken into account when designing pattern dimensions for symmetric and segmented printing. After mapping, symmetric pixel ordering (1,3,2) is implemented as shown in Figure 6 (c&d)(iv). The procedure retains the first pixel coordinate at the same location. The previous coordinate of the third pixel receives the second drop. The third drop is printed in between at the previous coordinate of the second drop. All of the pattern coordinates are rearranged with this repeated sequence array.

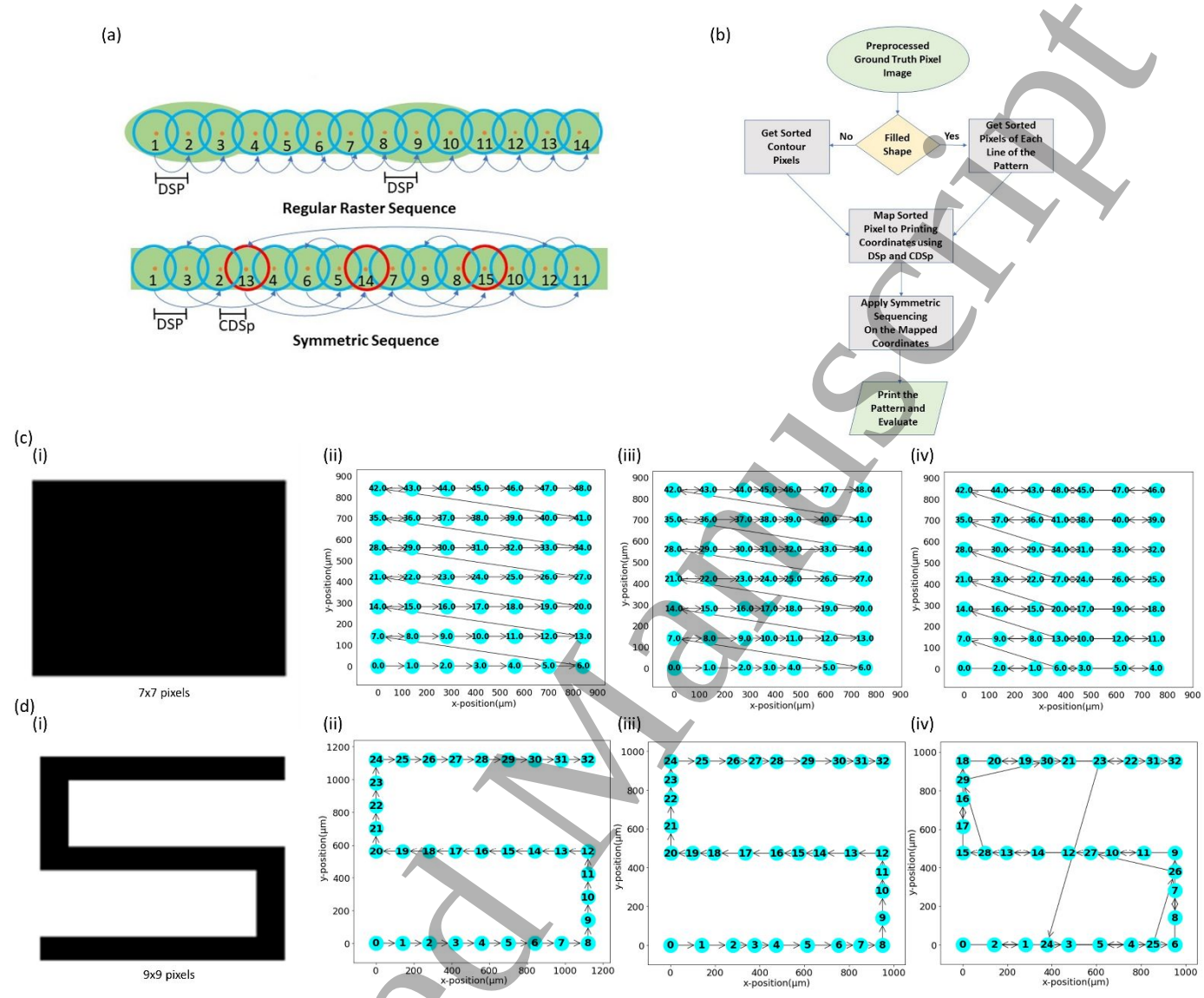

Figure 6: Symmetric and segmented printing methodology. (a) The top row shows a line of fourteen drops printed with traditional raster printing. It exhibits bulging at the start of the line and at regular intervals. The bottom row displays symmetric sequencing of fifteen drops with three segments and three connecting drops. The numbers indicate the order in which drops are printed. (b) Symmetric pixel sequencing algorithm. (c) Example filled pattern and (d) example non-filled pattern: (i) Ground truth image, (ii) network graph for raster sequencing with DSp=140 μm, (iii) symmetric pixel mapping before sequencing with DSp=140 μm, CDSp=0.7*DSP, (iv) final mapping with symmetric ordering

2.2.3 Corner Compensation

For filled shapes, the patterns can bulge in the corner, even though they are patterned through vectorization. A potential solution to this problem is the deletion of some pixels from the corner region before applying

raster or vector sequencing (see Figure S1) [18,27]. Here, a vision method is adopted to automatically apply this strategy and ultimately determine how many pixels should be deleted and from where. Initially, the original pattern is loaded, pre-processed, and then filtered through a bilateral filter to blur the image followed by the Harris corner detection procedure. Subsequent thresholding and tuning are applied to the detected corner positions to refine and get the exact coordinates. When the designed pattern dimension becomes small (approximately <15x15 pixels), the number of detected corners might be more than the real number and not in the precise location. For such cases, the corner detection efficiency is further improved with an additional distance measurement step to remove the extra corners within a threshold distance from each other. The detected corner pixels are removed from the design using a shift operation. The optimal shift depends on the input pattern. The design with removed corners is finally sequenced using vector or raster sequencing with the algorithms described in sections 2.2.1 and 2.2.2 to generate the relative coordinates for printing.

2.2.4 Matrix Vectorization

Another method to prevent unwanted ink flow within large, filled shapes, is sub-dividing them into a matrix of smaller blocks. Two different matrix vectorization schemes have been developed: one sequentially prints block by block; another interleaves the printing of the different blocks referred to as “multilevel” matrix by Tekin et al. [34]. The matrix block dimensions are defined in terms of the number of droplets in the x- and y-direction, such as 4x4 or 2x2. The optimum drop spacing and block size have the potential to generate prints with better pattern definition. Figure 7 (a)(i) shows an example of a rectangle pattern of six drops (x-direction) by eight drops (y-direction) sequenced with twelve 2x2 matrix blocks and Figure 7 (a)(ii) shows the same pattern sequenced with two 4x4 and two 2x4 matrix blocks. Figure 7 (b)(i&ii) are schematic illustrations of the 2x2 and 4x4 matrix vectorization print processes respectively at the block level. The order of drop deposition is denoted with blue numbers. Printing starts at the lower left corner and ends at the upper right. The first 2x2 matrix block, “block 1”, is printed completely with all four drops. Once “block 1” is finished, the printer moves to “block 2.” The four drops are deposited in raster sequence from low x, y coordinates to high x, y coordinates. The deposition is continued until the last of the 2x2 blocks, “block 12”, is complete. In the case of interleaving the matrix blocks drop by drop, as shown in Figure 7 (c)(i&ii), the first drops of each block are printed first (labelled as 1, 2, 3, ..., 12), then the second drops are printed (labelled as 13, 14, 15, …, 24). This continues until all the drops in the blocks have been sequenced.

To improve on this method, it can be combined with the contour printing method. The pattern contour border is printed first, and then the remaining center of the pattern is sequenced with the matrix vector method.

Schematic diagrams of contour matrix printing (block by block and interleaved drops) with 2x2 and 4x4 blocks are shown in Figure 7 (iii&iv). The contour drops are bounded with green rectangles, while the matrix blocks are shown with red lines.

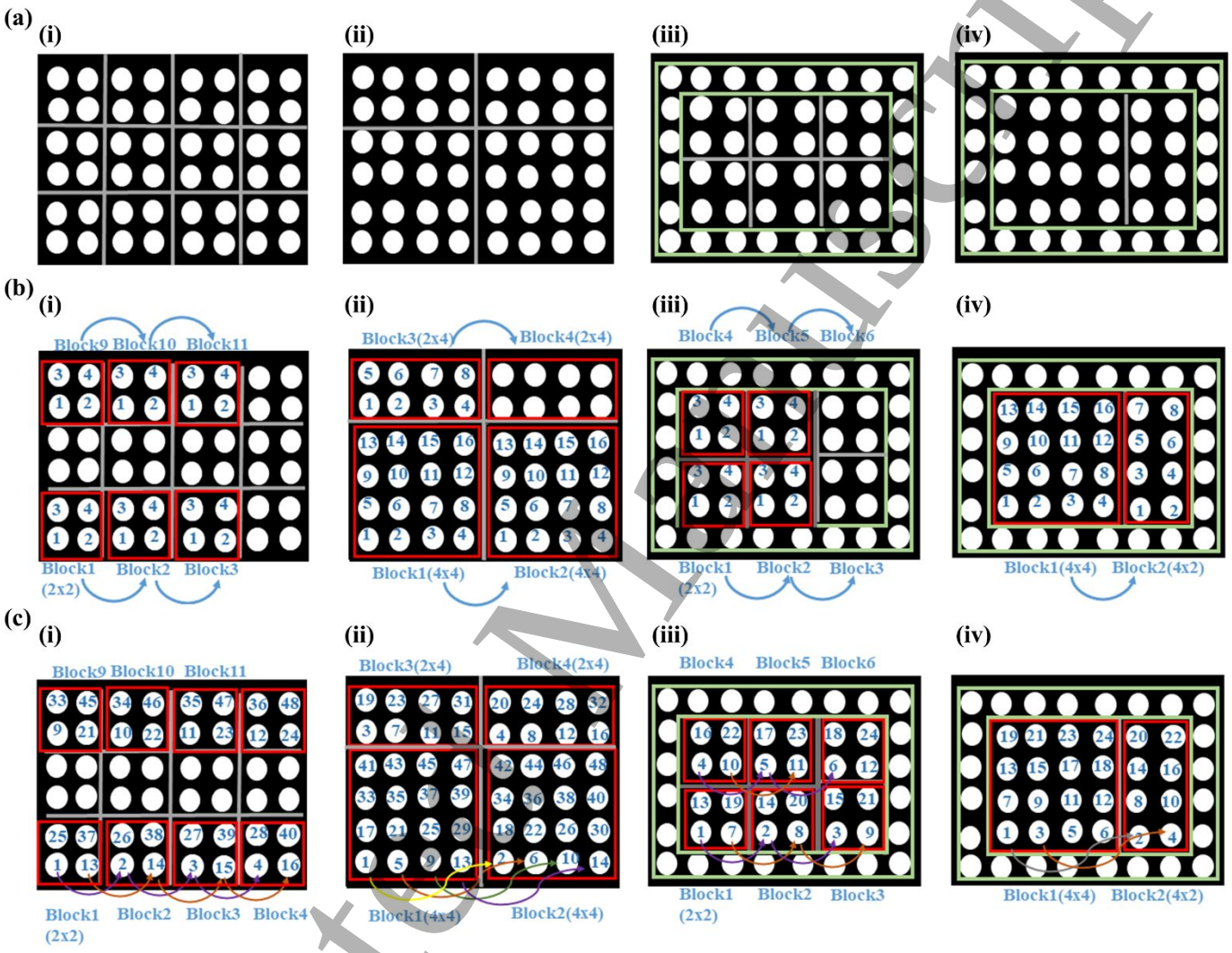

Figure 7: Schematic illustration of matrix and contour matrix vectorization, block by block and interleaved drops. (a) 6x8 drop rectangle pattern is divided into (i) twelve 2x2 matrix blocks, (ii) two 4x4 and two 2x4 matrix blocks, (iii) its contour and for the inside of the pattern 2x2 matrix blocks, (iv) its contour and for the inside of the pattern one 4x4 block and one 2x4 block. (b) Block by block matrix drop ordering. The sequence of drop deposition is denoted with a blue number. Red lines surround the matrix blocks. Once all the drops in the same block are deposited, the next block is sequenced until all blocks are complete. Subfigures (i)-(iv) correspond to the same pattern divisions as in (a). (c) Schematic illustration of matrix vectorization with interleaved drops between blocks. All the drops at the same location within all blocks are printed sequentially before moving to the next location. Subfigures (i)-(iv) correspond to the same pattern divisions as in (a).

In order to apply the matrix vectorization approach to arbitrary patterns, an automated method of partitioning the pattern into smaller repeating blocks is required. Many partitioning algorithms have been developed for geometry partitioning with rectangles [35,36] and triangles [37]. This method has already been used for very large-scale integrated circuit (VLSI) layout GDS files for data compression. For printed electronics, this approach is new. We studied two types of partitioning: by maximum area rectangle and by clockwise/ anticlockwise pattern partitioning, as shown in Figure 8. The maximum area method finds the largest rectangle within the pattern and removes it. This is repeated recursively until the entire pattern has been divided into successively smaller rectangular blocks. Another method is to segment the pattern clockwise or anticlockwise into rectangles. For the anticlockwise pattern partitioning, we start traversing from the top rightmost point of the pattern and traverse downward, with fixed x- and decreasing y-coordinate, until we reach any edge. After hitting the edge, we start traveling to the left until we hit another edge. Then, we start moving upward, until we reach the final edge. From there, we move in the right direction until reaching the starting point. This process is carried out recursively until all the block partitions are collected. The same partitioning process can be carried out from the top left corner in a clockwise way.

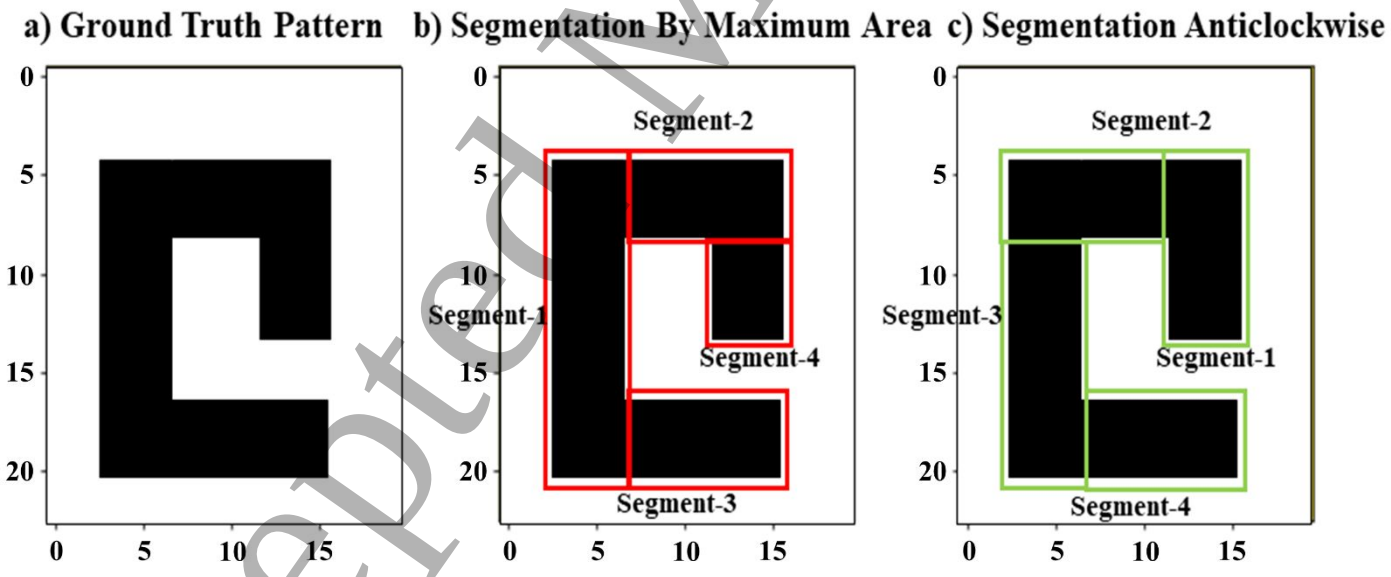

Figure 8: Segmenting intricate patterns before applying block vectorization. (a) Ground truth design pattern. (b) Segmenting pattern by maximum rectangular area. (c) Segmenting pattern with anticlockwise rectangles.

2.3 Experimental Methods

All the drop sequence generation methods were validated experimentally using a custom-built inkjet printer. Independent x- and y-stages allow vector motion and any desired drop sequence to be printed. A piezoelectric inkjet nozzle from Microfab Technologies Inc. (MJ-ATP-01-060, 60 μm diameter) was used. The substrate was glass (Fisherbrand™ Premium Cover Glass). Glass slides were cleaned in an ultrasonic bath for 30 minutes each in isopropanol and deionized water, and blow-dried in between and afterward. The ink is a commercial silver nanoparticle ink (ANP DGP 40LT-15C) with particle size 35 nm, viscosity 16 cP, and tri(ethylene glycol) monoethyl ether as the major solvent. The vapor pressure of the solvent is low, ensuring stable jetting without ink drying in the nozzle and avoiding the coin stacked morphology, which is undesirable in many microelectronic devices due to the resulting surface roughness. This ink-substrate combination was chosen as a representative test case because the printed patterns exhibit defects when using traditional raster printing. The goal of this work is to demonstrate that automated drop sequence generation methods can mitigate these problems even for unfavorable ink-substrate combinations. It is beyond the scope of this article to investigate other material combinations in detail; nevertheless, we also demonstrate successful printing with this method for a different ink, a water-based copper oxide ink (Novacentrix Metalon ICI-002HV), on the same substrate as supplementary information. After printing, the ink is subsequently dried for 30 minutes on a hotplate at 60°C. Images were taken with a camera mounted on the printer for each of the sequencing models to evaluate print quality. Printed pattern thickness was measured with a stylus profilometer (Alpha-Step D-600, KLA-Tencor, Milpitas, CA).

2.4 Printed Pattern Evaluation Methods

Before analyzing pattern quality, all the captured images of printed patterns are passed through background noise elimination and region of interest (ROI) cropping. Then, precision, recall, and accuracy are calculated and averaged.

2.4.1 Pre-Processing Steps for Printing Evaluation

Figure 9 (a) shows the pre-processing steps for defect detection from captured images. The images are taken by a camera mounted on the printer and passed through a pipeline for stitching if required. Noisy backgrounds require background subtraction and then denoising of the extracted foreground after RGB to gray scale conversion with the Rudin, Osher, and Fatemi algorithm [38]. There exist a number of feature-based image segmentation algorithms to subtract the noisy background from the foreground, such as color histogram-based, edges- or boundary-based (based on similarity), or region-based (discontinuities) [39,40]. Printed

electronics substrates, materials and patterns can be very diverse. Prior knowledge helps in extracting the foreground pattern. For instance, substrate background can be transparent (e.g. glass) or opaque (e.g. coated PET) with varying color of the foreground ink. Two case examples are shown in Figure 9 (b) with multiple different image segmentation algorithm results. The transparent substrate creates a noisy background. In this case, region-based clustering (k-means clustering, Gaussian mixture model, graph-cut) in Figure 9 (b)(ii-iv) or edge-based foreground processing in Figure 9 (b)(vii) alone are not sufficient to extract the printed pattern. A combined region-based (graph-cut) and edge-based approach ensure the extraction process works best on a transparent substrate. Graph-cut background elimination performs better than others in terms of detecting printed objects. With the help of the foreground and background color distribution of the image, two scribbles are set. Then each pixel is assigned a probability of belonging to foreground or background. A graph is constructed with the foreground scribble as the source and background scribble as the sink vertex. Each image pixel is placed as a node with edges as its distance from the source and the sink. Then, a graph-partitioning algorithm (fast maxflow min-cut) [41] separates the foreground from the background and returns a logical result as shown in Figure 9 (b)(iv). There are a few scattered background areas around the periphery of the extracted foreground, but they are detached after the edge-based foreground processing steps. The final extracted output in Figure 9 (b)(v) is a binary image, which corresponds well to the original pattern in Figure 9 (b)(i).

For an opaque background, the background elimination process is not needed, and the foreground pre-processing step alone can yield a well-defined extracted pattern, as shown in Figure 9 (b)(vii). The foreground pre-processing routine denoises the image, then thresholds it using adaptive Gaussian thresholding. Laplacian edge detection is carried out on the thresholded image, and potentially disconnected edges are connected through morphological transformation: closing followed by connected component analysis. Finally, the contour is determined from the connected edges, and this array is used as a mask on the original noisy image. The masked image pattern is now separated from the background noise and used for defect quantization.

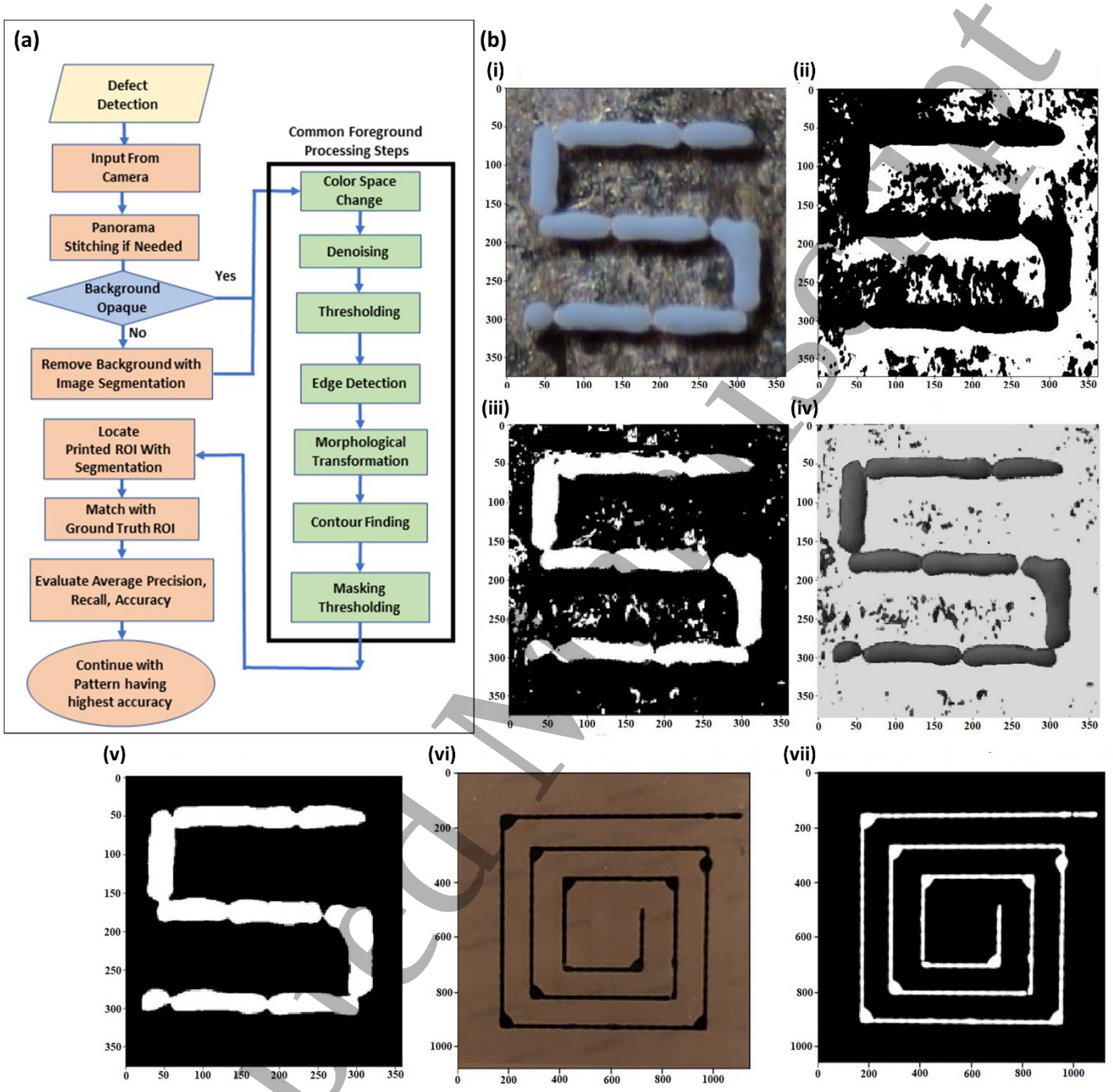

Figure 9: (a) Processing steps for defect detection in images of the printed patterns. (b) Comparison of segmentation approaches for background elimination and foreground processing. (i) Printed pattern on a transparent substrate on the metal stage of the printer resulting in a noisy background, (ii) Background reduction through k-means clustering-based image segmentation, (iii) Background reduction through Gaussian mixture model, (iv) Background reduction employing graph-cut model, (v) Output with processed foreground on top of graph-cut model separated background, (vi) Printed pattern captured on the same substrate put on a tabletop so that background appears opaque and not noisy, (vii) Binary output obtained only from foreground processing without requiring the background reduction step.

2.4.2 Defect Detection

The pre-processed, noise-reduced microscope images are fed through the final defect quantification pipeline. If the pattern dimensions are larger than the field of view of the microscope camera, it cannot be captured by a single image. Multiple images are taken by moving the pattern, and they are merged as a pre-processing step. Panorama image stitching is adopted to stitch multiple microscope images into one so that the test ROI can be easily compared with the ground truth pattern ROI (see Figure S2). A stitching method is constructed by using Oriented FAST and Rotated BRIEF (ORB) feature (keypoint) detector. Like other detectors (SIFT or SURF), it efficiently identifies unique features of the left image and matches them across to the right image with the brute force method. Once matched, a 2D projective transformation is carried out to put the two images on the same image. The denoised image undergoes background elimination and foreground processing (as shown in Figure 9). Histogram-based segmentation is applied further to crop the patterned ROI from both the ground truth and the test image and draw a bounding box around it. Cropped ground truth and test ROI are converted to the same scale for defect comparison.

3 Results and Discussion

3.1 Non-filled Printed Pattern Analysis for Different Printing Sequence Generation Algorithms

Representative experimental printing results for non-filled patterns are shown in Figure 10. Four small-scale patterns (L, S, Interdigitated, H) and one large-scale rectangular spiral pattern were investigated to study the impact of different vectorization methods on pattern quality. In all cases, printing was carried out using several values for drop spacing, and images of printed patterns are shown here only for optimum drop spacing. Quantitative evaluation of different DSp values is discussed in section 3.3. For all of the raster printing outcomes in Figure 10 (b), some bulging is observed at the beginning of line segments. The large-scale pattern in Figure 10 (b)(iv) also suffers from bulging at regular intervals along line segments. This repetitive bulging is not always observed for small-scale patterns in Figure 10 (b)(i-iii), but patterns also exhibit abrupt disconnects at the corners and intersections. This arises from abrupt position changes in the y-direction resulting in a pressure imbalance between the already deposited large fluid bead and the impinging drop with small radius at the next y-location. Consequently, these deviations from the intended pattern make raster printing a non-ideal choice for shapes comprising of single lines with frequent direction changes such as corners, T-junctions or crossing lines. Conversely, for vector printing, as shown in Figure 10 (c), improved patterning results can be observed. Small-scale patterns exhibit improved edges and corners with a noteworthy decrease in breakage at the junctions. Contrary to raster printing, adjacent drops are printed in immediate

succession. This means there is less time for the existing fluid bead to spread, thus reducing the problem of pressure imbalances. Additionally, there is less opportunity for drop misplacement. However, some bulging can still be noticed, especially for the large-scale spiral shape. The non-uniformity of large-scale patterns can be further improved by symmetric sequencing. Figure 10 (d)(iv) shows that the large-scale line pattern exhibits almost no bulging along lines with limited bulging at the corners. For the small-scale designs in Figure 10 (d)(i-iii), symmetric vectorization does not offer significant improvements over linear vectorization. Symmetric printing also requires reduced drop spacing due to the connecting drop spacing of 0.75 times the drop spacing in contrast to raster and linear vectorization resulting in wider lines.

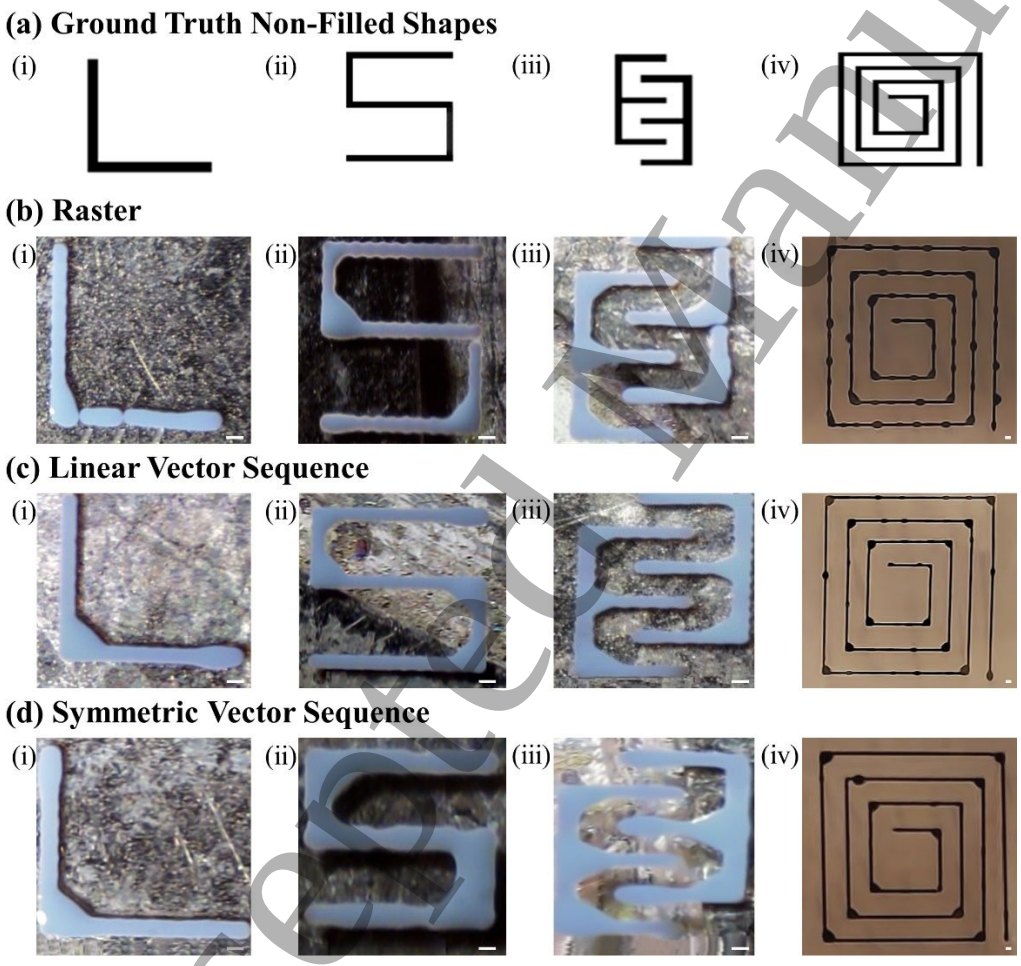

Figure 10: Optical micrographs of experimental printing results for three different sequencing techniques with the optimum DSp and CDSp for four different types of non-filled patterns. (a) Ground truth patterns, (b) Raster printing results with DSp=130 µm, (c) Linear vectorizing results with DSp=125 µm, (d) Symmetric vectorizing results with DSp=120 µm and CDSp=90 µm. Scale bars represent 200 µm.

3.2 Filled Printed Pattern Analysis for Different Printing Sequence Generation Algorithms

Similarly, pattern generation algorithm performance was evaluated for filled structures as shown in Figure 11. Four types of filled shapes (L, C, G, and Rectangle) were printed, and the results with optimized drop spacing are discussed here. Ground truth patterns are shown in Figure 11 (a). When filled patterns are raster printed, they exhibit a number of holes, and also, the edges are not uniform, as shown in Figure 11 (b). These problems with raster patterns can be overcome with contour-based vector sequencing as shown in Figure 11 (c&d). Both radial and parallel filling exhibit similar results. Both methods enhance the quality of filled shapes compared to raster printing. The biggest advantage of contour printing is even and uniform edges. The initial printing of the boundary confines the rest of the drops subsequently printed to fill the pattern. These mechanics control the fluid pressure within the borderline region and considerably reduce bulging and abrupt holes. Images of patterns printed with symmetric ordering are shown in Figure 11 (e). Symmetric printing improves the corners, although some inconsistencies around the edge profile and a few holes can be observed. Depending on the desired printed features, contour-based vectorizing with parallel or radial filling enhances not only the edge profile but also reduces pattern variance. However, these patterns suffer from reduced junction sharpness. For devices that can compromise in terms of edge smoothness but require strict corner localization, symmetric vectorization or block matrix printing are a better choice. The corner compensation algorithm was explored for filled structures, but because the results were considerably inferior to the other methods exhibiting holes, they are presented as supplementary information. Matrix sequencing (with and without contour printed first) produces promising results as shown in Figure 11 (f-h). When the contour is printed first and the inside is filled with a small block (2x2) matrix vectorization, the results are similar to radial or parallel filled contour printing. For contour matrix printing, the edges are smoother and less scalloping is visible. Interleaved printing of drops between blocks does not result in improved printing. Rather block-level printing outperforms interleaved printing for all the filled shapes, even without considering the effect of how the pattern is segmented into blocks. As the block size increases from 2x2 to 4x4, the contour matrix patterning result improves with less bulging at the corners. As the pattern scale size increases, matrix segmentation combined with contour-based block vectorization yields improved printing results. Further results comparing different block sequencing methods and block size can be found in the supplementary information.

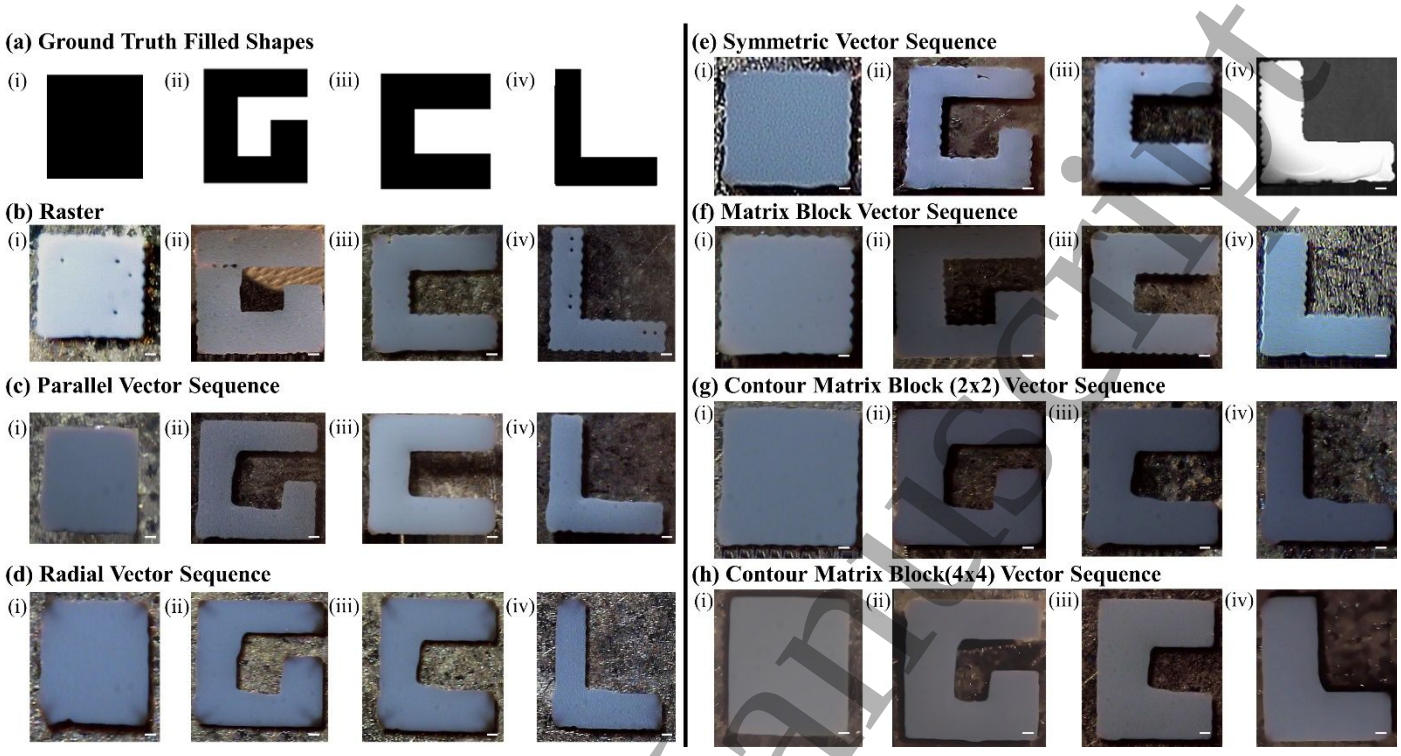

Figure 11: Optical micrographs of experimental printing results of six different sequencing techniques with optimized DSp and CDSp for four different types of filled shapes. (a) Ground truth patterns, (b) Raster printing results with DSp=120 µm, (c) Parallel contour vectorizing results with DSp=115 µm, (d) Radial contour vectorizing results with DSp=115 µm, (e) Symmetric vectorizing results with DSp=105 µm and CDSp=79 µm, (f) Block matrix vectorizing results without contour printing with DSp=110 µm, (g) Contour block (2x2) matrix vectorizing results with DSp=110 µm, (h) Contour block (4x4) matrix vectorizing results with DSp=110 µm. Scale bars represent 200 µm.

3.3 Quantitative Evaluation of Printed Patterns

Patterns generated with the different models were evaluated through computer vision-based defect detection, which is superior to traditional qualitative human inspection, which lacks standardization and efficiency. Each sequencing method's performance is evaluated based on average precision, recall and accuracy calculations (equations (3), (4), (5)) of two binary image sets (ground truth and test) based on the number of true positive (TP), false positive (FP), true negative (TN) and false negative (FN) pixels. Positive refers to white pixels (background), negative refers to dark pixels (pattern) in the binarized images. For example, a pixel is counted as a false negative if there is printed ink where there should not be any according to the ground truth design.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

When the non-print white (1) pixels in the ground truth image are not white (0) in the corresponding test image, it implies the printed patterns have bulged over the intended pattern boundaries. False negative is high and recall is low. Precision becomes low when false positive is high due to some black pixels (0) in the ground truth image being transformed to white pixels (1) in the test image. This means the pattern exhibits shrinking, i.e., hole defects. Accuracy increases with large true positive and true negative values, which means the number of black pixels in the printed image is close to the number of black pixels in the ground truth image and the white pixel numbers similarly match. Correspondingly, FP and FN are low, signifying reduced bulging, holes, and shrinking and good printed pattern quality.

Table 1 shows detailed results for the comparison of the different pattern generation methods applied to the test set for performance evaluation. Radial and parallel filling exhibit similar results; therefore, they were averaged together and their row labelled as contour vectorizing. Figure 12 gives an overview of key performance measures. These results are shown for the DSp that gives the best result for a particular method and pattern type. Filled shapes tend to exhibit lower recall (i.e., exhibit bulging defects) compared to non-filled shapes that are prone to lower precision (i.e., exhibit shrinking or disconnect defects). In the case of filled shapes, contour-based matrix printing as well as radial and parallel filling have the highest accuracy. Small-scale non-filled patterns are most challenging to print and benefit the most from contour vector sequencing. For large-scale non-filled shapes, the symmetric sequencing model outperforms other methods showing better average accuracy. These quantitative findings agree well with the qualitative findings described above. These quantitative metrics can be used in the future to identify the optimal sequencing method for different pattern types and as a general method to quantify print quality.

Table 1: Quantitative comparison of different patter generation methods. Optimal methods for different pattern classes (filled, small-scale non-filled, large-scale non-filled) are highlighted.

Pattern Type	***Shape Type***	***Precision***	***Recall***	***Accuracy (%)***	***Standard Deviation of Accuracy***
Raster	Filled (DSp: 120 µm)	0.935	0.773	88.7%	3.49%
	Non-Filled (DSp: 130 µm)	0.891	0.837	84.0%	1.77%
Contour Vector (Radial, Parallel Average)	Filled (DSp: 115 µm)	0.903	0.999	93.4%	2.91%
	Non-Filled (DSp: 125µm)				
	Small Scale	**0.931**	**0.871**	**88.5%**	**1.66%**
	Large Scale	0.919	0.825	85.6%	0.029%
	Average	0.925	0.848	87.1%	0.846%
Symmetric Vector	Filled (DSp: 105µm, CDSp: 0.75*DSp)	0.889	0.976	89.1%	3.35%
	Non-Filled (DSp: 120µm, CDSp: 0.75*DSp)				
	Small Scale	0.837	0.839	83.3%	1.165%
	Large Scale	**0.965**	**0.897**	**94.2%**	**0.011%**
	Average	0.901	0.868	88.8%	0.588%
Corner Compensation (1 Pixel)	Filled (DSp: 120 µm)	0.88	0.596	79.8%	2.07%
	Non-Filled (DSp: 130 µm)	0.42	0.351	51.6%	1.75%
Matrix (4x4 Block) (Clockwise)	Filled (DSP: 110 µm)	0.895	0.939	92.2%	2.87%
Contour Matrix (4x4 Block) (Clockwise)	Filled (DSP: 110 µm)	**0.916**	**0.983**	**94.9%**	**2.14%**

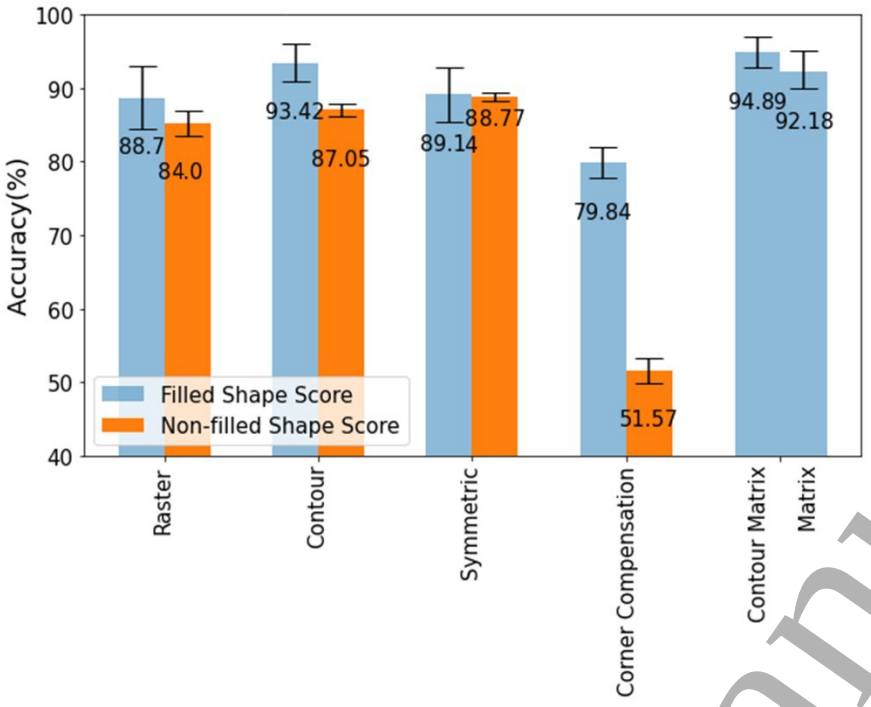

Figure 12: Accuracy of different pattern generation methods from image analysis.

3.4 Thickness Characterization

Pattern thickness and uniformity are important metrics for the performance of many printed circuits, e.g., determining resistance. Here, profilometry was used to characterize the thickness of the printed silver patterns and compare different drop sequencing methods. For other materials that are semi-transparent, this could also be directly integrated into the image-based pattern evaluation framework using color reflectometry [42]. Figure 13 (a) summarizes the profilometry results for four types of filled shapes (L, C, G, rectangle) and three small-scale non-filled patterns (L, S, interdigitated) with different drop sequencing methods. The average thickness varies between the patterns printed with different algorithms. For both filled and non-filled shapes, the thickness increases from raster to contour to symmetric printing. Filled shapes also exhibit a larger thickness than non-filled shapes. This occurs because the optimum DSp decreases from raster to contour to symmetric sequencing and from non-filled to filled shapes (see Table 1 for optimal DSp values). As a result, the deposited ink volume per unit area increases and printed thickness increases.

Thickness uniformity was quantified in terms of the surface roughness parameter Ra (arithmetic mean deviation) as shown in Figure 13 (b). The thickness non-uniformity exhibits similar trends as found with the image-based precision-recall pattern evaluation. Raster printed patterns exhibit the largest non-uniformities.

For filled shapes, contour matrix (4x4) printed block by block exhibits the smallest variations and the best pattern homogeneity. Again, radial and parallel contour printing exhibit very similar thickness variation. For non-filled shapes, raster printing also shows larger non-uniformity than contour and symmetric printing. Overall, filled shapes exhibit larger thickness variation than non-filled shapes due to their larger dimensions.

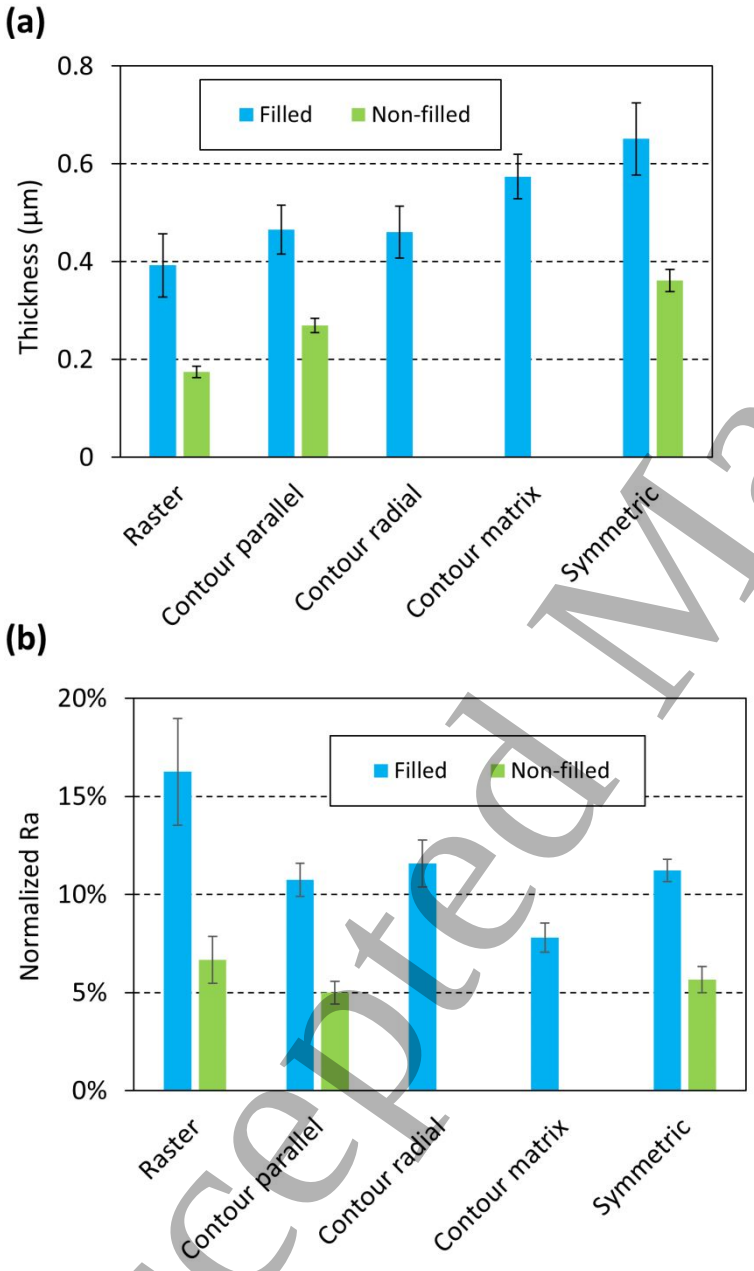

Figure 13: (a) Average printed thickness for filled and non-filled patterns sequenced with different methods using the ideal drop spacing for each method. For non-filled patterns, all contour methods are identical. (b) Roughness parameter Ra normalized by thickness for the same samples as in (a).

3.5 Feedback System

The defect detection system described in the previous section is not only useful to study the merit of different printing sequence generation methods, but also for providing useful feedback on the pattern design. Generating optimum printed patterns depends on the DSp and for symmetric printing CDSp, in addition to the drop sequencing method. This parameter needs to be re-optimized whenever a new ink-substrate combination is used or when printing onto inhomogeneous substrates with already existing printed layers. Incorrect DSp selection can lead to bulging, shrinking, scalloping, and beads [12]. Using the evaluation metrics described in the previous section as feedback can efficiently identify the optimum DSp and CDSp for a specific substrate and ink. This selection has always been difficult, especially for complex and diverse shapes. Previously, most published studies have identified the best DSp and CDSp only for single lines or simple patterns. Figure 14 (a) shows some of our DSp optimization results. Filled and non-filled shapes are averaged over different patterns for different values of DSp. The optimum DSp giving the highest accuracy is different for filled and non-filled shapes and the different sequencing methods. For filled shapes, raster, contour vectorization, symmetric and contour matrix printing should use a drop spacing of 120 μm, 115 μm, 105 μm and 110 μm, respectively, for this ink-substrate system. For non-filled shapes, raster, contour vectorization, and symmetric printing should use a drop spacing of 130 μm, 125 μm, and 120 μm respectively. Non-filled shapes require larger DSp than filled shapes.

Even with the optimum DSp and CDSp values and the optimum sequencing method, some of the printed features can still exhibit small abrupt holes, especially large-scale filled patterns. The accuracy-based feedback system can be used for backfilling abrupt holes at their centroid location. After finding the centroid of the void contour, the coordinates are merged, and the next time the printer prints the same pattern, it will compensate for the hole. Figure 14 (b) is an example of center coordinate detection for abrupt micron-scale holes. Figure 14 (b)(ii) shows that the center locations of the holes are determined in terms of x, y coordinates. These coordinates are further merged into the ready to print file as added drops. The improved printed shape is shown in Figure 14 (b)(iii). This method can serve as the last step in generating an optimized drop sequence after determining the optimal sequencing method and drop spacing using the above-described quantitative print quality metrics.

Another method for defect detection in complex printed micrometer-scale patterns is skeleton-based graph inspection similar to optical inspection methods for PCBs [28–30]. Figure 14 (c-f) shows defect patterns such as short and open circuits, bulging, and holes. Such defects can lead to catastrophic loss of functionality in

electronics systems. To analyze all connected components in the printed structure, Zhang's skeletonization [43] is adopted. The ROI is reduced to a 1-pixel wide representation by recursive identification and removal of border pixels as long as the structure's connectivity is conserved. A network graph is constructed on the skeletons using the sknw package [44] with the root and branching nodes and edge length as the distance between connecting nodes of the printed patterns. Graph-based inspection is performed on the skeletons by means of comparing the number of connected components as well as the distance between edge nodes in the ground truth and printed pattern. The shortest path distance within the network is utilized for short circuit tracing. A list of connected and disconnected nodes and their distance in micrometers is used to detect and visualize fault patterns. Comparing ground truth and printed pattern skeletons becomes much easier through fault tracing from one node to another. In Figure 14 (c)(i), the ground truth pattern is designed as two distinct lines without any connection between nodes 0 and 1. In the printed pattern (Figure 14 (c)(ii)), there exists a path between nodes 0 and 1 through nodes 2 and 3. The path distance is 418 μm. The printed pattern contains a single component signifying a short circuit fault. In Figure 14 (d)(i), the designed pattern is a two-component structure. The printed shape (see Figure 14 (d)(ii&iii)) is represented by a three-component skeleton with one separate track due to unwanted material deposition. A defect-free case is depicted in Figure 14 (e). This filled C-shape structure corresponds to Figure 11 (c)(iii) printed with DSp 115 μm and the contour algorithm. The ground truth and the printed pattern have the same number of connected components. The distance between the farthest nodes is nearly identical, with a 1.51% deviation. An interdigitated raster printing result with 130 μm DSp (corresponding to Figure 10 (b)(iii)) is tested for fault identification in Figure 14 (f). It demonstrates the presence of an open circuit fault in the constructed graph. The ground truth pattern in Figure 14 (f)(i) has two different connected skeletons with nodes 0,2,3,6 and 1,5,4,7. The printed skeleton in Figure 14 (f)(ii) is divided into five component skeletons clearly exhibiting open circuits. These results agree with the qualitative and quantitative analysis of pattern quality for different sequencing methods. Additionally, with the skeleton method, it is straightforward to find the location of a short or open circuit fault path in the printed pattern and optimize the design pattern. The skeleton method can be useful to detect catastrophic functional failures in printed circuits; however, it is less useful to detect more subtle differences in pattern quality such as corner rounding, which may also affect functionality depending on the device in question. Precision, recall and accuracy are more useful to quantitatively analyze high-quality drop sequencing methods that exhibit only very few catastrophic failures.

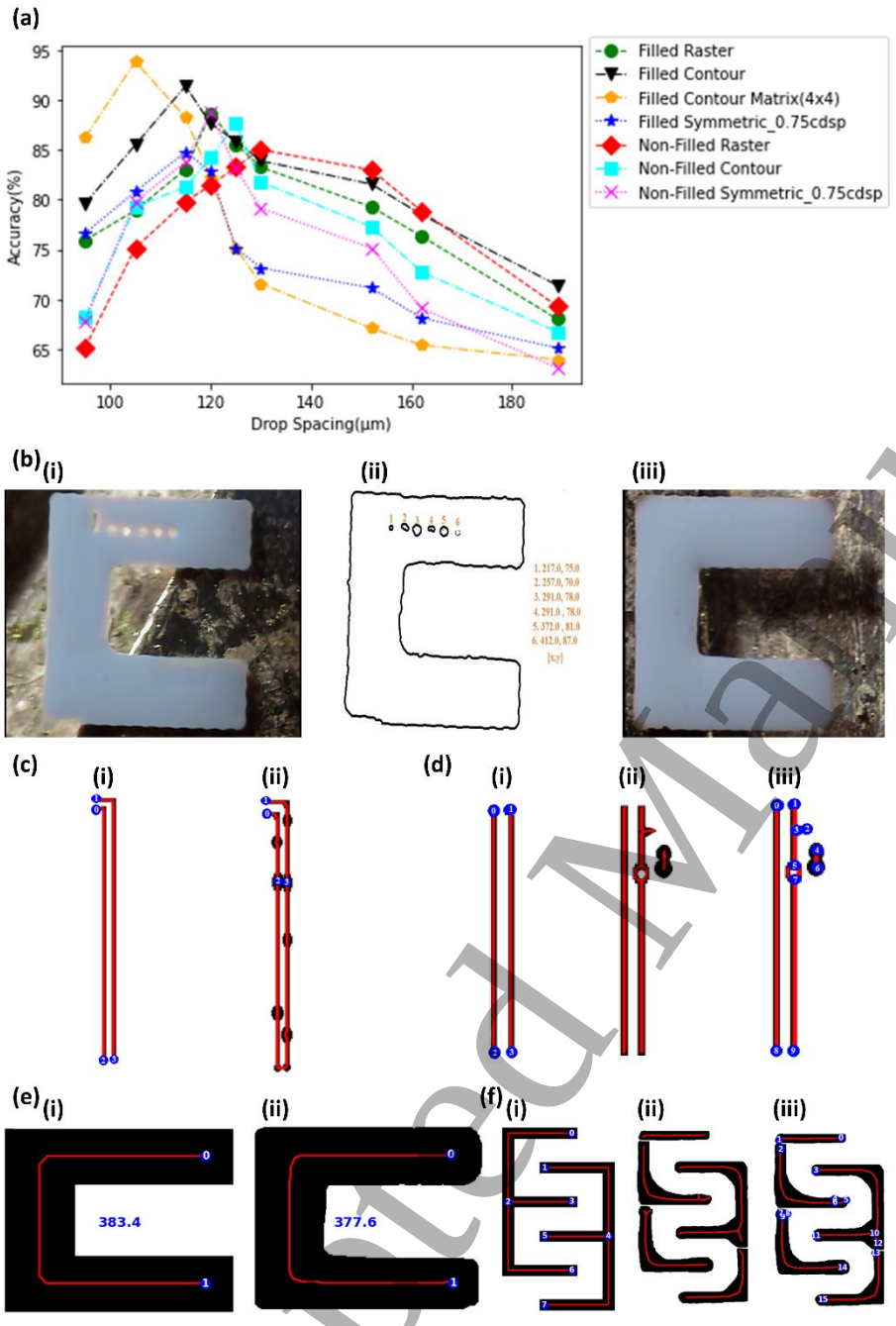

Figure 14: Pattern evaluation and error pruning using a graphical approach. (a) Optimum drop spacing selection using accuracy as the evaluation metric. (b) (i) Printed pattern with hole defects, (ii) Hole coordinates identification, (iii) Reprinted pattern with added drops at hole coordinates. (c) Short circuit path determination for two lines in close proximity. (i) Ground truth pattern, (ii) Printed lines are short circuited between node 2 and 3 due to significant bulging. (d) Void space and additional printed track tracing. (i) Ground truth design pattern, (ii) Skeleton for the printed lines, (iii) Graph nodes are created based on the skeleton connectivity. (e) (i) Ground truth pattern skeleton, (ii) Network graph for a filled shape with the same length of the current path. (f) (i) Ground truth interdigitated pattern skeleton components, (ii) Printed pattern skeleton components, (iii) Routing through pattern framework and open circuit path tracing.

4 Conclusion

A novel computer vision-based scheme is developed and evaluated for the control of inkjet printing at the micrometer scale. The pixels of a pattern image are taken as the input and converted to a drop sequence by means of feature mapping and shape detection algorithms. Finally, printed results are analyzed using automated defect detection. Four ways to generate the sequence of drops are compared. Contour based vectorization (parallel and radial filling) as well as matrix vectorization give the best results for filled shapes. Contour vectorization gives best results for non-filled shapes, which is improved for large-scale patterns by symmetric vectorization. Defect quantification and identification are vital for quality monitoring and final output judgment. The print output is evaluated with respect to the input image and they are compared to detect defects. The developed computer vision-based electronics printing methodology improves print quality for complex shapes in an automated fashion, which is not possible or requires extensive manual intervention with traditional methods. In the future, this algorithm can be extended to automatically combine different sequence generation methods for different parts of a layer that have different requirements based on automated feature extraction before printing as well as local defect detection and quantification after printing.

5 Acknowledgements

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), funding reference number STPGP 521480-18.

6 References

[1] Bonnassieux Y, Brabec C J, Cao Y, Carmichael T B, Chabinyc M L, Cheng K-T, Cho G, Chung A, Cobb C L, Distler A, Egelhaaf H-J, Grau G, Guo X, Haghiashtiani G, Huang T-C, Hussain M M, Iniguez B, Lee T-M, Li L, Ma Y, Ma D, McAlpine M C, Ng T N, Österbacka R, Patel S N, Peng J, Peng H, Rivnay J, Shao L, Steingart D, Street R A, Subramanian V, Torsi L and Wu Y 2022 The 2021 flexible and printed electronics roadmap *Flex. Print. Electron.* **6** 023001

[2] McKerricher G, Vaseem M and Shamim A 2017 Fully inkjet-printed microwave passive electronics *Microsyst Nanoeng* **3** 16075

[3] Correia V, Mitra K Y, Castro H, Rocha J G, Sowade E, Baumann R R and Lanceros-Mendez S 2018 Design and fabrication of multilayer inkjet-printed passive components for printed electronics circuit development *Journal of Manufacturing Processes* **31** 364–71

[4] Yasin T and Baktur R 2010 Inkjet printed patch antennas on transparent substrates *2010 IEEE Antennas and Propagation Society International Symposium* 2010 IEEE International Symposium Antennas and Propagation and CNC-USNC/URSI Radio Science Meeting (Toronto, ON: IEEE) pp 1–4

[5] Subramanian V, Frechet J M J, Chang P C, Huang D C, Lee J B, Molesa S E, Murphy A R, Redinger D R and Volkman S K 2005 Progress Toward Development of All-Printed RFID Tags: Materials, Processes, and Devices *Proc. IEEE* **93** 1330–8

[6] Rida A, Li Yang, Vyas R and Tentzeris M M 2009 Conductive Inkjet-Printed Antennas on Flexible Low-Cost Paper-Based Substrates for RFID and WSN Applications *IEEE Antennas Propag. Mag.* **51** 13–23

[7] Koo H S, Chen M, Pan P C, Chou L T, Wu F M, Chang S J and Kawai T 2006 Fabrication and chromatic characteristics of the greenish LCD colour-filter layer with nano-particle ink using inkjet printing technique *Displays* **27** 124–9

[8] Chang S, Liu J, Bharathan J, Yang Y, Onohara J and Kido J 1999 Multicolor Organic Light-Emitting Diodes Processed by Hybrid Inkjet Printing** *Advanced Materials* **11** 734–7

[9] Moonen P F, Yakimets I and Huskens J 2012 Fabrication of Transistors on Flexible Substrates: from Mass-Printing to High-Resolution Alternative Lithography Strategies *Advanced Materials* **24** 5526–41

[10] Chung S, Cho K and Lee T 2019 Recent Progress in Inkjet-Printed Thin-Film Transistors *Advanced Science* **6** 1801445

[11] Grau G, Kitsomboonloha R, Swisher S L, Kang H and Subramanian V 2014 Printed Transistors on Paper: Towards Smart Consumer Product Packaging *Advanced Functional Materials* **24** 5067–74

[12] Soltman D and Subramanian V 2008 Inkjet-Printed Line Morphologies and Temperature Control of the Coffee Ring Effect *Langmuir* **24** 2224–31

[13] Sowade E, Polomoshnov M and Baumann R R 2016 The design challenge in printing devices and circuits: Influence of the orientation of print patterns in inkjet-printed electronics *Organic Electronics* **37** 428–38

[14] Stucchi M, Bamal M and Maex K 2007 Impact of line-edge roughness on resistance and capacitance of scaled interconnects *Microelectronic Engineering* **84** 2733–7

[15] Duineveld P C 2003 The stability of ink-jet printed lines of liquid with zero receding contact angle on a homogeneous substrate *J. Fluid Mech.* **477**

[16] Stringer J and Derby B 2009 Limits to feature size and resolution in ink jet printing *Journal of the European Ceramic Society* **29** 913–8

[17] Stringer J and Derby B 2010 Formation and Stability of Lines Produced by Inkjet Printing *Langmuir* **26** 10365–72

[18] Diaz E, Ramon E and Carrabina J 2013 Inkjet Patterning of Multiline Intersections for Wirings in Printed Electronics *Langmuir* **29** 12608–14

[19] Hsiao W-K, Martin G D and Hutchings I M 2014 Printing Stable Liquid Tracks on a Surface with Finite Receding Contact Angle *Langmuir* **30** 12447–55

[20] Abunahla R, Rahman M S, Naderi P and Grau G 2020 Improved Inkjet-Printed Pattern Fidelity: Suppressing Bulges by Segmented and Symmetric Drop Placement *Journal of Micro and Nano-Manufacturing* **8** 031001

[21] Chen C-T and Hung T-Y 2016 Morphology and deposit of picoliter droplet tracks generated by inkjet printing *Journal of Micromechanics and Microengineering* **26** 115005

[22] Oh Y, Kim J, Yoon Y J, Kim H, Yoon H G, Lee S-N and Kim J 2011 Inkjet printing of Al2O3 dots, lines, and films: From uniform dots to uniform films *Current Applied Physics* **11** S359–63

[23] Kang H, Soltman D and Subramanian V 2010 Hydrostatic optimization of inkjet-printed films *Langmuir* **26** 11568–73

[24] Soltman D, Smith B, Morris S J S and Subramanian V 2013 Inkjet printing of precisely defined features using contact-angle hysteresis *Journal of Colloid and Interface Science* **400** 135–9

[25] Du Z, Xing R, Cao X, Yu X and Han Y 2017 Symmetric and uniform coalescence of ink-jetting printed polyfluorene ink drops by controlling the droplet spacing distance and ink surface tension/viscosity ratio *Polymer* **115** 45–51

[26] Soltman D, Smith B, Kang H, Morris S J S and Subramanian V 2010 Methodology for Inkjet Printing of Partially Wetting Films *Langmuir* **26** 15686–93

[27] Vila F, Pallares J, Ramon E and Teres L 2016 A Systematic Study of Pattern Compensation Methods for All-Inkjet Printing Processes *IEEE Transactions on Components, Packaging and Manufacturing Technology* **6** 630–6

[28] Ito M and Nikaido Y 1991 Recognition of pattern defects of printed circuit board using topological information *[1991 Proceedings] Eleventh IEEE/CHMT International Electronics Manufacturing Technology Symposium* Eleventh IEEE/CHMT International Electronics Manufacturing Technology Symposium (San Francisco, CA, USA: IEEE) pp 202–6

[29] Ito M, Nikaido Y and Hoshino M 1992 Divide And Merge Image Processing For Pattern Defect Analysis Of Printed Circuit Boards *Thirteenth IEEE/CHMT International Electronics Manufacturing Technology Symposium* Thirteenth IEEE/CHMT International Electronics Manufacturing Technology Symposium (Baltimore, MD: IEEE) pp 178–82

[30] Rau H and Wu C-H 2005 Automatic optical inspection for detecting defects on printed circuit board inner layers *Int J Adv Manuf Technol* **25** 940–6

[31] Soltman D 2011 *Understanding Inkjet Printed Pattern Generation* (University of California, Berkeley)

[32] Papari G and Petkov N 2011 Edge and line oriented contour detection: State of the art *Image and Vision Computing* **29** 79–103

[33] Iverson L A and Zucker S W 1995 Logical/linear operators for image curves *IEEE Trans. Pattern Anal. Machine Intell.* **17** 982–96

[34] Tekin E, de Gans B-J and Schubert U S 2004 Ink-jet printing of polymers - from single dots to thin film libraries *J. Mater. Chem.* **14** 2627

[35] Wu S-Y and Sahni S 1994 Fast Algorithms to Partition Simple RectilinearPolygons *VLSI Design* **1** 193–215

[36] Keil J M and Sack J-R 1985 Minimum Decompositions of Polygonal Objects *Machine Intelligence and Pattern Recognition* vol 2 (Elsevier) pp 197–216

[37] Neggazi B, Haddad M and Kheddouci H 2012 Self-stabilizing Algorithm for Maximal Graph Partitioning into Triangles *Stabilization, Safety, and Security of Distributed Systems* Lecture Notes in Computer Science vol 7596, ed A W Richa and C Scheideler (Berlin, Heidelberg: Springer Berlin Heidelberg) pp 31–42

[38] Getreuer P 2012 Rudin-Osher-Fatemi Total Variation Denoising using Split Bregman *Image Processing On Line* **2** 74–95

[39] Jaglan P, Dass R and Duhan M 2019 A Comparative Analysis of Various Image Segmentation Techniques *Proceedings of 2nd International Conference on Communication, Computing and Networking* Lecture Notes in Networks and Systems vol 46, ed C R Krishna, M Dutta and R Kumar (Singapore: Springer Singapore) pp 359–74

[40] Zaitoun N M and Aqel M J 2015 Survey on Image Segmentation Techniques *Procedia Computer Science* **65** 797–806

[41] Boykov Y and Kolmogorov V 2004 An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision *IEEE Trans. Pattern Anal. Machine Intell.* **26** 1124–37

[42] Bornemann N and Dörsam E 2013 A flatbed scanner for large-area thickness determination of ultra-thin layers in printed electronics *Opt. Express* **21** 21897

[43] Zhang T Y and Suen C Y 1984 A fast parallel algorithm for thinning digital patterns *Commun. ACM* **27** 236–9

[44] Yan X 2020 *Skeleton Network (sknw) https://github.com/Image-Py/sknw*