

Received June 18, 2020, accepted June 22, 2020, date of publication June 26, 2020, date of current version July 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3005184

Consistency Penalized Graph Matching for Image-Based Identification of Dendritic Patterns

ZAOYI CHI¹, ALI VALEHI¹, (Graduate Student Member, IEEE), HAN PENG¹,
MICHAEL KOZICKI², (Member, IEEE), AND ABOLFAZL RAZI¹, (Senior Member, IEEE)

¹School of Informatics, Computing, and Cyber Systems, Northern Arizona University, Flagstaff, AZ 86011, USA

²School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85281, USA

Corresponding author: Abolfazl Razi (abolfazl.razi@nau.edu)

This work is supported by the Arizona Board of Regents under Grant 1003329.


ABSTRACT Recently, physically unclonable functions (PUFs) have received considerable attention from the research community due to their potential use in security mechanisms for applications such as the *Internet of things* (IoT). The concept generally employs the fabrication variability and naturally embedded randomness of device characteristics for secure identification. This approach complements and improves upon the conventional cryptographic security algorithms by covering their vulnerability against counterfeiting, cloning attacks, and physical hijacking. In this work, we propose a new identification/authentication mechanism based on a specific implementation of optical PUFs based on electrochemically formed *dendritic patterns*. *Dendritic* tags are built by growing unique, complex, and unclonable nano-scaled metallic patterns on highly nonreactive substrates using electrolyte solutions. Dendritic patterns with 3D surfaces are technically impossible to reproduce, hence they can be used as the fingerprints of objects. Current optical PUF-based identification mechanisms rely on image processing methods that require high-complexity computations and massive storage and communication capacity to store and exchange high-resolution image databases in large-scale networks. To address these issues, we propose a light-weight identification algorithm that converts the images of dendritic patterns into representative graphs and uses a graph-matching approach for device identification. More specifically, we develop a probabilistic graph matching algorithm that makes linkages between the similar feature points in the test and reference graphs while considering the consistency of their local subgraphs. The proposed method demonstrates a high level of accuracy in the presence of imaging artifacts, noise, and skew compared to existing image-based algorithms. The computational complexity of the algorithm grows linearly with the number of extracted feature points and is therefore suitable for large-scale networks.

INDEX TERMS Optical PUF, graph matching, image identification, IoT security, identification tags.

I. INTRODUCTION

Visual information accounts for more than 90 percent of human perception of the surrounding world [1]. Therefore, most identification and authentication processes traditionally rely on image processing. Examples of image-based identification methods are countless. For instance, processing facial images [2], fingerprints [3], iris [4], outer ear shape [5], and gait analysis from video frames [6] are among the most popular methods developed for human identification. Identification tags are commonly used in supply chains, distribution systems, logistics, transportation, and related industries. For instance, barcodes, and QR codes are widely used to

translate text-based identification information (e.g., product ID, fabrication date, manufacturer, and the country of origin) into 1-D and 2-D patterns with universal mapping [7]. According to a recent survey conducted by Statista [8], in the US alone, an estimated 11 million households will scan a QR code in 2020. Barcodes and QR codes have been a successful ubiquitous identification method due to their advantages such as low-cost fabrication, affordable readers, high accuracy and light processing requirement, compared to more costly and intricate electronics-based identification methods, such as RFID. However, due to their plain 2-D structure, they are easily copiable, and hence are vulnerable to cloning attacks, and so anti-copy methods are currently under investigation [9], [10]). Their vulnerability to counterfeiting questions their utility, especially for more sensitive

The associate editor coordinating the review of this manuscript and approving it for publication was Yanjiao Chen .

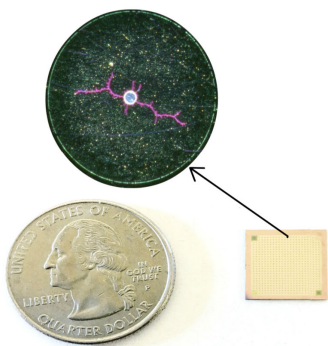


FIGURE 1. A sample dendrite object (magnified, top). The dendrite wafer panel (bottom left) includes $24 \times 24 = 576$ dendrite objects. The US Quarter Dollar is included for scale.

applications such as federal and military supply chains which are facing growing threats from the insertion of low-quality fake parts [11], [12]. To protect these sensitive supply chains, it is desired to use unclonable visual-tags as a trust mark for the item it is attached to. It is noteworthy that RFID and other microelectronics tags are highly susceptible to hostile cloning due to recent advances in cryptographic hacking methods involving side-channel attacks [13], focused ion beam (FIB) [14] edits, micro-probing, and chemical mechanical polishing (CMP) tear down after removing the passivation layer on the chip [15].

One major drawback of image-based identification methods (e.g., face recognition) is the need for huge storage and communication capacity for identification systems to maintain and exchange a valid copy of reference images. This can be troublesome when the identification system is scaled to include a large number of devices such as those found in the Internet of Things (IoT). Furthermore, image-based identification methods can be computationally expensive for IoT nodes with limited processing powers.

Dr. Kozicki's (co-author) team at Arizona State University (ASU) have invented a new proof-of-concept tagging technology, called *dendritic identifiers*, which involves electrodeposition of metallic structures on substrates containing electrolytes to provide unique information-rich patterns [16], [17]. A sample dendrite is shown in Figure 1. The inherent randomness of multi-scaled dendritic patterns 1 provides high-entropy object-specific identifiers. Further, due to the nature of the electrodeposited material which exhibits a 3D faceted surface, it is virtually impossible to clone any of the dendrites with existing technology. Therefore, these dendritic objects have great potential to be used in image-based identification methods.

In this paper, we propose a low-complexity and high-reliability image-based identification algorithm that has been developed for such *dendritic* patterns that use a simple cellphone-based imaging system. The core idea of the proposed method is the translation of the intricate dendritic patterns into representative graphs, which converts the computationally expensive image-based identification problem

into a much lighter graph matching algorithm without a significant degradation in the identification accuracy. This approach also eliminates the need for massive storage and communication capacities in large-scale networks. Using a novel feature extraction as well as a probabilistic matching approach, the proposed method is robust to imaging artifacts such as rotation, skew, scaling, noise, and scratches. We provide intensive simulations and compare the performance of the proposed algorithm to similar methods.

II. BACKGROUND INFORMATION AND RELATED WORK

Before proceeding with the details of the proposed solution, we provide a short review of Physical Unclonable Functions (PUFs), image-based identification methods, and graph-matching algorithms.

A. RELATED WORK ON OPTICAL PUFs

PUFs based on electronic devices are widely used as a reliable means of identification and authentication since they exploit the natural randomness and the fabrication variability of each device to produce as a unique and unclonable device identifier. Examples of electronic PUFs include ring oscillators [18], programmable delay lines [19], arbiters [20], and memory arrays [21]. A key advantage of the use of electronic PUFs for identification is that little or no additional cost is incurred at the device level as they use existing device elements. However, they cannot be considered fully secure. For instance, cloning attacks on static random access memories (SRAMs) have become possible by reprogramming the tendency of a cell using focused ion beam circuit edit [22]. In addition to the vulnerability of electronic PUFs to side-channel attacks [23], [24], there exist two limiting factors. Firstly, complicated custom-built circuitry is often required for the identification purpose that can make the reader costly compared to optical PUFs that can be interrogated by a software-based method through a simple camera. Secondly, electronic PUFs are usable only for specific microelectronic devices, hence they cannot be considered as a universal and low-cost solution for other applications. Optical PUFs use the similar concept of extracting unique, random, and high-entropy features from images for identification purposes. For instance, [25] proposes the use of a speckle fluctuation phenomenon caused by a laser passing through inhomogeneous transparent objects for the identification of mm-scaled objects. An array of commercial LCD arrays with 623 nm laser emission is used to implement an optical PUF that improves upon digital holography in terms of accuracy [26]. These methods provide high accuracy but require a precisely designed optical pipeline with accurate object placement in a lab setup, hence they are not suitable for commercial use. In this paper, we use the specific properties of dendritic patterns that include recognizable, but unclonable nano-scaled patterns to implement a highly reliable optical PUF. In contrast to laser-based methods, our method is fully software-based and uses commercial cameras, such as cellphone cameras.

B. IMAGE RECOGNITION TECHNIQUES

Image-based recognition methods involve several approaches and span a wide range of applications. They typically involve the sequential steps of detection, feature extraction, and matching. For instance, in face recognition methods, the steps include (i) detect and localize the face in the picture, (ii) extract numerical information or descriptive features from the face images and (iii) compare it with those extracted from one or multiple reference images [27]. The facial features include visual features, statistical pixel features, transform coefficient features, component-based and holistic representational features, and algebraic features. Recently, learning-based features that utilize dictionary learning from a large set of images have shown superior performance over conventional methods [28]. For instance, facial images may be projected into lower space using dimensionality reduction methods (e.g., Eigen-face [29]) to enhance the accuracy of recognition step.

As mentioned earlier, feature selection methods are an important part of image-based identification methods. Some feature selection methods have been developed for specific applications like face recognition [30], fingerprint identification [31], and cross-spectral biometric imaging [32], while others have been developed for general images. For instance, the popular scale-invariant feature transform (SIFT) algorithm and its variants perform image comparison by matching a refined selection of key-points obtained from both images based on the distance of Gaussian (DoG) method. Its performance is robust against the scale and orientation of the object by the use of local feature coordinates and multiple scales for the key points [33]. The speeded up robust features (SURF) technique can be viewed as an accelerated version of SIFT that uses Hessian matrix approximation (instead of DoG) to locate key points and uses Haar wavelet response (instead of the orientation histogram) to find the key-point orientation [33]. The features from accelerated segment test (FAST) is used to detect the features of input images. Unlike SIFT [34], FAST does not rely on DoG, rather it identifies the feature points when the summation of the absolute value of the difference between 9 or 12 continuous surrounding pixels and the middle position is greater than a threshold. However, FAST is not robust in noisy conditions. Another recently developed method is the maximally stable extremal regions (MSER) [35]. In this method the image is binarized frequently by decreasing thresholds. The regions that show less variations in consecutive binarization stages are identified as maximally stable regions.

Some other commonly used feature selection methods include Harris *et al.* [36], histogram oriented gradient (HOG), binary robust invariant scalable keypoints (BRISK) [37], and the modified version of Harris-Min Eigen [38].

For our work, considering the special topology of dendritic patterns, we locate feature points in terms of bifurcations and leaves of the dendritic *tree* which is naturally robust to noise, orientation, scratches, and skew. Our results show that this method outperforms similar feature selection and

descriptors methods when matching dendritic patterns since it is customized to the special morphology of dendrites.

The steps of feature selection, dimensionality reduction, and recognition can also be integrated into one algorithm. The most popular approach is using deep learning methods like convolutional neural networks. For instance, convolutional neural networks (CNN) have shown unprecedented power in exploiting deep local and global features from images. One milestone was Krizhevsky's very powerful CNN implementation, called ImageNet [39]. Since then many other implementations of CNN including AlexNet [39], VGG16 [40], VGG19, and GoogleNet [41] have been proposed to develop image based classification and object recognition tasks by stacking more layers and realizing more powerful and flexible architectures. A good review of these methods can be found in [42]. However, this is more relevant for classifying objects with similar image descriptors, hence not directly applicable to our case of optimal match finding where the goal is to find the most similar reference dendrite among a subset of reference dendrite images with completely random and unique patterns. Secondly, deep learning methods with thousands of parameters require a large dataset of high resolution images, and computationally expensive training phases, something not affordable in low-power tiny IoT devices, and in low-cost volume labeling technology. In this paper, we intend to design a light-weight algorithm that does not require too much computation power, and high storage for saving high resolution images to achieve an acceptable accuracy and flexibility.

C. REVIEW OF GRAPH MATCHING ALGORITHMS

Our proposed method converts the images of dendritic patterns into directed acyclic graphs (DAG) with tree structures. The extracted graph provides a numerical representation of the dendrite, hence graph-matching approaches can be used to evaluate the similarity between two dendrites for identification and authentication purposes. Graph matching is a powerful technique for similarity assessment for different applications such as object recognition [43], protein classification [44], face recognition [45], and finger print identification [46]. In our method, the similarity between two dendritic patterns is assessed by calculating the distance between matching nodes' parameters that quantify the length and orientation of corresponding branches.

In most applications, the matching problem boils down to finding graphs with similar structures that include exact and inexact matching, join matching of multiple graphs, higher-order matching, etc., [47]. The exact graph matching with zero-distortion node mapping, the so-called *graph isomorphism* is known to be in NP (neither P nor NP-complete). However, there exist special structures like planner trees that can be solved in linear time [48]. The majority of heuristic algorithms proposed for this problem are computationally intensive [49]. Another major challenge of using graph matching methods for image recognition is the sensitivity of extracted features to noise and image artifacts [50]. For instance, the majority of graph matching algorithms such

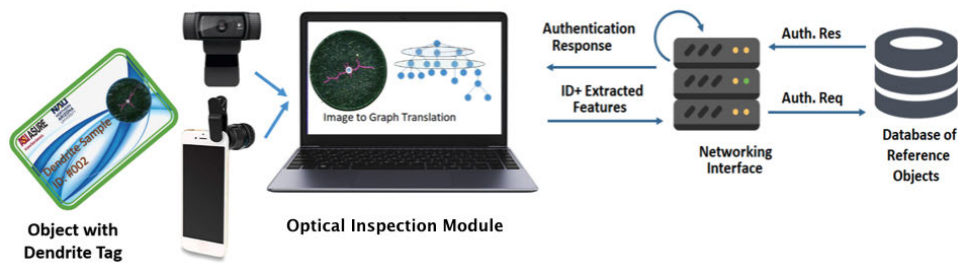


FIGURE 2. Conceptual block-diagram of dendrite-based identification/authentication system.

as greedy algorithms and labeling methods that employ full graph structures for comparison purposes demonstrate exponential complexity with the number of nodes for worst-case scenarios [51]. Graph matching problems can also be cast as nonlinear optimization problems. Using some sort of relaxation in labeling can significantly reduce the computational complexity compared to the exact labeling methods [52]–[54]. Another technique to reduce complexity is *subgraph matching* which is based on reducing the full graph matching to multiple smaller subgraph matching sub-problems [55].

In addition to the aforementioned complexity issue, most of these algorithms are appropriate only for topology matching of unweighted graphs but this does not apply to weighted graphs or our case of attributed graphs, where each node is associated with a weight representing the respective branch morphology.

In this paper, noting the special structure of dendritic patterns, we propose a low-complexity graph matching algorithm that uses an ad-hoc bottom-up method for graph matching. The core idea is to link similar nodes between the test and reference graphs based on their node metrics, and iteratively refine the linkages (by breaking the loose linkages) based on the calculated consistency scores of their local subgraphs. The proposed method has been customized for tree-based graphs and presents a computational complexity that grows linearly with the number of nodes, thereby being faster compared to the other more general methods.

The rest of this paper is organized as follows. In section III, the details of the proposed method of converting dendritic patterns into attributed graphs are given. The proposed graph-matching algorithm are elaborated in section IV. Section V includes experimental results to verify the robustness of the proposed method in the presence of various image distortions. Concluding remarks are provided in section VI.

III. METHODOLOGY

In this section, we elaborate on the details of the proposed approach of using dendritic patterns to implement a reliable mechanism for object identification and authentication in large-scale networks.

The conceptual block diagram of the system is shown in Figure 2. Each object has an embedded dendrite tag. The numerical information of tags is extracted and stored as a

representative tree structure in a network-based database during the registration phase. The dendritic tags can be used for both identification and authentication tasks, depending on the application-specific requirements. In the identification scenario, when a new object enters the network, a visual inspection is initiated by the reader at the entry point of the network. The unlabeled (anonymous) extracted numerical information (in terms of the representative tree) is exchanged with the database through secure communication. The test tree is compared against the reference trees and if matching with a similarity score above a predefined threshold is determined, the object is recognized as being valid and identification is thereby confirmed.

In the authentication scenario, the extracted tree along with its unique ID is sent by the reader to the network, and access is granted only if the similarity between the test tree and the associated reference tree is above the threshold, otherwise verification is denied. Both methods rely on the similarity of extracted numerical information between the test object and the corresponding record in the database. The identification scenario is more appropriate for small-scale networks while the authentication mechanism is suitable for both small-scale and large-scale networks since the object is compared only against the respective reference database record. Finally, note that the extracted numerical information can be combined with random numbers and temporary information to generate encryption keys for secure wireless communication (e.g., ciphering key (CK) in the non-access stratum (NAS) security algorithm in the long-term evolution (LTE) system [56], [57]). Regardless of the use case, the core of the image-based authentication includes two steps of (i) numerical representation of dendritic patterns in terms of graphs, and (ii) assessing the similarity between the test graph against a reference graph in the database using graph-matching algorithms. The following sections elucidate the operation of the proposed framework for the image-based authentication of dendritic tags.

A. IMAGE PRE-PROCESSING

The images taken from the dendrite object can be noisy due to imaging artifacts, the camera's accuracy, uneven illumination, and other factors. Also, the image quality may be compromised due to dust, dirt, scratches, etc. which can accumulate during use in operating environments. The goal

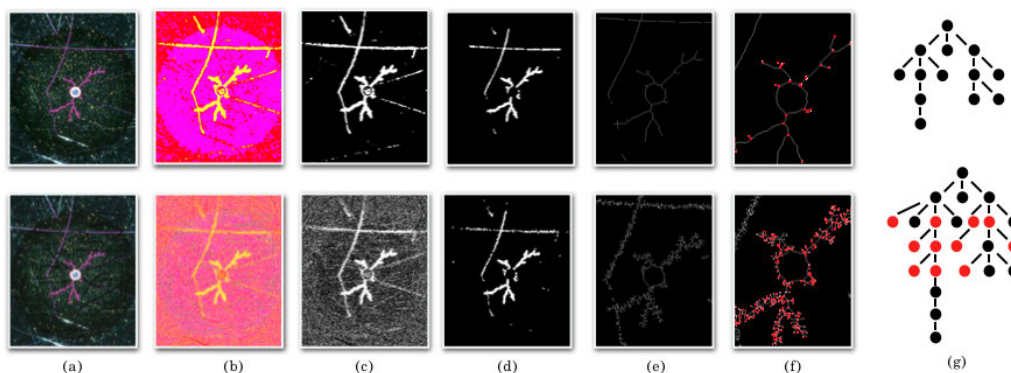


FIGURE 3. The image processing steps applied to the reference dendrite (top row) and the noisy version of the test dendrite (bottom row). The columns represent: (a) the original image, (b) the image in YCbCr space, (c) the image after segmentation, (d) the binarized image, (e) the extracted and thinned skeleton, (f) the extracted graph vertices (feature points), and (g) the representative graph.

of this stage is to clean up the images of artifacts and extract the skeleton of the dendritic pattern from the captured image. This skeleton is then converted to a representative tree and then used in the subsequent graph-matching algorithm. To illustrate the image pre-processing sequence, we selected a dendritic object from an array of objects created by the growth of silver dendrites on a microfabricated substrate. The captured dendritic images undergo the following sequential steps, as shown in Figure 3 for the test and the reference images:

1) YCbCr CONVERSION

The captured image is first converted to YCbCr color space. In this space, it is easy to represent colors in terms of one luminance component and two chrominance components. RGB space is not preferred for image segmentation because the space is not uniform and all components should be quantized with the same precision. On the other hand, YCbCr can mimic the properties of the human eye, which is more sensitive to the light intensity changes and less to the hue change. Therefore, YCbCr is preferred for image segmentation [58]. Considering the dendritic object shown in Figure 3, it is noticeable that the reference image (top image in Figure 3(b)) is much brighter than the bottom image which has high Gaussian noise. The high contrast of the image facilitates the dendritic skeleton extraction process.

2) DENOISING AND QUALITY ENHANCEMENT

In order to eliminate the noise and imaging, an adaptive median filter [59] is applied to smooth out the image and suppress the noise around the pattern. An exemplary image after denoising is shown in Figure 3(c).

3) BINARIZATION

By thresholding the pixel values with respect to the median of the pixel intensity histogram, the extracted pattern is converted to a monochrome image.

4) CONTIGUITY TEST

Noting that the main pattern is a tree-shaped connected structure, disjoint patterns should be removed. Therefore, a small object removal method (particle filtering) is applied to exclude the patterns, which are most likely due to scratches and imaging artifacts. The result of this step is shown in Figure 3(d).

5) CENTRE DETECTION

The dendritic patterns emerge from a centered circle. The center represents the root node and the level-2 nodes lie on the perimeter of the circle. To locate the circle, we use the Hough transformation [60] that provides the center and radius of the circle.

6) THINNING

To facilitate extraction of the coherent tree skeleton, the widths of branches are narrowed down to a unit-pixel width, as shown in Figure 3(e). The thinning is performed by following the root node towards the leaves of the tree and removing the extra pixels on both sides of the branches. The thinned tree is used to extract the skeleton and the bifurcation and leaf points of the tree (shown by red dots in Figure 3(f)).

7) KEYPOINT EXTRACTION

The extracted skeleton of the dendritic image is a binary image with two segments including the one-pixel-wide pattern and the background. At this step, we convert the extracted pattern to the representative graph. The tree is determined by pixels with three different types of feature points including (i) the root node (i.e., the center of the circle), (ii) seed points (the nodes at the first level, where branches emerge from the circle), (iii) the bifurcation nodes, and (iv) the leaves (terminating pixels). The rest of the pixels are considered regular points and do not contribute to the tree representation. we use keypoint and feature points interchangeably in this paper.

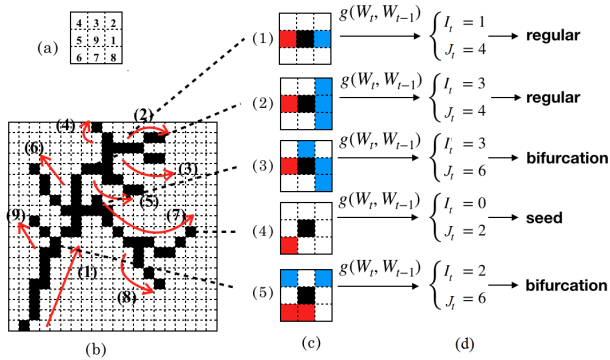


FIGURE 4. Overview of the proposed feature extraction method. **(a)** The 3×3 sliding window, **(b)** A sample dendritic partial after preprocessing and skeletonizing step, where the black pixels (1) represent the skeleton and white pixels (0) represent the background; **(c)** The 3×3 filter with pixel indexes shown by set \mathcal{W}_t at current position; red pixels represent the previous center of the filter and blue pixels represent the newly visited pixels by the window at its current position, i.e. $\mathcal{W}_t \setminus \mathcal{W}_{t-1}$; **(d)** The obtained indicators $[I_t, J_t] = g(\mathcal{W}_t, \mathcal{W}_{t-1})$ obtained by evaluating the filter at its current and previous positions to determine the type of pixels.

To identify these nodes, we scan a $3\text{pixel} \times 3\text{pixel}$ filter along the branches from the root node (center of the dendritic structure) towards the leaves of its tree-like structure using depth first traversal method, as shown in Figure 4. Note that we use the binarized image, where black and white pixels respectively represent the dendrite pattern and the background as shown in Figure 4(b). At each step, we move this 3×3 window one pixel in the direction of the branch (up, down, right, left) to visit and identify keypoints (bifurcation, and end nodes). Suppose that \mathcal{W}_t is a set of the index of black (foreground) pixels visited by the sliding window at time point t , where the center of the window is the pixel under investigation. The point type is determined by comparing the pixel values visited by the 3×3 filter at its current and previous positions, or equivalently by comparing the sets \mathcal{W}_t and \mathcal{W}_{t-1} . More specifically, we form two indicator functions. The first indicator I_t counts the number of newly visited black pixels after moving the filter to its current position (shown by blue color in 4(c)). If \mathcal{W}_t and \mathcal{W}_{t-1} respectively represent the set of black pixels visited by the filter at time points t and $t - 1$, then $I_t = |\mathcal{W}_t \setminus \mathcal{W}_{t-1}|$, where $|S|$ is the cardinality of set S . Intuitively, we have the following set of rules:

$$\begin{cases} I_t = 0 : & \text{Seed Point or End Point} \\ I_t = 1 : & \text{Regular Point} \\ 2 \leq I_t \leq 3 : & \text{Bifurcation Point or Regular Point} \\ I_t \geq 4 : & \text{Bifurcation Point} \end{cases} \quad (1)$$

There is no confusion between the seed points (first-level nodes) and the end points (leaves), as they are present, respectively, at the beginning and end of the tree traversal. However, since the bifurcation nodes can be confused with regular points for $2 \leq I_t \leq 3$, we define the second indicator J_t that computes the number of transitions from 0 (white) to 1 (colored) pixels and vice versa when traversing the 8 surrounding pixels of the filter (i.e., transition from

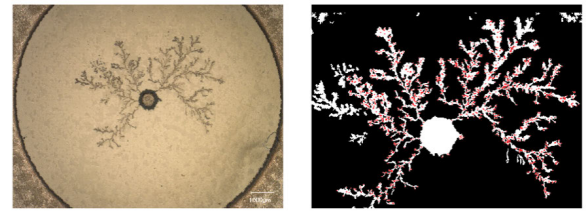


FIGURE 5. (Left): A silver dendrite grown on synthetic paper that had previously been soaked with a liquid electrolyte (Right): extracted features points are shown by red dots.

pixel 1 to pixel 2, from pixel 2 to pixel 3, ... and pixel 8 to pixel 1). More specifically, if we label the pixels of the current filter as shown in Figure 4(a) and represent the value of pixel i with $p_W(i, t)$, then J_t is defined as:

$$J_t = \sum_{i=1}^{n=8} |p_W(1 + \text{mod}(i, 8), t) - p_W(i, t)|, \quad (2)$$

where $\text{mod}(i, j)$ is the remainder of i divided by j . If $J_t \leq 4$ represents two branches emerging from the current point, one towards the root and one towards the leaf, it is therefore considered as a regular point. Otherwise, $J_t \geq 6$ represents a bifurcation point with more than two branches.

Figure 4(b) represents sample points of different types. In summary, the decision rules can be stated as:

$$\begin{cases} I_t = 0 : & \text{Seed Point or End Point} \\ I_t = 1 : & \text{Regular Point} \\ 2 \leq I_t \leq 3 \text{ and } J_t \leq 4 : & \text{Regular Point} \\ \text{else} : & \text{Bifurcation Point} \end{cases} \quad (3)$$

Figure 5 represents a dendritic image along with the feature points (seed, bifurcation, leaves) extracted using the method discussed above. As may be seen in this figure, the extracted keypoints can be different from the original pattern due to noise and imaging artifacts.

B. GRAPH REPRESENTATION OF DENDRITES

The extracted feature points determine the morphology of the extracted skeleton. At this stage, we complete the process by extracting the numerical information associated with the feature points that fully determine the tree. Each tree is represented by an attributed directed acyclic graph $\mathcal{G} = (V, E)$, where $V = \{n_1, n_2, \dots, n_N\}$ is the set of N vertices and $E = \{e_{ij} | i, j = 1, 2, \dots, N\}$ is the set of edges. We have $e_{ij} = 1$ if node n_i is the direct parent of node n_j or vice versa; otherwise we set $e_{ij} = 0$. To construct the attributed graph, we traverse the tree from the root node (the center of the circle) towards the leaves visiting and weighing all the feature points including seed, bifurcation, and leaves. Each node n_i (or equivalently the edge e_{ji} that terminates at this node) is assigned with a vector (l_i, θ_i) , where l_i denotes the distance from the current node to its parent and θ_i represents the orientation of the branch. For instance, if the position of n_i is (x_i, y_i) , and its parent is $n_j = \mathcal{P}(n_i)$ with position (x_j, y_j) , then

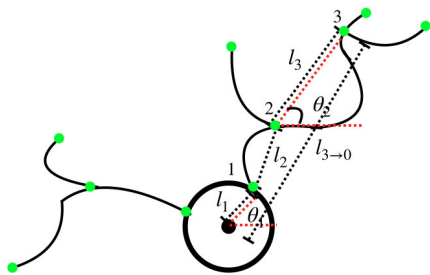


FIGURE 6. Numerical representation of the extracted skeleton through weighing the nodes. Each node n_i is associated with (l_i, θ_i) representing the length and orientation of the straight line connecting n_i to its parent $\mathcal{P}(n_i)$.

we have $l_i = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ and $\theta_i = \tan^{-1} \frac{y_i - y_j}{x_i - x_j}$. This concept is presented in Figure 6.

IV. GRAPH MATCHING ALGORITHM

In this section, the proposed consistency-penalized graph-matching (CPGM) algorithm is presented. The idea is to find the best mapping between the nodes of the test and reference graphs, respectively, denoted by $\mathcal{G}^{(t)}$ and $\mathcal{G}^{(r)}$. Once the mapping is established, the similarity score is calculated and this establishes the legitimacy of the test graph.

In [61], we proposed a graph-matching algorithm that takes a top-down approach. The idea was to start with the root node and make a linkage between the nodes of the test and reference trees at the first level such that the total distance between the paired nodes is minimized. Once the level-1 nodes are paired, we investigate each level-1 pair of nodes and make linkages between their children (the second-level nodes). The algorithm is continued until all nodes are mapped between the test and the reference trees. This method performs well in low-noise regimes but is extremely sensitive to the *level-shifting* errors due to noise and scratches. A scratch may cause a fake bifurcation node or may delete an existing branch from the test tree and consequently may compromise the level alignment. This error causes all the nodes in the emerging subgraph to be shifted up or down in the tree structure. In other words, this method suffers from the *error-propagation* issue, as all the nodes in the emerging sub-graphs of a miss-aligned node in the test tree are mapped to wrong nodes in the reference tree causing the algorithm to perform poorly. The impact is higher if the misalignment occurs in the lower levels. To avoid this issue, we proposed an embedded level re-alignment algorithm in [61] that attempts to realign the out-of-sync levels. However, this method can only repair 1-level shifts. Inspired by this observation, here we adopt a substantially different approach based on ad-hoc matching.

Notations: Before proceeding with the details of the proposed algorithm, we define our notations. We use $n_i^{(r)}$, $l_i^{(r)}$, $\theta_i^{(r)}$ to respectively denote the i^{th} node in the test tree, and the length and the orientation of the edge connects $n_i^{(r)}$ to its

parent. In general, we use subscripts to denote the node index and postscripts to denote the corresponding tree. Likewise, $N^{(t)}$, and $N^{(r)}$, represent the number of nodes in the test and reference trees. The set of the children of node n_i is given by $\mathcal{C}(n_i) = \{n_j | \mathcal{P}(n_j) = n_i\}$. Likewise, the set of the siblings of node n_i is given by $\mathcal{S}(n_i) = \{n_j | \mathcal{P}(n_j) = \mathcal{P}(n_i)\}$. The depth of node n_i is the minimum number of edges to reach the root node and denoted by $\text{dep}(n)$. The set of nodes at level i is defined as $D_i^{(t)} = \{n_j | \text{dep}(n_j) = i\}$. We also define notations for establishing linkages (pairing the nodes between the test and reference trees) that is the core of the proposed algorithm. We use $n_i^{(t)} \leftrightarrow n_j^{(r)}$ to show that nodes $n_i^{(t)}$ and $n_j^{(r)}$ are linked. We show the set of nodes with an active linkage in the test and reference trees by $\mathcal{L}^{(t)}$ and $\mathcal{L}^{(r)}$. Furthermore, $\mathcal{L}^{(t,r)} = \{(n, m) | n \in \mathcal{L}^{(t)}, m \in \mathcal{L}^{(r)}, \text{Link}(n, m) = 1\}$ denotes the set of active links between the test and reference trees. Finally, we define the set of free nodes as the complement of linked nodes $V_{\text{free}}^{(t)} = V^{(t)} \setminus \mathcal{L}^{(t)}$, and $V_{\text{free}}^{(r)} = V^{(r)} \setminus \mathcal{L}^{(r)}$, where $A \setminus B$ means the set of members of A excluding the members of B. We omit the postscript when it is clear from the context for notation convenience.

A. OVERVIEW OF THE ALGORITHM

The proposed algorithm is initialized by setting all the nodes as free nodes. Then, it takes an ad-hoc approach to make linkages between the free nodes of the test and reference trees based on their pairwise distances. Then, the consistency of the links in terms of their nodes' local subgraph including their parents, direct children and siblings is assessed. The consistency scores are compared against a predefined threshold; the links with higher consistency scores are more likely to represent correct matches, hence their linkages remain established. However, the nodes with less consistent sub-networks are more likely to be wrong matches, hence their links are broken. Next, we execute the mapping algorithm only for the free nodes with no active linkage. We repeat this algorithm until the mapping converges, the desired similarity achieved, or the maximum number of iterations is reached. The following are the details of the different components of the algorithm.

B. NORMALIZATION

To realize a fair and *scale-invariant* pairing mechanism and also avoid sensitivity to the units of the length and angle, we normalize the node metrics (l_i, θ_i) . More specifically, we use $l_i \rightarrow (l_i - \mu(l_i))/\sigma(l_i)$, and $\theta_i \rightarrow (\theta_i - \mu(\theta_i))/\sigma(\theta_i)$, where $\mu(\cdot)$ and $\sigma(\cdot)$ are the mean and standard deviation of parameters calculated using all nodes of the graph.

C. NODE PAIRING

An important part of the algorithm is making linkages between free nodes of the test and reference graph. The linkages are made based on the similarity of the associated node metrics (l_i, θ_i) as well as their depth level $\text{dep}(n)$.

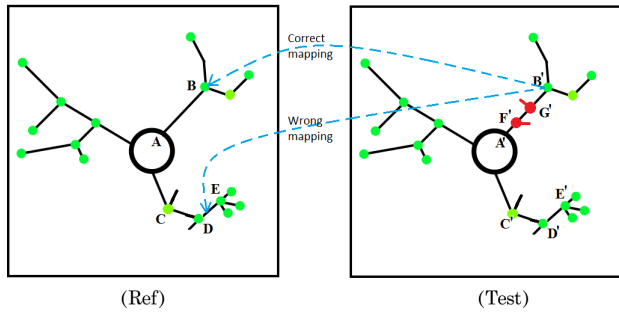


FIGURE 7. Demonstration of the use of consistency check to eliminate matching errors. The length of branch terminating at point B' is shrunk from $l_B = l_{AB}$ to $l_{B'} = l_{C'B'}$ due to the noise-related fake branch (red dots). Its level also is changed $\text{dep}(B') : 1 \rightarrow 3$, therefore this node B' in the reference tree is mistakenly mapped to node E with similar branch length and orientation in the reference tree instead of node B . The consistency verification identifies and filters out these wrong linkages.

More specifically, we define the following distance metric:

$$d(n_i^{(t)}, n_j^{(r)}) = (1 - \alpha) \left(\sum_{k=0}^{\min\{\text{dep}(n_i), \text{dep}(n_j)\} - 1} \left(\beta^k (l_{i,k}^{(t)} - l_{j,k}^{(r)})^2 \right) + \gamma (\theta_{i,k}^{(t)} - \theta_{j,k}^{(r)})^2 + \alpha |\text{dep}(n_i^{(t)}) - \text{dep}(n_j^{(r)})| \right)^{1/2}, \quad (4)$$

where α , β , and γ are tuning parameters between 0 and 1 to balance between the importance of the depth of the node, the length of the branch, and the orientation of the branch. Also, in order to mitigate bias to the insertion and deletion of fake branches that affect $l_i^{(t)}$, we use the accumulative length by summing over the links from the current node to the root node, while penalizing the relative importance of the link lengths as we move away from the current node. This factor is represented by $\sum_{k=0}^{\min\{\text{dep}(n_i), \text{dep}(n_j)\}} \left(\beta^k (l_{i,k}^{(t)} - l_{j,k}^{(r)})^2 \right)$ in (4), where $l_{i,k}^{(t)}$ is the distance of the k^{th} ancestor of node n_i to its $(k + 1)^{\text{th}}$ ancestor. In other words, we put more emphasis in the distance to the direct parent ($l_{i,0}^{(t)} - l_{j,0}^{(r)}$) and less on the k^{th} order ancestors by enforcing the penalization factor β^k .

In order to perform the pairing, we iteratively pairs free nodes (the nodes with no active linkages) between the test and references trees until the algorithm converges to a stable linkage. To this end, we use the *Hungarian* algorithm, also known as *Munkres assignment* [62] that can pair the elements of two sets based on a given distance metric without exhaustively examining all possibilities. In this paper, we use this algorithm to pair the nodes of $\mathcal{G}^{(t)}$ and $\mathcal{G}^{(r)}$ based on level-distance metric $d(n_i^{(t)}, n_j^{(r)})$ defined in (4).

D. CONSISTENCY ANALYSIS

The results of the *Munkres Assignment* in the previous section might not be accurate due to the noise-related graph topology changes. An illustrative example of such a scenario is presented in Figure 7, where one node in the test pattern undergoes substantial changes in its metric (i.e., $n_i^{(t)}$ in this case) due to fake branches caused by scratches. In such

Algorithm 1 Consistency-Penalized Graph Matching Algorithm

Inputs: test graph $\mathcal{G}^{(t)} = (V^{(t)}, E^{(t)})$; reference graph $\mathcal{G}^{(r)} = (V^{(r)}, E^{(r)})$; desired similarity score \mathcal{H}_{\min}

Outputs: Similarity score \mathcal{H} ; optimal linkage $\mathcal{L}^{(t,r)}$

Initialization:

Add all nodes to the set of free nodes $V_{\text{free}}^{(t)} = V^{(t)}$, and $V_{\text{free}}^{(r)} = V^{(r)}$; Equivalently $L^{(t)} = \{\}$, $L^{(r)} = \{\}$, $\mathcal{L}^{(t,r)} = \{\}$;

Set parameters $\alpha, \beta, \gamma, \alpha_c, \beta_c, \gamma_c, \text{Iter}_{\max}, T_c$

Set $\text{terminationFlag} \leftarrow \text{False}$; $t = 0$;

while NOT *terminationFlag* **do**

$t = t + 1$;

$\mathcal{L}_t^{(t,r)} = \text{Munkers}(V_{\text{free}}^{(t)}, V_{\text{free}}^{(r)}, d())$: Establish optimal linkage using Munkers Assignment and distance metric defined in (4)

for all links $L_{ij} = (n_i, n_j) \in \mathcal{L}_t^{(t,r)}$ **do**

Calculate consistency score $S_c(L_{ij}) = f(n_i, n_j)$ using (5)

Calculate link-break prob. $P_{\text{break}}(L_{ij})$ using (6)

Select link L_{ij} with Prob. $P_{\text{break}}(L_{ij})$

if link L_{ij} selected **then**

$\mathcal{L}_t^{(t,r)} \leftarrow \mathcal{L}_t^{(t,r)} \setminus \{(n_i, n_j)\}$ break the links

$\mathcal{L}^{(t)} \leftarrow \mathcal{L}^{(t,r)} \setminus \{n_i\}$

$\mathcal{L}^{(r)} \leftarrow \mathcal{L}^{(r)} \setminus \{n_j\}$

end

end

compute matching rate R_m using (7)

Check termination criteria:

compute similarity score \mathcal{H} using (8)

if $\mathcal{H} > \mathcal{H}_{\min}$ **or** $t \geq \text{Iter}_{\max}$ **or** $R_m \geq 90\%$ **or**

$\mathcal{L}_t^{(t,r)} = \mathcal{L}_{t-1}^{(t,r)}$ **then**

| $\text{terminationFlag} \leftarrow \text{True}$

end

end

scenarios, the node may be mapped to a wrong node with similar metrics in the reference pattern. Here, we propose the novel method of consistency-check for the paired nodes as follows. This step essentially examines the established linkages and provides a consistency score denoted by $S_c(L_{ij})$ for all links $L_{ij} \in \mathcal{L}^{(t,r)}$. The idea is to investigate the nodes $n_i^{(t)}$ and $n_j^{(r)}$ that make the linkage L_{ij} in terms of their parents ($\mathcal{P}(n_i^{(t)}), \mathcal{P}(n_j^{(r)})$), the sets of their children ($\mathcal{C}(n_i^{(t)}), \mathcal{C}(n_j^{(r)})$), and the sets of their siblings ($\mathcal{S}(n_i^{(t)}), \mathcal{S}(n_j^{(r)})$). If these local sub-networks of nodes $n_i^{(t)}$ and $n_j^{(r)}$ are linked, it indicates that their subnetworks create similar topology, and hence this linkage L_{ij} is confirmed. On the other hand, if the subnetworks are inconsistent, the linkage L_{ij} is not verified and is disconnected, and nodes $n_i^{(t)}$ and $n_j^{(r)}$ can join the next round of pairing as free nodes. Overall, the consistency score $S_c(L_{ij})$

increases with the number of paired relatives and is defined as:

$$\begin{aligned}
S_c(L_{ij}) &= f(n_i^{(t)}, n_j^{(r)}) \\
&= \alpha_c I[\mathcal{P}(n_i^{(t)}) \leftrightarrow \mathcal{P}(n_j^{(r)})] \\
&\quad + \beta_c \frac{\sum_{n_k \in \mathcal{C}(n_i^{(t)}), n_l \in \mathcal{C}(n_j^{(r)})} I[n_k \leftrightarrow n_l]}{\max(|\mathcal{C}(n_i^{(t)})|, |\mathcal{C}(n_j^{(r)})|)} \\
&\quad + \gamma_c \frac{\sum_{n_k \in \mathcal{S}(n_i^{(t)}), n_l \in \mathcal{S}(n_j^{(r)})} I[n_k \leftrightarrow n_l]}{\max(|\mathcal{S}(n_i^{(t)})|, |\mathcal{S}(n_j^{(r)})|)}, \quad (5)
\end{aligned}$$

where α_c , β_c , γ_c are tuning parameters, can be adjusted with cross-validation or can be simply set to $\alpha_c = \beta_c = \gamma_c = 1/3$. In this equation, $I()$ is the indicator function, and $n_i \leftrightarrow n_j$ means linkage between n_i and n_j , i.e. $(n_i, n_j) \in \mathcal{L}^{(t,r)}$.

The consistency verification step computes $S_c(L_{ij})$ for all active links $L_{ij} \in \mathcal{L}^{(t,r)}$, and splits the links into two sets. The link with consistency scores higher than a predefined threshold T_c remain valid. The links with lower consistency scores are excluded from the $\mathcal{L}^{(t,r)}$ and their respective nodes $n_i^{(t)}$ and $n_j^{(r)}$ added to the set of free nodes $V_{\text{free}}^{(t)}$ and $V_{\text{free}}^{(r)}$. To impose more randomness, a small part of verified links are also broken and their nodes are added to the set of free nodes. We break these links with probability $1 - S_c(L_{ij})$. In other words, the probability of breaking a link is:

$$P_{\text{break}}(L_{ij}) = \begin{cases} 1 & \text{if } S_c(L_{ij}) < T_c \\ 1 - S_c & \text{if } S_c(L_{ij}) \geq T_c \end{cases} \quad (6)$$

In the next iteration, the pairing operation is performed only among the set of free nodes. Once a full iteration is completed, we calculate the matching rate R_m as follows:

$$R_m = \frac{|\mathcal{L}^{(t,r)}|}{\min\{|V^{(t)}|, |V^{(r)}|\}}, \quad (7)$$

which gives the ratio of nodes that are linked with consistency scores above the threshold. Likewise, we define a global similarity score \mathcal{H} as:

$$\mathcal{H} = R_m \left(1 - \frac{\sum_{(n_i, n_j) \in \mathcal{L}^{(t,r)}} d(n_i, n_j)}{|\mathcal{L}^{(t,r)}|}\right), \quad (8)$$

which basically takes the average of similarity scores among the linked nodes after final matching. The term R_m is included to consider the dissimilarity of unmatched points.

The algorithm continues until one of the termination criteria is achieved: one of the following stopping criteria is met: (i) the majority of the nodes are paired with consistency score above threshold (i.e. $R_m \geq 90\%$), (ii) the mapping $\mathcal{L}^{(t,r)}$ does not significantly change for consecutive iterations, (iii) a desired similarity score is met $\mathcal{H} \geq \mathcal{H}_{\text{min}}$, or (iv) the maximum number of iterations $Iter_{\text{max}}$ is reached. A summary of the whole process is presented in Algorithm.1.

E. COMPUTATIONAL COMPLEXITY

The proposed graph-mapping algorithm completes the steps in $O(mKN^2)$ at worst case, where m is the number of reference objects in the reference database (if the ID is not already

available), K is the number of iterations, and N is the number of nodes in the test graph. Since the complexity grows linearly with the number of reference objects, the algorithm is easily scalable to large-scale networks. However, the complexity grows with N^2 . Therefore, a reasonable pattern granularity should be considered in the dendrite growth process, or a limited allowable tree depth should be considered when converting the captured image to the representative graph translation stage, to avoid unnecessarily heavy computations while maintaining a reasonable authentication specificity.

V. EXPERIMENTAL RESULTS

In this section, we present numerical results to evaluate the performance of the proposed algorithm in authenticating dendritic objects when subjected to imaging artifacts, noise, scratches, scaling, rotation and skew. Here, we use images taken from 50 dendrites produced at Arizona State University by the growth of silver electrodeposits on a solid electrolyte layer on a microfabricated substrate. We compare the proposed algorithm with the state of the art in terms of matching rate, identification error, representation quality, and storage size requirements. One key feature of the proposed numerical coding of the dendrites is its full reliance on the main structure of the dendrite through extraction of keypoints that results in a natural of robustness against other smaller patterns created by noise and imaging artifacts. This is demonstrated in different test scenarios.

A. KEYPOINT EXTRACTION

The first test compares the proposed keypoint extraction method with various popular feature mapping algorithms, including Harris, BRISK, FAST, Min Eigen, and SURF, when applied to the noise-free reference and noisy test images. The results of this test are shown in Figure 8. The reference and test images are overlapped and the keypoints extracted from the reference and test images are respectively shown by red and green markers. The feature points extracted by different methods are substantially different. This figure illustrates three key advantages for the proposed method. Firstly, the dendritic pattern is represented with much fewer keypoints compared to other methods. This significantly reduces the storage and communication throughput requirements in large-scale networks. Secondly, almost all of the features extracted by our method coincide with the dendritic pattern while those extracted by the other methods represent both the dendritic and scratch-related patterns. Thirdly, and more importantly, the matching rate between the keypoints extracted from the reference and test images using our method exhibit a much higher matching rate than other methods that result in a better identification rate.

B. MATCHING RATE

The proposed algorithm operates based on establishing linkages between the keypoints of test and reference image in an iterative fashion followed by finding a similarity score for the optimal matching (Figure 9). If the similarity score between

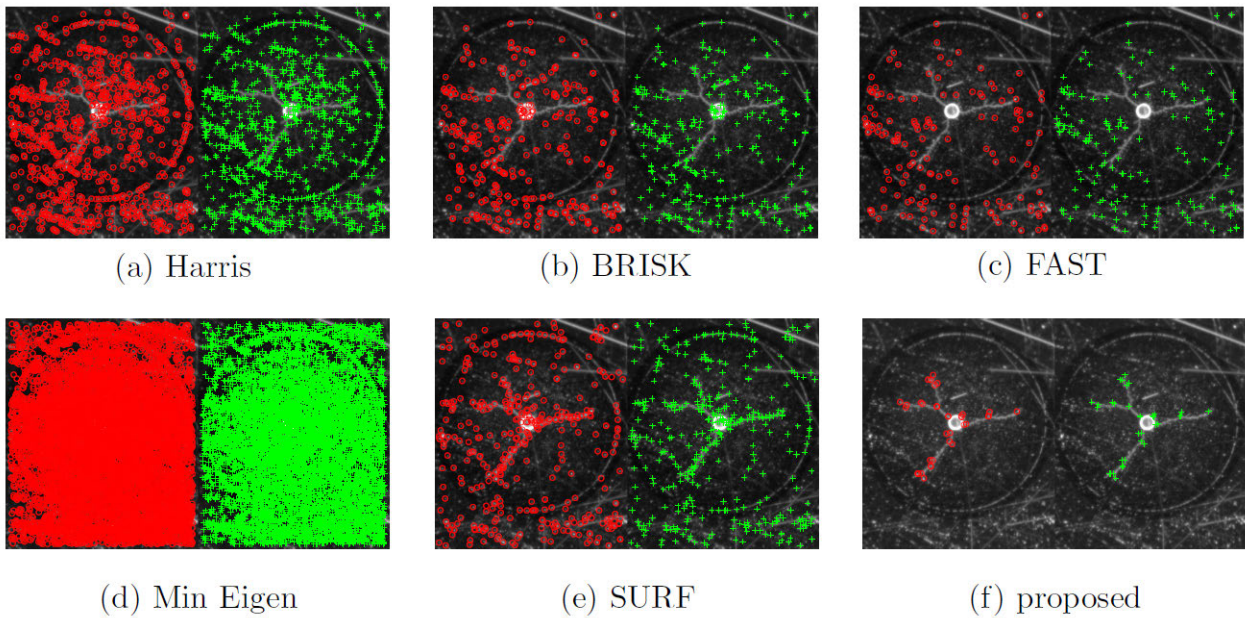


FIGURE 8. Descriptive key extracted by different feature matching algorithms. Green and red dots represent keypoints extracted from the reference and test images, respectively.

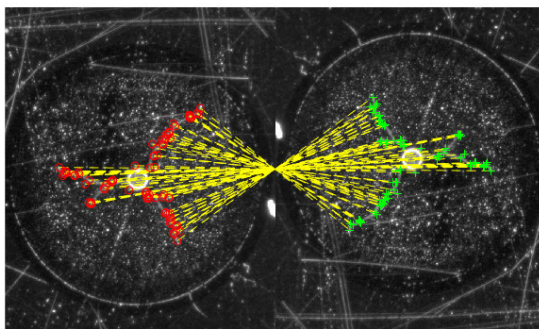


FIGURE 9. Keypoint mapping: test image is rotated 180° clockwise for a better illustration of linkage between the keypoints.

the test and the respective reference image is below a threshold, or lower than the similarity between the test and a different reference image, then the identification/authentication mechanism fails.

Figure 10 shows the similarity of 3 representative samples against 50 reference samples. The test image is obtained by adding noise to the reference images with three signal to noise (SNR) values. Case (a) represents the noise-free scenario ($SNR = \infty$), in which all test samples are correctly verified with a matching rate of $R_m = 100\%$. For case (b) with $SNR = 12.5 \text{ dB}$, the matching rate has reduced from 100% to about [25% – 45%] range, but the test samples are still correctly identified since their matching rates with the right reference samples are still higher than their matching rate with any other reference sample. However, if we increase the noise power, at some point the object identification may fail. For case (c) with the SNR of $SNR = 10.85$, the matching rate

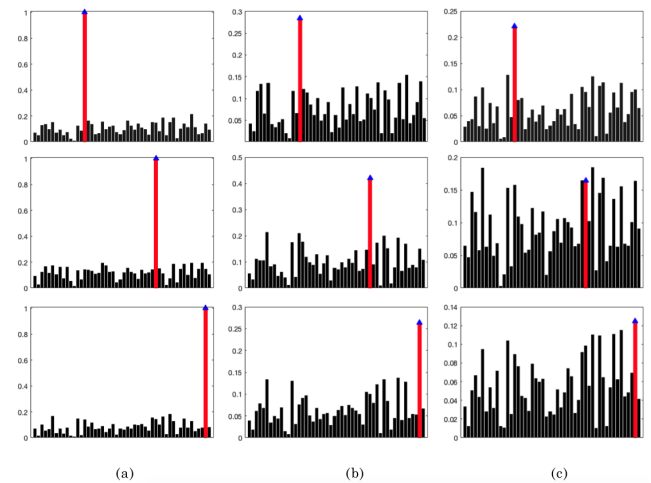


FIGURE 10. The matching rate between 3 test samples and 50 reference samples. The three rows from top to bottom represent random draw samples $ID\#15$, $ID\#34$, $ID\#49$. Each bar represents a reference image, and the red bar represents the corresponding reference image. The height of the bars represents the matching rate, namely the rate of robust and consistent links established between the test and reference graphs by the proposed algorithm. The test images are the noisy versions of the reference images with three SNR values including (a) noise-free (b) $SNR = 12.5 \text{ dB}$, (c) $SNR = 10.85 \text{ dB}$.

has dropped to the [10% – 25%] range, and only two of three samples are correctly identified.

C. COMPARATIVE ANALYSIS

We conducted another experiment to test the performance of the proposed method in comparison with other popular feature extraction and matching techniques when the test image is subject to rotation and Gaussian noise with different SNR values. To realize a fair comparison, other methods are

TABLE 1. Comparison of the proposed method with the popular feature extraction and matching methods for image identification at different noise levels. $R : 180^\circ$ represents a 180 degrees clockwise rotation. The results are in terms of identification accuracy (ACC) and the matching rate (MR) averaged over 50 samples. The numbers are in percentage (%).

Methods	SNR 19.13 ($R : 180^\circ$)		SNR 16.56		SNR 12.5		SNR 10.85		SNR 9.58	
	ACC	MR	ACC	MR	ACC	MR	ACC	MR	ACC	MR
Min Eigen + Exhaustive	98	0.37	98	0.24	84	0.07	46	0.029	10	0.014
FAST + Exhaustive	10	14.62	6.0	10.59	2.0	6.25	8.0	1.9	6.0	0.25
BRISK + Exhaustive	54	10.91	98	16.8	98	11.72	96	4.3	96	0.63
SURF + Exhaustive	100	55.53	98	52.9	92	38.89	94	27.26	54	16.29
Harris + Exhaustive	100	13.33	98	10.4	90	2.60	46	0.19	14	0.04
MSER + Exhaustive	38	92.96	24	85.8	4.0	29.67	18	14.69	8.0	3.49
proposed CPGM	100	83.58	100	84.2	100	29.67	98	17.65	86	9

TABLE 2. The number of keypoints extracted by different methods at different SNR levels used for results in Table 1.

Methods	SNR 19.13 (R)	SNR 16.56	SNR 12.5	SNR 10.85	SNR 9.58
Min Eigen	3.55e4	3.65e4	3.64e4	3.64e4	3.67e4
FAST	4.73e1	6.60e1	1.08e2	3.60e2	2.45e3
BRISK	1.08e4	1.26e2	1.74e2	4.42e2	2.60e3
SURF	1.91e2	1.94e2	2.04e2	2.17e2	2.36e2
Harris	4.74e2	4.99e2	6.62e2	4.28e3	1.18e4
MSER	1.05e1	1.17e1	1.96e1	3.36e1	1.17e2
CPGM	4.03e2	4.10e2	5.29e2	6.29e2	9.59e2

combined with exhaustive search for optimal performance. Table 1 compares the performance of the methods in terms of identification accuracy (ACC) and sum matching rate (MR). Likewise, Table 2 presents the number of extracted features (nF) on average for each method. The results of this section is obtained by averaging over 50 dendrite samples.

Table 1 suggests that only some of the algorithms are robust to rotation including our proposed algorithm, along with the Min Eigen, Fast, BRISK, SURF, Harris, and MSER techniques. A short description and references for these methods are provided in section II.B. Note that we use an exhaustive search when comparing the pairwise distances between the feature vectors in the reference and test images for the highest performance. Two feature vectors match when their distance is less than a predefined threshold.

The rest of the algorithms perform poorly with rotation which indicates an important weakness for their use in a poorly controlled viewing environment. In terms of robustness against noise, the proposed CPGM algorithm outperforms the majority of the methods in terms of identification accuracy and matching rate. Only BRISK shows slightly higher accuracy at extremely noisy regimes. However, as we see in Table 2, this slight improvement for BRISK comes at the cost of using more features by a factor of about 100, which is not affordable for fast image identification. Also, its lower matching rate of $R_m < 1\%$ is alarming. Although the correct reference dendrite (out of 50 samples) is selected as the matching reference for test objects in 96% of experiments, the accuracy can drop for larger datasets due to the BRISK's low matching rate between feature points. Harris, SURF and Min Eigen methods also perform reasonably well at high SNR regime, but their accuracy and feature matching rate drops dramatically with increasing noise variance and this makes them inappropriate for dendritic pattern recognition.

Further, the number of features required for the Min Eigen method is substantially larger than the proposed method that which makes it a poor choice for fast identification. Finally, the performance of MSER and FAST is not acceptable even at higher SNR regimes leaving them off the table for this application.

D. NOISE AND IMAGING ARTIFACTS

The performance of the identification stage can be compromised by the distortion of the test image. We model this distortion by noise to represent different effects such as scratches, dirt, camera inaccuracy, light reflection, low illumination intensity, and even motion-related blurring. Figure 11 shows the performance of the proposed CPGM algorithm for six samples. Figure 11 (left) shows the convergence of the algorithm when the signal to noise level is $SNR = 13.5$. As is evident from the figure, it takes about 10 iterations for the algorithm to converge to the best possible matching rate. Figure 11 (right) presents the final matching rate versus different SNR values for six individual samples as well as the average across all samples.

E. SCRATCH

Scratch is a normal type of distortion occurs when people carry, touch, and scan the dendrite tags. These scratches can appear as fake branches in the dendrite structure and hence negatively affect the authentication algorithm. In this test, we will prove our proposed algorithm is flexible to handle the normal scratch and fake branch scenarios. Fig 12 shows the performance of the proposed algorithm using the original samples (a and c) and samples with severe scratches and multiple fake branches (b and d). When applying the algorithm to the original and scratched paired samples (a) and (b), more than 83.75% of points are correctly matched within

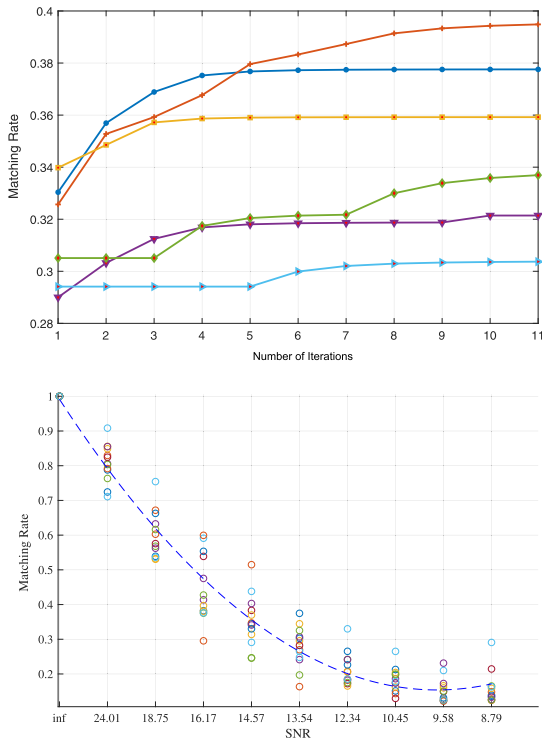


FIGURE 11. (Top): the convergence of the algorithm in terms of matching rate (R_m) versus iterations for six different samples, each of which is represented by a different color. (Bottom): the achieved matching rate after the convergence point versus SNR for twelve individual samples (represented by circles of different colors) as well as the average over all samples, represented by a dashed line.

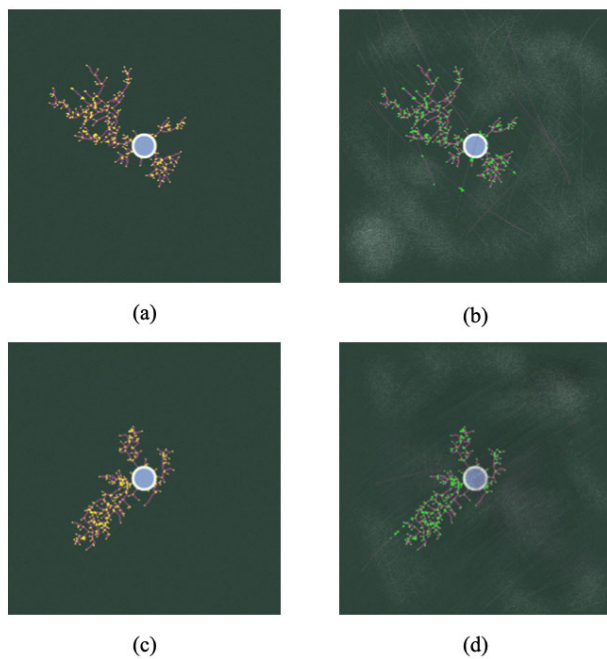
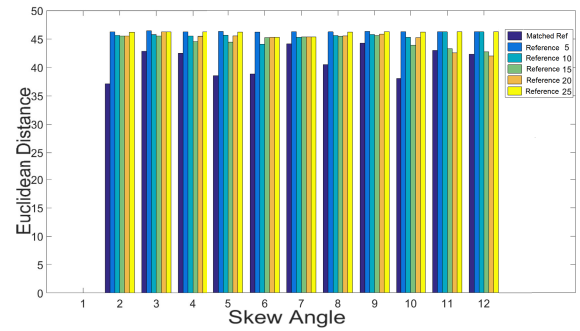
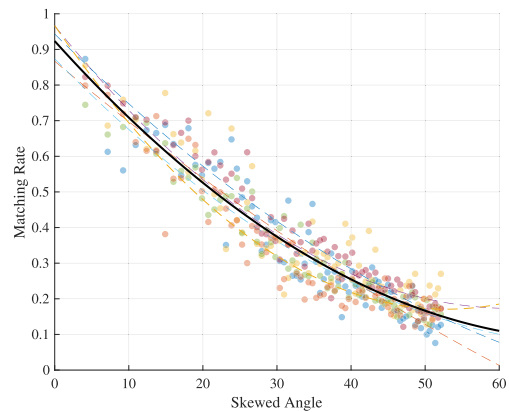


FIGURE 12. (a) and (c): the original dendritic samples with identified nodes marked as yellow; (b) and (d): the scratched samples with identified nodes marked as green.

10 iterations. Likewise, the matching rate for the pair (c) and (d) is 87.32% within 10 iterations. When comparing the



(a)



(b)

FIGURE 13. (a) The average Euclidean distances between the matching keypoints of a representative skewed image and the matched reference as well as five randomly selected reference images; (b) The matching rate between the test image and corresponding reference image for skewness ranging from 0° to 60° at noise-free regime. Dots of different colors represent the test results for different test samples.

scratchy samples with 100 reference objects, both successfully are identified and matched to the corresponding reference original images. Both tests prove that even though the fake branches slightly reduce the feature matching rate, but the proposed graph matching algorithm is powerful and flexible enough to find the right reference images by matching the rest of the nodes that retain distinctive multi-dimensional features.

F. SKEW

Another important source of image distortion is *skew* that is caused by the misalignment between the camera focus line and object exposure that changes the angle of view. This problem is more commonly observed when pictures are taken by handheld and cellphone cameras. Even a completely noise-free but skewed image may fail the identification stage if the matching algorithm is not well designed to mitigate this issue. One advantage of the proposed algorithm is its natural robustness against skewness since the overall structure of the dendritic pattern and the metrics associated with the extracted keypoints do not substantially change with the skew. The behavior of the proposed algorithm when processing skewed images is shown in Figure 13.

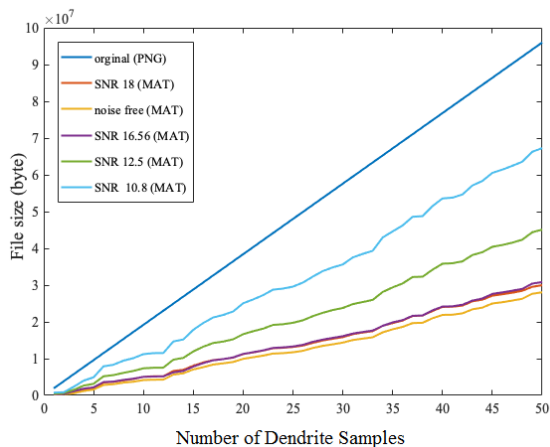


FIGURE 14. Comparison of file sizes required to store original images for multiple samples in PNG format as well as the size of MATLAB files containing the representative graphs with numerical content in different noise regime.

Figure 13a shows the performance of the proposed algorithm in terms of identification accuracy when the test image is skewed with 1 to 12 degrees. A skewed test image compared against the reference datasets and the achieved similarity score in terms of the average Euclidean distance between the matched points of the test image and the right reference image as well as five randomly selected reference images are shown. This experiment shows that a skew up to 10 degrees is tolerable by the algorithm, while the resulting Euclidean distance for the matched reference is not the minimum for skews over 10 degree. For instance, Ref#20 mistakenly identified as the matching reference for skew of 11° and 12°. The tolerance of this algorithm to skewness obviously depends on many factors including the size of the reference dataset, and the granularity of the dendritic patterns. Figure 13a shows the performance of the proposed algorithm in terms of another key metric, namely the keypoint matching rate (R_m) when the test image undergoes skew from 0° to 60°. The results are shown for six individual samples as well as the average for all six samples. The similarity score decreases with skew as expected. Our algorithm retains a 40% matching rate in 30-degree skew, which is much higher than the average successful matching rate of other methods in Table. 1. It seems that the typical skew up to 10° remains fully in the safe zone.

G. COMPRESSION EFFICIENCY

Here, we assess the storage requirement of the proposed method. The original images can be compressed and stored using common formats such as PNG. Here, the proposed method stores the reference objects in terms of graphs that contain numerical representations of the dendritic patterns with storage requirements shown in Fig 14. Note that the proposed method does not replace image compression methods but adds yet another layer of compression by translating the image into attributed graph format with small storage requirements, as a desirable feature for large-scale networks.

Figure 14 shows that that the required file size to store samples as representative graphs is approximately average 4500 times lower than typical compressed methods like PNG in typical noise condition. This considerable compression rate comes without significant compromise in the identification accuracy. This is another key advantage of the proposed method that is scalable to large-scale networks without the need for massive information exchange and storage requirements to maintain and update the reference databases.

VI. CONCLUSION

This paper considers the use of the unique key features of dendrite patterns for image-based identification and authentication. These dendritic tags can be used as optical PUFs to add an additional layer of security to IoT devices. Dendritic tags can mitigate counterfeiting and cloning attacks since they are not clonable with the existing technology due to their 3D facet and nano-scaled granularity. However, we noted that there exists no customized method for authenticating such devices with acceptable performance and affordable complexity that scales to large-scale networks.

We addressed this issue by offering a new graph-based image identification method that operates by matching the keypoints extracted from nano-scaled dendritic patterns. Due to the use of keypoint extraction, skeletonizing, and small pattern filtering methods, the algorithm is robust to image scratches. Also, the probabilistic nature of the graph matching algorithm with the proposed consistency check in an iterative fashion provides robustness against noise, skew, and image distortion. Overall, in typical conditions, the identification accuracy remains in the excellent range of 95% to 100%. Although we applied the proposed identification algorithm to our custom-built dendrite samples, it is a general method that can be used for image authentication problems when the image features can be well represented by numerical graphs.

REFERENCES

- [1] (Sep. 2014). *Humans Process Visual Data Better*. [Online]. Available: <http://www.t-sciences.com/news/humans-process-visual-data-better>
- [2] H. Koshimizu, M. Tominaga, T. Fujiwara, and K. Murakami, "On KANSEI facial image processing for computerized facial caricaturing system PICASSO," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 1999, pp. 294–299.
- [3] A. Kumar and Y. Zhou, "Human identification using finger images," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2228–2244, Apr. 2012.
- [4] K. W. Bowyer and M. J. Burge, *Handbook Iris Recognition*. New York, NY, USA: Springer, 2016.
- [5] B. Moreno, A. Sanchez, and J. F. Velez, "On the use of outer ear images for personal identification in security applications," in *Proc. IEEE 33rd Annu. Int. Carnahan Conf. Secur. Technol.*, Oct. 1999, pp. 469–476.
- [6] A. Kale, N. Cuntoor, B. Yegnanarayana, A. Rajagopalan, and R. Chellappa, "Gait analysis for human identification," in *Proc. Int. Conf. Audio Video Biometric Person Authentication*. Hilton Rye Town, NY, USA: Springer, Jul. 2003, pp. 706–714. [Online]. Available: <https://link.springer.com/book/10.1007/978-3-540-31638-1>
- [7] (Jul. 2019). *Guide to Barcodes Vs. QR Codes: In-Depth Comparison and Analysis of Both Label Types*. [Online]. Available: <https://www.mpofcinci.com/blog/barcode-vs-qr-code/>
- [8] (2018). *QR Code Statistics 2018: Up-to-Date Numbers on Global QR Code Usage*. [Online]. Available: <https://scanova.io/blog/qr-code-statistics/>
- [9] T. D. Pawlik and M. T. Olm, "Decoder for barcodes with anti-copy feature," U.S Patent 8 893 974, Nov. 25, 2014.

- [10] T. D. Pawlik and M. T. Olm, "System for detecting reproduction of barcodes," U.S. Patent 13 690 180, Jun. 5, 2014.
- [11] K. Garska. (2018). *Two-Factor Authentication (2fa) Explained: 2D Barcode Authentication*. [Online]. Available: <https://blog.identityautomation.com/two-factor-authentication-2fa-explained-2d-barcode-authentication>
- [12] R. Metzger. (Aug. 2018). *Federal Supply-Chain Threats Quietly Growing*. [Online]. Available: <https://www.federaltimes.com/opinions/2018/08/13/federal-supply-chain-threats-quietly-growing/>
- [13] Y. Zhou and D. Feng, "Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing," *IACR Cryptol. ePrint Arch.*, vol. 2005, p. 388, Dec. 2005.
- [14] C. Helfmeier, C. Boit, and U. Kerst, "On charge sensors for FIB attack detection," in *Proc. IEEE Int. Symp. Hardware-Oriented Secur. Trust*, Jun. 2012, pp. 128–133.
- [15] S. O. Skorobogatov, "Semi-invasive attacks: A new approach to hardware security analysis," Univ. Cambridge Comput. Library, Cambridge, U.K., Tech. Rep., Apr. 2005. [Online]. Available: <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-630.pdf>
- [16] M. Kozicki, "Dendritic structures and tags," U.S. Patents 9773 141, Dec. 5, 2017.
- [17] M. K. Y. Y. C. W. Zhi Zhao and N. Chamele, "Photochemical synthesis of dendritic silver nano-particles for anti-counterfeiting," *J. Mater. Chem.*, vol. 20, no. 7, pp. 6099–6104, 2019.
- [18] M. Gao, K. Lai, and G. Qu, "A highly flexible ring oscillator PUF," in *Proc. The 51st Annu. Design Autom. Conf. Design Autom. Conf.*, 2014, pp. 1–6.
- [19] M. Majzoubi, F. Koushanfar, and S. Devadas, "FPGA PUF using programmable delay lines," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Dec. 2010, pp. 1–6.
- [20] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh, "Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs," in *Proc. Int. Conf. Reconfigurable Comput. FPGAs*, Dec. 2010, pp. 298–303.
- [21] A. Iyengar, K. Ramclam, and S. Ghosh, "DWM-PUF: A low-overhead, memory-based security primitive," in *Proc. IEEE Int. Symp. Hardware-Oriented Secur. Trust (HOST)*, May 2014, pp. 154–159.
- [22] C. Helfmeier, C. Boit, D. Nedospasov, and J.-P. Seifert, "Cloning physically unclonable functions," in *Proc. IEEE Int. Symp. Hardware-Oriented Secur. Trust (HOST)*, Jun. 2013, pp. 1–6.
- [23] D. Merli, D. Schuster, F. Stumpf, and G. Sigl, "Side-channel analysis of PUFs and fuzzy extractors," in *Proc. Int. Conf. Trust Trustworthy Comput.* Pittsburgh, PA, USA: Springer, 2011, pp. 33–47. [Online]. Available: <https://link.springer.com/conference/trust>
- [24] X. Xu and W. Burleson, "Hybrid side-channel/machine-learning attacks on PUFs: A new threat?" in *Proc. Des., Autom. Test Eur. Conf. Exhib. (DATE)*, 2014, p. 349.
- [25] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, Sep. 2002.
- [26] N. Noor, V. Manthina, S. Muneer, A. Agrios, A. Gokirmak, and H. Silva, "Zno nanoforest optical pufs," in *Proc. Mater. Res. Soc. (MRS)*, 2018, pp. 25–30.
- [27] R. Jafri and H. R. Arabnia, "A survey of face recognition techniques," *J. Inf. Process. Syst.*, vol. 5, no. 2, pp. 41–68, 2009.
- [28] Y. Xu, Z. Li, J. Yang, and D. Zhang, "A survey of dictionary learning algorithms for face recognition," *IEEE Access*, vol. 5, pp. 8502–8514, 2017.
- [29] B. Moghaddam, C. Nastar, and A. Pentland, "A Bayesian similarity measure for direct image matching," in *Proc. 13th Int. Conf. Pattern Recognit.*, 1996, pp. 350–358.
- [30] U. Bakshi and R. Singhal, "A survey on face detection methods and feature extraction techniques of face recognition," *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 3, no. 3, pp. 233–237, 2014.
- [31] A. Shrivastava and D. K. Srivastava, "Fingerprint identification using feature extraction: A survey," in *Proc. Int. Conf. Contemp. Comput. Informat. (IC3I)*, Nov. 2014, pp. 522–525.
- [32] M. Oktiana, F. Arnia, Y. Away, and K. Munadi, "Features for cross spectral image matching: A survey," *Bull. Elect. Eng. Inform.* vol. 7, no. 4, pp. 552–560, Dec. 2018.
- [33] M. Guerrero, "A comparative study of three image matcing algorithms: Sift, surf, and fast," Utah State Univ., Logan, UT, USA, Tech. Rep., 2011. [Online]. Available: <https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=2029&context=etd>
- [34] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [35] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, Sep. 2004.
- [36] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, p. 5244.
- [37] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2548–2555.
- [38] J. Shi and Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1994, pp. 593–600.
- [39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
- [42] M. Wang and W. Deng, "Deep face recognition: A survey," 2018, *arXiv:1804.06655*. [Online]. Available: <http://arxiv.org/abs/1804.06655>
- [43] M. A. Abdulrahim and M. Misra, "A graph isomorphism algorithm for object recognition," *Pattern Anal. Appl.*, vol. 1, no. 3, pp. 189–201, Sep. 1998.
- [44] M. A. Lozano and F. Escolano, "Protein classification by matching and clustering surface graphs," *Pattern Recognit.*, vol. 39, no. 4, pp. 539–551, Apr. 2006.
- [45] L. Wiskott, J.-M. Fellous, N. Kruger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," in *Proc. Int. Conf. Image Process.*, 1999, pp. 456–463.
- [46] D. K. Izenor and S. G. Zaky, "Fingerprint identification using graph matching," *Pattern Recognit.*, vol. 19, no. 2, pp. 113–122, Jan. 1986.
- [47] J. Yan, X.-C. Yin, W. Lin, C. Deng, H. Zha, and X. Yang, "A short survey of recent advances in graph matching," in *Proc. ACM Int. Conf. Multimedia Retr.*, 2016, pp. 167–174.
- [48] M. R. Garey and D. S. Johnson, *Computer Intractability*, vol. 29. New York, NY, USA: WH Freeman, 2002.
- [49] F. Emmert-Streib, M. Dehmer, and Y. Shi, "Fifty years of graph matching, network alignment and network comparison," *Inf. Sci.*, vols. 346–347, pp. 180–197, Jun. 2016.
- [50] S. Gold and A. Rangarajan, "Graph matching by graduated assignment," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1996, pp. 239–244.
- [51] B. Gallager, "Matching structure and semantics: A survey on graph-based pattern matching," in *Proc. AAAI FS*, vol. 6, 2006, pp. 45–53.
- [52] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. Syst., Man, Cybern.*, vols. SMC–6, no. 6, pp. 420–433, Jun. 1976.
- [53] L. S. Davis, "Shape matching using relaxation techniques," *IEEE Trans. Pattern Anal. Mach. Intell.*, vols. PAMI–1, no. 1, pp. 60–72, Jan. 1979.
- [54] R. A. Hummel and S. W. Zucker, "On the foundations of relaxation labeling processes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vols. PAMI–5, no. 3, pp. 267–287, May 1983.
- [55] D. Eppstein, "Subgraph isomorphism in planar graphs and related problems," in *Proc. 6th Annu. ACM-SIAM Symp. Discrete Algorithms*, Philadelphia, PA, USA, 1995, pp. 632–640. [Online]. Available: <http://dl.acm.org/citation.cfm?id=313651.313830>
- [56] Netmanias. (Jul. 2013). *LTE Security I: Concept and Authentication*. [Online]. Available: <https://www.netmanias.com/en/?m=view&id=techdocs&no=10425>
- [57] P. Sahu. (Mar. 2011). *LTE Security Architecture*. [Online]. Available: <http://www.3gpteinfo.com/lte-security-architecture/>
- [58] G. P and V. Rajini, "YIQ color space based satellite image segmentation using modified FCM clustering and histogram equalization," in *Proc. Int. Conf. Adv. Electr. Eng. (ICAEE)*, Jan. 2014, pp. 1–5.
- [59] H. Hwang and R. A. Haddad, "Adaptive median filters: New algorithms and results," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 499–502, Apr. 1995.

- [60] S. J. K. Pedersen, "Circular Hough transform," *Aalborg Univ. Vis., Graph., Interact. Syst.*, vol. 123, no. 6, pp. 1–6, Nov. 2007. [Online]. Available: https://cdn.manesht.ir/9961___Simon_Pedersen_CircularHoughTransform.pdf and <https://www.semanticscholar.org/paper/Circular-Hough-Transform-Pedersen/27621c8db35f7ea1ae6ba495e2dac-214e1e84b74>
- [61] A. Valehi, A. Razi, B. Cambou, W. Yu, and M. Kozicki, "A graph matching algorithm for user authentication in data networks using image-based physical unclonable functions," in *Proc. Comput. Conf.*, Jul. 2017, pp. 863–870.
- [62] H. W. Kuhn, "The hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.



ZAOYI CHI received the B.S. degree in electrical engineering from the Chongqing University of Posts and Telecommunications, China, in 2018. He is also a Graduate Student in electrical engineering at Northern Arizona University. His research activities center around the fields of machine learning, computer vision, IoT security, and deep learning, supervised by advisor Dr. Abolfazl Razi.



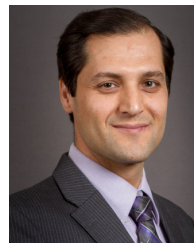
ALI VALEHI (Graduate Student Member, IEEE) received the B.Sc. degree from the Azad University of Tehran, Iran, in 2015, and the M.Sc. degree from the Northern Arizona University (NAU), in 2017, both in electrical engineering. His research interests include wireless communications, statistical analysis, machine learning, computer vision, and robotic perception and control.



HAN PENG received the B.Sc. degree in electrical engineering from Northern Arizona University (NAU), in 2016, and the B.Sc. degree in electronic and information engineering from the Chongqing University of Posts and Telecommunications (CQUPT). He is currently pursuing the Ph.D. degree with the School of Informatics, Computing and Cyber Systems, NAU. His current research interests include developing machine learning algorithms for IoT and UAV networks. In particular, he is working to develop algorithms to predict network topology changes over time and use it to enhance the performance of networking protocols.



MICHAEL KOZICKI (Member, IEEE) has been a Professor of electrical engineering with the Arizona State University, since 1985. He served several years in the semiconductor industry, and is the inventor of the technology behind the commercialized memory known as CBRAM. He has produced several hundred papers, presentations, and patents (56 granted in the US and several dozen internationally) that have been cited around 12,000 times to date. His patented inventions led to his election as a Fellow of the National Academy of Inventors, in 2015. His research interests include cyber and physical security using ionic devices, and he is currently working on US Air Force and National Science Foundation-sponsored projects in this area. He is a frequently invited speaker at conferences worldwide and has extensive international ties, including Visiting Professor at his alma mater, The University of Edinburgh, U.K., and Adjunct Professor at the Gwangju Institute of Science and Technology, South Korea. He also holds the professional designation of Chartered Engineer in the UK. He recently completed a two-year term as a Fulton Entrepreneurial Professor, is a founder of several start-up companies, and has served as the Chief Scientist of Silicon Valley Corporation Adesto Technologies in their pre-IPO days.



ABOLFAZL RAZI (Senior Member, IEEE) received the B.S. degree from Sharif University, in 1998, the M.S. degree from Tehran Polytechnic, in 2001, and the Ph.D. degree from the University of Maine, in 2013, all in electrical engineering. He held two postdoctoral positions in the field of machine learning and predictive modeling at Duke University, from 2013 to 2014, and Case Western Reserve University, from 2014 to 2015. He is currently an Assistant Professor with the School of Informatics, Computing and Cyber Systems, Northern Arizona University (NAU). His current research interests include smart connected communities, biomedical signal processing, wireless networking, Internet of Things, and predictive modeling. He is a recipient of several competitive awards, including the Best Research of MCI, in 2008, the Best Graduate Research Assistant of the Year Award from the College of Engineering, University of Maine, in 2011, and the Best Paper Award from the IEEE/CANEUS Fly By Wireless Workshop, in 2011.

...