Journal of Manufacturing Systems 65 (2022) 605–621

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Technical Paper

SplitCode: Voronoi-based error exaggeration for authentication of manufactured parts

Riddhi R. Adhikari [a], Karim A. ElSayed [b], Ergun Akleman [c,d], Jitesh H. Panchal [b], Vinayak Krishnamurthy [a,d,*]

[a] J. Mike Walker'66 Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, USA
[b] School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA
[c] Department of Visualization, Texas A&M University, College Station, TX 77843, USA
[d] Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA

ARTICLE INFO

Keywords:
Voronoi tessellation
Topology change
Authentication
Counterfeiting
Coordinate error distributions
Additive manufacturing

ABSTRACT

Counterfeiting in manufacturing is a crucial problem that has the potential to cause economic losses to both small and large businesses, including the aerospace, automotive, and medical industry. Existing techniques for preventing counterfeiting are based on external modification of the manufacturing process which incurs extra cost, and limits their use in everyday applications. In our work, we take advantage of the inherent characteristics of the machine configuration (machine and process parameters) to identify whether a given part is manufactured by a certain machine or class of machines. Each machine configuration has a unique coordinate error distribution which is indicative of its precision and bias. The overarching idea is to differentiate between error distributions of two machine configurations in order to determine whether the positional errors in a given part come from the same distribution as that of the machine configuration. To be able to differentiate between two machine configurations robustly, we propose a novel topological transformation technique based on the principle of Voronoi tessellation that exaggerates the difference between their error distributions. We present a methodology for authentication of machined parts and validate it numerically and experimentally through the example of additive manufacturing. This research work also offers various opportunities of further exploration in terms of part design, algorithm of *SplitCode*, imaging and post processing methods and statistical variations.

1. Introduction

1.1. Broader context and motivation

Ensuring security in manufacturing systems is among the central challenges within the purview of cyber–physical production systems (CPPS). The seamless integration of software and hardware components and increased use of networking capabilities has also resulted in a wide variety of avenues for attacks and threats on CPPS. Much of the discussion regarding secure manufacturing focuses primarily on information-related breaches and the potential approaches, such as decentralized computing and block-chains, for addressing them [1]. However, we posit that security in CPPS additionally requires the integration of both computational and physical sub-systems of CPPS. In this paper, our aim is to present an embodiment of this integrated notion of secure manufacturing. We specifically focus on counterfeiting and introduce a methodology for quantitative assessment of part authenticity.

Counterfeiting is a significant problem that is not only prevalent in consumer goods, but also in high-end manufactured products. The problem affects all manufacturing sectors, including the automotive, aerospace, defense, and pharmaceutical industries. Counterfeiting in the automotive sector, for example, is a lucrative business, estimated to generate billions of dollars in sales in the US every year [2–4]. Counterfeit mechanical parts, such as engine mounts, seat-belts, brakes, fasteners, bearings, valves, and springs not only result in lost corporate revenue and lost taxes for the government, but also increase the likelihood of accidents and even fatalities [5–7]. Further, IBM has consistently shown that the manufacturing industry has been facing

* Corresponding author at: J. Mike Walker'66 Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, USA.
E-mail address: vinayak@tamu.edu (V. Krishnamurthy).

https://doi.org/10.1016/j.jmsy.2022.10.005
Received 23 June 2022; Received in revised form 25 August 2022; Accepted 7 October 2022
Available online 3 November 2022

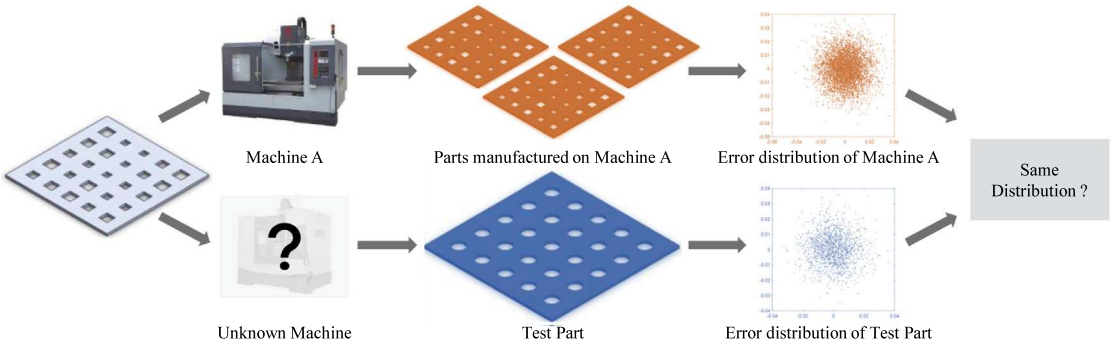


Fig. 1. To find if the test part is manufactured by the reference machine, we compare the machine's reference error distribution and the test part's error distribution to see if they belong to the same distribution.

high security attack rates over the years, with being the second most-attacked industry and experiencing the most data theft attacks (33% of all data theft attacks) in 2020 [8].

While there has been significant research in counterfeit prevention in electronic parts, research on counterfeit prevention of non-electronic hardware and materials is still in its infancy [9]. Existing approaches for counterfeit mitigation include standards to provide guidance for acquisition of authentic materials (e.g., SAE AS6174 [10]), attachment of RFID tags [11,12], chemical and DNA tagging [13,14], watermarking [15], and information embedding in additively manufactured (AM) parts [16–18]. These approaches for ensuring the authenticity of parts rely on modifying the manufacturing process.

An alternate approach is to take advantage of the inherent randomness of manufacturing processes, and unique characteristics of materials, and manufacturing equipment to assess whether or not a part is authentic. The idea of using statistical properties of manufactured components is widely used in physically unclonable functions (PUFs) [19,20] for electronic parts. Commonly used PUFs involve evaluating integrated circuit (IC) responses to a given challenge [20]; the responses are chosen such that they vary randomly between instances of the IC even under identical manufacturing conditions. Examples include measuring the spatially varying capacitance of a coating or the delay between different gates of an IC [21]. In this paper, we apply the underlying principle of leveraging randomness in manufacturing configuration (machines and processes) to non-electronic parts. To do so, we introduce a novel methodology to encode the distribution of process errors in part instances manufactured with different parameters on a single machine, and across multiple machines.

1.2. Problem

Broadly, the problem of part authentication may refer to several distinct cases such as verifying whether a part was created on a (1) *uniquely identifiable machine and process parameters* or (2) *uniquely identifiable "class" of machines and process parameters*. We consider counterfeiting scenarios where a given part is either manufactured with a different class of machines than intended, or using different process parameters than those prescribed, or a combination of the two. Therefore, given a manufactured part, the problem we seek to address is to determine whether this part was manufactured using a given manufacturing configuration (class of machine, process, and parameters).

Any combination of machine, process, and process parameters leads to coordinate errors in the tool-tip caused due to the intrinsic build of the machine, vibrations during the process, inherent coordinate biases etc. These errors manifest in the manufactured part in the form of geometric deviations with respect to the designed CAD model. Therefore, each part carries unique characteristics (similar to a fingerprint) associated with its corresponding manufacturing configuration.

We hypothesize that the geometric deviations across multiple instances of a given part manufactured using the same manufacturing configuration belong to some common underlying error distribution that is unique to the manufacturing configuration. Consider a "*reference*" error distribution for a given manufacturing configuration that is determined by evaluating geometric deviations for multiple instances of a given part with some geometric features. Given a "*test*" part with the same features, our technical problem (Fig. 1) translates to determining the statistical similarity between the reference and test distributions.

1.3. Challenge & approach

Consider a part that has been designed with a specific set of geometric features (e.g. holes of different sizes and shapes) to be manufactured using a specific combination of machine, process, and process parameters. One can measure and characterize the manufacturing error by computing the geometric deviations of such features with respect to the ideal (CAD) model. For instance, one could estimate center-to-center distances for a circular hole or corner-to-corner deviations for polygonal holes, etc. Measurements of these features (corners, centers, etc.) could be performed using a variety of methods (e.g. photography, photogrammetry, scanning, CMM etc.) depending on the type (shape, material, etc.) of the part. The key challenge here is that the coordinate error distributions of two similar but distinct manufacturing configurations can be indistinguishable making authentication prohibitively difficult. Furthermore, the process of measuring geometric deviations (from imaging or other means) adds additional errors to the distribution that may be inseparable from the manufacturing errors.

One possible way to view this problem is that of creating discriminatory features from two given distributions and it may be possible to use feature engineering techniques from machine learning. However, such techniques are limited because of the need for fine tuning learning parameters and inherent lack of explainability. We propose a robust and novel geometric methodology that exaggerates the differences between error distributions of two machines. Our methodology is based on the application of Voronoi tessellation [22] to transform coordinate errors of the tool-tip into a topologically-induced error distribution [23,24]. The idea behind our approach is that Voronoi tessellation of sites arranged in rectangular grids admits a 4-valency vertex configuration that is extremely sensitive to any noise in the location of the sites. Specifically, any perturbation to the Voronoi sites in a rectangular grid results in the splitting of the 4-valency vertex into an edge—hence the name *SplitCode*. This interesting property, when viewed from a statistical standpoint, leads to a unique method for exaggerating the coordinate error distributions. As such, this approach is general in that it can be applied to any type of manufacturing process with any machine and process parameters.

1.4. Contributions

The prime purpose of our work is to authenticate parts printed with different manufacturing configurations. For that, we want to be able to measure the deviations resulting from the mechanical structure of machines and process parameters, and to model the connection between deviations from different manufacturing configurations and the manufactured parts. To achieve this, we make the following contributions:

1. Our primary contribution in this work is to introduce a new concept that we call topologically transformed error distribution, which is defined as a statistical exaggeration of differences between two coordinate distributions.
2. Our second contribution is to apply this new concept to manufacturing problems by developing a novel algorithm, which we call *SplitCode*. By using the well-known principle of Voronoi tessellation, *SplitCode* successfully exaggerates the differences between the coordinate error distributions of different manufacturing configurations.
3. Our third contribution is a part authentication scheme based on *SplitCode* by identifying the geometric representations that lead to the maximum exaggeration between two coordinate error distributions.

To evaluate our methodology, we present a numerical validation of our scheme by investigating the effect of different known distributions on the authentication method. As such, the proposed method is universally applicable to authentication with any type of a manufacturing process. However, to demonstrate our approach in action, we specifically present an experimental case study with additive manufacturing. A comparative analysis of our topological transformed error distributions with respect to direct coordinate error demonstrates the efficacy of our approach.

2. Background and related work

Given that we investigate our approach through additive manufacturing (AM), we will review relevant research specific to AM as the domain of application in terms of the potential threats and authentication methods explored in literature.

2.1. Potential threats in additive manufacturing

Additive manufacturing (AM), commonly known as *3D Printing*, is the process of manufacturing 3D parts in a layer-by-layer fashion based on a computer-aided design (CAD) file [25,26]. It is a multi-step process combining both automated and manual workflows. The process begins with designing a computer-aided model of the part to be printed and storing design in a format compatible with 3D printers, such as STereoLithography (STL), Additive Manufacturing File (AMF) and 3D Manufacturing Format (3MF). The information stored is used to create a G-code with commands to control the position of the printing nozzle and the bed. The G-code is optimized based on different parameters, such as machine specification, material, support structure, layer height, and printing speed. After printing, the part is post processed to obtain a finished product. Yampolskiy et al. [25,27] discuss in their work how various elements in this process can be compromised and manipulated depending on their role in the 3D printing process.

For AM, major threats have been categorized into three main groups—theft of technical data, sabotage, and illegal part manufacturing [28]. Technical data includes part specification, printing and post-processing parameters, which could be comprised through theft or reverse engineering [29–32]. Few examples of sabotage targets include: the CAD model, G-code, printing material, printing equipment, manufactured part or the printing environment [25,33–35]. The third major threat, unauthorized part manufacturing, arises due to the large availability of 3D digital blueprint files online and increasing accessibility of high quality AM equipment [36,37].

These cyber–physical attacks adversely affect the AM industry and parts, resulting in slow adaptation of the technology, economical losses, and immature failure of additively manufactured parts [33], along with endangering human lives [27,36]. These attacks give rise to counterfeiting of 3D printed parts, which has disrupted key industries, including aerospace, automotive, and medical industries [38–40]. Researchers have extensively explored this problem and introduced several authentication methods, which are discussed in the following section.

2.2. Authentication methods in additive manufacturing

Many authentication methods utilize external appearance of parts by surface tagging [15,16,41,42] while others take advantage of the inner structure [43–46]. Depending on the where the information, or "tag", is located, different equipment can be used for capturing this information for authentication, such as cameras [16], scanners [15], micro-computed tomography (micro-CT) scanner [43], or infrared (IR) cameras [47].

The technique proposed by Harrison et al. [48] is an example of surface tagging. They present barcodes in the form of physical notches on the surface of the printed part which when swiped by fingernails produce a unique sound wave, which is captured using a microphone. Such methods affect the appearance of the 3D printed parts. Gupta et al. [49] propose a method where additional features are introduced in the CAD model because of which the quality of the part is maintained only when specific printing conditions are met and in all other cases produce lower quality parts. The disadvantage of this method is that poor quality of 3D printed parts cannot be measured directly but is only detected on premature failure of the part. Chen et al. [43] discuss embedding of QR codes in multiple layers to eliminate the effect of codes on mechanical properties of the 3D printed part. This method requires usage of costly micro-CT equipment for authentication, increased effort in designing, and high infill percentage of parts. Recently, Brandman et al. [50] propose the notion of physical hash for detecting if a part being 3D printed is according to design specifications in real-time. Their idea is to print a QR code derived as a hash string from the nominal process parameters and tool-path and print it alongside the original geometry for in situ measurement.

To address the issue of authentication in low infill parts, Kubo et al. [44] present a method of using resonant properties of objects. 3D printed parts are assigned unique resonant properties by changing their internal structure and can be differentiated even if they have similar appearance. The change in the internal structure is captured with acoustic sensing. Kubo et al. extend their work by varying the infill patterns during slicing to create unique resonant properties, which reduces the effort of 3D modeling [45]. Sandborn et al. [51] propose a method for detecting counterfeiting by measuring impedance identity of a part using piezoelectric sensors.

Li et al. make an important argument [52] that 3D printers possess unique fingerprints resulting from its hardware imperfections. In their work, they model a connection between the fingerprints and the texture on the 3D printed parts. Dogan et al. [42] also utilize the patterns appearing on the 3D printed objects due to slicing instead of adding extra features to the part. Their method differentiates between patterns that inherently arise from varying the slicing parameters. Similarly, Delmotte et al. [15] and ElSayed et al. [41] locally vary printing parameters (layer height and printing speed, respectively) to introduce subtle changes on parts' surfaces for authentication. Taking inspiration from these works, we propose a method called *SplitCode* to address some of the existing challenges.

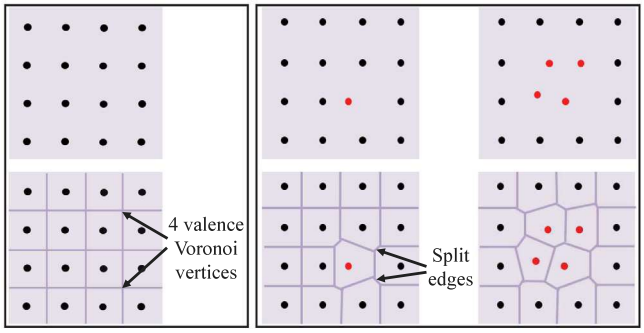


Fig. 2. Change in the Voronoi site location leads to change in topology as four valence vertices split into Voronoi edges.

2.3. Our work

The premise of our proposed approach is that authentication of parts is intimately linked with the part quality which embeds the natural randomness of the manufacturing process itself. However, it is the quantification of part quality that poses a technical challenge. To our knowledge, there is no benchmark or competing methodology other than semi-manual quality checks of manufactured parts. On the other hand, digital methods proposed in previous works either simply apply standards, or involve new hardware additions (such as RFID), or require one to modify a given manufacturing process (i.e. may not scale to different types of manufacturing processes). In contrast, our work presents a systematic methodology for tackling part counterfeiting. Because the methodology is purely based on applying statistical transformation to geometric information to assess quality of token parts, it is independent of the manufacturing process.

3. Conceptual preliminaries

Our authentication methodology entails the statistical comparison of geometric deviations of a test part with a reference distribution that characterizes a given manufacturing configuration. Therefore, our authentication approach relies on the ability to better distinguish distributions coming from two different manufacturing configurations (i.e. two different combinations of machine, process, and parameters). To achieve this, we utilize the concept of Voronoi tessellation to transform the original coordinate distributions such that the difference between two distributions coming from two different configurations is exaggerated thereby enabling robust authentication. Here, we discuss the key concepts underlying our methodology.

3.1. Voronoi tessellation

A Voronoi tessellation of some given spatial domain is a way to partition the domain with seed geometric entities (called Voronoi "*sites*") into mutually exclusive and exhaustive regions (called Voronoi cells) such that each region contains exactly one generating site and every point in a given region is closer to its generating site than to any other. As such, the sites can take any form (points, lines, curves, simplicial complexes, etc.). However, if the Voronoi sites are points in Euclidean space, the Voronoi cells are always convex. We specifically consider 2D Euclidean domains wherein each Voronoi cell is a convex polygon with Voronoi edges (the edges of the polygon) and Voronoi vertices (the corners of the polygon). In particular, we are interested in the topology of the Voronoi tessellation as captured by the number of edges incident on (i.e. the valency of) a given Voronoi vertex in the tessellation.

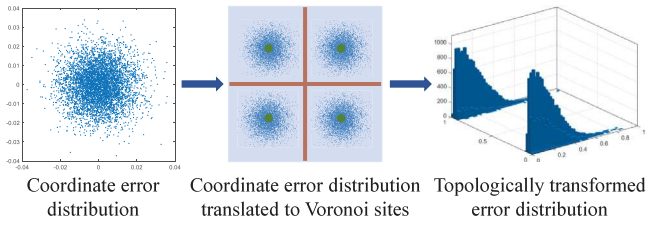


Fig. 3. Topological transformation of two coordinate error distributions results in exaggeration between them.

3.2. Topological transitions in Voronoi cells

While it is obvious that the locations of the sites dictate the cell geometry, the interesting aspect of the tessellations that is relevant to us is that the site locations also affect the cell topology through edge collapses and vertex splits. Consider a special case wherein the sites are arranged in an infinite rectangular grid on the plane. For this site configuration, the tessellation results in an infinite rectangular grid (Fig. 2a)—each cell is a rectangle and each vertex has valency of four.[1] Now note that any perturbation in even one of the sites results in at least one vertex to split into an edge. For the rectangular site arrangement, this topological shift is guaranteed to occur (Fig. 2b). It is this sensitivity to the sites that we leverage to exaggerate the error distribution.

3.3. Topologically transformed error distribution

Based on the principle of topological sensitivity, we can design a transformation that enables us to distinguish two seemingly indistinguishable error distributions. To perform such a transformation, consider a 2D Euclidean domain comprised of four Voronoi sites located at the corners of a square (without loss of generality). As noted earlier, the Voronoi tessellation for this arrangement is a single 4-valency vertex at the center with four edges extending to point at infinity. Now, perturbing each of the four sites using a given coordinate error distribution results in a new distribution of the split edge at the center (Fig. 3). Based on this principle, any coordinate error distribution is transformed into an equivalent statistical distribution of the split edge resulting from the topological change. It is this *topological transformation* that leads to the exaggeration of the difference between two coordinate error distributions. We call this new distribution the *topologically transformed error distribution*.

4. *SplitCode*: Algorithm

The process of computing topologically transformed error distribution from the original reference and test error distributions comprises of the following steps:

Step 1. We start with four Voronoi sites arranged along the vertices of a square inside a 2D Euclidean square domain (Fig. 4a). These four points represent the ideal locations of the geometric features in a manufactured part.

Step 2. We compute the Voronoi tessellation for this arrangement which results into an exhaustive grid of quad cells with a 4-valency vertex at the center. We refer to this vertex as the center point.

[1] This is not the only arrangement that results in a 4-valency vertex. There is at least one more arrangement that results in a trapezoidal tessellation with all 4-valency vertices. However, an exhaustive listing of such arrangements is out of scope of this paper.

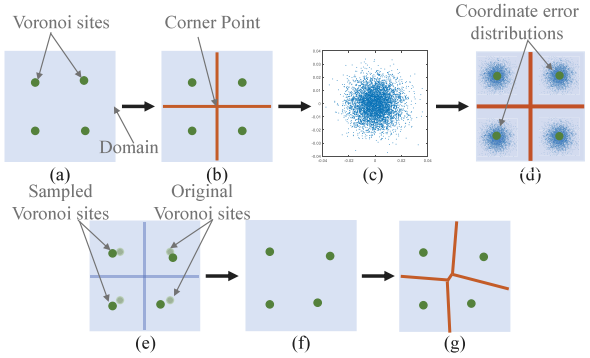


Fig. 4. *SplitCode*: Algorithm.

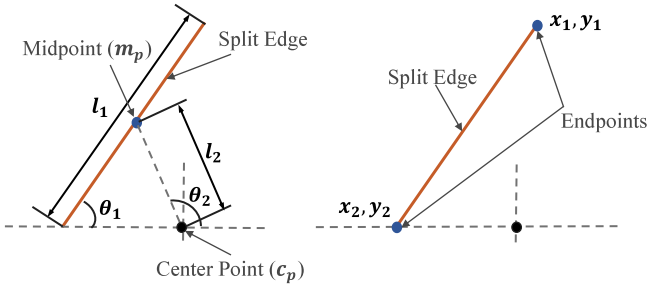


Fig. 5. Split Edge can be represented either by its length, angle, and location of the midpoint or by the location of its two endpoints.

Step 3. We then consider a coordinate error distribution (Fig. 4c) that characterizes (gives us an estimate of precision and bias for) a particular manufacturing configuration.

Remark: In a real-world scenario, the coordinate error distribution would typically by obtained from manufacturing sample parts and computing feature errors with respect to the ideal (image of the CAD part). Here, a feature could be a geometric entity such as a point, edge, corner, etc and the deviation of the measured locations of these features from their ideal locations is the distribution (example: Fig. 4c).

Step 4. Given a coordinate error distribution (Fig. 4c), we simply copy it to each of the four Voronoi sites. Here, by *copying*, we mean that the error distribution, whose mean is the origin, is translated such that the mean is a given Voronoi site. Thus, we get four translated distributions with same standard deviation but different mean locations, all referring to the same manufacturing configuration (Fig. 4d).

Remark: Therefore, the image in Fig. 4d is essentially a set of four copies of the image in Fig. 4c with the means centered at the four Voronoi sites.

Step 5. We then sample one point each from all the four translated distributions giving us four new Voronoi sites that could be imagined to be the actual location of features in a manufactured part and hence, shifted from the original Voronoi sites (Fig. 4e,f).

Step 6. In the final step, we again compute the Voronoi diagram for the newly sampled Voronoi sites. As the new sites are randomly located in the domain, the domain gets decomposed into four unequal Voronoi cells as oppose to a grid of quad cells (Fig. 4g). Consequently, the four valence center point in the grid splits into a line with three valence endpoints when the Voronoi sites are displaced. This line is known as the Split Edge.

Through this process we transform the error resulting from a manufacturing configuration into a split edge. It is important to note here that even small changes in the location of the Voronoi sites (small manufacturing error) can be very easily detected and accurately captured in the resulting Voronoi tessellation. This is possible because of the topology change occurring as the center point, a four valence vertex splits into the split edge. Hence, we utilize the split edge as our leading point for further work.

4.1. Split edge representation

Note that a topologically transformed error distribution essentially represents the probability of a set of line segments (the split edges) in 2D-space. Given that a line segment can be defined in more than one way (e.g. two end-points, length–angle–midpoint, etc.), there is no unique way to compute the distribution. We consider two different ways to represent the split edge (Fig. 5) and thereby the error distribution. The first representation is in terms of the length of the edge (l_1), its angle (θ_1) with respect to the horizontal axis, and the polar coordinates ($l_2,\ \theta_2$) of the midpoint (m_p). The second representation is simply the Cartesian coordinates of the two end-points of the edge ((x_1, y_1) and (x_2, y_2)). We experiment with these two combinations of parameters to select the representation that best differentiates between reference distributions coming from two different manufacturing configurations.

In both split edge representations, we require four parameters to fully define the split edge i.e. either $(l_1, \theta_1, l_2, \theta_2)$ or (x_1, y_1, x_2, y_2). In other words, the topologically transformed error distribution is a 4-variate distribution. However, we note that there is a natural separation of the parameters into two bi-variate distributions. In the first case, we get one bi-variate distribution from the length and angle of the edge (l_1, θ_1) and another from the mid-point coordinates (l_2, θ_2). In other words, one distribution depicts the probability of occurrence of a split edge with a specific length and angle and the second probability distribution represents the probability of occurrence of a split edge with its midpoint at a specific location. Similarly, for the second representation, the location of each of the end points provides a bi-variate representation. Here, the first probability distribution gives the probability of occurrence of split edge with its first endpoint at a specific location and the second distribution exhibits probability of occurrence of split edge with its second endpoint at a specific location.

Regardless of how we represent the split edge, the two bi-variate probability distributions resulting from the split edges can be mapped to the original coordinate error distribution and are called *topologically transformed error distributions* (Fig. 6). It is entirely possible that a given split edge representation may lead to better exaggeration while another may lead to indistinguishable results. Therefore, a necessary step in the design of our authentication scheme is to systematically investigate different edge parametrizations and their corresponding ability to differentiate between similar coordinate error distributions.

5. Design of authentication scheme

Our authentication scheme is predicated upon the ability to distinguish two distributions coming from two different manufacturing configurations. In simpler terms, we wish our scheme to be able to tell apart two distinct ***combinations*** of machines, processes, and process parameters. As a concrete example, suppose we produce multiple instances $\{i_i, i_2, \ldots, i_n\}$ using manufacturing configuration I and $\{j_i, j_2, \ldots, j_m\}$ produced using manufacturing configuration J. Note that each of these part instances result in an error distribution, namely $P_I = \{(P^1(i_1), P^2(i_1)), \ldots, (P^1(i_n), P^2(i_n))\}$ and $P_J = \{(P^1(i_1), P^2(i_1)), \ldots, (P^1(i_n), P^2(i_n))\}$. Here, $(P^1(.), P^2(.))$ signify the two bi-variate distributions we get from the split edge.

Our goal is to determine the representation of the split edge such that the sets P_I and P_J admit two separable clusters. In other words, instance distributions coming from configuration I should be clustered together and should be separable from the cluster generated from the instance distributions of configuration J. To select the right split edge representation, we conducted a simulated experiment as detailed in the following sections.

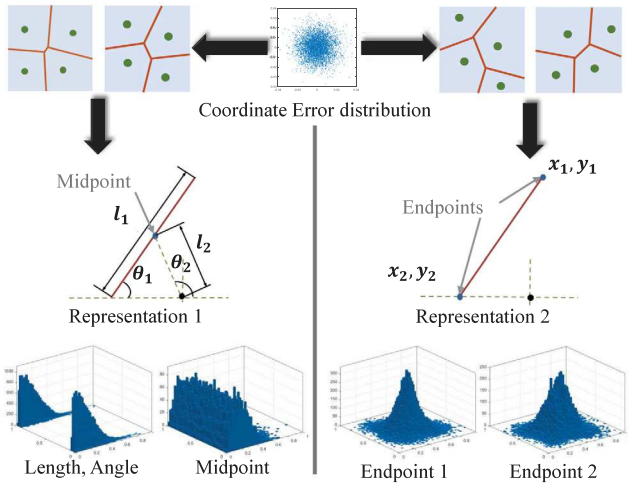


Fig. 6. When multiple sets of 4 Voronoi sites are sampled from an error distribution, multiple split edges are generated. Length, angle, midpoint, and endpoints of these split edges are represented through bi-variate probability distributions called topologically transformed error distributions.

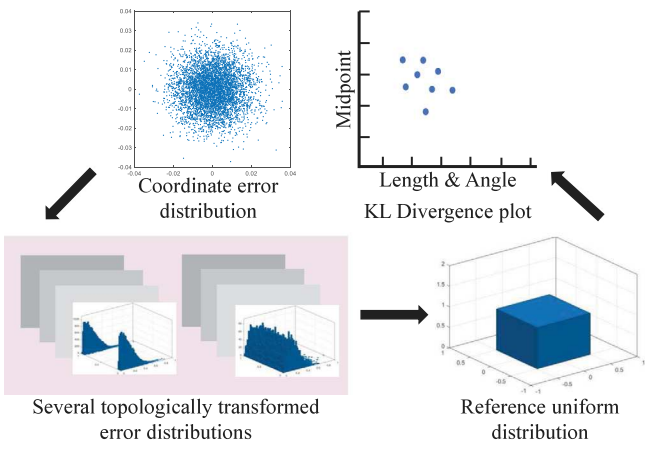


Fig. 7. Difference between topologically transformed error distributions and reference uniform distribution is shown in the distribution space. In this example, each point is the ordered pair $\left(D_{KL}(P^1 \parallel Q_1), D_{KL}(P^2 \parallel Q_2)\right)$ where P^1 is the bi-variate distribution of the length and angle of the split edge and P^2 is the bi-variate distribution of the coordinates of the mid-point of the edge.

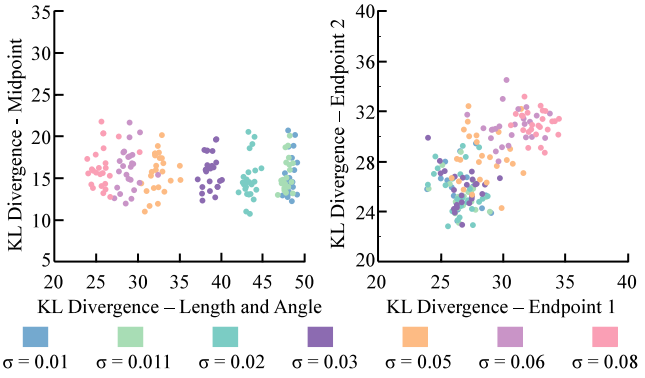


Fig. 8. D_{KL} Plots generated from length, angle, midpoint and two endpoints of split edges for error distributions with same mean and varying standard deviation.

5.1. Mapping part instances to the distribution space

In order to select the right split edge representation, we need a method that maps a set of instance distributions to a common space – *a distribution space* – so as to observe whether instances produced by a given manufacturing configuration cluster together and whether different configurations lead to separable clusters. Given a pair $(P^1(i), P^2(i))$ of the two bi-variate distributions from some part instance i, we can map this to an ordered pair $\left(D_{KL}(P^1(i) \parallel Q_1), D_{KL}(P^2(i) \parallel Q_2)\right)$. Here, $D_{KL}(A\|B)$ is the Kullback–Leibler divergence (also known as the statistical distance or relative entropy) of a probability distribution B from A. In our case, we choose Q_1 and Q_2 as two uniform distributions that span the maximum intervals for the corresponding variables as follows:

1. **For split edge representation 1:**
   The variables for Q_1 are $0 \le l_1 \le \sqrt{2}$ (length of diagonal of the unit square) and $0 \le \theta_1 \le 2\pi$. Therefore the height $h(Q_1) = \frac{1}{4\pi\sqrt{2}}$.
   The variables for Q_2 are $0 \le l_2 \le \frac{1}{\sqrt{2}}$ (half-length of diagonal of the unit square) and $0 \le \theta_2 \le 2\pi$. Therefore the height $h(Q_2) = \frac{1}{2\pi\sqrt{2}}$.
2. **For split edge representation 2:**
   The variables for Q_1 and Q_2 are $-0.5 \le x_1, y_1, x_2, y_2, \le 0.5$ (the min and max coordinates in the unit square). Therefore the heights $h(Q_1) = h(Q_2) = \frac{1}{2}$.

For a given manufacturing configuration I with distributions $P_I = \{(P^1(i_1), P^2(i_1)), \ldots, (P^1(i_n), P^2(i_n))\}$, this results in a set of points where each pair $\{(P^1(i_k), P^2(i_k))\}$ maps to $(D_{KL}(P^1(i_k) \parallel Q_1), D_{KL}(P^2(i_k) \parallel Q_2))$ in the distribution space for $1 \le k \le n$ (Fig. 7). In the first representation method, we use D_{KL} to map the probability distributions generated from length-angle of the split edges and the midpoint to the distribution space. In the second representation, we map the two probability distributions generated from the two endpoints of split edges. Therefore, we can use this process to compute two different mappings of the same set of part instances. Our goal in subsequent sections is to design a series of experiments to compare which representation enables better differentiation between part instances produced by two different manufacturing configurations.

5.2. Experiment design for split edge representation

If we know the function of error distribution generated from a given manufacturing configuration I, we can create multiple instances $\{i_i, i_2, \ldots, i_n\}$ of the same manufacturing configuration by randomly sampling points from the original error distribution. Error distribution of each instance leads to two bi-variate distributions $(P^1(i), P^2(i))$ which can be further mapped to a distribution space in terms of an ordered pair $\left(D_{KL}(P^1(i) \parallel Q_1), D_{KL}(P^2(i) \parallel Q_2)\right)$. To find the representation of split edge that consequently gives rise to separable clusters of ordered pairs $\left(D_{KL}(P^1 \parallel Q_1), D_{KL}(P^2 \parallel Q_2)\right)$ for each manufacturing configuration, we simulate error distributions resulting from different manufacturing configurations. Application of *SplitCode* on these simulated error distributions generates two different mappings as a result of two representations of the split edge. These mappings can be compared to find the representation of the split edge that simplifies differentiation between error distributions of different manufacturing configurations. For these simulations, we assume coordinate error distributions to be bi-variate normal distributions with a specific mean and standard deviation value. We perform comparison between the two representations of the split edge for the following two groups of coordinate error distributions:

1. Bi-variate normal distributions with same mean and varying standard deviation - In the first group, we generate seven

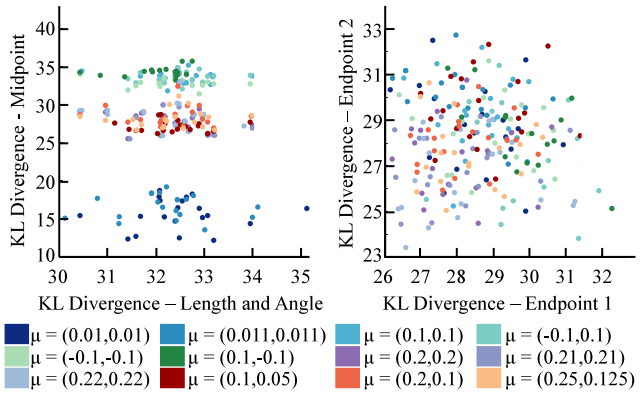


Fig. 9. D_{KL} Plots generated from length, angle, midpoint and two endpoints for error distributions with varying mean and same standard deviation.

bi-variate normal distributions with their mean at the origin and standard deviation as 0.01, 0.011, 0.02, 0.03, 0.05, 0.06 and 0.08. The co-variance for these distributions is zero. All seven distributions consist of 5000 points. This group of coordinate error distributions imitates the scenario where different manufacturing configurations produce part instances with different precision.

2. Bi-variate normal distributions with same standard deviation and varying mean - Second group comprises of bi-variate normal distributions with a standard deviation of 0.05 and varying means. The co-variance value of these distributions is zero and they too consist of 5000 points each. This group talks more about the difference in bias resulting from different manufacturing configurations.

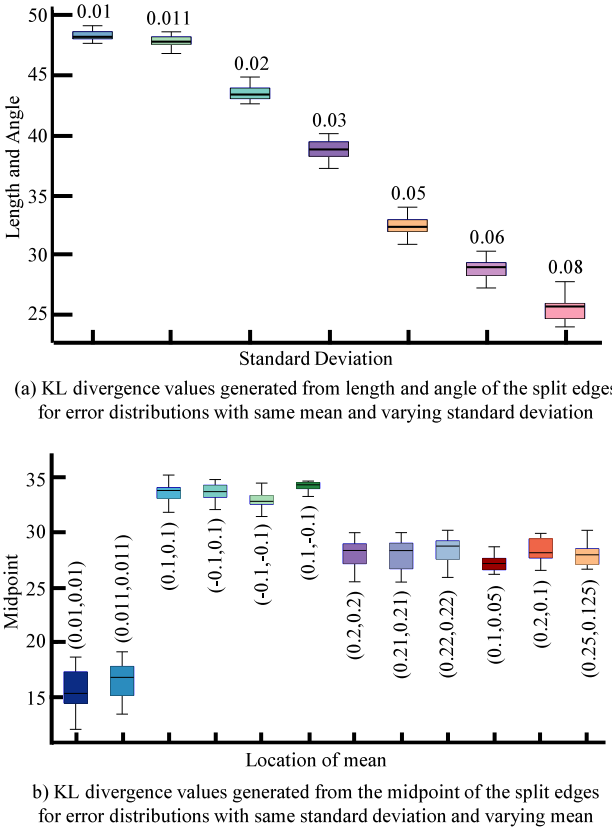
The location of mean for these distributions are selected in three categories. In category 1, we first select few locations in the first quadrant along the line defined by equation $x = y$, at an increasing distance from the origin such as $(0.01, 0.01)$, $(0.011, 0.011)$, $(0.1, 0.1)$, $(0.2, 0.2)$, $(0.21, 0.21)$ and $(0.22, 0.22)$. The goal here is to understand how the nature of bi-variate distributions change with increasing distance of mean location from origin. In the second category, we select mean locations in the first quadrant along a line with slope of 0.5 such as $(0.1, 0.05)$, $(0.2, 0.1)$ and $(0.25, 0.125)$. This is to investigate the variation in bi-variate distributions with the change in the slope of the line along which mean location is chosen. Finally, we select some locations in all four quadrants equidistant from the center and the two axes in the third category. In this category, the means of error distribution are located at $(0.1, 0.1)$, $(-0.1, 0.1)$, $(-0.1, -0.1)$ and $(0.1, -0.1)$. This would help us to understand how bi-variate distributions change with the quadrants in which mean is located.

We generate 20 instances for each manufacturing configuration which gives rise to 20 pairs of bi-variate distributions and thus, a set of 20 points in each mapping in the distribution space. Detailed comparison between mappings obtained from two representations of the split edge for the aforementioned groups is given in the following section.

5.3. Selection of the split edge representation

5.3.1. Same mean and varying standard deviation

We observe that for this group of error distributions, when bi-variate distributions generated from the length, angle, and midpoint of split edge are mapped to the distribution space, we get clusters of points belonging to instances generated with the same manufacturing configuration (Fig. 8). In addition to that, clusters belonging to different manufacturing configurations are clearly separable. On the other hand,



(a) KL divergence values generated from length and angle of the split edges for error distributions with same mean and varying standard deviation

b) KL divergence values generated from the midpoint of the split edges for error distributions with same standard deviation and varying mean

Fig. 10. D_{KL} generated for the length and angle, and the midpoint of the split edges are shown for error distributions with varying standard deviation and varying mean respectively.

when endpoints are used to represent split edges, points belonging to one manufacturing configuration forms a close cluster. However, these clusters cannot be clearly separated for different manufacturing configurations as they overlap each other (Fig. 8). Thus, part instances produced with different manufacturing configurations resulting in error distributions with same mean but varying standard deviation can be better distinguished by using length, angle, and midpoint representation of the split edge.

We note that for distributions with same mean, as the standard deviation of the error distribution increases, location of the sampled Voronoi sites move farther from the center point. As a result, we see a clear variation in the length of the split edge with increasing standard deviation but not in its location. Consequently, we observe a monotonically decreasing behavior in the $\left(D_{KL}(P^1 \parallel Q_1)\right)$ value obtained from the length and angle bi-variate distribution as the deviation in the error distribution increases (Fig. 10a). The median $\left(D_{KL}(P^1 \parallel Q_1)\right)$ value for error distribution with lowest deviation of 0.01 is 48.1490 and the $\left(D_{KL}(P^1 \parallel Q_1)\right)$ for the highest deviation distribution is 25.8009. In between these two distributions, the $\left(D_{KL}(P^1 \parallel Q_1)\right)$ value decreases to 47.7912, 43.3260, 38.6587, 32.4051, and 29.0111 for coordinate error distributions with a standard deviation of 0.011, 0.02, 0.03, 0.05, and 0.06 respectively. However, we do not observe much variation in the $\left(D_{KL}(P^2 \parallel Q_2)\right)$ values generated from midpoint bi-variate distributions across all coordinate error distributions (Fig. 8).

5.3.2. Same standard deviation and varying mean

Mappings obtained for distributions in this group tell us that, when split edge representation 1 (length, angle, midpoint) is used, bi-variate

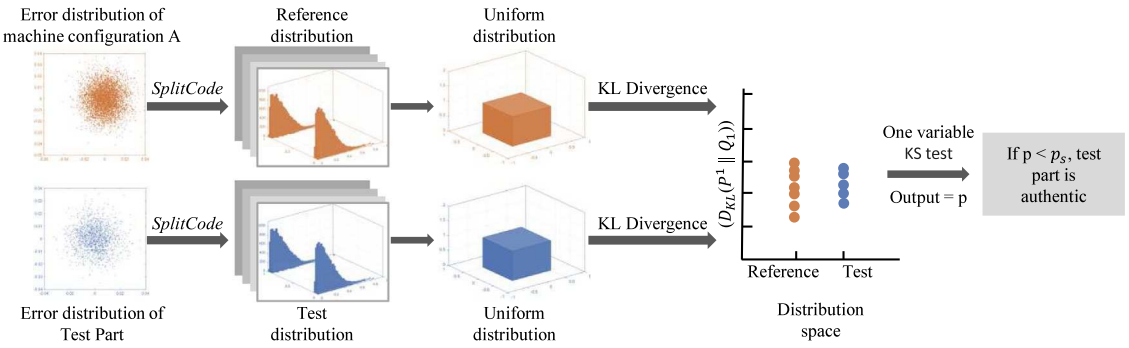


Fig. 11. Authentication procedure: *SplitCode* is applied on error distributions of a manufacturing configuration and the test part to obtain topologically transformed error distributions (reference and test distribution). Reference and test distribution are compared with a uniform distribution in terms of KL divergence and are mapped to a distribution space as $(D_{KL}(P^1 \parallel Q_1))$. Later $(D_{KL}(P^1 \parallel Q_1))$ values obtained from reference and test distributions are tested against each other through a one variable KS test to find if the test part is printed with the given manufacturing configuration.

distributions of instances coming from one manufacturing configuration map to points in a close separable cluster in the distribution space (Fig. 9). However, bi-variate distributions resulting from the second representation of the split edge are randomly mapped to points in the distribution space. We do not get separable cluster for each manufacturing configuration. Instances produced with all manufacturing configurations contribute to one common cluster in the distribution space making differentiation between different manufacturing configurations difficult (Fig. 9). Thus, for part instances produced with different manufacturing configurations resulting in error distributions with varying mean but same standard deviation, length, angle, and midpoint representation of the split edge offers better differentiation.

When the location of the mean of error distributions with same standard deviation change, location of the split edge is affected. No difference is observed in the length of the split edge. Hence, in this case the midpoint of the split edge is more important than its length. Thus for distributions with same mean but varying standard deviation, we observe a pattern in the $(D_{KL}(P^2 \parallel Q_2))$ values generated from midpoint bi-variate distributions (Fig. 10b). The median $(D_{KL}(P^2 \parallel Q_2))$ value for distributions with their mean at $(0.01, 0.01)$ and $(0.011, 0.011)$ is 15.4593 and 16.9105 respectively. Then as the distance of the mean of error distributions from the origin increases such as for error distributions with their mean at $(0.1, 0.1)$, $(-0.1, 0.1)$, $(-0.1, -0.1)$ and $(0.1, -0.1)$, the median $(D_{KL}(P^2 \parallel Q_2))$ value also increases to 33.8169, 33.7143, 32.8356 and 34.3296 respectively. After this we observe a drop in $(D_{KL}(P^2 \parallel Q_2))$ values as the error distributions move further from the origin. We get a median $(D_{KL}(P^2 \parallel Q_2))$ value of 28.3907, 28.3804, 28.7866, 27.2097, 28.1860, 27.9994 for error distributions with their mean at $(0.2, 0.2)$, $(0.21, 0.21)$, $(0.22, 0.22)$, $(0.1, 0.05)$, $(0.2, 0.1)$, $(0.25, 0.125)$ respectively.

This shows that even after the first drop, $(D_{KL}(P^2 \parallel Q_2))$ continues to increase with increasing distance of the mean from the origin (Fig. 10b). We also note that $(D_{KL}(P^2 \parallel Q_2))$ values for error distributions that are equidistant from the origin and axes in all four quadrants lie in a very close range and $(D_{KL}(P^2 \parallel Q_2))$ does not change significantly with the slope of line on which mean of error distributions are located (Fig. 10b). However, not much variation was observed in the $(D_{KL}(P^1 \parallel Q_1))$ values across all coordinate error distributions (Fig. 9).

5.3.3. Analysis

Our numerical simulations show that the length and angle (l_1, θ_1) of the split edge and the coordinates of the midpoint (l_2, θ_2) exhibit exaggeration to distinguish between coordinate error distributions of different manufacturing configurations. Not only that, a direct comparison of the original coordinate error distributions results in a near constant value (31.2792) of the KL-divergence making it prohibitively difficult to perform authentication. Therefore, we get better characterization of manufacturing configurations using our topologically transformed error distributions specifically with the length, angle, and midpoint of the split edge. This consequently shows that by applying topological transformation enabled by *SplitCode*, we can successfully distinguish part instances and their error distributions produced with different manufacturing configurations.

A more important observation to make is that each of these parameter pairs (length-angle vs. midpoint) exaggerate different types of properties between distributions. Specifically, the length and angle (l_1, θ_1) of the split edge responds well to the standard deviation of the coordinate error distribution (Fig. 10(a)). On the other hand, the midpoint (l_2, θ_2) responds better to the location of the mean of the original coordinate error distributions (Fig. 10(b)). For a focused investigation of our approach, we will consider distributions that differ in their spreads (i.e. standard deviation). This case signifies manufacturing configurations with different precision. In the rest of the paper, we will focus on the distribution space populated based on the length and angle (l_1, θ_1) of the split edge.

5.4. Overview of authentication procedure

Our authentication scheme is formulated as a statistical comparison between the topologically transformed error distribution of a test part, i.e. the *test distribution*, with a *reference distribution* that characterizes a given manufacturing configuration (class of machine, process, and parameters) (Fig. 11). The *reference error distribution* is defined for a combination of a part geometry and a given manufacturing configuration and is obtained as follows:

1. Manufacture multiple instances of the part geometry with the given manufacturing configuration.
2. For each manufactured instance,
   (a) Measure the geometric deviations. For example, if the part geometry consists a set of circular holes, the deviations could be the difference between the estimated hole center with the ideal center (as defined in the CAD model of the part).
   (b) Generate the coordinate error distribution by combining all deviations into one sample.
3. Combine the coordinate error distributions for all the manufactured instances into a single distribution.
4. Apply the *SplitCode* algorithm to generate the topologically transformed error distribution. This final distribution is called the reference distribution.

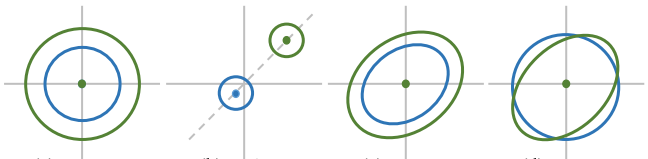


Fig. 12. *SplitCode* was numerically validated for four types of known distributions—(a) Distributions with same mean, zero co-variance, varying standard deviation, (b) Distributions with same standard deviation, zero co-variance, varying mean, (c) Distributions with same mean, non-zero co-variance, varying standard deviation, (d) Distributions with same mean, same standard deviation, varying co-variance.

5.4.1. Authentication test

Given the reference and test distributions, we begin by mapping them to the distribution space (Fig. 11). As highlighted in our design of scheme, both the reference and test distributions are computed based on the length and angle of the split edge. Recall that these are both bi-variate distributions and result in a uni-variate set of points in the distribution space (after mapping using KL divergence) (Fig. 11). Subsequently, we perform one-variable Kolmogorov–Smirnov (KS) test in order to determine whether the test distribution comes from the same manufacturing configuration as the reference distribution (Fig. 11). The $p-$ value of the KS test quantifies whether the reference and test samples come from the same underlying distribution function (the null hypothesis).

We accept the null hypothesis (the two distributions come from a common underlying distribution) if the $p \geq p_s$ where p_s is a significance threshold for the reference manufacturing configuration. Typically, $p < 0.05$ is considered statistically significant to reject the null hypothesis in most statistical tests. For instance, in our case, $p < 0.05$ would mean that the test instance was produced using a manufacturing configuration different from the one represented by the reference. In other words, if $p < 0.05$, then the test instance is not authentic. However, we note that a constant significant threshold is not suitable for comparing an arbitrary pair of distributions. The threshold may be different for different manufacturing configurations. Therefore, given a reference distribution for a manufacturing configuration under question, our authentication scheme additionally prescribes steps to determine its intrinsic threshold as discussed below.

5.4.2. Significance threshold

Significance threshold is the threshold value that helps us to decide if a particular part is manufactured with a given manufacturing configuration or not. Every manufacturing configuration has a different significance threshold just as it leads to a different error distribution. To find significance threshold for a given manufacturing configuration, we follow a simple procedure. We simply perform a statistical (KS) test between a reference and a test error distribution where the test is known to be generated using the same manufacturing configuration as the reference. The $p-$ value (p_s) resulting from this test is used as the significance threshold for the reference error distribution. The basic idea behind this is that when a test error distribution of a part produced with a different manufacturing configuration are statistically compared with the reference error distribution, the $p-$ value for the KS test should be lower than (p_s) as the two distributions in consideration do not belong to the same distribution. Hence, (p_s) obtained by statistically comparing reference error distribution and known test error distribution of the same manufacturing configuration is considered as the significance threshold (p_s) for that manufacturing configuration.

6. Numerical validation of authentication with *SplitCode*

Our first objective for validating the *SplitCode* authentication scheme is to understand its ability to authenticate a test instance with respect to a reference. To achieve this objective, we conducted a series of comparative experiments wherein we consider two manufacturing configurations A and B. We then numerically simulate reference and test distributions for each of these configurations.

6.1. Experimental approach

We follow the following experimental approach:

- Authentication without *SplitCode*: We generate coordinate error distributions A^*_{ref}, A^*_{test}, B^*_{ref}, and B^*_{test} in the original space of geometric deviations in 2D Euclidean space. We then conduct pairwise comparisons between the reference and test distributions using a two-variable KS test.
  Similar to our approach in SplitCode, we first find the significance thresholds, namely p^*_{sA} for A^*_{ref}, p^*_{sB} for B^*_{ref}. Our experiment (*cross test*) is then to compute p^*_{AB} and p^*_{BA}, which are the $p-$ values obtained from comparing B^*_{test} with respect to A^*_{ref} and vice-versa respectively.
- Authentication with *SplitCode*: Here we use our authentication scheme (Section 5.4) after topological transformation of the errors to obtain A_{ref}, A_{test}, B_{ref}, and B_{test} in the distribution space using KL-divergence for length and angle (l_1 and θ_1). We then conduct pairwise comparisons between the reference and test distributions using a one-variable KS test.
  We first find the significance thresholds, namely p_{sA} for A_{ref}, p_{sB} for B_{ref}. Our experiment (*cross test*) is then to compute p_{AB} and p_{BA}, which are the $p-$ values obtained from comparing B_{test} with respect to A_{ref} and vice-versa respectively.

Here, A^*_{ref}, and A^*_{test} have the same mean, standard deviation and co-variance. Similarly, B^*_{ref}, and B^*_{test} have the same mean, standard deviation and co-variance. A^*_{ref}, and B^*_{ref} comprise of 5000 points whereas, A^*_{test}, and B^*_{test} consist of 500 points. The difference in the number of points between reference and test error distributions is to emulate the scenario that reference error distribution is captured from multiple instances whereas a test error distribution is captured from one.

For every cross test that gives a $p-$ value less than the significance threshold for the reference distribution in consideration, we say that we get one true negative result (for example when $p^*_{AB} < p^*_{sA}$, $p^*_{BA} < p^*_{sB}$ before transformation or $p_{AB} < p_{sA}$, $p_{BA} < p_{sB}$ after transformation). Given two manufacturing configurations, when both the cross test results are true negative, we can assert that authentication is perfectly possible for that case. If any one or both cross tests result in $p-$ value greater than or equal to their corresponding significance threshold, we get one or two false positive results respectively. In such a situation, we state that the authentication has failed.

6.2. Experiment design

We perform numerical experiments for four cases of bi-variate normal distributions as follows:

Case 1 (Fig. 12a): Same means ($\mu_a = \mu_b$), different standard deviations ($\sigma_a \neq \sigma_b$), and zero co-variances ($\sigma_{a12} = \sigma_{b12} = 0$).
Case 2 (Fig. 12b): Different means ($\mu_a \neq \mu_b$), same standard deviations ($\sigma_a = \sigma_b$), and zero co-variances ($\sigma_{a12} = \sigma_{b12} = 0$)
Case 3 (Fig. 12c): Same means ($\mu_a = \mu_b$), different standard deviations ($\sigma_a \neq \sigma_b$), and non-zero equal co-variances ($\sigma_{a12} = \sigma_{b12} \neq 0$)
Case 4 (Fig. 12d): Same means ($\mu_a = \mu_b$), same standard deviations ($\sigma_a = \sigma_b$), and different co-variances ($\sigma_{a12} \neq \sigma_{b12}$)

Table 1
Authentication results for numerically validated distributions is shown. Rows 1–4 represent the four cases illustrated in Fig. 12.

Category of Distributions	Condition	Cases	Before Transformation			After Transformation		
			Significance Threshold	Cross Test Results	Possibility of Authentication	Significance Threshold	Cross Test Results	Possibility of Authentication
Same Mean, Varying Standard Deviation, Zero Co-variance	$\mu_A = \mu_B$, $\sigma_A \neq \sigma_B$, $\sigma_{A12} = \sigma_{B12} = 0$	$\sigma_B - \sigma_A = 0.0001$	$p^*_{sA} = 0.5522$, $p^*_{sB} = 0.5522$	$p^*_{AB} = 0.5982$, $p^*_{BA} = 0.4633$	No	$p_{sA} = 0.9050$, $p_{sB} = 0.1794$	$p_{AB} = 0.9239$, $p_{BA} = 0.3112$	No
		$\sigma_B - \sigma_A = 0.001$	$p^*_{sA} = 0.5522$, $p^*_{sB} = 0.5522$	$p^*_{AB} = 0.0501$, $p^*_{BA} = 0.0505$	Yes	$p_{sA} = 0.9050$, $p_{sB} = 0.0626$	$p_{AB} = 0.00002$, $p_{BA} = 0.0199$	Yes
Same Standard Deviation, Varying Mean, Zero Co-variance	$\mu_A \neq \mu_B$, $\sigma_A = \sigma_B$, $\sigma_{A12} = \sigma_{B12} = 0$	$\mu_B - \mu_A = 0.001$	$p^*_{sA} = 0.5522$, $p^*_{sB} = 0.5522$	$p^*_{AB} = 0.5295$, $p^*_{BA} = 0.3280$	Yes	$p_{sA} = 0.0463$, $p_{sB} = 0.0034$	$p_{AB} = 0.0463$, $p_{BA} = 0.0034$	No
		$\mu_B - \mu_A = 0.01$	$p^*_{sA} = 0.5522$, $p^*_{sB} = 0.5522$	$p^*_{AB} = 0.0005$, $p^*_{BA} = 10^{-05}$	Yes	$p_{sA} = 0.0012$, $p_{sB} = 0.0001$	$p_{AB} = 0.0463$, $p_{BA} = 10^{-07}$	No
Same Mean, Varying Standard Deviation Non-zero Co-variance	$\mu_A = \mu_B$, $\sigma_A \neq \sigma_B$, $\sigma_{A12} = \sigma_{B12} \neq 0$	$\sigma_B - \sigma_A = 0.001$	$p^*_{sA} = 0.2975$, $p^*_{sB} = 0.2742$	$p^*_{AB} = 0.2591$, $p^*_{BA} = 0.3580$	No	$p_{sA} = 0.0410$, $p_{sB} = 0.0656$	$p_{AB} = 0.0410$, $p_{BA} = 0.1535$	No
		$\sigma_B - \sigma_A = 0.01$	$p^*_{sA} = 0.2975$, $p^*_{sB} = 0.1091$	$p^*_{AB} = 0.0516$, $p^*_{BA} = 0.0009$	Yes	$p_{sA} = 0.0410$, $p_{sB} = 0.4333$	$p_{AB} = 0.0001$, $p_{BA} = 0.0006$	Yes
Same Mean, Same Standard Deviation, Varying Co-variance	$\mu_A = \mu_B$, $\sigma_A = \sigma_B$, $\sigma_{A12} \neq \sigma_{B12}$	$\sigma_{B12} - \sigma_{A12} = 0.02$	$p^*_{sA} = 0.5522$, $p^*_{sB} = 0.2742$	$p^*_{AB} = 0.1635$, $p^*_{BA} = 0.1769$	Yes	$p_{sA} = 0.0248$, $p_{sB} = 0.0656$	$p_{AB} = 0.0146$, $p_{BA} = 0.0656$	No

The aforementioned cases are selected to explore authentication between possible pairs of manufacturing configurations with varying error distributions. Two manufacturing configurations can either vary in their precision, bias or both. In statistical terms, coordinate error distributions of two manufacturing configurations can have different means, standard deviations or co-variances. In terms of physical interpretation, the first and third cases (Figs. 12a,c) essentially represent a scenario with two machines that have different precision. An example could be two 3D printers of the same make (or even the same 3D printer) and model but one printing at a higher speed than the other. The second case (Fig. 12b) represents a scenario with two machines of the same make and model working at same speeds but with different biases (which may be caused due to different initialization or calibration errors). The final fourth case (Fig. 12d) represents a scenario with completely different machines, printing at possibly different speeds. These scenarios are investigated to develop a deeper understanding of the feasibility of *SplitCode* for authentication.

6.3. Results: Case 1 ($\mu_a = \mu_b$, $\sigma_a \neq \sigma_b$, $\sigma_{a12} = \sigma_{b12} = 0$)

In this category we will study if topological transformation is required for comparing two coordinate error distributions with same bias but different precision. Here we study two different cases. Distributions with higher standard deviation signify error distribution of manufacturing configurations producing part instances with less precision.

All error distributions considered in this category have their mean at $(0, 0)$ and have zero co-variance. In the first case, we have A^*_{ref} and A^*_{test} with a standard deviation of 0.01 and a B^*_{ref} and B^*_{test} with a standard deviation of 0.0101. The difference in the standard deviations of A and B is 10^{-4}. Before topological transformation, as the $p-$ value (p^*_{BA}) for cross test is less than the significance threshold for B^*_{ref} (p^*_{sB}), we get one true negative result (Table 1). However, the $p-$ value (p^*_{AB}) is greater than the significance threshold for A^*_{ref} (p^*_{sA}) which gives us one false positive result (Table 1). Hence, authentication is not possible for this case without topological transformation. After performing topological transformation, results of both the cross tests (p_{AB}, p_{BA}) turn out to be greater than their corresponding significance threshold (Table 1) and thus, leads to two false positive results. Hence, when the difference in standard deviation of two distributions with same co-variance and their mean at the origin is 10^{-4}, authentication is not possible both before and after performing topological transformation.

We increase the difference in the standard deviations of A and B to 10^{-3} in case 2. In this case, we have A^*_{ref} and A^*_{test} with a standard deviation of 0.01 and B^*_{ref} and B^*_{test} with a standard deviation of 0.011. Here $p-$ value obtained for the two cross tests (p^*_{AB}, p^*_{BA}, p_{AB}, p_{BA}) are lower than their corresponding significance thresholds both before and after topological transformation (Table 1). This gives us a perfect authentication case as we get two true negative results before transformation as well as after. Thus, we observe that for manufacturing configurations resulting in parts with different precision, when the difference between the standard deviation of their error distributions is as low as 10^{-4} authentication is not possible with or without topological transformation. However, when the difference in the two standard deviation values is in the range of 10^{-3} or maybe higher, we can successfully authenticate 3D printed parts both before and after topological transformation.

6.4. Results: Case 2 ($\mu_a \neq \mu_b$, $\sigma_a = \sigma_b$, $\sigma_{a12} = \sigma_{b12} = 0$)

In this category we study two cases where the difference in the mean locations of two error distributions in the first case is lower than the second. Here, we have error distributions with different biases where distributions with their mean locations closer to the origin resemble manufacturing configurations generating lesser error and error distributions with mean locations farther from the origin showcase higher error. Particularly for this category of distributions, we focus on the distribution space populated based on the midpoint (l_2, θ_2) of the split edge as the two manufacturing configurations have different bias.

We have A^*_{ref} and A^*_{test} with their means at $(0.01, 0.01)$ and B^*_{ref} and B^*_{test} with their means at $(0.0101, 0.011)$ in the first case. The difference in the location of means of A and B is 10^{-3}. In the second case, the difference in the mean location of A^*_{ref}, A^*_{test} and B^*_{ref}, B^*_{test} increases to 10^{-2} as for the same A^*_{ref}, A^*_{test}, the means of B^*_{ref}, B^*_{test} are shifted to $(0.11, 0.11)$. All distributions in this category have same standard deviation of 0.05 and a co-variance of zero in both directions.

Before topological transformation, we observe that the cross tests give $p-$ values (p^*_{AB}, p^*_{BA}) that are lower than their significant threshold values in both the cases (Table 1). As a result, authentication is possible before transformation when the difference in the mean location of distributions generated from two different manufacturing configurations is 10^{-3} or 10^{-2}. However, after performing topological transformation we notice that for case one, $p-$ value (p_{AB}, p_{BA}) for both cross tests are same as their corresponding significance threshold values giving us two false positive results (Table 1). Similarly, on increasing the difference in the mean locations for A and B, we get one false positive and one true negative results as in the second case cross test result p_{AB} is greater than the significance threshold for A_{ref} (p_{sA}) and cross test result (p_{BA}) is less than the significance threshold for B_{ref} (p_{sB}) (Table 1). Thus, when two manufacturing configurations give rise to coordinate error distributions with same standard deviation but varying mean locations, authentication is only possible before applying *SplitCode*.

6.5. Results: Case 3 ($\mu_a = \mu_b$, $\sigma_a \neq \sigma_b$, $\sigma_{a12} = \sigma_{b12} \neq 0$)

This category is a variation of the first category. These error distributions have a different bias than the error distributions in the first category as they have a non-zero co-variance in both directions. They form an elliptical shape as oppose to circular distributions in the first two categories. Similar to the first category, error distributions with lower standard deviation here refer to manufacturing configuration resulting in lesser error.

Here also we consider two cases. For both cases we generate A^*_{ref} and A^*_{test} with a standard deviation of 0.05. Then we generate B^*_{ref} and B^*_{test} with a standard deviation of 0.051 in the first case and 0.06 in the second case respectively. Thus, the difference in standard deviation of A & B increases from 10^{-3} to 10^{-2} as we move from the first to the second case. All error distributions considered in this category have their mean at the origin and a co-variance of 0.0004.

In the first case where difference in the standard deviation of A & B is 10^{-3}, the cross test result (p^*_{AB}) is lower than the significant threshold for A^*_{ref} (p^*_{sA}) but the other cross test result (p^*_{BA}) is greater than its corresponding significant threshold (p^*_{sB}). As a result, we get one true negative and one false positive result before transformation (Table 1). Whereas after transformation, we get two false positive results as (p_{AB}) is same as (p_{sA}) and (p_{BA}) is greater than (p_{sB}). Hence, authentication is not possible before or after topological transformation in this particular case. When the difference in the standard deviation of A & B increases to 10^{-2}, not only before transformation but also after transformation all cross tests give $p-$ values that are lower than their corresponding significant thresholds (Table 1). We get two true negative results before transformation as well as after transformation for the second case. Hence, when two manufacturing configurations result in coordinate error distributions that have same bias but a difference of 10^{-3} in their precision, authentication is not possible before or even after application of *SplitCode*. However, when the difference in their precision increases to 10^{-2}, authentication is possible before transformation as well as after transformation.

6.6. Results: Case 4 ($\mu_a = \mu_b$, $\sigma_a = \sigma_b$, $\sigma_{a12} \neq \sigma_{b12}$)

Finally, in the last category we compare two error distributions with same mean, same standard deviation and different co-variance, which means that the two manufacturing configurations associated with these error distributions result in part instances with same precision but different bias. We consider A^*_{ref} and A^*_{test} with a co-variance of zero. Then we have another pair of B^*_{ref} and B^*_{test} error distributions B that have a co-variance of 0.0004 in both the directions. All four error distributions have their means at the origin $(0, 0)$ and have a standard deviation of 0.051. In this case, we have a circular distribution A and an elliptical distribution B.

On testing A^*_{ref} against B^*_{test} before transformation and A_{ref} against B_{test} after transformation we get $p-$ values (p^*_{AB}, and p_{AB}) that are lower than their corresponding significance threshold values for A^*_{ref} (p^*_{sA}) and A_{ref} (p_{sA}) respectively. Thus, we get one true negative before transformation, as well as after transformation (Table 1). When we perform a KS test between B^*_{ref} and A^*_{test}, we get (p^*_{BA}) lower than the significance threshold for B^*_{ref} (p^*_{sB}) before transformation (Table 1). Now we have two true negative results before transformation. However, after transformation we get a $p-$ value (p_{BA}) which is same as the significance threshold for B_{ref} (p_{sB}) (Table 1). We consider this result as a false positive which takes our total to one true negative and one false positive after transformation. From these results, we can see that for error distributions with same mean and standard deviation but a difference of 0.0004 in the co-variance, authentication is possible before transformation but not possible after transformation.

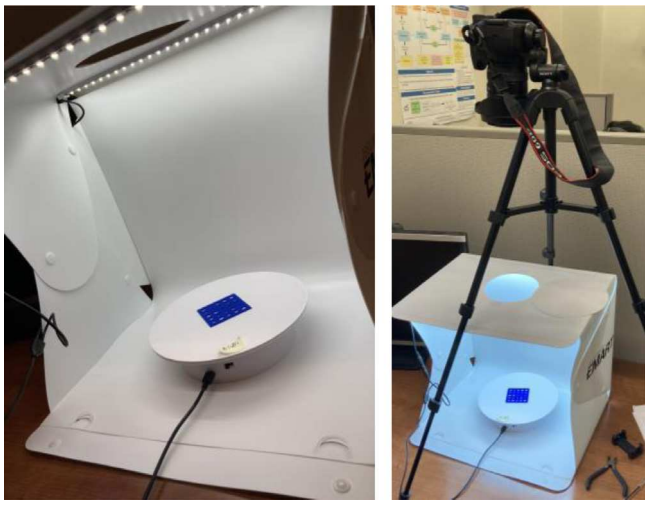


Fig. 13. The printed part was placed on a rotating table in a photo studio light box and multiple images of the part at different orientations are captured.

7. Experimental validation of authentication with *SplitCode*

From the results gathered during numerical simulations, we understand that the performance and accuracy of *SplitCode* depends on the nature of relative coordinate error distributions and the difference between them. We observe that *SplitCode* enables authentication in some cases where two reference coordinate error distributions have same biases but different precision. We also notice that for certain coordinate error distributions with different biases, authentication is possible before application of *SplitCode* but fails after the topological transformation. To check if we get similar results for actual manufactured parts, we perform some physical experiments using additive manufacturing process. These experiments provide insight on the nature of possible coordinate error distributions that one can get for various printer configurations. In this section, we describe about the experiments conducted in detail and present their results.

For these experiments, we assume that error measured with respect to the CAD model of the design in multiple prints printed with a given printer configuration, contribute to the reference error distribution of that printer configuration. Similarly, if we have a test print printed with an unknown printer configuration, error measured in the test print gives us the test error distribution.

7.1. Experimental setup

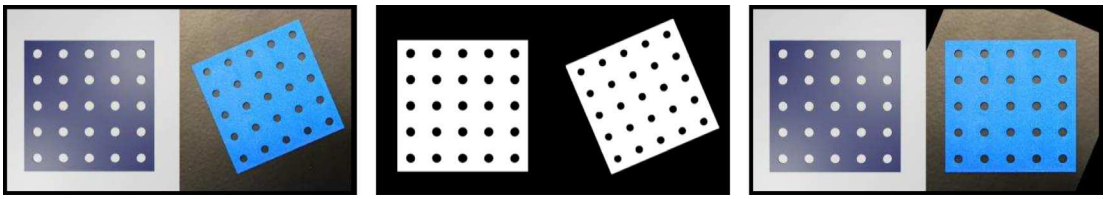
We first started with 3D printing the parts which were then imaged, pre-processed, and registered and were used for computing reference and test error distributions. We used three different printers in our experiments. These printers, their slicing software, and the printing parameters used for fabricating the parts are reported in Table 2. All parts were printed using a PLA (polylactic acid) filament and imaged using Canon EOS 60D DSLR camera with an EF 17–40 mm F/4 USM lens (Fig. 13). In order to remove bias from imaging at a specific part orientation, each part was imaged 15 times, on an average, after rotating it by 24° after each capture. Additionally, we utilized a photo studio light box (Fig. 13) to further remove any discrepancies that could be introduced by variation in the lighting and shadows.

7.2. Design of parts

We used two types of part designs in our experiments which are as follows:-

Table 2
Printers, parts, and their printing parameters used for experiments.

Printer	Parts	Slicer	Layer height (mm)	Nozzle (bed) temperature (C)	Perimeter (infill) printing speed (mm/s)	Infill density (%)
1. Creality Ender 3	a. 4 parts	Cura 4.8.0	0.16	200 (60)	37.5 (75)	100
	b. 2 parts		0.16	200 (60)	18.75 (37.5)	100
2. LulzBot TAZ 6	c. 4 parts	Cura LulzBot edition	0.16	200 (60)	37.5 (75)	100
3. Prusa i3 MK3S	d. 2 parts	PrusaSlicer 2.3.3	0.10	210 (60)	40 (80)	100
	e. 2 parts		0.07	210 (60)	30 (40)	100



(a) Captured images (b) Pre-processed images (c) Registered images

Fig. 14. Images of ideal CAD model and the printed part when (a) captured, (b) pre-processed and (c) registered.

1. Square part with 25 square holes
We designed a square part of dimensions 55 × 55 × 1.2 mm with 25 square holes on it (Fig. 17). This design was created in SOLIDWORKS 2019. The centers of these square holes were arranged in a 5 × 5 grid and the length of their sides were selected randomly from 3 mm, 3.5 mm, 4 mm 4.5 mm and 5 mm. For this part design, we measured manufacturing error at the vertices of the square holes in the images of printed parts with respect to their ideal locations in the CAD image. The error calculated was then utilized to generate coordinate error distributions.

2. Square part with 25 circular holes
We designed another square part of dimensions 50 × 50 × 1.2 mm in SOLIDWORKS 2019 which contained 25 circular holes of varying diameters on it. We chose five different diameters for the circular holes—2 mm, 2.5 mm, 3.5 mm, 4 mm, and 5 mm (Fig. 17). The centers of the holes were arranged in 5 × 5 grid such that in every alternate row starting from the first one, the circular holes were arranged in the increasing order of their diameter from left to right and in the remaining rows they were arranged in the decreasing order of the diameters from left to right. Such an arrangement of holes is chosen to check if the diameter and the location of hole affects the error in printing that hole. For this design, we compute error at the centers of the circular holes in the printed part images with respect to their ideal locations in the CAD image and generate coordinate error distributions.

7.3. Image processing

Given that we captured an average of 15 images per print with different orientations, we must register each captured image to its "ideal" CAD image to calculate the deviations and error distributions (Fig. 14). To achieve this, we utilize MATLAB's Image Processing Toolbox version R2020b.[2] First, we pre-process (Fig. 14b) the captured images and the "ideal" render by applying a global threshold to binarize using Otsu's method [53]. We then apply a morphological erosion followed by a dilation to remove noise while maintaining the hole features. Finally, we align (Fig. 14c) each captured image with the "ideal" one using phase correlation [54] and register them using MATLAB's Intensity-Based Automatic Image Registration technique.[3]

[2] Available: https://www.mathworks.com/products/image.html.
[3] Overview: https://www.mathworks.com/help/images/ref/imregister.html.

7.4. Error measurement and computation of coordinate error distribution

For parts with circular holes, all holes in both the print image and the CAD image were detected using circle detection function (imfindcircles) in MATLAB and each hole in the captured image of the print is mapped to its corresponding hole in the image of the CAD model (Fig. 15). We then find out the centers of the two holes. We finally measure the length and angle of the line joining the centers of the two holes in the printed part image and CAD image. Thus, we get the errors introduced while printing that particular hole with a given printer configuration. In the same manner we compute error for all the 25 holes in all images of the print and use that to create the coordinate error distributions of the print. It was observed that parts with circular holes led to elliptical or circular coordinate error distributions (Fig. 17).

For parts with square holes, a corner detection function (detectHarrisFeatures) was used in MATLAB to detect all the corners of the holes in the captured images. At this stage, multiple factors come into picture in the error computation process. Firstly, because of the noise in the captured images, we do not get straight line edges for the square holes when we process the images in MATLAB (Fig. 16a). As a result, multiple points are detected in the images by the corner detection function (Fig. 16b). We map all the detected points on the ideal CAD image and find the points that are closest to the actual corners of the square holes on the CAD image. An important point to note here is that, because of the fundamental nature of printing and the thickness of the printing filament, as the printing direction changes by 90° we do not get a sharp corner. Instead we get fillets. Secondly, when we take images of these prints, shadows increase the probability of detecting a fillet in the location of a corner.

Because of the above two reasons, the points detected by the corner detection function often lie on the edges that are incident on the corner (Fig. 16c). Naturally, points that are closest to the corners in the CAD image also lie on one of the two incident edges. As a result, we get a diamond-shaped (Fig. 17) error distribution for parts with square holes.

7.5. Design of physical experiment

In our experiments we focus mainly on the following two problems:-

1. Application of topological transformation to perform authentication between parts printed on two different printers (either at same or different speeds).
2. Application of topological transformation to identify parts printed on the same printer but with different speed and precision.

In total, we perform three experiments as described below (Fig. 17):-

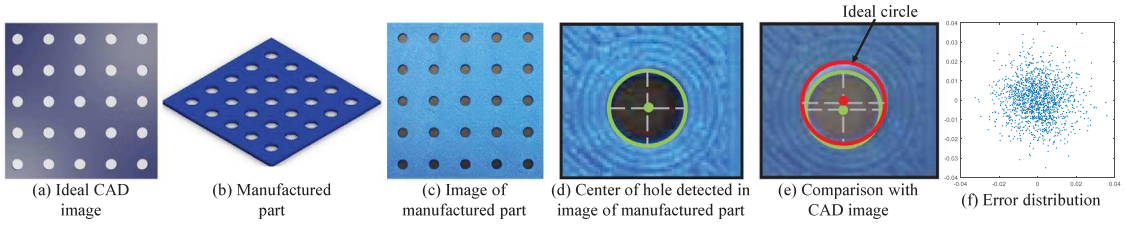


Fig. 15. Error distribution is generated by measuring the deviations in the image of the printed part with respect to the rendered ideal CAD image.

1. In the first experiment we want to understand if the exaggeration caused by topological transformation helps in detecting a counterfeited print printed on a printer other than the authentic printer at the same speed. For this, we print four prints each of the design with circular holes on the printers Creality Ender 3 (a1, a2, a3, & a4) and LulzBot TAZ 6 (b1, b2, b3, & b4). Both the printers print at the same speed. Three prints from each printer (a1, a2, a3 & b1, b2, b3) were used to generate the reference error distribution for the printers and the remaining prints (a4 & b4) were used as test parts. Here we perform cross tests between test part printed on LulzBot TAZ 6 (b4) and reference error distribution of Creality Ender 3 (a1, a2, a3). Similarly, we also perform a cross test between test part printed on Creality Ender 3 (a4) and reference error distribution of LulzBot TAZ 6 (b1, b2, b3).

2. For the second experiment, we use two prints (a1 & a2) printed at a lower speed on Creality Ender 3 as test prints and compare them with the four prints (b1, b2, b3, & b4) printed at a higher speed on LulzBot TAZ 6. Design of part with circular holes is used in this experiment. First we perform a KS test between error distributions coming from the first three prints (b1, b2, & b3) and the fourth print (b4) printed on LulzBot TAZ 6. This KS test gives us the significance threshold for the printer LulzBot TAZ 6 at 37.5 mm/s. We then perform a cross test between the test prints printed on Creality Ender 3 (a1 & a2) and the reference error distribution of LulzBot TAZ 6 printer generated from b1, b2, & b3.

3. In the last experiment our goal is to utilize topological transformation for detecting parts printed on the same authentic printer but at different speeds. To explore this situation, we wanted to select a design where change in error specifically due to change in speed could be measured. We know that when the direction of printing changes, printing speed plays an important role in maintaining the accuracy of printing. Hence, we selected the design with square holes and measured printing error at the corners of the square holes where the direction of printing changes. We used printer Prusa i3 MK3S and printed two prints (a1 & a2) at a lower speed and two at a higher speed (b1 & b2). We first find the significance threshold of the printer at lower speed by performing KS test between the error distributions generated from prints a1 & a2. Later, prints a1 & a2 together contribute to the reference error distribution of the printer at lower speed and we perform cross test between the reference error distribution and error distributions of the prints printed at a higher speed (b1 & b2).

7.6. Results

7.6.1. Experiment 1

In this experiment, coordinate error distributions for prints printed on Creality Ender 3 have an elliptical shape (Fig. 17). These error distributions are identical to the distributions with their mean at the origin and non zero co-variance. Orientation of the error distributions of these four prints roughly tell us about the possibility that they are

Fig. 16. As a combined effect of printing, imaging and pre-processing, corners of square holes detected in the image of printed part lie on one of the edges incident on the actual corners in the CAD image.

all printed with the same printer configuration. On the other hand, coordinate error distributions of prints printed on LulzBot Taz 6 have circular shape (Fig. 17). These distributions can be related to bi-variate normal distributions with their mean at the origin and co-variance equal to zero. Thus, this experimental case resembles to the simulated case where we had one elliptical and one circular reference and test distribution.

We will consider the Creality Ender 3 as Printer A and the LulzBot Taz 6 as printer B. Before transformation we first find the significance threshold for printer A as 4.14×10^{-05} and printer B as 0.0104. With these values as reference, we perform the cross tests. We get a p value of 10^{-13} for the cross test between reference error distribution A and test error distribution B which is lower than significance threshold of A. For the other cross test between reference error distribution B and test error distribution A we get $p = 10^{-12}$. This value is also lower than the significance threshold of printer B. Hence, we clearly get two true negative results before transformation indicating that test print A is not printed on printer B and test print B is not printed on printer A. From these results, we can tell that authentication for this experimental case is possible without topological transformation.

After topological transformation, we get a significance threshold of 0.0029 for printer A and 0.01 for printer B. On performing cross test between reference distribution A and test distribution B we get $p = 10^{-12}$. Similarly, cross test between reference distribution B and test distribution A also results in $p = 10^{-12}$. p values resulting from both the cross tests are lower than their corresponding significance thresholds. Hence, we get two true negative results after transformation too. We can therefore say that, for this experimental case, authentication is possible both before and after topological transformation. Hence, if we have two printer configurations with significant difference in the mean location of their reference error distributions (significantly different bias), we can statistically compare reference and test error distributions without performing topological transformation on them for authentication.

7.6.2. Experiment 2

Error distributions generated in this experiment tells us that by reducing the speed of printing in a cheaper printer (Creality Ender 3), it is possible to minimize the difference between error distributions

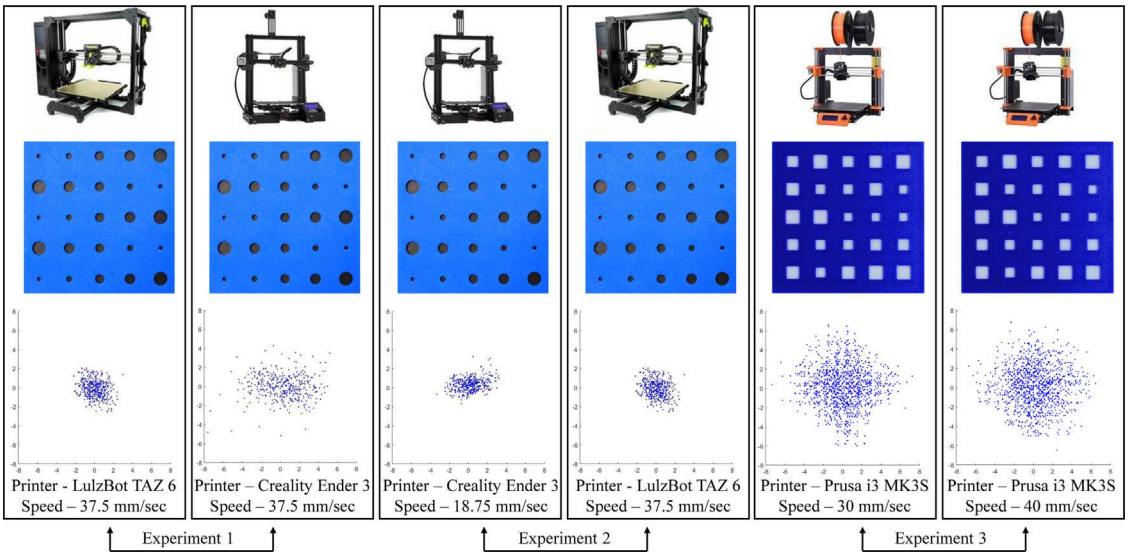


Fig. 17. In our experiments, we used a combination of five printer configurations to print parts with either circular or square holes, parts with circular holes generated elliptical error distributions whereas parts with square holes resulted in pyramid-shaped error distribution.

for prints printed by a cheaper printer and the prints printed by an expensive printer at a higher speed. Here, the significance threshold of printer (LulzBot TAZ 6) is 0.0104 before transformation and 0.01 after transformation. We then compare each of the two test error distributions from prints printed on Creality Ender 3 with the reference error distribution of LulzBot TAZ 6. Before transformation, cross test results for the two prints are $p = 10^{-12}$ and $p = 10^{-10}$ respectively. These two values are lower than the significance threshold value before transformation. Now after performing topological transformation, we again perform cross tests between the two test distributions and the reference distribution and we get $p = 0.1733$ and $p = 4.61 \times 10^{-07}$ for the two test prints. We notice that one of the two cross tests give us a p value that is greater than the significance threshold of the reference printer configuration and thus, we get one false positive result after transformation. Hence, in this case where the error distributions of two printer configurations vary only slightly in terms of precision and bias, it is possible to authenticate parts before topological transformation but not possible after.

7.6.3. Experiment 3

In the third experiment, we get a significance threshold of 0.0051 for the prints printed at 30 mm/s on the printer before transformation. As soon as we get this value, we test the two prints printed at higher speed (b1 & b2) individually with the reference error distribution generated from prints printed at lower speed. When we perform a KS test between print b1 test error distribution and the reference error distribution, we get a p value of 0.4277. This value is higher than the significance threshold for the printer configuration in consideration. For the KS test between print b2 test error distribution and the reference error distribution, we get a p value of 0.1671. This test also gives us higher p value than the significance threshold. Thus, we get two false positive results as they falsely indicate that prints b1 and b2 are printed with the same printer configuration (speed) as that of prints a1 and a2. We can say that in this case authentication is not possible before topological transformation.

After topological transformation, we get a significance threshold of 0.9748 for prints printed at lower speed (a1 & a2). The first cross test is performed between print b1 test distribution and the reference distribution. We get $p = 0.0946$ for this test. Similarly, we perform a cross test between print b2 test distribution and the reference distribution. This test gives a p value of 0.00013. Both the cross tests tell us that the prints b1 and b2 are not printed with the same printer configuration (same speed) as prints a1 and a2. Hence, we get two true negative results and a perfect authentication case after transformation. Thus, in this case of identifying parts with square holes that have different error distributions due to being printed at a different speed, topological transformation is necessary for authentication as authentication fails before transformation.

8. Discussion

8.1. Part design

In terms of part design, currently we are using centers of circular holes and corners of square holes for estimating error. We have already noticed in our experiments that both the geometry of the hole and the features used for error measurement affect the nature of error distributions and eventually, authentication. In future, an interesting problem is to study the effect of different geometric figures in more detail by selecting figures like triangles, pentagons or hexagons. Experimenting with different features such as centroid, edges or midpoints of edges will give us an understanding on how the estimation of error is affected by the features at which it is calculated. It is also possible to arrange geometric figures in various patterns such as in a triangular or hexagonal grid. Different arrangement of features for the same part will give different error distributions thus, giving us a more accurate representation of the precision and bias of printers.

In our current implementation, we demonstrated *SplitCode* with 2D features such as circle centers and corners. An exciting avenue for research is to extend these ideas to 3D shapes directly. For instance, instead of computing error distributions from planar features that are always on the top surface of a token part, one can measure deviations along straight and curved edges that are spatially distributed on a part. For this, it would also be important to compute the topological transformation in 3D space. In this scenario, the split edge would essentially become a split discrete surface (composed of multiple planar faces). A statistical representation of this, in itself, is a challenging theoretical and computational issue that requires a deeper investigation.

8.2. SplitCode algorithm

There is great scope of exploration within the algorithm of *SplitCode*. Our algorithm is based on a simple principle of topology change in uniform quad grid structures. Changing this quad grid to other nonuniform

Voronoi grid structure will generate different topologically transformed error distributions and provide different amount of exaggeration between two error distributions. In order to increase the sensitivity of topological transformation, arranging Voronoi sites in a circle is worth trying. Our method assumes that imprecise tool movement is the only cause of error in manufactured parts. But in reality, even this error could be affected by factors like temperature of the surroundings, temperature of the machine bed and raw material quality. In order to properly measure error only due to inexact movement of the tool, we could use higher dimensional Voronoi sites like straight lines or curves. These variations in the *SplitCode* algorithm will provide an understanding of which factors affect topological transformation in a positive sense to achieve meaningful exaggeration between two error distributions.

8.3. Statistical variations

Once exaggeration between two error distributions is achieved, the tests used for authentication also affect the performance of authentication. In our work, we use statistical tests that allow use of only two features of the split edge. Instead of considering probability of one split edge to check if the reference and test error distributions belong to the same distribution, using joint probability of a grid of split edges can prove to be a more descriptive representation of an error distribution and hence, improve the accuracy of authentication. Use of multiple split edges and multiple features can be made possible by replacing statistical tests with machine learning algorithms. Another way to get more reliable authentication is to consider location dependent error for a given part. Because of the hardware imperfections, machine error can vary from one location to the other. Experimenting with location dependent error will offer a way to differentiate between machines that have similar overall error distribution but different error distributions when compared location wise.

8.4. Methods for measuring deviations

Since the process of error measurement in manufactured parts start with capturing images of the part and processing them, these processes greatly influence the measured error. As seen in our experiments, image processing and feature detection methods can even change the shape of the error distribution. Hence, it is important to correctly detect the features at which error is estimated. In future, use of better post processing methods for RGB images will facilitate accurate detection of features. While taking images, surrounding lights and shadows tend to sway the measurement of error. Implementing different imaging methods like 3D scans or micro scan can help in eliminating these issues. Finally, utilizing non imaging techniques like acoustic, vibrations or optics for detecting features and errors is also another way to ensure that the error measured is solely caused due to machining.

An important note here is that error distributions may or may not change with different manufacturing processes. Depending on the shape of error distribution we can modify the *SplitCode* algorithm, identify the features that offer maximum exaggeration of error, use suitable imaging and post processing methods, and select proper statistical or machine learning tests for authentication. We should understand here that the overall concept and technique of authentication remains same for all machining processes with some necessary modifications.

8.5. A note on computational complexity

In a real-world setting, the *SplitCode* procedure would involve two components, namely, pre-computation of reference distribution and the actual authentication test. Therefore, the main factor that will dictate the time taken for our procedure would be during authentication, which would involve: (1) imaging (which is semi-manual), (2) image registration, (3) coordinate error computation using feature detection, (4) topological transformation using Voronoi edge split, and (5) Kolmogorov–Smirnov (KS) test for authentication. Image registration and feature detection methods are now quite standard [55] and can be done for very large image sizes within fractions of a second, especially with GPU acceleration [56]. As for the topological transformation, we note that the error exaggeration based on Voronoi decomposition essentially samples exactly 4 points for each computation. Given that the sweep-line algorithm for 2D Voronoi is $O(k\ log(k))$ [57], it is constant in our case since $k = 4$. Therefore, for a sample of n points, the time complexity for error exaggeration is simply $O(n)$. Finally, the complexity of the Kolmogorov–Smirnov for bi-variate (two-dimensional) data is reported to be $O(n^2)$ [58,59].

9. Conclusion

We presented a technique to exaggerate differences between error distributions of two machine configurations based on a combination of Voronoi Tessellation with statistical analysis. Our method enables characterization of machine configurations by characterizing their bias and precision and thereby enables authentication of parts manufactured with them. In contrast to machine learning approaches, our method is principled (i.e. based on fundamental geometric reasoning) and does not require any parameter tuning, and formulates the authentication task in terms of a single confidence value ($p-$ value).

Experiments showed that when the problem is to identify or authenticate a part manufactured with a secondary manufacturing configuration that has a different bias than the part manufactured with the original manufacturing configuration, authentication is possible after application of *SplitCode*. However, when there is significant difference in the bias of two configurations, authentication could be achieved even without applying *SplitCode*. In another scenario wherein there is minor difference between the bias of two manufacturing configurations, authentication is possible before applying *SplitCode*, but it fails after transformation. On the other hand, in cases where the problem is to identify a part produced with a manufacturing configuration that varies in precision, authentication is possible both before and after application of *SplitCode* if the difference between the two precision levels is significant. When the difference between two precision levels is less, depending on the geometry of features used, authentication is not possible before but is possible after applying *SplitCode*.

Although this technique has been experimentally validated only for 3D printed parts in our work, it is applicable to any process where some form of error distribution can be obtained either experimentally or through modeling. Moreover, the technique could be further used to infuse encoded information in the form of synthetic machine noise and other means. Given the potential for extension of this technique, we believe that this work shows initial steps toward a rich research direction in the domain of manufacturing security and authentication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by the National Science Foundation (NSF), USA Award #2048182 (Engineering Design and Systems Engineering Program). The authors would like to thank Dr. Adam Dachowicz for his valuable comments and feedback. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] Suvarna M, Yap KS, Yang W, Li J, Ng YT, Wang X. Cyber–physical production systems for data-driven, decentralized, and secure manufacturing—A perspective. Engineering 2021;7(9):1212–23. http://dx.doi.org/10.1016/j.eng.2021.04.021, URL https://www.sciencedirect.com/science/article/pii/S2095809921002988.

[2] Counterfeit parts: Increasing awareness and developing countermeasures. 2011, https://www.aia-aerospace.org/report/counterfeit-parts-increasing-awareness-and-developing-countermeasures/, accessed: 2018-05-08.

[3] OECD/EUIPO. Trade in counterfeit and pirated goods: Mapping the economic impact. OECD Publishing Paris; 2016, http://www.oecd.org/gov/risk/trade-in-counterfeit-and-pirated-goods-9789264252653-en.htm, accessed: 2018-05-08.

[4] The theft of American intellectual property: Reassessments of the challenge and Unitied States policy. The National Bureau of Asian Research; 2017, Update to the 2013 IP Commission Report, The Commission on the Theft of American Intellectual Property.

[5] Dowling J. Toyota trying to locate thousands of counterfeit airbag parts that could prove deadly in a crash. 2015, http://www.news.com.au/news-story/1f12a255d1093e070b1c9be616ad65ff, accessed: 2018-05-08.

[6] McCormack R. Boeing's planes are riddled with Chinese counterfeit electronic components. 2012, http://www.manufacturingnews.com/news/counterfeits615121.html, accessed: 2018-05-08.

[7] Lucyshyn W, Gansler JS, Rigilano J. Addressing counterfeit parts in the DoD supply chain. Tech. rep., Monterey, California: Naval Postgraduate School; 2013.

[8] IBM. IBM X-Force threat intelligence index. 2020.

[9] Counterfeit Parts: DOD Needs to improve reporting and oversight to reduce supply chain risk. United States Government Accountability Office (GAO); 2016, https://www.gao.gov/products/GAO-16-236. Accessed: 2018-05-08.

[10] SAE international standards: Counterfeit avoidance, detection, mitigation and disposition. SAE International; 2017.

[11] Domdouzis K, Kumar B, Anumba C. Radio-Frequency Identification (RFID) applications: A brief introduction. Adv Eng Inform 2007;21(4):350–5.

[12] Strassner M, Fleisch E. The promise of auto-ID in the automotive industry. MIT; 2003, https://www.dolcera.com/ipmapdemo/rfid/documents/MLB-AUTOID-BC001.pdf. Accessed: 2018-05-08.

[13] Gooch J, Daniel B, Abbate V, Frascione N. Taggant materials in forensic science: A review. TRAC Trends Anal Chem 2016;83:49–54.

[14] sigNatureDNA. 2017, http://adnas.com/signature_dna/, accessed: 2018-05-08. [link]. URL http://adnas.com/signature_dna/.

[15] Delmotte A, Tanaka K, Kubo H, Funatomi T, Mukaigawa Y. Blind watermarking for 3D printed objects by locally modifying layer thickness. IEEE Trans Multimedia 2019;1. http://dx.doi.org/10.1109/tmm.2019.2962306.

[16] Maia HT, Li D, Yang Y, Zheng C. LayerCode: optical barcodes for 3D printed shapes. ACM Trans Graph 2019;38(4):1–14.

[17] Wei C, Sun Z, Huang Y, Li L. Embedding anti-counterfeiting features in metallic components via multiple material additive manufacturing. Addit Manuf 2018;24:1–12.

[18] Tiwari A, Villasenor EJ, Gupta N, Reddy N, Karri R, Bukkapatnam STS. Protection against counterfeiting attacks in 3D printing by streaming signature-embedded manufacturing process instructions. In: Proceedings of the 2021 workshop on additive manufacturing (3D printing) security. AMSec '21, New York, NY, USA: Association for Computing Machinery; 2021, p. 11–21. http://dx.doi.org/10.1145/3462223.3485620.

[19] Maes R. Physically unclonable functions: Constructions, properties and applications. Springer; 2013.

[20] Suh GE, Devadas S. Physical unclonable functions for device authentication and secret key generation. In: Proceedings of the 44th annual design automation conference. ACM; 2007, p. 9–14.

[21] Maes R, Verbauwhede I. Physically unclonable functions: A study on the state of the art and future research directions. In: Towards hardware-intrinsic security. Springer; 2010, p. 3–37.

[22] Kaplan CS. Voronoi diagrams and ornamental design. In: Proceedings of the first annual symposium of the international society for the arts, mathematics, and architecture (ISAMA'99). 1999, p. 277–83.

[23] Gómez-Gálvez P, Vicente-Munuera P, Tagua A, Forja C, Castro AM, Letrán M, Valencia-Expósito A, Grima C, Bermúdez-Gallardo M, Serrano-Pérez-Higueras Ó, et al. Scutoids are a geometrical solution to three-dimensional packing of epithelia. Nature Commun 2018;9(1):1–14.

[24] Mughal A, Cox S, Weaire D, Burke S, Hutzler S. Demonstration and interpretation of 'scutoid'cells formed in a quasi-2D soap froth. Phil Mag Lett 2018;98(8):358–64.

[25] Yampolskiy M, Schutzle L, Vaidya U, Yasinsac A. Security challenges of additive manufacturing with metals and alloys. In: International conference on critical infrastructure protection. Springer; 2015, p. 169–83.

[26] Wong KV, Hernandez A. A review of additive manufacturing. Int Sch Res Not 2012;2012.

[27] Yampolskiy M, Skjellum A, Kretzschmar M, Overfelt RA, Sloan KR, Yasinsac A. Using 3D printers as weapons. Int J Crit Infrastruct Prot 2016;14:58–71.

[28] Yampolskiy M, King WE, Gatlin J, Belikovetsky S, Brown A, Skjellum A, Elovici Y. Security of additive manufacturing: Attack taxonomy and survey. Addit Manuf 2018;21:431–57.

[29] Chen F, Mac G, Gupta N. Security features embedded in computer aided design (CAD) solid models for additive manufacturing. Mater Des 2017;128:182–94.

[30] Chhetri SR, Faezi S, Canedo A, Al Faruque MA. Thermal side-channel forensics in additive manufacturing systems. In: 2016 ACM/IEEE 7th international conference on cyber-physical systems. ICCPS, IEEE; 2016, p. 1.

[31] Al Faruque MA, Chhetri SR, Canedo A, Wan J. Acoustic side-channel attacks on additive manufacturing systems. In: 2016 ACM/IEEE 7th international conference on cyber-physical systems. ICCPS, IEEE; 2016, p. 1–10.

[32] Hojjati A, Adhikari A, Struckmann K, Chou E, Tho Nguyen TN, Madan K, Winslett MS, Gunter CA, King WP. Leave your phone at the door: Side channels that reveal factory floor secrets. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016, p. 883–94.

[33] Belikovetsky S, Yampolskiy M, Toh J, Gatlin J, Elovici Y. Dr0wned–cyber-physical attack with additive manufacturing. In: 11th {USENIX} workshop on offensive technologies ({WOOT} 17). 2017.

[34] Sturm L, Williams C, Camelio J, White J, Parker R. Cyber-physical vunerabilities in additive manufacturing systems. Context 2014;7(8):951–63.

[35] Sturm LD, Williams CB, Camelio JA, White J, Parker R. Cyber-physical vulnerabilities in additive manufacturing systems: A case study attack on the. STL file with human subjects. J Manuf Syst 2017;44:154–64.

[36] Sternstein A. Things can go kaboom when a defense contractor's 3-D printer gets hacked. Nextgov 2014. September 11.

[37] Yampolskiy M, King W, Pope G, Belikovetsky S, Elovici Y. Evaluation of additive and subtractive manufacturing from the security perspective. In: International conference on critical infrastructure protection. Springer; 2017, p. 23–44.

[38] Chen L, He Y, Yang Y, Niu S, Ren H. The research status and development trend of additive manufacturing technology. Int J Adv Manuf Technol 2017;89(9–12):3651–60.

[39] Leal R, Barreiros F, Alves L, Romeiro F, Vasco J, Santos M, Marto C. Additive manufacturing tooling for the automotive industry. Int J Adv Manuf Technol 2017;92(5):1671–6.

[40] Liu R, Wang Z, Sparks T, Liou F, Newkirk J. Aerospace applications of laser additive manufacturing. In: Laser additive manufacturing. Elsevier; 2017, p. 351–71.

[41] ElSayed KA, Dachowicz A, Panchal JH. Information embedding in additive manufacturing through printing speed control. In: Proceedings of the 2021 workshop on additive manufacturing (3D printing) security. 2021, p. 31–7.

[42] Dogan MD, Faruqi F, Churchill AD, Friedman K, Cheng L, Subramanian S, Mueller S. G-ID: Identifying 3D prints using slicing parameters. In: Proceedings of the 2020 CHI conference on human factors in computing systems. 2020, p. 1–13.

[43] Chen F, Luo Y, Tsoutsos NG, Maniatakos M, Shahin K, Gupta N. Embedding tracking codes in additive manufactured parts for product authentication. Adv Energy Mater 2019;21(4):1800495.

[44] Kubo Y, Eguchi K, Aoki R, Kondo S, Azuma S, Indo T. FabAuth: Printed objects identification using resonant properties of their inner structures. In: Extended abstracts of the 2019 CHI conference on human factors in computing systems. 2019, p. 1–6.

[45] Kubo Y, Eguchi K, Aoki R. 3D-printed object identification method using inner structure patterns configured by slicer software. In: Extended abstracts of the 2020 CHI conference on human factors in computing systems. 2020, p. 1–7.

[46] Wei C, Sun Z, Huang Y, Li L. Embedding anti-counterfeiting features in metallic components via multiple material additive manufacturing. Addit Manuf 2018;24:1–12.

[47] Suzuki M, Matumoto T, Takashima Y, Torii H, Uehira K. Information hiding inside 3-d printed objects by forming high reflectance projections. In: Proceedings of the international conference on video and image processing. 2017, p. 146–50.

[48] Harrison C, Xiao R, Hudson S. Acoustic barcodes: passive, durable and inexpensive notched identification tags. In: Proceedings of the 25th annual ACM symposium on user interface software and technology. 2012, p. 563–8.

[49] Gupta N, Chen F, Tsoutsos NG, Maniatakos M. ObfusCADe: Obfuscating additive manufacturing CAD models against counterfeiting. In: Proceedings of the 54th annual design automation conference 2017. 2017, p. 1–6.

[50] Brandman J, Sturm L, White J, Williams C. A physical hash for preventing and detecting cyber-physical attacks in additive manufacturing systems. J Manuf Syst 2020;56:202–12.

[51] Sandborn M, Olea C, White J, Williams C, Tarazaga PA, Sturm L, Albakri M, Tenney C. Towards secure cyber-physical information association for parts. J Manuf Syst 2021;59:27–41.

[52] Li Z, Rathore AS, Song C, Wei S, Wang Y, Xu W. PrinTracker: Fingerprinting 3D printers using commodity scanners. In: Proceedings of the 2018 ACM sigsac conference on computer and communications security. 2018, p. 1306–23.

[53] Otsu N. A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 1979;9(1):62–6.

[54] Reddy BS, Chatterji BN. An FFT-based technique for translation, rotation, and scale-invariant image registration. IEEE Trans Image Process 1996;5(8):1266–71.
[55] Atherton TJ, Kerbyson DJ. Size invariant circle detection. Image Vis Comput 1999;17(11):795–803.
[56] Fung J, Mann S. OpenVIDIA: parallel GPU computer vision. In: Proceedings of the 13th annual ACM international conference on multimedia. 2005, p. 849–52.
[57] Fortune S. A sweepline algorithm for Voronoi diagrams. Algorithmica 1987;2(1):153–74.
[58] Lopes RH, Reid I, Hobson PR. The two-dimensional Kolmogorov-Smirnov test. 2007.
[59] Gonzalez T, Sahni S, Franta WR. An efficient algorithm for the Kolmogorov-Smirnov and Lilliefors tests. ACM Trans Math Softw 1977;3(1):60–4.