Composites: Part A 213 (2027) 110351

Contents lists available at ScienceDirect

Composites Part A

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

Voronoi-based vacuum leakage detection in composite manufacturing

Christoph Brauer [a],*, Arne Hindersmann [a], Timo de Wolff [b]

[a] Deutsches Zentrum für Luft- und Raumfahrt, Institut für Systemleichtbau, Ottenbecker Damm 12, 21684 Stade, Germany
[b] Technische Universität Braunschweig, Institut für Analysis und Algebra, Universitätsplatz 2, 38106 Braunschweig, Germany

ARTICLE INFO

Dataset link: Upper Wing Cover Leakage Data (Original data)

Keywords:
Composite manufacturing
Leakage detection
Refined Voronoi diagram
Voronoi diagram

ABSTRACT

In this article, we investigate vacuum leakage detection problems in composite manufacturing. Our approach uses Voronoi diagrams, a well-known structure in discrete geometry. The Voronoi diagram of the vacuum connection positions partitions the component surface. We use this partition to narrow down potential leak locations to a small area, making an efficient manual search feasible. To further reduce the search area, we use a refined form of Voronoi diagrams known as ordered order-k Voronoi diagrams. We evaluate both variants using a novel dataset consisting of several hundred one- and two-leak positions along with their corresponding flow values. Our experimental results demonstrate that Voronoi-based predictive models are highly accurate and have the potential to resolve the leakage detection bottleneck in composite manufacturing.

1. Introduction

Fiber composite materials are widely used in the production of aerospace components and other applications due to their combination of low weight and high stiffness. Prepreg materials are a specific type of fiber composite that is pre-impregnated with epoxy resin. They are commonly used to produce large-scale, high-quality structures. However, the final properties of fiber composite parts depend on more than just the raw materials used. The production process and its parameters also play a significant role.

In prepreg manufacturing, the component is typically sealed in an airtight cavity and cured under elevated temperature and pressure in an autoclave. The cavity consists of a rigid mold and a flexible vacuum bag, which is sealed to the mold edges using tacky tape. Additional auxiliary materials, such as breather layers, facilitate the evacuation of air from the cavity. For large components, multiple vacuum connections are typically required to maintain an approximately uniform pressure distribution across the component surface.

If the vacuum bag, tacky tape, mold, or vacuum connection leaks during this process, various effects may occur which affect the quality of the components. On the one hand, larger leaks in particular can lead to a reduction in effective pressure. This reduces the compression of the laminate, thereby lowering the fiber volume content. On the other hand, leaks cause an increase in pressure within the cavity, which in turn reduces the force that can act on the trapped air. Consequently, the removal of air is impeded, resulting in porosities within the laminate. These porosities serve as starting points for crack formation under mechanical stress [1].

To prevent this, the cavity is tested for leaks before curing. In principle, either pressure or volume flow measurements can be used for this purpose. In the pressure measurement method, the connection between the vacuum pump and the cavity is closed, and the subsequent pressure increase is evaluated. If the pressure rise over time is small, the cavity can be considered airtight [1,2]. For volume flow measurement, a flow sensor is integrated into the vacuum line to determine the volume flow rate. An airtight cavity is indicated by a negligible or zero flow [1,3]. While both pressure and volume flow measurements provide an indication of the cavity's tightness, neither can identify the precise location of a leak. Therefore, additional methods must be used to detect leaks.

The following approaches are well established in the literature and are also applied in industrial environments. The helium leak test uses a chamber that is placed on the vacuum bag and filled with helium. A sensor measuring the helium concentration is installed upstream of the vacuum pump. If a high helium concentration is detected, a leak is likely present in the area covered by the chamber. The chamber is then moved to the next section until the entire component has been inspected [2]. The ultrasonic leak test exploits the fact that small leaks often generate high-frequency noise due to incoming air. A directional microphone is used to convert these inaudible frequencies into lower, audible ones. Since the microphone is portable, even large components can be examined. The thermographic leak test is based on the Joule–Thomson effect: leaks cause localized cooling of the surrounding air. An infrared camera visualizes these cooled areas and thus identifies the leak locations [4].

* Corresponding author.
E-mail addresses: christoph.brauer@dlr.de (C. Brauer), arne.hindersmann@dlr.de (A. Hindersmann), t.de-wolff@tu-braunschweig.de (T. de Wolff).

https://doi.org/10.1016/j.compositesa.2026.110351
Received 15 July 2026; Received in revised form 23 September 2026; Accepted 7 October 2026
Available online 22 October 2026
1359-835X/© 2026 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
Open access funding enabled and organized by Projekt DEAL.

All of these methods allow for efficient localization of leaks in relatively small or geometrically simple components. However, larger components, or those with complex geometrical features, typically require significantly more time and effort to inspect. As a result, leakage detection can become a bottleneck in the production of large structures such as wing covers.

One strategy to address this challenge is to employ an additional technique that narrows the search area on large components to a size that can be inspected more quickly. In the past, separate flow measurements at different vacuum connections were used for this purpose. The underlying intuition is that air follows the path of least resistance from a leak location to the nearest vacuum connection. Consequently, most of the airflow is expected to follow this path, resulting in the highest value being indicated by the flow meter at the corresponding connection. In other words, if leaks are present, at least one of them is likely to be located closest to the vacuum connection with the largest measured flow. This concept can be formalized using a well-established, classical object from discrete geometry called Voronoi diagram.

Previous literature on methodologies for transforming volumetric flow measurements into leak localization predictions includes knowledge-based approaches as well as machine learning models. In [3], the authors investigate a hand-crafted mapping between order relation patterns in three flow measurements and specific subregions of a triangular surface. The studies [1,5,6] rely on the same dataset captured on a 1.5-square-meter, quadratic experimental part with four vacuum connections. In [1], the authors subdivide the quadratic surface into regular grids of varying granularity and use machine learning-based classification models to predict whether each grid cell contains a leak. This principally enables the localization of multiple leaks simultaneously. The authors of [5] integrate symmetry information into regression neural networks to predict single-leak coordinates. The comparative study [6] considers potential flow theory, different regression models and a volumetric flow matching method, all for the task of single-leak detection. The authors of [7] also use an approximately square experimental setup (81 cm by 72 cm) and investigate different machine learning models for leak localization. The master's thesis [8], which was supervised by the authors of this work, investigates regression neural networks for multi-leak detection using the same dataset as in this article.

Despite these advances, the existing approaches leave an important gap regarding the geometry-independent localization of leaks on large and geometrically complex components. Existing knowledge-based approaches rely on manually constructed mappings, while several machine learning approaches employ regular grids or are developed and evaluated for specific experimental geometries. Consequently, the spatial partition used for the prediction is not necessarily adapted to the geometry of the component and the arrangement of its vacuum connections. Moreover, the applicability of a purely geometry- and knowledge-based approach to industrial-scale components has not yet been systematically investigated.

In this paper, we propose leakage detection models based on Voronoi diagrams. Our work generalizes the partitioning approach used in [3], whose construction on a triangular surface can be interpreted as a Voronoi diagram, to arbitrary two-dimensional surfaces. Furthermore, we extend our approach to refined Voronoi diagrams, a generalization of Voronoi diagrams that allows for a finer surface partition. Similar to [5] our approach incorporates prior information about the part geometry without being limited to a specific geometry, such as a square. Technically, our predictive models are classification models. However, unlike [1], they are purely knowledge- and geometry-based, and, instead of using a regular grid, we use a Voronoi-based subdivision that exploits the geometric features of the part and the vacuum setup. Our numerical experiments, conducted on an industrial-scale wing dataset collected at the German Aerospace Center's Institute of Lightweight Systems in Stade, Germany, demonstrate that Voronoi diagrams are well-suited for leakage localization and that our methods predict Voronoi cells containing leaks with high confidence. We provide our wing dataset and the code necessary to reproduce the subsequently reported results at

https://github.com/chrbraue/leakage_detection.

The remainder of this paper is structured as follows. In Section 2, we present our mathematical model of the vacuum setup and introduce classical and refined Voronoi diagrams, as well as respective predictive models for leak localization. Section 3 includes the experimental evaluation of the proposed methods. In Section 4, we discuss and interpret the obtained results with a focus on practical implications. Section 5 concludes the main part of the paper. Subsequently, Appendix A introduces basic concepts from discrete geometry, while Appendix B provides additional information on classical Voronoi diagrams. Appendix C offers a broader definition of refined Voronoi diagrams and an algorithm for constructing them.

The following notation is used throughout the paper. The surface of the vacuumized part is represented by a connected two-dimensional point set F, and $\mathcal{P}$ denotes a discrete two-dimensional point set representing the positions of vacuum connections. Vectors (namely $\boldsymbol{p}_i \in \mathcal{P}$, flow measurements $\boldsymbol{x} \in \mathbb{R}^k$, and leak positions $\boldsymbol{y} \in \mathbb{R}^2$) are printed in bold, and scalars are in normal font. We denote covers of F by $\boldsymbol{V} = \{V_1, \dots, V_c\}$ (or similar), where the elements $V_1, \dots, V_c \subset F$ are disjoint subsets that together exactly cover F. To denote functions that map a vector of flow values to an element of a cover, we use $f : \mathbb{R}^k \to \boldsymbol{V}$. Furthermore, we abbreviate $[k] = \{1, \dots, k\}$ and use $[k] \times [k]$ to denote the Cartesian product of the same set with itself. Finally, $\|\cdot\|$ refers to the Euclidean norm. Terms are explained in more detail below as needed.

2. Methodology

In this section, we present our mathematical model of the vacuum setup. The model incorporates the surface of the evacuated component, the leak location, the locations of the vacuum connections, and the sensor data collected at these connections. We further introduce predictive models that reduce the search area for leaks from the entire surface to smaller subregions. We provide only the mathematical details necessary to understand the proposed methods. The appendix contains rigorous definitions and derivations of the underlying mathematical structures.

2.1. Mathematical model of the vacuum setup

The methods presented in this work focus on flat components or components with negligible curvature. Accordingly, the surface of the evacuated component is modeled as a two-dimensional set $F \subset \mathbb{R}^2$. If the actual surface is not perfectly flat, then F denotes its two-dimensional projection. The positions of the vacuum connections are represented as a finite set of points

$$\mathcal{P} = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_k\} \subset F \,,$$

and a leak in the vacuum bag is represented by a two-dimensional coordinate vector $\boldsymbol{y} = (y_1, y_2) \in F$. Since leaks are typically very small, their area or diameter is neglected in this model. Alternatively, $\boldsymbol{y}$ may be interpreted as the geometric center of a leak.

To describe the relationship between a leak and the corresponding flow measurements, we consider a leak located at position $\boldsymbol{y} \in F$. This leak causes characteristic flow values $\boldsymbol{x} = (x_1, \dots, x_k)$ that are measured at the vacuum connections located at $\boldsymbol{p}_1, \dots, \boldsymbol{p}_k$. Although measurement noise and minor variations between different realizations of the same vacuum setup may cause small deviations in the measured values, we do not explicitly take these into account.

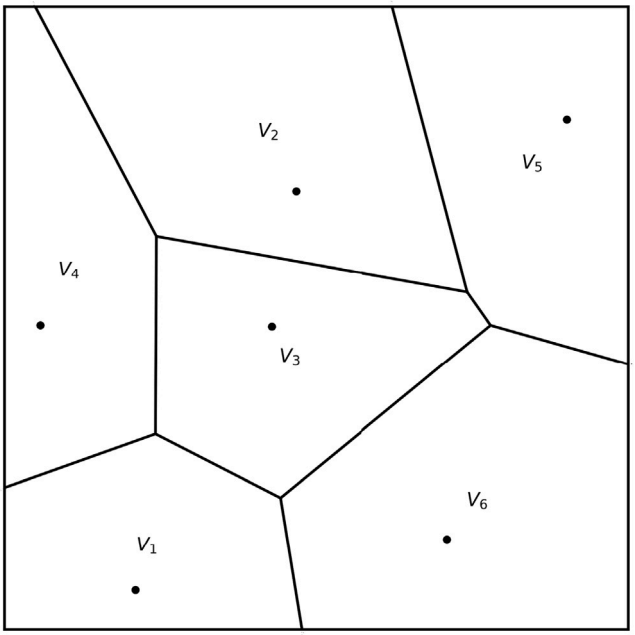


Fig. 1. Classic Voronoi diagram of a random set of points.

2.2. Predictive models for leak localization

We present two approaches for narrowing the search area for potential leaks from the entire component surface to smaller, more manageable regions. This idea is motivated by both practical experience and mathematical considerations. As discussed above, methods such as helium, ultrasonic, and thermographic leak testing can efficiently locate leaks in small areas, but are less effective for large or complex components such as wing covers. From a mathematical perspective, developing a model that predicts exact leak coordinates is inherently difficult, even in the case of a single leak. Instead, it is often more meaningful to determine a confidence region, which effectively narrows down the search area. When multiple leaks occur simultaneously, as is often the case in practice, the problem is even more complex. The same set of flow measurements can then possibly be caused by a different number or assortment of leaks. Under appropriate assumptions, this ambiguity can be proven rigorously, but a detailed analysis is beyond the scope of this paper.

Let $\boldsymbol{V} = \{V_1, \dots, V_c\}$ denote a cover of the surface F, that is,

$$\bigcup_{i=1,\dots,c} V_i = F \ .$$

In Sections 2.2.1 and 2.2.2, we introduce two concepts for constructing such a cover based on Voronoi diagrams, together with functions $f : \mathbb{R}^k \to \boldsymbol{V}$ that map a measurement vector $\boldsymbol{x}$ to a subset $f(\boldsymbol{x}) = V_i$ of the surface that is predicted to contain a leak. The choice of the cover $\boldsymbol{V}$ is purely geometric and knowledge driven; it does not require any observed data. In Section 2.2.3, we present strategies that generalize the two concepts to the multi-leak case.

2.2.1. Classic Voronoi diagrams

Voronoi diagrams, first introduced by Georgy Voronoi in 1908 and by Dirichlet in 1850 for the special cases of two and three dimensions, are a well-known construct in discrete geometry [9]. Voronoi diagrams have numerous applications in fields such as meteorology [10], biology [11], and crystallography [12]. In meteorology, Alfred H. Thiessen famously used them to predict rainfall as early as 1911. In that field in particular, Voronoi diagrams are also referred to as Thiessen polygons. Below, we present the standard definition of a two-dimensional Voronoi diagram, as it is commonly found in the literature. For a more in-depth introduction to discrete geometry and Voronoi diagrams, we refer to Appendices A and B, respectively. Since the following section introduces a refinement of Voronoi diagrams, we refer to the version defined here as the classic Voronoi diagram.

Definition 1. Let $\mathcal{P} = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_k\} \subset \mathbb{R}^2$ be a finite set of points. For each $i \in [k]$, we define the corresponding classic Voronoi cell as

$$V_i = \left\{ \boldsymbol{y} \in \mathbb{R}^2 \;\middle|\; \|\boldsymbol{y} - \boldsymbol{p}_i\| \leq \|\boldsymbol{y} - \boldsymbol{p}_j\| \text{ for all } j \neq i \right\}.$$

We call the collection

$$\boldsymbol{V}(\mathcal{P}) = \{V_1, \dots, V_k\}$$

the classic Voronoi diagram of $\mathcal{P}$.

Example 2. Fig. 1 illustrates the classic Voronoi diagram of a random set of points $\mathcal{P} = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_6\}$ sampled from a uniform distribution on $F = [-1, 1]^2$. As can be seen, the Voronoi diagram $\boldsymbol{V}(\mathcal{P}) = \{V_1, \dots, V_6\}$ partitions the surface into six subsets. The line segment between two Voronoi cells is the set of points equidistant from the two respective points from $\mathcal{P}$. Accordingly, the nodes at which three line segments meet are equidistant from three points.

We introduce our first predictive model based on classic Voronoi diagrams. This model relies on the assumption that, if a leak is present, then the largest flow is measured at the vacuum connection closest to the leak. Conversely, if flow measurements are available and the leak location shall be inferred, then we expect the leak to lie within the region of points that are closest to the vacuum connection exhibiting the largest flow. Hence, the Voronoi diagram constructed from the vacuum connection locations provides a natural geometric framework to formalize and implement this search rule. If x_i is the largest entry in the flow vector $\boldsymbol{x}$, then the corresponding Voronoi cell V_i is predicted to contain the leak.

Definition 3. Let $\boldsymbol{V}(\mathcal{P}) = \{V_1, \dots, V_k\}$ denote the classic Voronoi diagram of the vacuum connection locations $\boldsymbol{p}_1, \dots, \boldsymbol{p}_k$. Define

$$i^*(\boldsymbol{x})$$

as the index corresponding to the largest entry of the measurement vector $\boldsymbol{x} \in \mathbb{R}^k$:

$$x_{i^*(\boldsymbol{x})} \geq x_j \quad \text{for all} \quad j \in [k] \setminus \{i^*(\boldsymbol{x})\}$$

Then,

$$f_1(\boldsymbol{x}) = V_{i^*(\boldsymbol{x})}$$

defines a function $f_1 : \mathbb{R}^k \to \boldsymbol{V}(\mathcal{P})$, referred to as the classic Voronoi predictor, which maps a measurement vector $\boldsymbol{x} \in \mathbb{R}^k$ to the Voronoi cell associated with the vacuum connection exhibiting the largest flow measurement.

2.2.2. Refined Voronoi diagrams

Various generalizations of the classic Voronoi diagram have been proposed and applied in the literature. These include generalizations with respect to the underlying metric (other than the Euclidean ℓ_2-norm), cells defined relative to objects other than single points (e.g., line segments), and extensions to spatial dimensions beyond two; see, for example [9], for an overview. Among these, the most extensively studied variant is arguably the higher-order Voronoi diagram, also referred to as the k-nearest neighbor Voronoi diagram. These objects were first introduced in [13], and remain an active field of research; see also, for example, [14,15]. Order-d Voronoi cells are defined such that each point within a given cell shares the same d nearest sites, irrespective of their order.

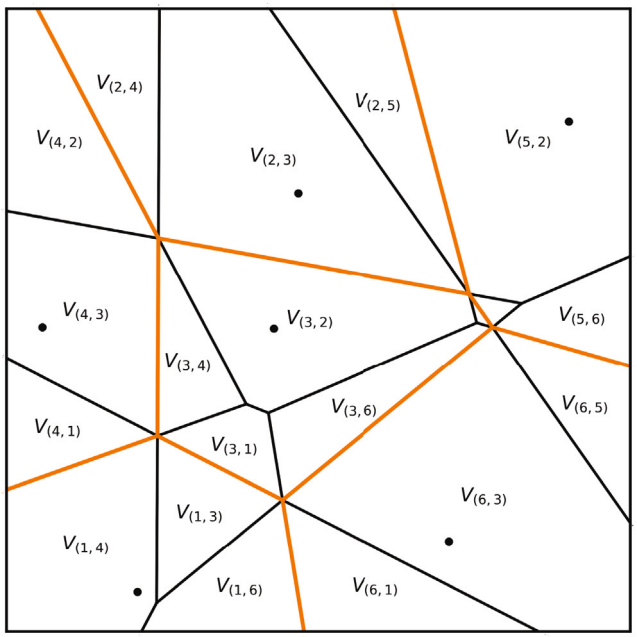


Fig. 2. Order-two refined Voronoi diagram of a random set of points.

In this work, we use refined Voronoi diagrams, also known as ordered order-k Voronoi diagrams [16], which constitute a generalization and refinement of both classic and higher-order Voronoi diagrams. The key distinction is that, in refined Voronoi diagrams, the ordering of the nearest sites is explicitly taken into account — each cell corresponds to a particular permutation of the d nearest sites. In what follows, we present the definition of order-two refined Voronoi diagrams, which are also employed in our empirical experiments. We provide a more general definition for arbitrary order-d refined Voronoi diagrams, along with details on their construction, in Appendix C.

Definition 4. Let $\mathcal{P} = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_k\} \subset \mathbb{R}^2$ be a finite set of points and

$$T_2 = \left\{ (i_1, i_2) \in [k] \times [k] \;\middle|\; i_1 \neq i_2 \right\}$$

the set of ordered index pairs without repetition. For each $\boldsymbol{t} = (i_1, i_2) \in T_2$, we define the corresponding order-two refined Voronoi cell as

$$V_{\boldsymbol{t}} = \Big\{ \boldsymbol{y} \in \mathbb{R}^2 \;\Big|\; \|\boldsymbol{y} - \boldsymbol{p}_{i_1}\| \leq \|\boldsymbol{y} - \boldsymbol{p}_{i_2}\| \leq \|\boldsymbol{y} - \boldsymbol{p}_j\| \quad \text{for all } j \in [k] \setminus \{i_1, i_2\} \Big\}.$$

We call the collection

$$\boldsymbol{V}_2(\mathcal{P}) = \{V_{\boldsymbol{t}}\}_{\boldsymbol{t} \in T_2}$$

the order-two refined Voronoi diagram of $\mathcal{P}$.

Example 5. Fig. 2 revisits the example introduced in Fig. 1. The classic Voronoi diagram is shown in orange. Consider the central cell V_3, which has five edges. In the refined Voronoi diagram (black), V_3 is subdivided into five refined Voronoi cells. Each of these subcells, denoted $V_{(3,j)}$, corresponds to one of the five neighboring cells and contains the points that are closest to $\boldsymbol{p}_3$ and second-closest to $\boldsymbol{p}_j$. While V_3 is adjacent to each other classic Voronoi cell, this is not true for the remaining five cells. Each of them has only three adjacent cells and, accordingly, three edges; thus, each is subdivided into three refined Voronoi cells. For instance, the cell V_1 is subdivided into $V_{(1,3)}$, $V_{(1,4)}$ and $V_{(1,6)}$, whereas $V_{(1,2)}$ and $V_{(1,5)}$ are empty. The connection to the corresponding higher-order (2-nearest neighbor) Voronoi diagram is as follows: each pair of refined Voronoi cells $V_{(i,j)}$ and $V_{(j,i)}$ is separated by an edge that also belongs to the classic Voronoi diagram. The corresponding higher-order Voronoi diagram can then be obtained by merging each such pair of cells. Informally, this means that removing all orange edges in Fig. 2 yields the corresponding higher-order (2-nearest neighbor) Voronoi diagram.

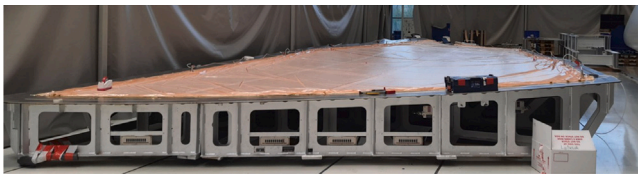


Fig. 3. Upper wing cover mold.

With the definition of a refined Voronoi diagram in hand, introducing our second predictive model is straightforward. Building on our earlier assumption that a leak is closest to the vacuum connection with the largest flow, we extend this idea as follows: We expect the leak to be closest to the vacuum connection with the largest flow and second-closest to the connection with the second-largest flow. This notion of first- and second-closest proximity is precisely captured by order-two refined Voronoi diagrams. The following definition adapts the concept of the Voronoi predictor from classic Voronoi diagrams to order-two refined Voronoi diagrams.

Definition 6. Let $\boldsymbol{V}_2(\mathcal{P}) = \{V_{\boldsymbol{t}}\}_{\boldsymbol{t} \in T_2}$ denote the order-two refined Voronoi diagram of the vacuum connection locations $\boldsymbol{p}_1, \dots, \boldsymbol{p}_k$. Define

$$\boldsymbol{t}^*(\boldsymbol{x}) = (i_1^*(\boldsymbol{x}), i_2^*(\boldsymbol{x}))$$

as the indices corresponding to the two largest entries of $\boldsymbol{x}$, sorted in descending order:

$$x_{i_1^*(\boldsymbol{x})} \geq x_{i_2^*(\boldsymbol{x})} \geq x_j \quad \text{for all} \quad j \in [k] \setminus \{i_1^*(\boldsymbol{x}), i_2^*(\boldsymbol{x})\}$$

Then,

$$f_2(\boldsymbol{x}) = V_{\boldsymbol{t}^*(\boldsymbol{x})}$$

defines a function $f_2 : \mathbb{R}^k \to \boldsymbol{V}_2(\mathcal{P})$, referred to as the order-two refined Voronoi predictor, which maps a measurement vector $\boldsymbol{x} \in \mathbb{R}^k$ to the order-two refined Voronoi cell associated with the vacuum connections exhibiting the two largest flow measurements.

2.2.3. Strategies for multi-leak localization

Both the Voronoi and refined Voronoi predictors map a set of flow values to a single region predicted to contain a leak. In practice, however, the vacuum bag often contains more than one leak, and the total number of leaks is unknown. To apply our predictive models to such situations, we introduce the strategy of repeatedly applying a Voronoi predictor. If more than one leak is possible, this strategy assumes that the region predicted by the Voronoi predictor contains at least one of them. Once the first leak is located and repaired, flow measurements change until a new flow equilibrium is established. Then, we use another application of the Voronoi predictor to locate the next leak. We repeat this procedure until the total flow drops below a certain threshold, which indicates that the setup is sufficiently tight.

While the above-mentioned strategy involves fixing a leak before predicting the next one based on another measurement, it is, of course, desirable to use a single initial measurement to detect multiple leaks simultaneously. We can, for example, predict the classic Voronoi cells corresponding to the two connections with the largest flows to detect two leaks at once with a single measurement. This principle is similar to that of the refined Voronoi predictor, except the prediction consists of two classic Voronoi cells instead of one refined Voronoi cell. However, this strategy's capability is limited if the number of leaks is unknown. Furthermore, it is not equally applicable to the classic and refined

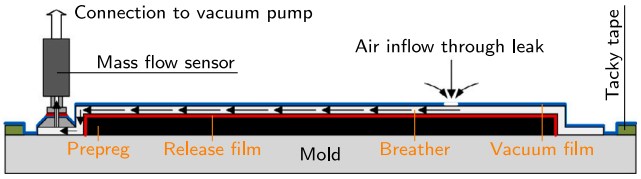


Fig. 4. Schematic cross-sectional view.

Voronoi predictors. Therefore, we do not consider it in our experimental evaluation. Another way to predict multiple leaks based on a single measurement is training a deep learning [17] model. Multi-label classification models, as used in combination with regular grid cells in [1], are principally suitable for this purpose. Each grid cell – or, in our case, each classic or refined Voronoi cell – then corresponds to one class. We will discuss the applicability of these models further below. However, implementing machine learning models for our current use case is beyond the scope of this paper and will be the subject of future work.

3. Experiments

We evaluate the predictive models introduced in the previous section using the upper wing cover mold shown in Fig. 3. This mold, manufactured for research purposes, measures approximately 5.2×16 meters. It is industrial in scale, except for the missing wingtip at the narrower end. Before data acquisition, we first cover the tool surface with breather material (Airtech Ultraweave 1332), followed by vacuum film (Airtech WL 600 V). Then, we seal the setup airtight along the boundary with tacky tape (Airtech GS 213-3). Finally, we install ten vacuum connections (Airtech Multi-Valve-407) equipped with flow sensors (Vögtlin Instruments GmbH red-y smart GSM) at specified positions around the perimeter. Fig. 4 shows a schematic cross-sectional view of a typical vacuum setup. We assume that the airflow caused by a leak occurs only within the breather. This assumption is based on the fact that the release film (red) between the breather and the prepreg (black) acts as a barrier. For this reason, we omit the prepreg material from the actual experimental setup.

3.1. Data acquisition

Once the setup is airtight and evacuated, the process to acquire a measurement is as follows: While the vacuum pump is running, we use a hypodermic needle with an inner diameter of 0.27 mm to pierce the vacuum bag at a specific position, as illustrated in Fig. 5. Air begins to flow through the vacuum connections due to the induced leak. After a short period, the airflow stabilizes. We record the equilibrium values (see Fig. 6) along with the leak position. Finally, we seal the leak with adhesive tape and repeat the process at the next location.

Fig. 7 shows the detailed positions of the measurement points. To generate leak locations, we use a regular grid with a step size of 250 mm, and select a random grid node for each trial. To efficiently produce two-leak samples, we take an intermediate step between two single-leak trials. Instead of immediately fixing a previously recorded single leak, we first introduce a second leak at the next selected grid node. Then, we record the equilibrium airflow along with the positions of both open leaks. After this step, we fix the previously recorded single leak, and measure the equilibrium airflow corresponding to the second single leak. We repeat this procedure as needed. Using this approach, we generated a total of 413 single-leak examples and 440 two-leak examples (some positions appear in more than one two-leak pair).

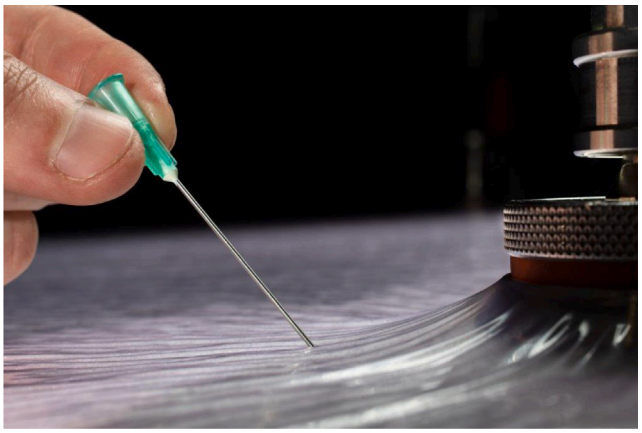


Fig. 5. Manual leak introduction using a hypodermic needle.

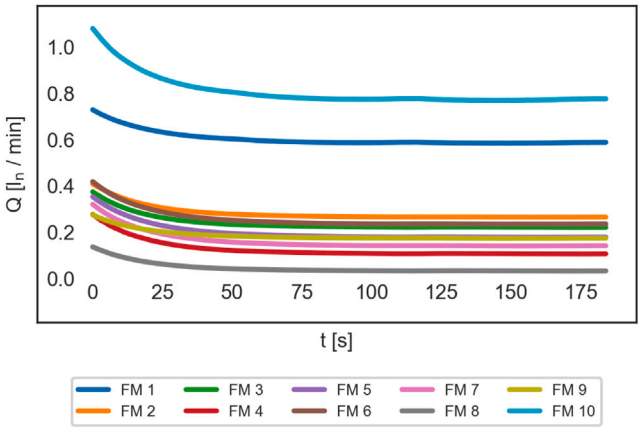


Fig. 6. Flow measurements over time (FM = Flow Meter)

3.2. Metrics

We evaluate our predictive models in terms of two metrics: accuracy (ACC) and mean Euclidean distance (MED). Accuracy is defined as the proportion of correct predictions among all evaluated samples. Formally, this is expressed as follows:

$$\mathrm{ACC} = \frac{\text{number of correct predictions}}{\text{number of samples}}$$

The Euclidean distance is commonly used to measure the discrepancy between two vectors, e.g., between true and predicted coordinates. However, since we are considering categorical predictions, in which each Voronoi cell corresponds to one category, the use of the Euclidean distance requires further explanation. Ultimately, we evaluate Euclidean distances between true leak coordinates and their projections onto predicted Voronoi cells. The Euclidean projection of a point $\boldsymbol{y}$ onto a Voronoi cell V_i is defined as

$$\mathrm{proj}_{V_i}(\boldsymbol{y}) = \arg\min_{\boldsymbol{z} \in V_i} \|\boldsymbol{z} - \boldsymbol{y}\| .$$

In other words, the Euclidean projection of $\boldsymbol{y}$ onto V_i is the point in V_i that is closest to $\boldsymbol{y}$ in terms of Euclidean distance. Based on this, we use the following definition of mean Euclidean distance in our evaluation:

$$\mathrm{MED} = \frac{1}{\text{number of samples}} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \text{samples}} \|\boldsymbol{y} - \mathrm{proj}_{f(\boldsymbol{x})}(\boldsymbol{y})\|$$

This is precisely the mean distance between the actual leak coordinates and their projections onto the predicted Voronoi cells, where

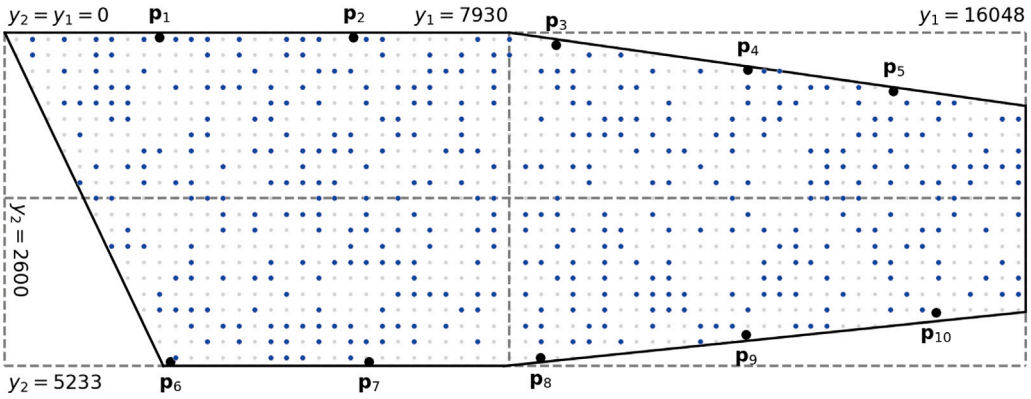


Fig. 7. Experimental data.

either $f = f_1$ (classic Voronoi predictor) or $f = f_2$ (refined Voronoi predictor). For each sample $(\boldsymbol{x}, \boldsymbol{y})$, there are two possible cases. First, if the Voronoi predictor makes a correct prediction, then the actual leak position is inside the predicted Voronoi cell. In this case, $\mathrm{proj}_{f(\boldsymbol{x})}(\boldsymbol{y}) = \boldsymbol{y}$, and the Euclidean distance for that sample is zero. Second, if the Voronoi predictor makes an incorrect prediction, then the term $\|\boldsymbol{y} - \mathrm{proj}_{f(\boldsymbol{x})}(\boldsymbol{y})\|$ is non-zero, indicating the exact distance between the actual leak position and the predicted Voronoi cell. The factor $1/(\text{number of samples})$ indicates that we are averaging over all samples, including both correct and incorrect predictions. Alternatively, it may be of interest to know the error restricted to incorrect predictions only. In that case, the only necessary change is replacing $1/(\text{number of samples})$ with $1/(\text{number of incorrect predictions})$. In our evaluation below, we report both values. The required projections are approximated by 100 iterations of Dykstra's projection algorithm [18]. The selected number of 100 iterations was tuned by hand and ensures convergence of the method according to our observations.

3.3. Data cleaning

There are several possible reasons why the data can contain outliers. For example, a momentary malfunction of a flow meter, vacuum connection, or software can cause implausible measurement records. Additionally, human error on the part of the experimenter can be a factor. Collecting the data took approximately 250 h, requiring high concentration. Sources of human error include manually introduced leaks at positions different from the experimental design, improperly repaired leaks that cause biased measurements in subsequent steps, and typos or mix-ups when entering steady-state flows in a table. Prior to evaluation, we aim to remove such outliers from the data. To that end, we also use the Euclidean projection approach described above. Specifically, if the distance between a true leak position and the associated predicted classic Voronoi cell is greater than a manually chosen threshold of two meters, then the respective sample is considered an outlier and removed from the data. Fig. 8 shows the identified outliers, which include five single-leak samples and one two-leak sample. For the two-leak samples, a sample is considered an outlier if both leaks are more than two meters from the corresponding classic Voronoi cell of the sensor with the largest flow. For the sake of transparency, all evaluations were conducted on both the original and cleaned data, and we report all corresponding results below.

3.4. Experimental results on single-leak data

In this section, we present the experimental results obtained using classic and refined Voronoi predictors on single-leak data. As mentioned earlier, the original single-leak dataset contains 413 samples, five of which were identified as outliers during the cleaning process. Thus, the cleaned single-leak dataset contains 408 samples.

Fig. 9 provides an overview of the true leak locations in the cleaned single-leak dataset, along with the associated predicted Voronoi cells. Table 1 reports the corresponding accuracies and mean Euclidean distances, also those for the original single-leak dataset. Here, we discuss the results on the cleaned single-leak dataset further.

Figs. 10 and 11 illustrate confusion matrices obtained using Voronoi predictors on cleaned single-leak data. The entries in the matrices are based on the frequency with which a predictor misclassifies pairs of classes in both directions. The absolute counts for each combination are then normalized either column-wise (Figs. 10(a) and 11(a)) or row-wise (Figs. 10(b) and 11(b)). In the case of column normalization, the values on the diagonal of the confusion matrix correspond to class-specific precision, or the estimated probability that a prediction of a particular class is correct. In the row-wise case, the diagonal values correspond to class-specific recall, or the estimated probability of a correct prediction given a particular true class.

The classic Voronoi predictor achieves an accuracy of 91.67%. As Fig. 9(a) shows, most leak locations are marked with circles indicating correct predictions. Most of the incorrectly predicted samples, marked with crosses, have leak locations that are only marginally outside the predicted classic Voronoi cell. The mean Euclidean distance between the true leak locations and the predicted classic Voronoi cells for the incorrect predictions is 20.58 cm. This distance must be considered relative to the total part area, which is approximately 16×5.2 m. False positive predictions concentrate on Voronoi cells V_1 and V_{10}, which lie in the upper left and lower right corners, respectively. This can also be seen in the confusion matrix in Fig. 10(a), where the first and last diagonal entries indicate low precision for these two cells. Consequently, the classic Voronoi cells V_2, V_5, V_6 and V_9, which encompass the true leak locations corresponding to these false positive predictions, exhibit low recall, as indicated by the diagonal values in the confusion matrix in Fig. 10(b).

The refined Voronoi predictor achieves an accuracy of 74.75% on the single-leak data. As with the classic Voronoi predictor, Fig. 9(b) shows that the largest deviations between the true leak locations and the predicted refined Voronoi cells occur at the left and right boundaries of the part's surface. The confusion matrix in Fig. 11(a) shows that the precision values of these cells are significantly lower than the average. The mean Euclidean distance with respect to incorrect predictions is 25.23 cm. Relative to the classic Voronoi predictor, this corresponds to a 22.59% increase.

Furthermore, the confusion matrices in Fig. 11 reveal a side effect of using refined Voronoi diagrams for leak detection: On the one hand, there are Voronoi cells that are so small they do not contain any true leak positions and also are not predicted. Specifically, these are $V_{(2,6)}$, $V_{(6,2)}$, $V_{(3,7)}$, $V_{(7,3)}$, $V_{(3,9)}$, $V_{(9,3)}$, $V_{(5,9)}$, and $V_{(9,5)}$, and the diagonal entries in the confusion matrices corresponding to these cells are zero. Using $V_{(2,6)}$ and $V_{(6,2)}$ as an example, Fig. 9(a) shows that the corresponding classic Voronoi cells V_2 and V_6 are diagonally adjacent

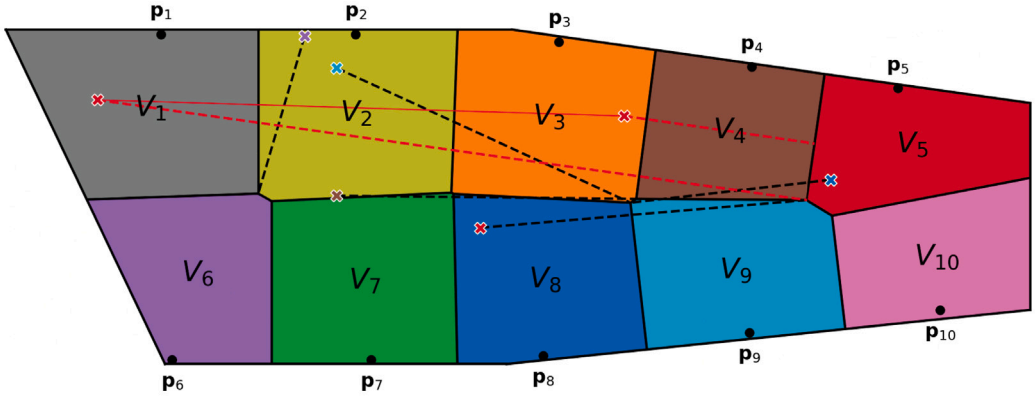
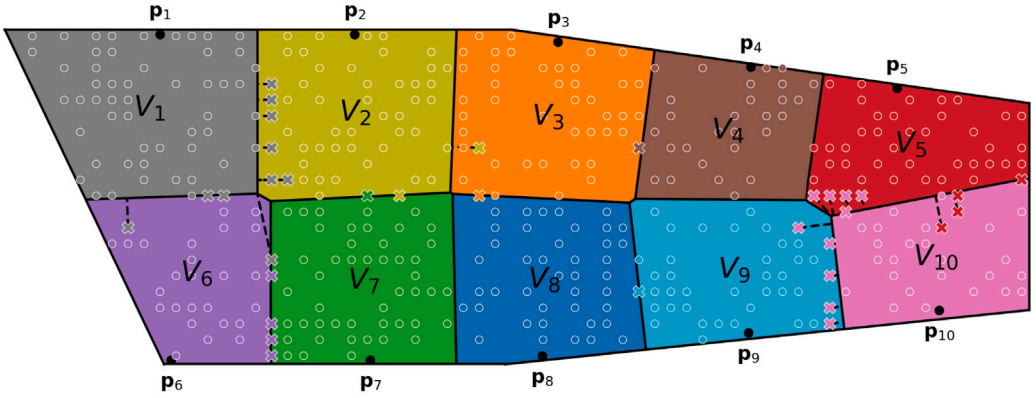
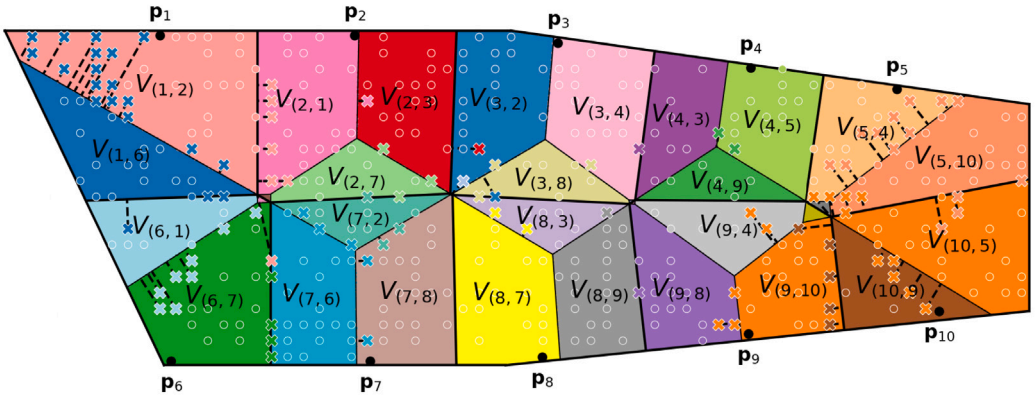


Fig. 8. Results of outlier detection based on the Euclidean distance between the true leak positions and the classic Voronoi cell corresponding to the largest flow. Distances correspond to the lengths of the dashed lines (black for single-leak and red for multi-leak), and a sample is considered an outlier if the distance is greater than two meters.



(a) Classic Voronoi predictor

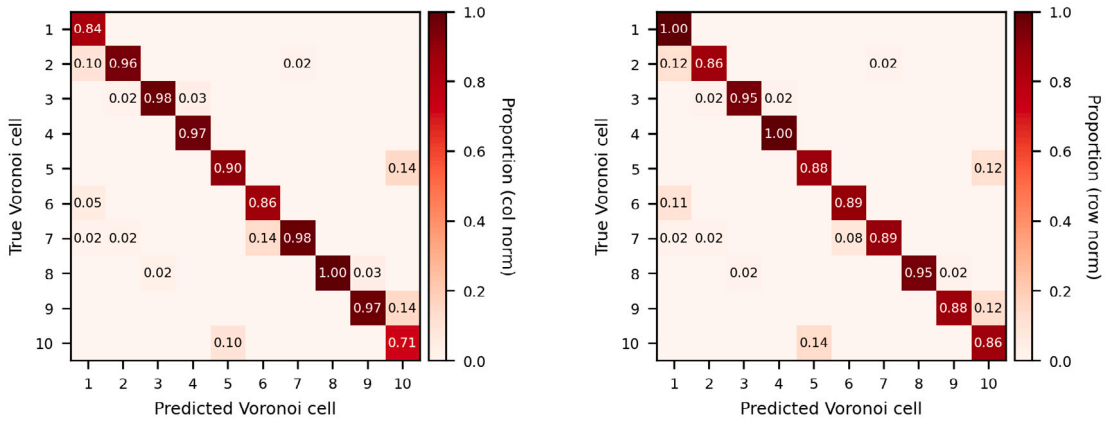


(b) Refined Voronoi predictor

Fig. 9. Evaluation of Voronoi predictors on cleaned single-leak data. Each circle or cross indicates a leak position in the cleaned data. The circles represent correct predictions; that is, the actual leak position is within the Voronoi cell that was predicted based on the recorded flow data. Crosses represent incorrect predictions. Dashed lines connect the actual leak positions (crosses) and their associated projections on the predicted classic Voronoi cells.

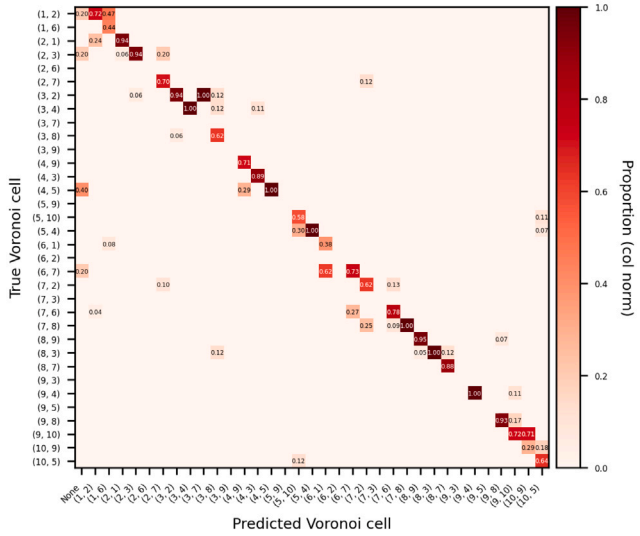
Table 1
Evaluation on single-leak data.

Voronoi type	Dataset	Accuracy (%)	Mean Euc. dist. (full dataset, cm)	Mean Euc. dist. (incorrect only, cm)
Classic	Original	90.56	6.69	70.84
Classic	Cleaned	91.67	1.71	20.58
Refined	Original	73.85	11.53	45.69
Refined	Cleaned	74.75	6.13	25.23

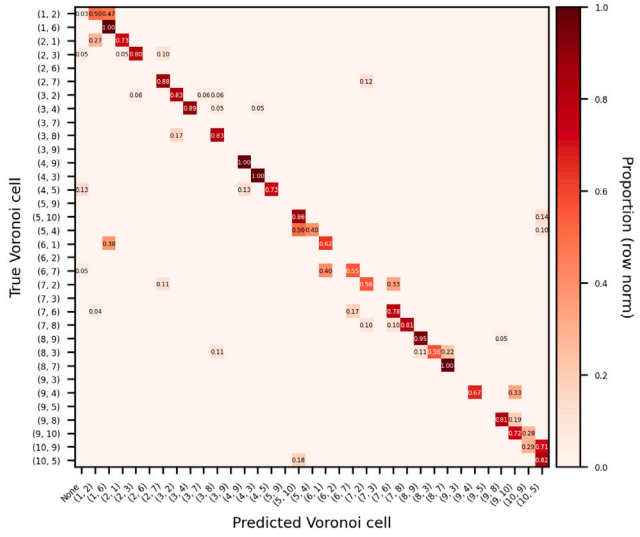


(a) Column-wise normalization (b) Row-wise normalization

Fig. 10. Confusion matrices obtained with the classic Voronoi predictor on single-leak data, based on the frequency with which a predictor misclassifies pairs of classes in both directions, and normalized such that either columns (left) or rows (right) sum up to one.



(a) Column-wise normalization



(b) Row-wise normalization

Fig. 11. Confusion matrices obtained with the refined Voronoi predictor on single-leak data.

and have a relatively short common edge. Consequently, the refined cells corresponding to their direct neighbors comprise most of the original area of V_2 and V_6. On the other hand, in five cases, the refined Voronoi predictor identifies cells that do not exist in the given refined Voronoi diagram, namely $V_{(2,10)}$, $V_{(4,10)}$ (twice), $V_{(6,10)}$, and $V_{(1,4)}$. These predictions, in the following referred to as invalid predictions, are summarized in the *None* category in Fig. 11 (which has a corresponding column but no corresponding row in the confusion matrices because there are no corresponding true Voronoi cells) and illustrated in Fig. 12 (they are not contained in Fig. 9(b)). Notably, all corresponding true leak positions are located very close to one particular vacuum connection.

3.5. Experimental results on multi-leak data

Now, we evaluate the use of classic and refined Voronoi predictors with multi-leak data. By definition, both predictors are designed to detect single leaks. In Section 2.2.3, we introduced the repeated leak localization strategy to expand the capabilities of our predictors for multi-leak detection. The original dataset contains 440 two-leak examples. One example was removed during data cleaning of the two-leak data. Seven more examples were removed because they included leak positions that were removed during the cleaning of the single-leak data. This left 432 examples in the cleaned two-leak dataset. After these steps, each position of a leak in a two-leak example also appears as a single-leak example. This enables us to simulate the repeated leak detection strategy as follows: First, we apply a Voronoi predictor to the flow values induced by the two leaks and evaluate the predictor's ability to identify one of the two leaks. If successful, we feed the flow induced by the remaining leak to the same Voronoi predictor and evaluate the second step.

Table 2 summarizes the results for multi-leak data. Step 1 accuracy is defined as the percentage of samples for which at least one true leak position falls within the predicted cell. The step 1 Euclidean distance is the minimum distance between the two leak positions and the predicted cell, i.e., it is non-zero only if both true leak positions are outside the predicted cell. Both step 2 accuracy and Euclidean distance are computed only among samples for which step 1 is successful (i.e., at least one true leak position is within the predicted cell) and are based on the remaining leak position and its associated flow values. Fig. 13 illustrates the step 1 predictions for both Voronoi predictors. Again, the crosses indicate incorrect predictions, i.e., that none of the two true leak positions in a sample is within the predicted cell in step one. For each incorrectly predicted sample, only the leak position with the smaller distance to the predicted cell is shown.

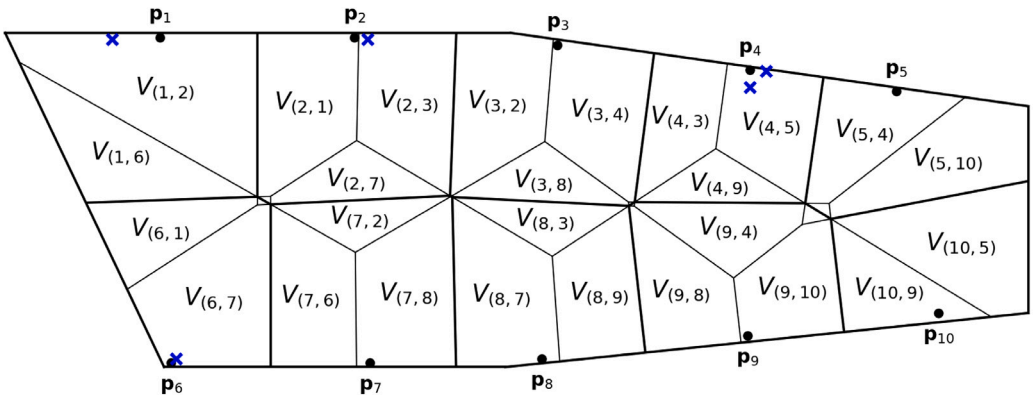
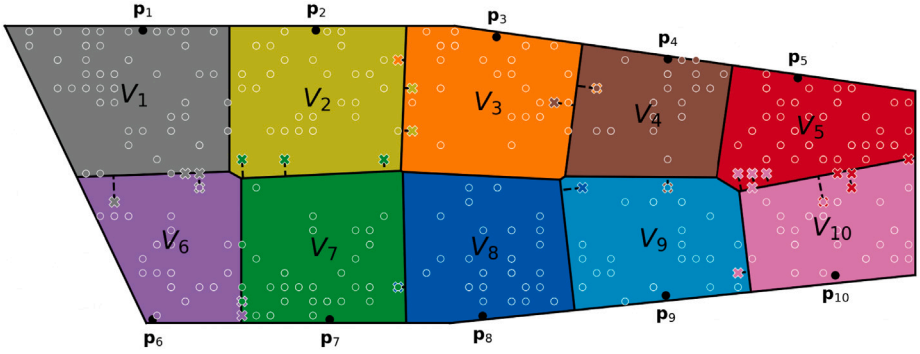


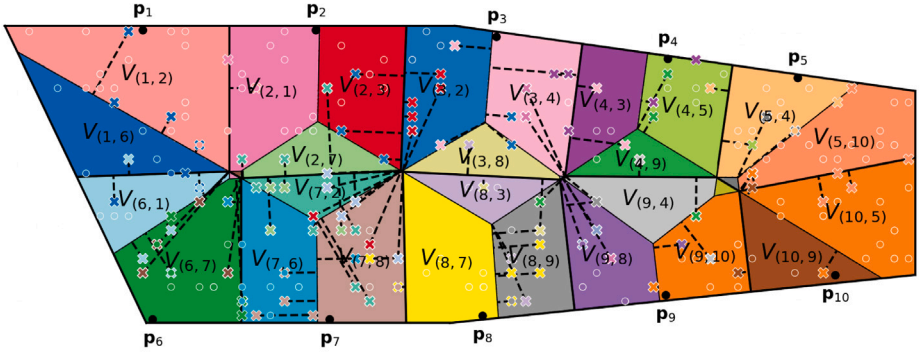
Fig. 12. Invalid predictions of the refined Voronoi predictor on the single-leak data. Each cross represents a case where the refined Voronoi predictor identifies a cell that does not exist in the given refined Voronoi diagram. Corresponding predictions are $V_{(1,4)}$ (top left), $V_{(2,10)}$ (top middle), $V_{(4,10)}$ (top right, 2×), and $V_{(6,10)}$ (bottom).

Table 2
Evaluation on multi-leak data.

Voronoi type	Dataset	Accuracy (%)		Mean Euc. dist. (full dataset, cm)		Mean Euc. dist. (incorrect only, cm)	
		Step 1	Step 2	Step 1	Step 2	Step 1	Step 2
Classic	Original	92.95	87.04	1.97	8.42	27.93	64.99
Classic	Cleaned	93.06	88.31	1.30	2.50	18.76	21.35
Refined	Original	33.86	70.47	28.89	12.54	65.07	43.15
Refined	Cleaned	33.80	71.23	29.20	6.07	65.63	21.45



(a) Classic Voronoi predictor



(b) Refined Voronoi predictor

Fig. 13. Evaluation of Voronoi predictors on cleaned multi-leak data. Each circle or cross indicates a leak position in the cleaned data. The circles represent a correct prediction in step 1 of the repeated leak localization approach, meaning at least one of the two true leak positions in the same sample is within the predicted cell. The crosses represent cases in which neither true leak position is within the predicted cell in step one. Only the leak position with the smaller distance to the predicted cell and its projection onto the predicted cell are shown, connected by a dashed line.

The classic Voronoi predictor achieves 93.06% accuracy in step 1, even surpassing the accuracy on single-leak data. Step 2 accuracy is 88.31%. Similarly, the mean Euclidean distances are of the same magnitude as in the single-leak case. At first glance, Figs. 9(a) and 13(a) appear similar. Upon closer inspection, it becomes apparent that the density of the marked leak positions in Fig. 13(a) is lower toward the horizontal center of the part. The missing leak positions correspond to incorrect step 1 predictions and are those with the greater distance to the predicted cell among the two leak positions in their sample.

The refined Voronoi predictor achieves an accuracy of 33.80% in step 1 and 71.23% in step 2. The significant discrepancy between the accuracy of the refined Voronoi predictor in the single-leak case and the accuracy in step 1 of the two-leak case is primarily due to invalid predictions, i.e., predictions for which the predicted refined Voronoi cell does not exist in the diagram. Of the 432 examples in the cleaned two-leak data, the refined Voronoi predictor produces 146 correct predictions in step 1 (corresponding to the 33.80% accuracy), 169 invalid predictions (39.12%, not included in Fig. 13(b)), and 117 valid but incorrect predictions (27.08%). Since invalid predictions do not correspond to any cell in the refined Voronoi diagram, the respective samples were excluded from the calculation of the reported mean Euclidean distances. Among the valid predictions, the mean Euclidean distances of the refined Voronoi predictor in step 1 are significantly larger than the mean Euclidean distances in the single-leak case. This can also be seen by comparing Figs. 9(b) and 13(b).

Before concluding this section, we investigate the observed invalid predictions further. We hypothesize that these predictions may contain useful information about the spatial distribution of multiple leaks, which the refined Voronoi predictor is currently unable to exploit. This is because the predictor establishes a one-to-one correspondence between the indices of the two vacuum connections with the largest measurements and the corresponding refined Voronoi cell. If a corresponding refined Voronoi cell does not exist, the prediction cannot be used to draw conclusions about leak locations. One possible solution is to relax the aforementioned one-to-one correspondence where appropriate. While developing further prediction models is beyond the scope of this paper, the following analysis aims to provide initial insights and directions for future development.

Fig. 14 shows a histogram of the 50 invalid prediction categories that occur when the refined Voronoi predictor is applied to the cleaned two-leak data. The related Voronoi diagram has 34 cells, which corresponds to the number of valid prediction categories. Since there are 90 possible orderings of two out of ten measurements, only six theoretically possible invalid prediction categories do not occur. These missing categories are $(4,7)$, $(6,3)$, $(6,10)$, $(8,6)$, $(8,10)$, and $(10,3)$.

Fig. 15 shows the spatial distribution of leak locations associated with the three most frequent invalid prediction categories $(1,3)$, $(5,2)$, and $(10,1)$. The leak locations in Fig. 15(a) show that all two-leak samples associated with the prediction $(1,3)$ feature one leak location in the classic Voronoi cell V_1 (except in one case where the leak location is in V_6 but very close to the shared edge of V_1 and V_6) and one leak location in the classic Voronoi cell V_3. Moreover, it can be observed that most leak locations even concentrate in or close to refined Voronoi cells $V_{(1,6)}$ and $V_{(3,4)}$. Note that two of the leak locations in V_3 appear in two samples which is why there are nine locations in V_1 but only seven locations in V_3. Basically, the same statements hold for the leak locations illustrated in Figs. 15(b) and 15(c).

3.6. Search area reduction

We calculate the areas of all classic and refined Voronoi cells using the shoelace formula [19] to quantify the reduction in search area provided by Voronoi diagrams and refined Voronoi diagrams. The largest classic Voronoi cell V_1 has an area of $8.7\,\mathrm{m}^2$, and the smallest classic Voronoi cell V_{10} has an area of $5.79\,\mathrm{m}^2$. The mean area of all classic Voronoi cells is $6.93\,\mathrm{m}^2$, corresponding to an average reduction in search area of 90% relative to the full surface area of $69.3\,\mathrm{m}^2$ (note that there are ten classic Voronoi cells that form an exact partition of the full surface).

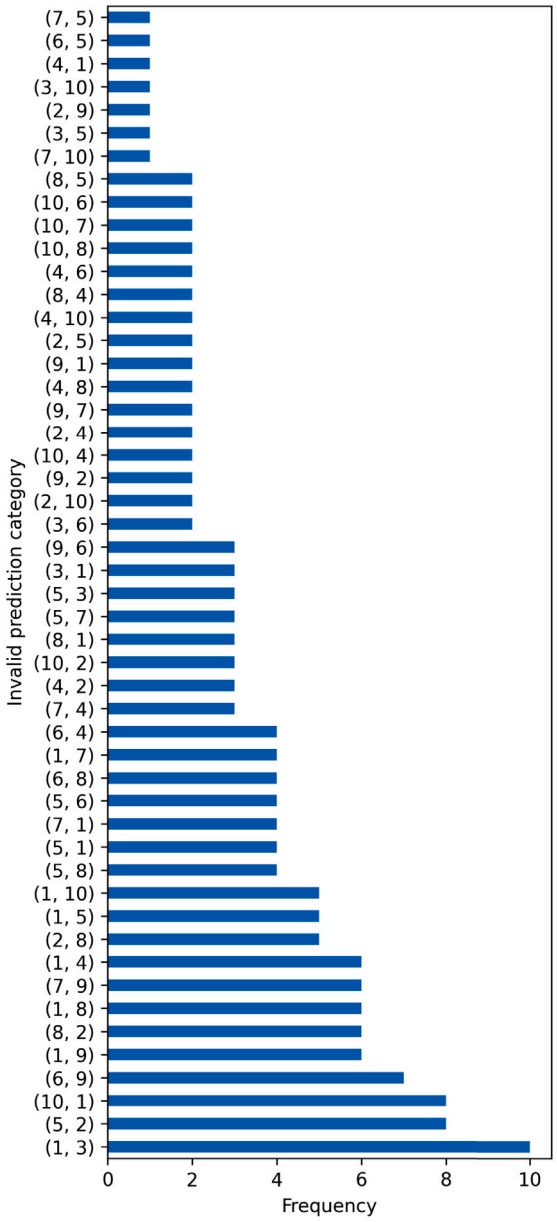
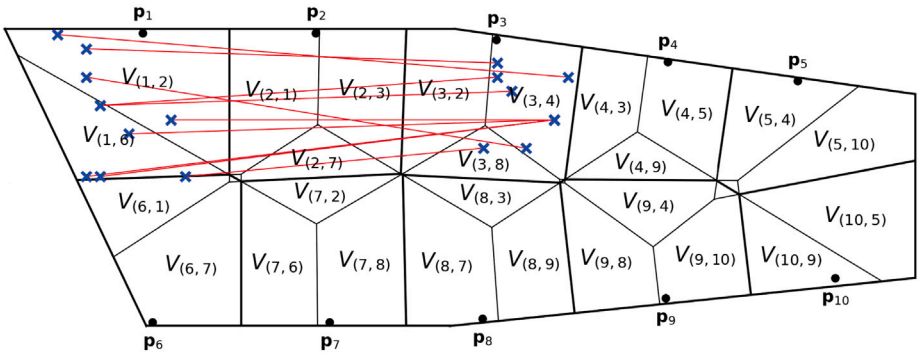


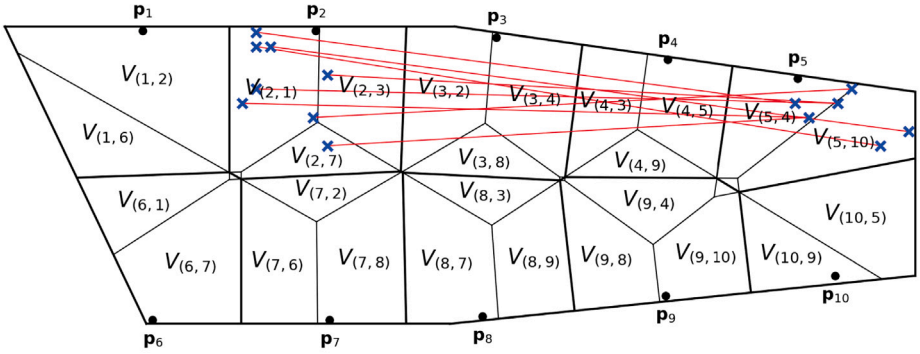
Fig. 14. Histogram of invalid prediction categories for the refined Voronoi predictor.

Among the 34 refined Voronoi cells, the average reduction in search area is $1 - 1/34 = 97.06\%$. The largest refined Voronoi cell $V_{(1,2)}$ has an area of $5.7\,\mathrm{m}^2$, while the smallest refined Voronoi cell $V_{(7,3)}$ has an area of only $0.0003\,\mathrm{m}^2 = 3\,\mathrm{cm}^2$. Cell $V_{(7,3)}$ corresponds to locations $\boldsymbol{p}_7$ and $\boldsymbol{p}_3$ which are only diagonally adjacent. Therefore, the locations closest to $\boldsymbol{p}_7$ and second-closest to $\boldsymbol{p}_3$ are limited to a very small area. All cell areas are given in Table 3.

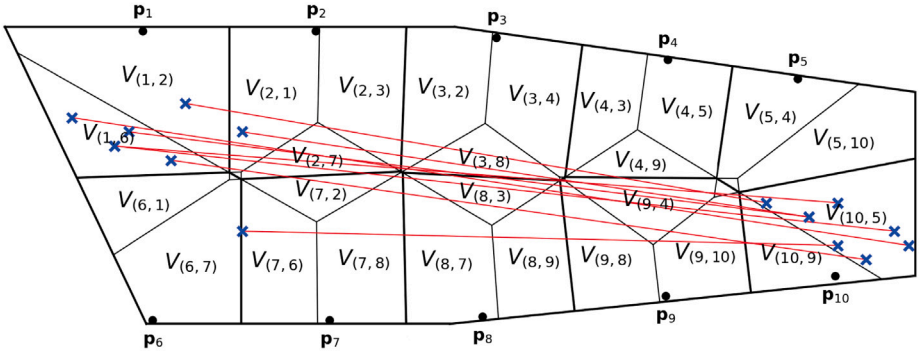
The area reduction in Table 3 is calculated using the formula $(1 - \text{cell area}/\text{full area}) \times 100$, while the speedup factor corresponds to full area/cell area. Note that the reported speedup factors assume that search time is proportional to search area, i.e., that the subsequent leak test has no overhead. However, it is reasonable to assume that there is at least some constant overhead, e.g., for preparation and instrument setup. Although quantifying these technology-dependent overheads is beyond the scope of this work, this should be kept in mind when viewing Table 3.



(a) Invalid prediction (1, 3)



(b) Invalid prediction (5, 2)



(c) Invalid prediction (10, 1)

Fig. 15. Spatial distribution of leak locations for the three most frequent invalid prediction categories of the refined Voronoi predictor.

4. Discussion

Here, we discuss the experimental results reported in Section 3 and their potential implications for practical leak detection. For the sake of brevity, we focus on the main aspects.

Classic Voronoi predictor

Our results indicate that the classic Voronoi predictor reliably performs on both one-leak and two-leak data. In practice, the restriction of leak positions to classic Voronoi cells already facilitates subsequent techniques, such as ultrasonic, helium, or thermographic leak tests a lot. In the event of an incorrect prediction, the true leak position is usually very close to the predicted cell (on average, less than 20 cm). Therefore, most incorrect predictions can be accounted for by introducing a small safety margin to the predicted classic Voronoi cell.

Refined Voronoi predictor

The results obtained with the refined Voronoi predictor require further differentiation. This is particularly because of the significant discrepancy between the results for one-leak and two-leak data.

One-leak data The accuracy is lower than that of the classic Voronoi predictor, which is expected because of the smaller cells. However, the same statement as above holds true: A small security margin around the predicted refined Voronoi cells captures a significant portion of all incorrect predictions as well (the mean distance is around 25 cm). For practical applications, this indicates that a prioritized search within and around a refined Voronoi cell is likely effective before extending the search area to a classic Voronoi cell if necessary.

Two-leak data While the classic Voronoi predictor has similar accuracy in steps one and two as in the single-leak case, this does not hold for the refined Voronoi predictor. The refined Voronoi predictor's step 1 accuracy is only 33.8%, which is less than half of its step 2 accuracy. This suggests that the refined Voronoi predictor has limited practical use in cases involving multiple leaks.

Invalid predictions

The poor accuracy of the refined Voronoi predictor in step 1 of the repeated strategy is largely due to invalid predictions, which account

Table 3
Voronoi cell areas (in m^2), search area reduction (in %), and theoretical search speedup for a full surface area of $69.3176\,m^2$.

Voronoi type	Cell index	Cell area	Area reduction	Speedup factor
Classic	1	8.7038	87.44	7.96
Classic	2	8.0459	88.39	8.62
Classic	3	7.6138	89.02	9.10
Classic	7	7.5336	89.13	9.20
Classic	8	7.2091	89.60	9.62
Classic	9	6.8330	90.14	10.14
Classic	6	6.0068	91.33	11.54
Classic	4	5.7964	91.64	11.96
Classic	5	5.7878	91.65	11.98
Classic	10	5.7874	91.65	11.98
Refined	(1, 2)	5.6956	91.78	12.17
Refined	(6, 7)	4.0111	94.21	17.28
Refined	(10, 5)	3.7197	94.63	18.64
Refined	(7, 8)	3.4684	95.00	19.99
Refined	(8, 7)	3.4374	95.04	20.17
Refined	(2, 1)	3.4313	95.05	20.20
Refined	(2, 3)	3.1992	95.38	21.67
Refined	(5, 10)	3.1889	95.40	21.74
Refined	(3, 2)	3.1366	95.47	22.10
Refined	(3, 4)	3.1180	95.50	22.23
Refined	(1, 6)	3.0082	95.66	23.04
Refined	(7, 6)	2.9048	95.81	23.86
Refined	(8, 9)	2.5966	96.25	26.70
Refined	(5, 4)	2.5535	96.32	27.15
Refined	(9, 8)	2.5530	96.32	27.15
Refined	(4, 5)	2.5289	96.35	27.41
Refined	(9, 10)	2.4176	96.51	28.67
Refined	(4, 3)	2.1383	96.92	32.42
Refined	(10, 9)	2.0677	97.02	33.52
Refined	(6, 1)	1.9822	97.14	34.97
Refined	(9, 4)	1.7871	97.42	38.79
Refined	(2, 7)	1.4021	97.98	49.44
Refined	(3, 8)	1.3559	98.04	51.12
Refined	(8, 3)	1.1751	98.30	58.99
Refined	(7, 2)	1.1601	98.33	59.75
Refined	(4, 9)	1.1291	98.37	61.39
Refined	(9, 5)	0.0727	99.90	953.47
Refined	(5, 9)	0.0454	99.93	1526.82
Refined	(6, 2)	0.0135	99.98	5134.64
Refined	(2, 6)	0.0132	99.98	5251.33
Refined	(3, 9)	0.0029	100.00	23902.62
Refined	(9, 3)	0.0027	100.00	25673.19
Refined	(3, 7)	0.0004	100.00	173294.00
Refined	(7, 3)	0.0003	100.00	231058.67

for 169 out of 432 samples. However, our investigation of the most frequently observed invalid prediction categories suggests that these predictions contain useful information that the refined Voronoi predictor does not exploit. First, our results indicate that invalid predictions are a relatively robust indicator of the presence of multiple leaks. Specifically, only five invalid predictions were observed in the one-leak case, versus 169 invalid predictions in the two-leak case. Furthermore, in the two-leak case, it is often even possible to correctly predict the classic Voronoi cells containing the leaks in a single step. For 144 of the 169 invalid predictions (85.21%), the predicted tuple (i_1, i_2) points to classic Voronoi cells V_{i_1} and V_{i_2} which both contain a leak. For additional 20 of the 169 invalid predictions (11.83%), only V_{i_1} contains a leak. And in 5 of 169 cases (2.96%), only V_{i_2} contains a leak. This procedure corresponds to the simultaneous search strategy briefly discussed above. As also mentioned above, the simultaneous strategy is not applicable when the number of leaks is unknown. However, one important takeaway for future methods is that the output of the refined Voronoi predictor – namely, the order of measurements – is a valuable feature to take into account.

We summarize our discussion as follows:

- The classic Voronoi predictor is highly accurate and robust, even in multi-leak scenarios. It reduces the search area by a factor inversely proportional to the number of vacuum connections.
- The refined Voronoi predictor is not sufficiently accurate to be directly useful in multi-leak scenarios. However, its predictions reliably indicate the presence of multiple leaks and constitute a valuable feature to be exploited by future methods.

5. Conclusion

In this paper, we investigated the use of Voronoi diagrams for leakage detection in vacuum setups for composite manufacturing. In addition to the classic Voronoi diagram, which is based on the positions of individual vacuum connections, we used a refined version that considers the order of distances from a point to multiple connections. We derived knowledge-based predictors from both variants that can detect single or multiple leaks with a suitable repeated search strategy. These predictors were evaluated and compared using one- and two-leak scenarios generated on an industrial-scale wing cover mold.

The classic Voronoi predictor reliably narrows down the search area for a leak from the entire mold surface to a single Voronoi cell. While the refined Voronoi predictor is less precise as a standalone predictor, our analysis revealed that the underlying order of flow measurements contains important information that can enable multi-leak detection and enhance predictive models in future work. One potential enhancement is the use of data-based strategies. First, such strategies can adapt Voronoi diagrams to experimental data, i.e., shift nodes and edges slightly to increase accuracy. Second, classification models based on flow data and Voronoi diagrams can likely improve upon the knowledge-driven predictors presented in this work, especially when the number of leaks is unknown.

In this study, we examined a single mold geometry, a single vacuum connection layout, and a single leak size caused by a fixed needle with a constant inner diameter. Although the proposed methodology can be applied to any geometry and connection setup, it has yet to be validated in variable setups. Previous work has found that airflow depends linearly on needle diameter. However, investigating the immediate impact of variable needle diameter on the predictive accuracy of our methods is an open question for future work. Finally, an important industrial use case and future direction is leakage detection on potentially curved surfaces that incorporate flow barriers or additional flow aids below the vacuum bag. Both prevent isotropic flow. Examples of this include setups with stringers or other reinforcing elements, additional hoses, and areas without breather material. We consider this work to be the foundation for the aforementioned future directions that are the subject of current activities or will be part of future research.

CRediT authorship contribution statement

Christoph Brauer: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Arne Hindersmann:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Conceptualization. **Timo de Wolff:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

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.

Appendix A. Discrete geometry

We review some basic concepts and notations from discrete geometry. For further details, see introductory sources such as [20,21].

Let A be an arbitrary subset of $\mathbb{R}^n$. The convex hull of A is given as

$$\operatorname{conv}(A) = \left\{ \sum_{i=1}^{k} \lambda_i \boldsymbol{a}_i \;\middle|\; \boldsymbol{a}_i \in A,\ \lambda_i \geq 0,\ \sum_{i=1}^{k} \lambda_i = 1,\ k \in \mathbb{N} \right\}.$$

Moreover, A is called convex if $\operatorname{conv}(A) = A$. Geometrically, A is convex if and only if it holds for every line segment connecting two points $\boldsymbol{a}, \boldsymbol{b} \in A$ that the entire line segment $[\boldsymbol{a}, \boldsymbol{b}]$ is also contained in A.

A dual approach for constructing convex sets is given by the intersection of half spaces. An $(n-1)$-dimensional subspace H of $\mathbb{R}^n$ is called an (affine) hyperplane. H is uniquely determined by a linear form, i.e.,

$$H = \{\boldsymbol{x} \in \mathbb{R}^n \mid \langle \boldsymbol{a}, \boldsymbol{x} \rangle = b\}$$

for some $\boldsymbol{a} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ and $b \in \mathbb{R}$. The vector $\boldsymbol{a} \in \mathbb{R}^n$ in this representation is unique up to nonzero multiples. In fact, $\boldsymbol{a}$ is a normal vector to the hyperplane H (and hence to all hyperplanes parallel to H).

A hyperplane H defines a positive and a negative halfspace

$$H^+ = \{\boldsymbol{x} \in \mathbb{R}^n \mid \langle \boldsymbol{a}, \boldsymbol{x} \rangle \geq b\} \quad \text{and}$$
$$H^- = \{\boldsymbol{x} \in \mathbb{R}^n \mid \langle \boldsymbol{a}, \boldsymbol{x} \rangle \leq b\}.$$

For a given convex set A, H is called a supporting hyperplane if $A \cap H \neq \emptyset$ and $A \subseteq H^+$.

Every intersection of (possibly infinitely many) positive halfspaces yields a convex set. And, vice versa, every convex set is given by the intersection of the positive half-spaces of all its supporting hyperplanes. Moreover, if a convex set is representable as the intersection of finitely many halfspaces, it is called a polyhedron, and if it is moreover bounded it is called a polytope.

Example 7. Consider the three-dimensional prism over a triangle shown in Fig. 16. The volume illustrated is defined by the inequalities $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$, $x_3 \leq 1$ and $x_1 + x_2 \leq 1$. Each inequality can be represented in terms of a positive halfspace. For example,

$$\left\{ \boldsymbol{x} \in \mathbb{R}^3 \;\middle|\; x_1 \geq 0 \right\} = \left\{ \boldsymbol{x} \in \mathbb{R}^3 \;\middle|\; \left\langle \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{x} \right\rangle \geq 0 \right\}.$$

The prism is, by definition, the set of points that satisfy all five inequalities or, in other words, the intersection of the five corresponding positive halfspaces.

Polyhedra carry a significant amount of combinatorial properties and data. For example, let $P \subset \mathbb{R}^n$ be a polyhedron. A subset F of P is called a face of P if there exists a supporting hyperplane H such that $F = P \cap H$. One can show that every face of a polyhedron is a polyhedron itself. The dimension of F, short $\dim(F)$ is given as the dimension of the smallest affine subspace of $\mathbb{R}^n$ that contains F. If P itself is of dimension n, then faces of dimension $n-1$ are called facets, faces of dimension 1 are called edges, and faces of dimension 0 are called vertices.

In several contexts, it is useful to consider assembles of polyhedra, which intersect in a structure preserving way. Let us make this precise. Let $P_1, \ldots, P_k \subset \mathbb{R}^n$ be a collection of polyhedra. Then the collection $C = \{P_1, \ldots, P_k\}$ is called a polyhedral complex if it satisfies the following conditions:

1. $\emptyset \in C$.
2. If $P_i \in C$, then all faces of P_i are also contained in C.
3. If $P_i, P_j \in C$, then $P_i \cap P_j$ is a face of both P_i and P_j.

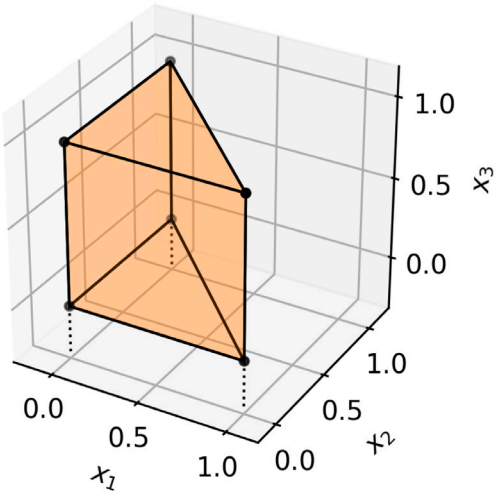


Fig. 16. Three-dimensional prism over a triangle.

Appendix B. Classic Voronoi diagrams

This section of the appendix provides additional information about classic Voronoi diagrams [9]. Specifically, we informally discuss an algorithm for constructing classic Voronoi diagrams of arbitrary point sets. First, recall Definition 1 from above:

Definition (*Recall of Definition 1*)**.** Let $\mathcal{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k\} \subset \mathbb{R}^2$ be a finite set of points. For each $i \in [k]$, the corresponding classic Voronoi cell is defined as

$$V_i = \left\{ \boldsymbol{y} \in \mathbb{R}^2 \;\middle|\; \|\boldsymbol{y} - \boldsymbol{p}_i\| \leq \|\boldsymbol{y} - \boldsymbol{p}_j\| \text{ for all } j \neq i \right\}.$$

The collection

$$\boldsymbol{V}(\mathcal{P}) = \{V_1, \ldots, V_k\}$$

is called the classic Voronoi diagram of $\mathcal{P}$.

Proposition 8. *Let $\mathcal{P} \subset \mathbb{R}^2$ be a finite point set. Its classic Voronoi diagram is a polyhedral complex. Moreover, it can be computed in $O(|\mathcal{P}| \log |\mathcal{P}|)$ time. [22, Chapter 7]*

There are different algorithms for computing classic Voronoi diagrams. Below, we briefly review an algorithm that uses the Delaunay triangulation, which is dual to the classic Voronoi diagram of the same set of points. More precisely, the classic Voronoi diagram and Delaunay triangulation graphs of the same points are dual to each other. As a consequence, given the Delaunay triangulation of $\mathcal{P}$, the corresponding classic Voronoi diagram can be derived straightforwardly. The graph representation of a classic Voronoi diagram consists of the edges and vertices of the classic Voronoi cells. The Delaunay triangulation graph can be obtained by connecting any two points in $\mathcal{P}$ with a straight line if they have the following property: There exists a circle that passes through both points and contains no other points from $\mathcal{P}$. Therefore, the points in $\mathcal{P}$ are the vertices of the Delaunay triangulation graph, and the edges are the connecting lines just defined.

Example 9. We consider the classic Voronoi diagram from Example 2 again. Fig. 17 shows the same classic Voronoi diagram (black) together with the corresponding Delaunay triangulation (orange).

Before explaining how to recover a classic Voronoi diagram from the associated Delaunay triangulation, we describe a common algorithm for computing the Delaunay triangulation of a two-dimensional point collection. Two dimensions are sufficient for our use case. However,

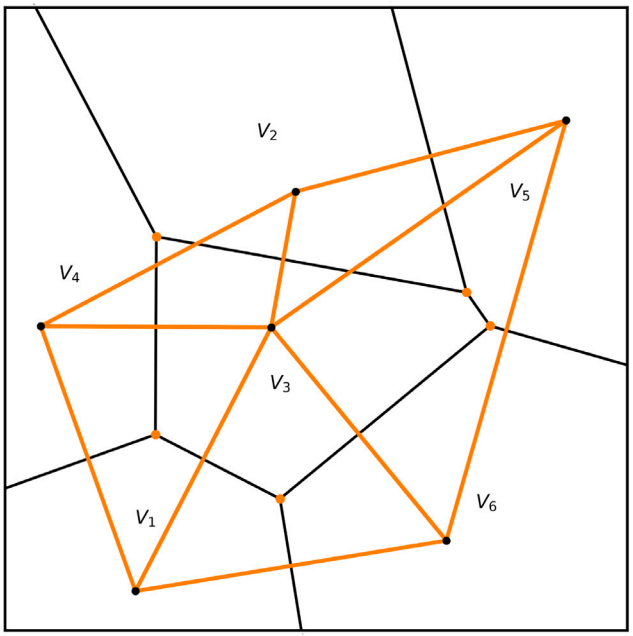


Fig. 17. Classic Voronoi diagram and Delaunay triangulation of a random set of points.

the same algorithm can be extended to higher dimensions in principle. To ensure a unique Delaunay triangulation, we must assume that $\mathcal{P}$ is in general position, i.e., that no three points in $\mathcal{P}$ are co-linear and no four points in $\mathcal{P}$ are co-circular.

To compute the Delaunay triangulation, we first transform all points in $\mathcal{P}$ into three dimensions, i.e., for each point $\boldsymbol{p} = (p_1, p_2)$, we compute $\boldsymbol{p}' = (p_1, p_2, p_1^2 + p_2^2)$. Next, we compute the convex hull $\mathrm{conv}(\mathcal{P}')$ of the associated three-dimensional point collection $\mathcal{P}'$. Given our assumption that $\mathcal{P}$ is in general position, all facets of $\mathrm{conv}(\mathcal{P}')$ are triangles. Now, for each facet of the convex hull, we consider an outer surface normal vector. That is, a vector $\boldsymbol{n} \in \mathbb{R}^3$ that is perpendicular to the face, points outward from the convex hull and is unique up to scaling. If $\boldsymbol{n}$ points downward, i.e., if n_3 is negative, then the face corresponds to a triangle in the Delaunay triangulation which we obtain by projecting the three vertices of the triangle back to two dimensions. In other words, we map the vertices, which are points in $\mathcal{P}'$, back to the corresponding points in $\mathcal{P}$.

The final step is to deduce the associated classic Voronoi diagram from the Delaunay triangulation. First, the circumcenters of the Delaunay triangles are exactly the vertices of the classic Voronoi diagram. Fig. 17 illustrates the circumcenters as orange points. Note that depending on the type of triangle, the circumcenter may also lie outside the triangle. To construct the edges of the classic Voronoi diagram, we distinguish between interior and boundary edges of the Delaunay triangulation. Interior edges are shared by two adjacent triangles, while boundary edges belong to only one triangle. Connecting the circumcenters of all triangles that share an interior edge yields all finite classic Voronoi edges (see the lines connecting the orange points in Fig. 17). Additionally, each boundary edge of the Delaunay triangulation corresponds to an infinite classic Voronoi edge (a ray, to be exact). Each infinite edge starts at the circumcenter of the respective Delaunay triangle and points outward of the Delaunay triangulation perpendicular to the respective boundary edge. This completes the computation of the classic Voronoi diagram graph.

As Proposition 8 states, classic Voronoi diagrams are polyhedral complexes. In other words, each classic Voronoi cell V_i is a polyhedron that can be represented as the intersection of a finite number of positive halfspaces. We need this halfspace representation for our numerical implementation, so we briefly discuss how to deduce it from the Delaunay triangulation as well. We consider an arbitrary point $\boldsymbol{p}_i \in \mathcal{P}$, which is by definition also a vertex of the Delaunay triangulation. Hence, $\boldsymbol{p}_i$ is contained within a number of Delaunay edges and, as just discussed, each Delaunay edge containing $\boldsymbol{p}_i$ corresponds to an edge of the classic Voronoi cell V_i. Consider any point $\boldsymbol{p}_j \in \mathcal{P}$ connected to $\boldsymbol{p}_i$ by a Delaunay edge. The bisector of $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ is the set of points equidistant from $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. Formally, we represent the bisector as $\{\boldsymbol{y} \in \mathbb{R}^2 \mid \|\boldsymbol{y} - \boldsymbol{p}_i\| = \|\boldsymbol{y} - \boldsymbol{p}_j\|\}$. At the same time, the bisector is a hyperplane perpendicular to the line connecting $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. In other words, the vector $\boldsymbol{p}_i - \boldsymbol{p}_j$ is a normal vector of the bisector and the bisector has a hyperplane representation $\{\boldsymbol{y} \in \mathbb{R}^2 \mid \langle \boldsymbol{p}_i - \boldsymbol{p}_j, \boldsymbol{y}\rangle = b\}$. The classic Voronoi cell V_i can be represented as the intersection of positive halfspaces corresponding to bisectors of $\boldsymbol{p}_i$ and adjacent points in the Delaunay triangulation. We define

$$\boldsymbol{a}_{ij} = \frac{\boldsymbol{p}_i - \boldsymbol{p}_j}{\|\boldsymbol{p}_i - \boldsymbol{p}_j\|} \quad \text{and} \quad b_{ij} = \frac{\langle \boldsymbol{a}_{ij}, \boldsymbol{p}_i + \boldsymbol{p}_j\rangle}{2}$$

and therewith

$$H_{ij}^+ = \left\{ \boldsymbol{y} \in \mathbb{R}^2 \;\middle|\; \langle \boldsymbol{a}_{ij}, \boldsymbol{y}\rangle \geq b_{ij} \right\} .$$

The positive halfspace H_{ij}^+ originates from the bisector of $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ and covers the side of the bisector on which $\boldsymbol{p}_i$ lies. The only deviation from the above motivation is that the normal vector $\boldsymbol{a}_{ij}$ now has norm one, and of course the order of i and j matters. Finally, let $\mathcal{N}_i$ denote the index set such that $j \in \mathcal{N}_i$ if and only if there exists a Delaunay edge connecting $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, then

$$V_i = \bigcap_{j \in \mathcal{N}_i} H_{ij}^+$$

is the aimed representation of V_i as an intersection of positive halfspaces.

Appendix C. Refined Voronoi diagrams

In the context of our leakage localization problem, the points determining the Voronoi cells are the locations of the vacuum connections. The idea behind Voronoi predictors is that a leak should be closest to the particular sensor that captured the largest flow. This concept can be refined in case there are more than two sensors. Namely, it is likewise natural to assume that a leak should be in second closest proximity to the sensor that captured the second largest flow, and so on. This idea can be formalized using ordered order-k Voronoi diagrams [16], which we refer to as refined Voronoi diagrams in this work.

Definition 10. Let $k > 2$ and $d \leq k$. Further, let

$$T_d = \left\{ \boldsymbol{t} = (i_1, \ldots, i_d) \mid i_\alpha \in [k] \text{ and } i_\alpha \neq i_\beta \text{ for } \alpha \neq \beta \right\}$$

be the set of length d tuples from $[k]$ without repetition. For any $\boldsymbol{t} \in T_d$, let

$$S(\boldsymbol{t}) = \left\{ i_1, \ldots, i_d \right\} \quad \text{and} \quad |\boldsymbol{t}| = |S(\boldsymbol{t})|$$

be the set of elements in $\boldsymbol{t}$ and its cardinality. Then, we define the corresponding refined Voronoi cell as

$$V_{\boldsymbol{t}} = \left\{ \boldsymbol{y} \in \mathbb{R}^2 \;\middle|\; \|\boldsymbol{y} - \boldsymbol{p}_{i_1}\| \leq \cdots \leq \|\boldsymbol{y} - \boldsymbol{p}_{i_{|\boldsymbol{t}|}}\| \leq \|\boldsymbol{y} - \boldsymbol{p}_j\| \text{ for all } j \in [k] \setminus S(\boldsymbol{t}) \right\} .$$

We call the collection of all refined Voronoi cells based on d-tuples the refined Voronoi diagram of order d of $\mathcal{P}$ and denote it by $\boldsymbol{V}_d(\mathcal{P}) = (V_{\boldsymbol{t}})_{\boldsymbol{t} \in T_d}$.

Example 11. $V_{\boldsymbol{t}}$ is the region of points that are closest to $\boldsymbol{p}_{i_1}$, second closest to $\boldsymbol{p}_{i_2}$, and so on. Note that $\boldsymbol{V}_1(\mathcal{P}) = (V_{(1)}, \ldots, V_{(k)})$ is exactly the classic Voronoi diagram $\boldsymbol{V}(\mathcal{P}) = (V_1, \ldots, V_k)$. The refined Voronoi diagram of order two $\boldsymbol{V}_2(\mathcal{P})$ corresponding to the classic Voronoi diagram in Fig. 17 is illustrated in Fig. 2. The case $d = k$ corresponds to a maximal refinement of Voronoi regions by means of k-tuples $\boldsymbol{t} \in T_k$.

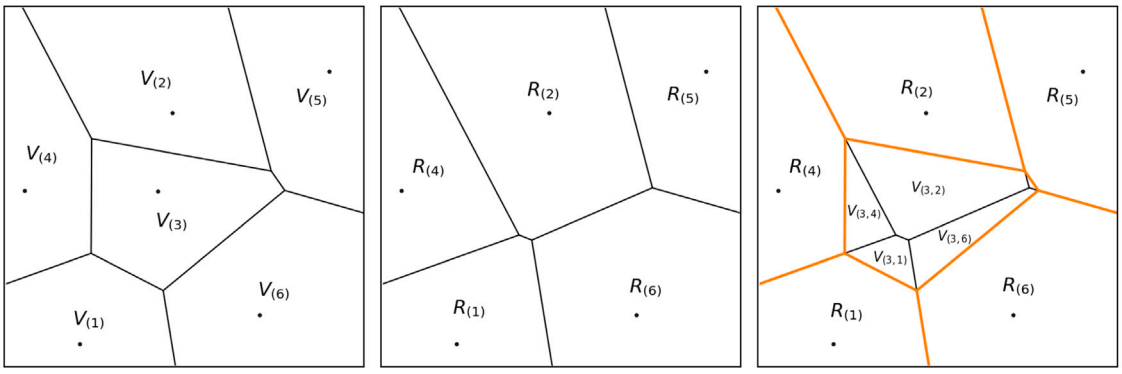


Fig. 18. Subsequent steps of Algorithm 1 applied to a random example.

Algorithm 1 Computation of $(V_t)_{t\in T_{d+1}}$ given $(V_t)_{t\in T_d}$

for all $(i_1,\dots,i_d)\in T_d$ **do**
  Compute Voronoi diagram $(R_i)_{i\in[k]\setminus\{i_1,\dots,i_d\}}$
  for all $j\in[k]\setminus\{i_1,\dots,i_d\}$ **do**
    $V_{(i_1,\dots,i_d,j)} = V_{(i_1,\dots,i_d)}\cap R_j$
  end for
end for

In what follows, we assume that an algorithm to compute classic Voronoi diagrams is available. Based on that, we derive a scheme that can be used to compute refined Voronoi diagrams recursively. The proposed scheme is outlined in Algorithm 1. Assuming that $\boldsymbol{V}_d(\mathcal{P})$ for $d\le k-1$ is given as input, the refined Voronoi diagram of subsequent order $\boldsymbol{V}_{d+1}(\mathcal{P})$ results as output.

Each tuple $(i_1,\dots,i_d)$ is visited during the outer for loop, and the inner for loop then provides all refined Voronoi cells for tuples $(i_1,\dots,i_d,j)$ with $j\in[K]\setminus\{i_1,\dots,i_d\}$. To that end, a standard Voronoi diagram with reduced point set

$$\boldsymbol{V}\left(\mathcal{P}\setminus\left\{\boldsymbol{p}_{i_1},\dots,\boldsymbol{p}_{i_d}\right\}\right) = \left(R_i\right)_{i\in[k]\setminus\{i_1,\dots,i_d\}}$$

is computed first. Each new Voronoi region of subsequent order is then obtained by taking the intersection

$$V_{(i_1,\dots,i_d,j)} = V_{(i_1,\dots,i_d)}\cap R_j\ .$$

The interpretation of that step is as follows: We have previously computed the cell $V_{(i_1,\dots,i_d)}$. All points in that cell are closest to $\boldsymbol{p}_{i_1}$, second closest $\boldsymbol{p}_{i_2}$, and so on. In the current step, we are looking for the $k-d$ subsets of $V_{(i_1,\dots,i_d)}$ that are $d+1$-st closest to $\boldsymbol{p}_j$ for $j\in[k]\setminus\{i_1,\dots,i_d\}$. These are obtained through the intersections with R_j.

Example 12. Fig. 18 illustrates the transition from $d=1$ to $d=2$ using the example from Fig. 2. The cell $V_{(3)}$ is subdivided into $V_{(3,1)}$, $V_{(3,2)}$, $V_{(3,4)}$, $V_{(3,5)}$ and $V_{(3,6)}$.

As discussed above, classic Voronoi diagrams are polyhedral complexes, and their cells can be represented as the intersection of a finite number of positive halfspaces. These statements also hold for refined Voronoi diagrams. To understand how a refined Voronoi cell can be represented as the intersection of halfspaces in the case of $d=2$, consider the definition according to the inner loop of Algorithm 1:

$$V_{(i,j)} = V_i\cap R_j$$

In other words, $V_{(i,j)}$ is the intersection of V_i and R_j, both of which are classic Voronoi cells, the latter only with respect to a reduced point set. Let $\mathcal{N}_j'$ include the indices of the neighbors of $\boldsymbol{p}_j$ in the Delaunay triangulation of the reduced point set. Then, analogous to above, the two cells can be represented as

$$V_i = \bigcap_{\mu\in\mathcal{N}_i} H_{i\mu}^+ \quad\text{and}\quad R_j = \bigcap_{\nu\in\mathcal{N}_j'} H_{j\nu}^+$$

and hence,

$$V_{(i,j)} = \left(\bigcap_{\mu\in\mathcal{N}_i} H_{i\mu}^+\right)\bigcap\left(\bigcap_{\nu\in\mathcal{N}_j'} H_{j\nu}^+\right)\ .$$

Thus, the set of halfspaces to represent $V_{(i,j)}$ is simply the union of the respective sets for V_i and R_j.

Data availability

Research Link Provided

Upper Wing Cover Leakage Data (Original data) (GitHub)

References

[1] Haschenburger A. Influence and detection of vacuum bag leakages in composites manufacturing (Ph.D. thesis), Delft University of Technology; 2022, http://dx.doi.org/10.4233/uuid:03637286-f682-4602-9890-2ac1c0102599, URL https://elib.dlr.de/190099/.
[2] Haschenburger A, Heim C. Two-stage leak detection in vacuum bags for the production of fibre-reinforced composite components. CEAS Aeronaut J 2019;10(3):885–92.
[3] Haschenburger A, Menke N, Stüve J. Sensor-based leakage detection in vacuum bagging. Int J Adv Manuf Technol 2021;116(7):2413–24.
[4] Ucan H, Bölke J, Krombholz C, Gobbi H, Meyer M. Robotergestützte leckageerkennung an vakuumaufbauten mittels thermografie. In: Deutscher luft- und raumfahrtkongress. 2011.
[5] Brauer C, Lorenz D, Tondji L. Group equivariant networks for leakage detection in vacuum bagging. In: European signal processing conference. IEEE; 2022, p. 1437–41.
[6] Haschenburger A, Onorato L, Sujahudeen MS, Taraczky DS, Osis A, Bracke ARS, Byelov MD, Vermeulen FI, Oosthoek EHQ. Computational methods for leakage localisation in a vacuum bag using volumetric flow rate measurements. Prod Eng 2022;16(6):823–35.
[7] Esmaeili YR, Cosco B, Pan Y, Najjaran H. Intelligent vacuum bagging leakage location prediction. Int J Adv Manuf Technol 2025;136(5):2877–91.
[8] Naveenachandran SV. Data-based leakage detection and uncertainty quantification in the manufacturing of large-scale CFRP components [Master's thesis], TU Braunschweig; 2023.
[9] Aurenhammer F, Klein R, Lee D-T. Voronoi diagrams and delaunay triangulations. World Scientific; 2013.
[10] Brassel KE, Reif D. A procedure to generate thiessen polygons. Geogr Anal 1979;11(3):289–303.
[11] Indermitte C, Liebling TM, Troyanov M, Clémençon H. Voronoi diagrams on piecewise flat surfaces and an application to biological growth. Theoret Comput Sci 2001;263(1–2):263–74.
[12] Blatov VA. Voronoi–Dirichlet polyhedra in crystal chemistry: Theory and applications. Crystallogr Rev 2004;10(4):249–318.
[13] Shamos MI, Hoey D. Closest-point problems. In: Proceedings of the 16th annual symposium on foundations of computer science. IEEE; 1975, p. 151–62.
[14] Chazelle B, Edelsbrunner H. An improved algorithm for constructing k-th order voronoi diagrams. In: Proceedings of the first annual symposium on computational geometry. 1985, p. 228–34.
[15] Chan TM, Cheng P, Zheng DW. An optimal algorithm for higher-order voronoi diagrams in the plane: The usefulness of nondeterminism. In: Proceedings of the annual ACM-SIAM symposium on discrete algorithms. SIAM; 2024, p. 4451–63.

[16] Okabe A, Boots B, Sugihara K, Chiu SN. Spatial tessellations: Concepts and applications of voronoi diagrams, 2nd ed.. Wiley series in probability and statistics, John Wiley & Sons; 2000.
[17] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016.
[18] Boyle JP, Dykstra RL. A method for finding projections onto the intersection of convex sets in Hilbert spaces. In: Advances in order restricted statistical inference. Springer; 1986, p. 28–47.
[19] O'Rourke J. Computational geometry in C. 2nd ed.. Cambridge University Press; 1998.
[20] Grünbaum B, Klee V, Perles MA, Shephard GC. Convex polytopes, vol. 16, Springer; 1967.
[21] Ziegler GM. Lectures on polytopes. New York: Springer; 2007.
[22] de Berg M, Cheong O, van Kreveld M, Overmars M. Computational geometry: Algorithms and applications. 3rd ed.. Springer; 2008.



Dr. Christoph Brauer is a staff scientist at the German Aerospace Center (DLR) in Stade, Germany, where he serves as deputy head of the Production Technologies department at the Institute of Lightweight Systems. His research interests include mathematical optimization, machine learning, and image and signal processing, with applications in lightweight engineering. He received his doctorate from TU Braunschweig in 2018 under the supervision of Dirk Lorenz. He is also a lecturer at TU Braunschweig, an external member of the Applied Algebra Group at the Institute of Analysis and Algebra at TU Braunschweig, and an external member of the Inverse Problems and Imaging Group at the Center for Industrial Mathematics at the University of Bremen.



Dr. Arne Hindersmann is a staff scientist at the German Aerospace Center (DLR) in Stade, Germany. His research interest focuses on the manufacture and production of fiber-reinforced plastics for use in the aerospace industry and in the manufacture of wind turbine blades. He received his doctorate from TU Braunschweig in 2016 under the supervision of Martin Wiedemann.



Prof. Dr. Timo de Wolff leads the Applied Algebra Group at the Institute of Analysis and Algebra at TU Braunschweig. His main research interests are nonnegativity and polynomial optimization. He received his doctorate from Goethe University Frankfurt in 2013 under the supervision of Thorsten Theobald. Before establishing the Applied Algebra Group in Braunschweig, he held positions at Technische Universität Berlin, Texas A&M University, Saarland University, and Goethe University Frankfurt. He is an alumnus of the DFG Emmy Noether Programme (2017–2023) and Die Junge Akademie, the German Young Academy (2019–2024), where he served as chair in 2022/23.