

Research paper

Vision incorporated MULTichannel Feature Fusion Template Matching (MUFF-TM) and real-time sub-pixel coordinate localization for 2D textile surface in ultrasonic tacking systems

Bimsara Sandaruwan Kahandawala^{a,c,†}, Rizma Nalmi^{d,†}, Buddhika Amila Kumara Sodige^e, Kasun Subasinghage^e, Bhagya Nathali Silva^{a,c}, Ruchire Eranga Wijesinghe^{b,c}, Seong Tak Woo^{f,*}

^a Department of Information Technology, Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe 10115, Sri Lanka

^b Department of Electrical and Electronic Engineering, Faculty of Engineering, Sri Lanka Institute of Information Technology, Malabe 10115, Sri Lanka

^c Center for Excellence in Informatics, Electronics & Transmission (CIET), Sri Lanka Institute of Information Technology, Malabe 10115, Sri Lanka

^d Department of Textile and Apparel Engineering, Faculty of Engineering, University of Moratuwa, Sri Lanka

^e Department of Materials and Mechanical Technology, Faculty of Technology, University of Sri Jayewardenepura, Homagama 10206, Sri Lanka

^f Department of Electronic Engineering, Dong Seoul University, Seongnam 13117, Republic of Korea

ARTICLE INFO

Keywords:

Ultrasonic tacking
Template matching
Tacking automation
Computer vision
Garment Template

ABSTRACT

Temporary stitches are essential in apparel manufacturing as they temporarily secure fabric pieces to prevent misalignment during machine sewing and ensure high-quality results. Manual ultrasonic tacking machines were introduced to enhance the precision of this process; however, the necessity for expert operators remains a major constraint. This work introduces a real-time system for ultrasonic tacking machines that combines vision-guided single-modal MULTichannel Feature Fusion Template Matching (MUFF-TM) to autonomously identify and align tacking points on textiles with sub-pixel spatial accuracy. To overcome the limitations of classic feature-based algorithms on smooth and deformable fabrics, the proposed method utilizes macro-contour extraction and equidistant boundary sampling rather than relying on unstable local textures. Experimental results demonstrate that MUFF-TM achieves a 100% target detection rate with a highly stable Mean Absolute Error (MAE) of under 5 pixels across various dynamic conditions, including changes in orientation, illumination, scale, and non-rigid deformation. By significantly outperforming traditional algorithms (SIFT, SURF, and ORB) in spatial precision, the developed software interface offers a robust, versatile, and scalable solution for advancing automated precision in the apparel industry.

1. Introduction

The apparel industry is one of the slowest automating industries and highly dependent on the use of manual labor [1]. The automation of the textile industry faces numerous challenges as it involves fabrics from various shapes, sizes, and patterns. The irregularities in fabrics and lack of defined edges make it further difficult to automate the process. Hence, temporary stitching (tacking or basting) is considered as a critical step in the textile industry, which is used to temporarily secure fabric pieces in place prior to permanent stitching [1,2]. Although this process ensures proper alignment and fit of the garment, thereby enhancing the overall

quality of the final product, it is constrained by high time consumption and complexity.

As a key step in apparel industry, the technology behind tacking has evolved significantly to enhance efficiency and precision, while attempting to reduce labor-intensive operations. Manual tacking (hand-needling fabric pieces together) has been widely used despite its drawbacks, which is labor-intensive, time-consuming, and highly subjective, leading to variability in stitch placement. Errors in tacking, such as inconsistent stitch positioning and lack of skill can result in sewing mistakes during the final assembly, adversely affecting the final quality of the apparel. This manual approach poses challenges in high-volume

* Corresponding author.

E-mail address: stwoo@du.ac.kr (S.T. Woo).

[†] These authors contributed equally to this work.

<https://doi.org/10.1016/j.rineng.2026.111491>

Received 19 June 2025; Received in revised form 10 April 2026; Accepted 9 June 2026

Available online 10 June 2026

2590-1230/© 2026 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

production environments, where consistency and speed are essential [3]. Despite technological advancements, the need for skilled labor remains a bottleneck in completing the tacking process. Consequently, production speed fails to reflect the extent of technological improvement [4].

The invention of ultrasonic machines is considered as one of the notable advancements in the apparel industry, which aimed to improve the precision of garment manufacturing [5]. It marked a considerable reduction in manual labor by replacing hand-needling with machine-operated tacking. However, current ultrasonic tacking machines still require skilled operators to manually adjust the fabric and determine precise tacking positions. A fully automated tacking process holds the potential to revolutionize garment production by boosting speed and consistency, while minimizing reliance on skilled labor [5,6]. Moreover, the extreme diversity among garment styles, fabric types, patterns specific tacking points, and orientations make automated tacking process highly challenging. Existing ultrasonic tacking machines lack the adaptability to autonomously handle fabric variations, highlighting the need for a smart solution that can identify fabric orientation and determine precise tacking points in real time [7].

Recent advancements in textile automation have increasingly incorporated computer vision technologies to reduce reliance on manual operations. Vision-based systems have been widely applied in fabric defect detection, pattern recognition, and process monitoring. According to [8], the textile sector is transitioning from manual inspection toward smart manufacturing technologies, as manual inspection suffers from fatigue-related inconsistencies and achieves only around 70% detection accuracy. It has also been reported that implementing AI-based automated inspection can increase supply chain profits by approximately 8.6% compared to traditional systems [8]. Furthermore, automation is now extending to more complex production tasks such as automated sewing, where machine vision is used to detect seam lines and generate stitching paths without human intervention [9].

The integration of deep learning techniques in textile manufacturing has demonstrated strong performance in fabric defect detection, key point localization, wrinkle estimation, and pattern classification [10–12]. The Detectron2-based system developed using Mask R-CNN with ResNeXt-101 backbone achieved a 93–97% mean average precision and 97% recall rates for multi-texture defect detection [13]. Architectures, such as VGG-16 and ResNet have been applied to fabric defect classification, with VGG-16 achieving higher accuracy (73.91%) than ResNet (67.59%) in specific transfer-learning scenarios, despite ResNet's advantages in addressing vanishing gradients [10]. Additional approaches using ResNet512-based CNNs have demonstrated 96.5% accuracy with 98.5% precision [14]. Similarly, [12] successfully deployed YOLOv11 on edge computing platform for real-time industrial inspection. The evolution of YOLO architectures from v1 to v11 has been particularly significant, incorporating attention mechanisms and Transformer integration that enhance detection capabilities [15]. Other approaches, such as Mobile-UNet and DCGAN, have shown effectiveness in pixel-level defect detection and unsupervised localization tasks [11]. While these methods learn rich, high-level fabric features, they typically require large, annotated datasets, substantial computational resources, and often exhibit sensitivity to domain shifts commonly found in industrial environments [11].

With respect to template matching algorithms, classical pixel-based methods have inherent limitations, as they generally require the target image to maintain the same orientation and involve high computational complexity [16]. By contrast, modern invariant feature-based algorithms, including Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), and Oriented FAST and Rotated BRIEF (ORB), provide significantly improved robustness to rotation, scale, and illumination changes. SIFT offers highly stable feature matching and strong performance under complex visual conditions [17,18]. SURF demonstrates strong robustness under challenging image transformations and exhibits better rotation invariance, blur resistance, and warp transform

performance than SIFT, whereas SIFT performs better under scale and saturation variations [18]. ORB is the fastest among the three, operating approximately 10 to 40 times faster than SIFT and SURF [9,11]. Although each algorithm has unique strengths, none performs optimally across all scenarios, where SIFT suffers from slower processing speed [17], ORB provides the lowest robustness among the three [16], and SURF is slower than ORB and generally less accurate than SIFT [19,20]. However, none of these methods were designed to explicitly compensate for fabric deformation or to map predefined tacking coordinates onto deformed textile surfaces, which are fundamental requirements in automated ultrasonic tacking applications.

While visual automation exists for inspection, and non-rigid matching algorithms are well established in computer vision, there remains a critical gap in adapting these methods to the real-time, high-speed requirements of automated tacking. To address these limitations, this study proposes the single-modal MULTichannel Feature Fusion Template Matching (MUFF-TM) algorithm, a lightweight, training-free, single-modal vision framework designed for orientation-invariant and deformation-compensated real-time sub-pixel coordinate localization on textile surfaces. The proposed system is specifically developed for integration into automated ultrasonic tacking systems operating under industrial conditions with a single RGB camera, without requiring training data. Existing template matching techniques mainly rely on strict pixel-wise alignment or rigid geometric transformations, which are often insufficient when dealing with the natural deformation characteristics of textile materials. Similarly, invariant feature-based methods such as SIFT, SURF, and ORB offer robustness to rotation and scale variations but lack explicit mechanisms for deformation compensation and predefined coordinate mapping on textile surfaces. To overcome these limitations, the MUFF-TM algorithm integrates a single-modal multi-channel contour feature fusion strategy with a deformation-compensated coordinate reconstruction mechanism, achieving orientation-invariant template matching and sub-pixel localization of predefined tacking coordinates on deformable textile surfaces.

The scope of the MUFF-TM algorithm is defined for automated localization of predefined coordinates on two-dimensional textile surfaces within ultrasonic tacking systems. The core technical contribution lies in achieving real-time sub-pixel coordinate localization. This is accomplished by employing Harris corner detection for landmark extraction, a pixel-wise XOR-based similarity metric for orientation correction, and a nearest neighbor matching strategy to compensate for geometric discrepancies between the template and target. Subsequently, the matched points are transformed into the system's coordinate frame using homography, and the resulting coordinates are transmitted to the Electronic Control Unit (ECU) [21,22]. The MUFF-TM algorithm is experimentally validated for fabrics with texture variations, non-rigid deformations resulting from cutting or handling processes, and in-plane rotations due to manual placement variations. However, the current framework does not address three-dimensional deformations, severe fabric distortions, or overlapped fabric placements, which remain directions for future work. Beyond ultrasonic tacking, the underlying principles of the MUFF-TM can be extended to other textile applications, such as automated sewing operations, fabric pattern verifications, and textile defect detections, where robust and training-free visual localization is required. By combining deformation-compensated localization with real-time sub-pixel accuracy, the proposed system provides an industrial-ready solution for precise tacking point localization on deformable fabrics.

Through this, the study makes a novel contribution to the field of textile automation. The core contributions are summarized as follows:

1. A *rotation-invariant, deformation-aware coordinate mapping method that corrects textile-induced shape deviations, overcoming the limitations of rigid template matching.*

2. A sub-pixel localization mechanism optimized for industrial ultrasonic tacking that enables precise, real-time coordinate transformation for Cartesian motion control, under practical production constraints.
3. A single-modal multichannel segmentation strategy using RGB channel-specific edge features for robust fabric contour extraction under varying textures and illumination.

2. Materials and methods

2.1. The Overall System Operation

Figure 1 illustrates the automated ultrasonic tacking machine and the functionality of the algorithm. Figure 1(a-c) depict the conventional manual operation of ultrasonic tacking, where an operator precisely positions the fabric, aligns the tacking points, and manually activate the ultrasonic horn to complete each tacking operation. Figure 1(e) and (f) show the developed machine, which aims to address the limitations of conventional methods.

As shown in Figure 1(d), the MUFF-TM algorithm compares a user-defined template image against a captured target image to detect and localize fabric pieces on the worktable. During the template preparation stage, the user defines the template image and specifies the tacking point coordinates on the fabric surface according to a predefined protocol. These coordinates serve as predefined tacking locations that the algorithm subsequently maps onto each detected fabric piece in the target image. To mitigate the practical challenges of industrial textile production environments including different fabric orientations, shape deformations, and diverse surface textures and patterns, this study

considers the inherent material properties of textiles, as fabrics are naturally stretchable and shrinkable materials with a high tendency to deform due to handling and external forces [23].

The schematic of the overall operation is illustrated in Figure 1(g). Fabrics placed on the worktable for tacking operation were captured by a camera and the images were transferred to the algorithm for analysis. The algorithm then processes the images and identifies tacking locations relative to the camera coordinate system. The detected coordinates were then transformed into real-world coordinate system using a homography matrix [21] and sent to the Electronic Control Unit (ECU) via serial communication. Next, the ECU uses received coordinates and sends G-code commands to the Cartesian robot. Consequently, the robot positions the ultrasonic tacking tool at each designated location and performs the tacking operation.

2.2. Template Image Construction and Coordinate Definition for Tacking

The flowchart shown in Figure 2 outlines the steps involved in defining the template image and tacking points. The process defines tacking point coordinates on a template image, which are subsequently mapped onto the target image through feature-based geometric alignment between the template and target images, even when the fabric exhibits different orientations, shifts, or slight shape variations. As shown in the flowchart (Figure 2), the algorithm takes an input image of the fabric specimen after it has been placed on the worktable. The input image may include irrelevant background objects or noise that can affect subsequent processing. Therefore, a fabric segmentation step (illustrated in Figure 3) was applied to isolate the fabric area for accurate analysis.

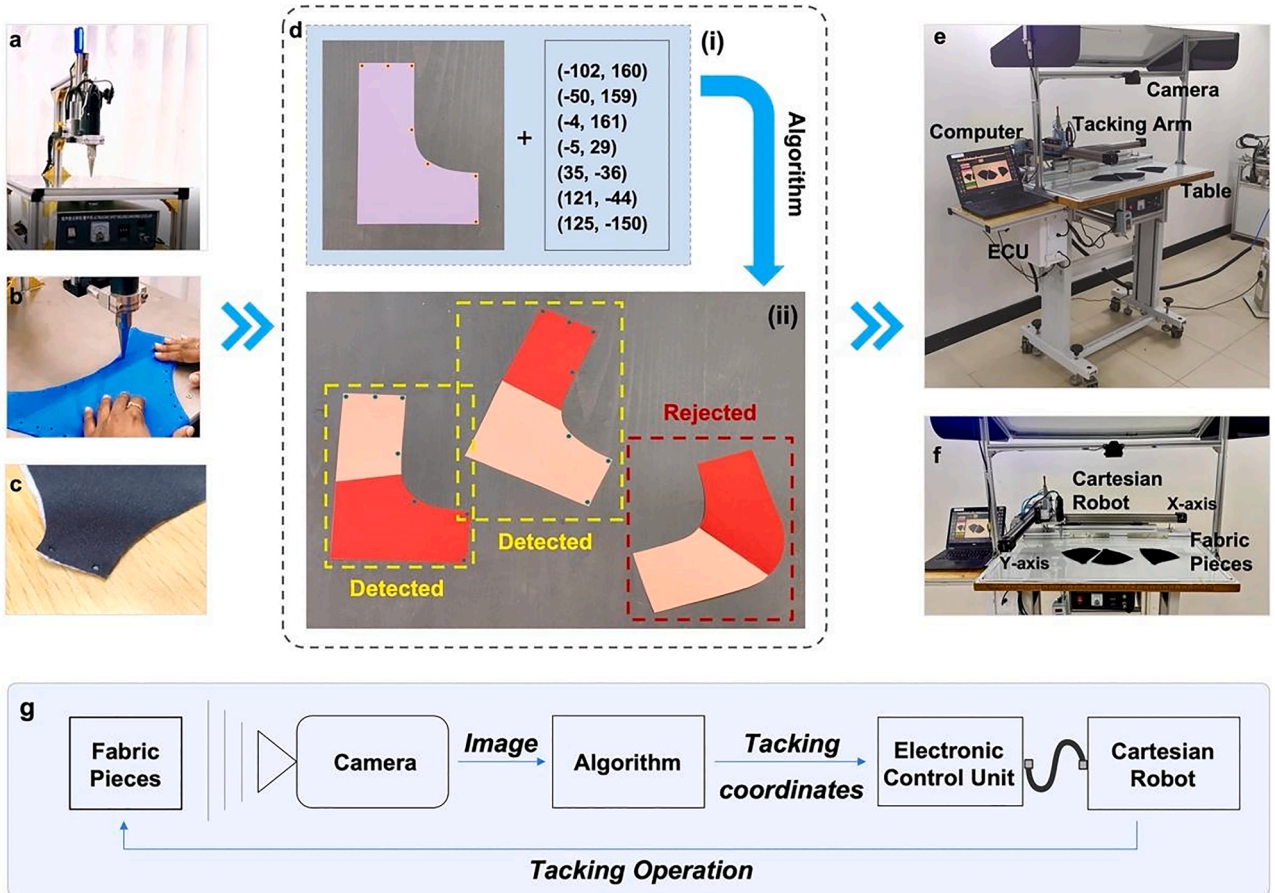


Figure 1. Comprehensive methodology and system configuration. (a) conventional manual ultrasonic tacking machine, (b) manual tacking operation, (c) manually tacked fabric pieces, (d) overall functionality of the proposed algorithm, (e) overview of developed machine, (f) closeup view of the developed machine, and (g) system architecture.

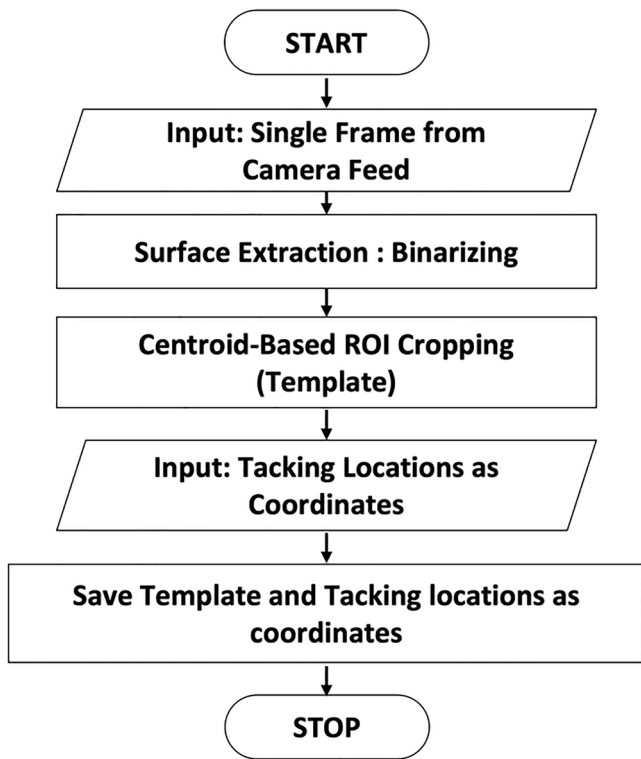


Figure 2. Flowchart – Defining fabric shape and tacking locations.

Figure 3(a) shows image processing pipeline for shape extraction, where the input image is an RGB image, which then separated into three individual red, green, and blue color channels as shown in Figure 3(b). In this study, multichannel processing was chosen for segmentation over single-channel methods due to the critical color information in fabric images, as textile surfaces exhibit considerable color variability, diverse surface patterns, and subtle boundary contrasts that vary significantly across different fabric types and industrial illumination conditions. Prior

research has shown that converting color images to grayscale can result in the loss of color information, leading to errors in regions where visually distinct colors have similar grayscale values [24]. This limitation is particularly pronounced in textile applications, where fabric edges frequently exhibit low luminance contrast but high chromatic contrast, most commonly in colored and patterned fabric. Multichannel correlation processing enables element-wise transformations that enhance inter-channel differences, increasing the independence of information across channels [25], thereby preserving the spectral boundary information necessary for reliable fabric surface extraction. Although multichannel processing introduces additional computational complexity, numerous studies have demonstrated that the resulting improvements in accuracy, robustness, and information preservation make it superior to single-channel approaches for image processing and template matching applications [26].

Each color channel is then convolved with the Scharr X and Scharr Y operators to compute the gradient magnitudes along the horizontal and vertical directions. The resulting gradient images are thresholded to produce binary edge maps for each color channel, as illustrated in Figure 3(d) [22]. These binary images are then combined using a bitwise ‘OR’ logic operation, producing a comprehensive image that includes all possible objects within the captured image (illustrated in Figure 3(f)). Consequently, the combined image underwent contour detection, where all contours except the largest one are removed, considering the largest contour as the fabric shape. The resulting contour is filled to produce a binary mask containing only the segmented fabric as shown in Figure 3(g). The binary mask then processes through a bounding box-based Region of Interest (ROI) cropping operation, as shown in the flowchart of Figure 2, to produce the final template image.

The cropping process begins by detecting the largest contour and computing its centroid $c = (X_c, Y_c)$, and the diameter d of its smallest enclosing circle. A square cropping region is then defined such that it fully contains the largest contour, the detected fabric, with an additional margin. The crop size is calculated as,

$$s = 1.2d \quad (1)$$

And the top-left corner coordinate of the crop is calculated by shifting the centroid by half the crop size as,

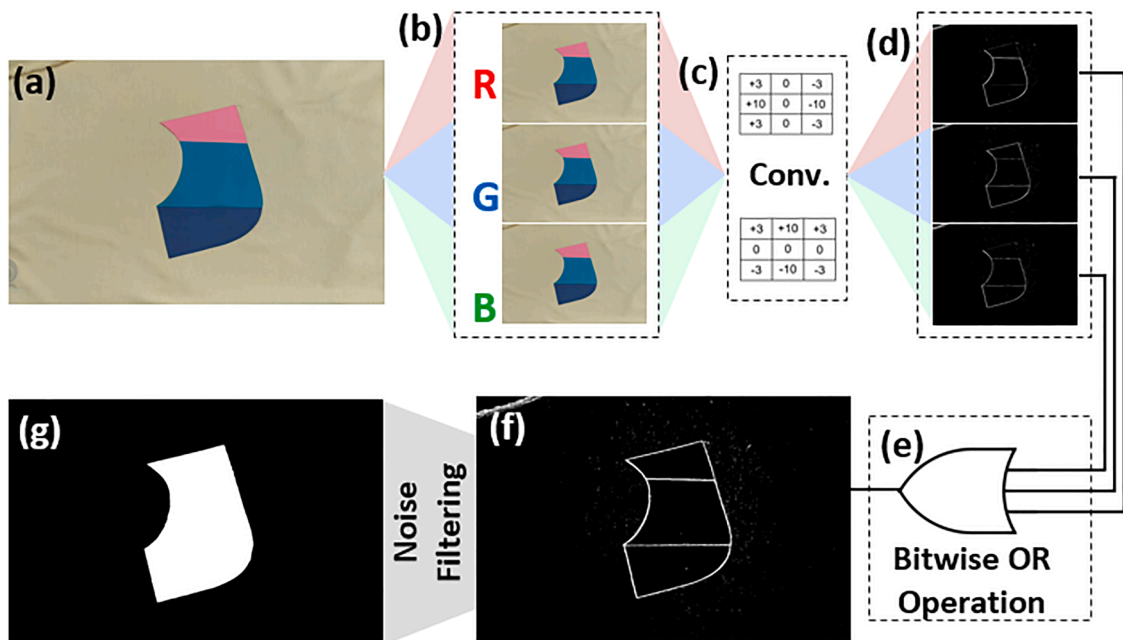


Figure 3. Illustration of the image processing pipeline for shape extraction. (a) input fabric image. (b) RGB channel decomposition. (c) convolution for edge enhancement with Sobel filter. (d) edge detection in each channel. (e) bitwise OR operation to combine edges. (f) edge map after combination. (g) final binary shape mask after noise filtering.

$$(X_t, Y_t) = c - \left(\frac{s}{2}, \frac{s}{2} \right) \quad (2)$$

The cropped window, the template image is therefore a square of width and height s ,

$$w = h = s \quad (3)$$

The 1.2 scaling factor extends the crop beyond the object boundary, preventing edge loss and accommodating shape, noise, and orientation variations for consistent fabric capture.

The user is then prompted to input tacking locations on the segmented fabric shape (binary mask) within the cropped template. These locations are defined as coordinates in a template-based coordinate system, where the image center is set as the origin. Finally, the cropped binary template image is converted into a binary matrix, and both this matrix and the user-defined tacking coordinates were stored for later use in template matching and coordinate identification. For a complete, step-by-step execution workflow, including all inputs, outputs, and key logic, please refer to the pseudocode provided in Algorithm S1 of the Supplementary Material.

2.3. Template matching and coordinate identification

Once the binary template image (matrix) and tacking point coordinates have been prepared and stored, they can be utilized to detect and map the predefined tacking coordinates onto target fabric instances. To perform the tacking operation, the operator places multiple fabric workpieces on the worktable and inputs a command to the system. Upon receiving the command, the algorithm retrieves the previously stored binary template matrix and predefined tacking coordinates from the template preparation stage, setting up the reference information required for target fabric detection and coordinate mapping. The algorithm then captures an image from the camera feed and processes it as

the target image. The flowchart in Figure 4 illustrates the complete template matching and tacking point identification process. As a preliminary step, object segmentation is performed on the target image following the same procedure described in Figure 3, producing a binarized image containing all segmented objects present on the worktable, including fabric workpieces. Each segmentation is then individually cropped using the bounding box method, following the same approach adopted during template image construction.

To standardize the crop size with respect to the template, the algorithm uses the side length l of the template image. For each detected contour, a square crop is defined by centering the template-sized window on the contour centroid (X_{c1}, Y_{c1}) ,

$$X_t, Y_t = (X_{c1} - l/2), (X_{c1} - l/2) \quad (4)$$

$$w = h = l \quad (5)$$

This method ensures that each cropped region from the target image has the same dimensions as the template. After cropping, each binary image is further refined by removing all contours except the largest contour within each cropped region. As shown in Figure 4, the algorithm processes each cropped fabric image (binary image) independently. For each crop, it performs three key steps: orientation estimation (β), shape classification, and coordinate mapping. If the shape matches the template, the predefined tacking coordinates are calculated and mapped accordingly.

2.3.1. Estimation of Target Contour Orientation with Respect to the Template

In this stage, the algorithm estimates the orientation of the contour in each cropped target image (binary image) with respect to the template. The estimation of the orientation (β) with respect to the template is performed to introduce the rotation invariance into the algorithm prior to similarity evaluation and coordinate mapping, which is essential

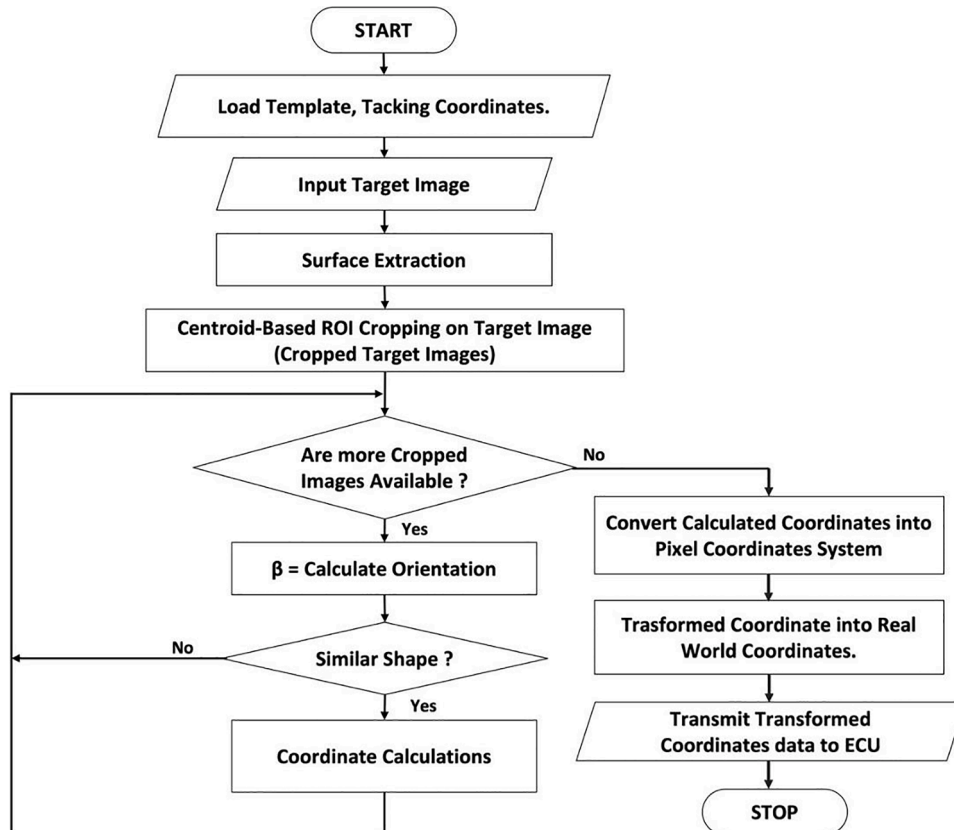


Figure 4. The flow chart illustrating the process of surface extraction.

under practical industrial conditions. In ultrasonic tacking operations, fabric workpieces are placed on the worktable without strict angular constraints, resulting in-plane rotations of the fabric pieces with respect to the template. The algorithm uses equation 6 to calculate the β value:

$$\beta = \underset{0 < \theta < 360}{\operatorname{argmin}} \left[\sum_{q=1}^m \sum_{r=1}^m (I_{\theta ij} \oplus T_{ij})_{qr} \right] \quad (6)$$

where β is represents the computed orientation angle of the target image, while θ is the rotation angle varying from 0° to 360° in 1° increments, I_{θ} represents the rotated target image at θ angle, T is the template image, m is the side length (pixels) of the template, and the subscript i, j denotes pixel coordinates in the image.

The algorithm determines the optimal orientation for β by rotating the target image by θ degrees around its center. For each angle, the dissimilarity is calculated using the 'XOR' ($\oplus$) logic operation, which identifies pixel-wise differences between the rotated target image and the template. The total dissimilarity was computed by summing these differences across all pixels. The angle (θ) that yields the minimum dissimilarity is selected as the estimated orientation (β).

2.3.2. Classification of Fabric Shape against the Template

Once the orientation estimation is completed, the algorithm performs a shape classification step to discard cropped binary images that do not contain the same segmented fabric shape as the template. Without this step, irrelevant objects or incorrectly placed present in the target image could be processed during coordinate mapping phase. This can result in wasted computational resources due to mapping calculations and tacking operation on invalid locations. These operational failures can damage the fabric, misfire the tacking tool, or damage the tacking tool, resulting in production loss and safety risks. To mitigate these issues, we introduce this step as validation and efficiency mechanism into the algorithm.

The dissimilarity between the target and template was quantified using equation 7. The dissimilarity quantification is performed according to the equation given below:

$$D = \left[\sum_{q=1}^m \sum_{r=1}^m (I_{\beta ij} \oplus T_{ij})_{qr} \right] \quad (7)$$

where I_{β} represents the rotated target image at β angle found previously, T represents the template image, m is the pixel length of the template, and the subscript i, j denotes pixel coordinates in the image, and D is the qualified dissimilarity between template and target. The classification decision is based on the following threshold condition:

$$\left\{ \begin{array}{l} \text{Rejected, if } \left(\frac{m \times m}{D} \right) > \tau \\ \text{Accepted, Otherwise} \end{array} \right. \quad (8)$$

Here, τ is the classification threshold, set to 0.2 based on empirical evaluation to ensure the cropped target image maintains a sufficient level of similarity to the template. According to equation 8, if the cropped target image is accepted by the classification process, it proceeds to the next stage; otherwise, it is discarded, and the algorithm moves to the next cropped target image.

2.3.3. Deformation-compensated Coordinate mapping procedure

The next phase of the algorithm is the coordinate mapping phase, which is only applied to cropped images that were validated and from the previous classification phase. Since fabrics are flexible and shrinkable materials, they are prone to deformation during cutting and handling processes [26]. Therefore, the estimated orientation (angle β) alone is insufficient to transform the coordinates accurately, because fabrics may undergo stretching, shrinking, or other local deformations. To address this, we introduce a deformation-compensated coordinate mapping method that handles these variations and accurately maps the tacking points onto the target fabric. Figure 5 illustrates a scenario

where the template (Figure 5(a)) and target (Figure 5(b)) are in the same orientation, but the target is slightly deformed. The deformation can be observed from the differences in the corresponding corner coordinates as shown in Table 1.

To compensate for these local deformations, the algorithm first identifies stable reference points on both the template and target shapes. Corners are selected as reference points because, once the orientation is estimated, they provide consistent and spatially distributed control points that can be reliably paired between the template and target images to incorporate localized deformations into the coordinate mapping calculations. For the corner detection, Harris corner detection is applied, since it reliably identifies points of high gradient variation along the fabric boundary [27,28]. As illustrated in Figure 5, the corresponding corners detected on the template and target are labeled as C1, C2, C3, and C4. As shown in Table 1, there are observable discrepancies between these coordinates; the deformation column shows the pixel-wise differences in x and y , confirming that the target is not identical to the template.

Since the template shape defines the expected geometric structure of the fabric, the reference points on the template act as reference locations for establishing correspondences during matching, and the template corners serve as reliable structural points without spurious detections. However, additional or missing corner detections may occur on the target boundary due to segmentation artefacts or minor irregularities during fabric handling processes. These are implicitly filtered during the nearest-neighbor correspondence stage after orientation alignment, where candidate matches are accepted only if the Euclidean distance between template and target reference points satisfies a predefined threshold. This constraint ensures that only geometrically consistent reference point pairs are retained for deformation modelling, while unmatched or spatially inconsistent detections are excluded.

To ensure robust spatial coverage and accurate modeling of local deformations, at least one reference point is defined in each quadrant of template coordinate system, so that all regions of the fabric are represented, and no deformation is neglected during coordinate mapping. For an adequate spatial representation of the shape geometry, reference points are distributed across the coordinate space using a quadrant-based strategy, preventing concentration of reference points in limited regions and improving stability when modelling localized deformation effects.

If a quadrant does not include at least one corner in the template, the algorithm computes intersection points between the template shape boundary and the diagonal axes and uses these intersections as additional reference points to ensure adequate spatial representation. The corresponding reference points on the target are determined using the same quadrant and diagonal intersection method. In the instance shown in Figure 5, no corners were present in Quadrants 1 and 4 of the template; hence, intersection points B1 and B2 were introduced as reference points, and their corresponding locations on the target were obtained using the same construction. These boundary-intersection points serve as supplementary geometric landmarks for regions with smooth contours, where distinct curvature-based corners are absent, ensuring reliable reference coverage for deformation-compensated coordinate mapping across diverse fabric shapes.

The two sets of reference points from the template and target were then matched using a nearest-neighbor approach [18]. These matched reference points serve as anchors for modelling deformation and ensuring that predefined template coordinates can be accurately mapped onto the target. In the final phase, each user defined tacking points relative to the template P is mapped onto the target using a geometric formulation based on polar representation around a reference point, the law of cosines, decomposition into parallel and perpendicular components, and transformation to global coordinates, followed by orientation correction.

Let C_A denote the matched reference point on the target corresponding to the template reference point that is closest to the tacking

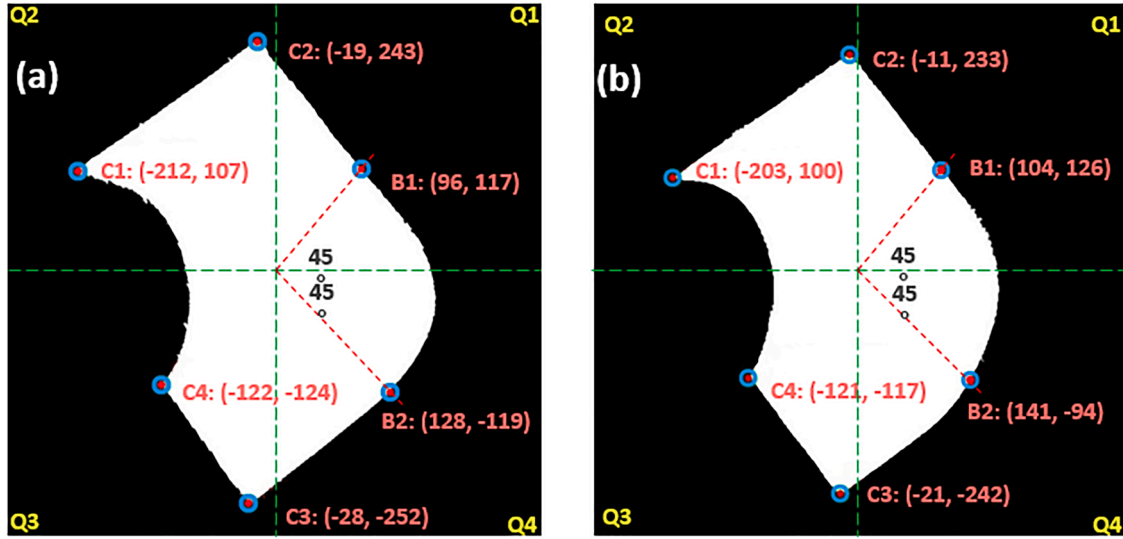


Figure 5. Instance for deformations in target image compares to template. (a) Template (b) Rotated target image by β angle (c) Matched reference point pairs.

Table 1

The matched reference point pairs.

Reference Point	Reference Points on Template	Reference Points on Target	Deformation
C1	(-212,107)	(-203, 100)	(Δx : -9, Δy :7)
C2	(-19, 243)	(-11, 233)	(Δx : -8, Δy :14)
B1	(96, 117)	(104,126)	(Δx : -8, Δy : -9)
B2	(128, -119)	(141, -94)	(Δx : -13, Δy : -25)
C3	(-28, -252)	(-21, -242)	(Δx : -7, Δy : -10)
C4	(-122, -124)	(-121, -117)	(Δx : 2, Δy : -7)

point C_T in Euclidean distance, and O denotes the origin (0, 0). The essential geometric variables are formally defined as follows:

In the final phase, the algorithm maps each tracking point coordinate, defined relative to the template, onto the target using the equations (9)-(15).

Equations (9) and (10) provide definitions as:

$$a = \|P - C_A\|, b = \|C_T - O\| \quad (9)$$

$$\alpha = \text{atan2}(C_y - C_x), \gamma = \angle OCP \quad (10)$$

Using the cosine rule, the deformation-compensated radial distance is,

$$d^2 = a^2 + b^2 - 2ab \cos \gamma \quad (11)$$

and the corresponding parallel component is obtained as,

$$f = \sqrt{d^2 - a^2 \sin^2 \gamma} \quad (12)$$

The tacking point is then reconstructed in the global coordinate system using

$$x' = f \cos \alpha - (a \sin \gamma) \cdot \sin \alpha \quad (13)$$

$$y' = f \sin \alpha - (a \sin \gamma) \cdot \cos \alpha \quad (14)$$

Finally, an orientation correction is applied by rotating (x' , y') by the estimated β , producing the mapped coordinates on the target

$$[X'' \ Y''] = [\cos \beta \ -\sin \beta \ \sin \beta \ \cos \beta] \quad (15)$$

Here, (X'' , Y'') give the final mapped location of the tracking point on the target specimen.

Using these equations, the algorithm maps each tracking point, defined relative to the template to its corresponding target in the input image. However, the mapped coordinates belong to the coordinate systems of cropped target images. The next step of the process is to transform these coordinates into main pixel coordinate system through a transformation matrix. Following this transformation, the coordinates were mapped to the real-world coordinate system using the homography matrix. The homography matrix defines the relationship between the real-world coordinate system and the pixel coordinate system [21]. The transformed coordinates were transmitted to the ECU of the Cartesian robot. Upon receiving the coordinates, the ECU precisely controlled the movement of the X and Y axes to each designated position and executes the tracking operation. For a complete, step-by-step execution workflow, including all inputs, outputs, and key logic, please refer to the pseudocode provided in Algorithm S2 of the Supplementary Material.

2.4. Validation Experiments

2.4.1. Validation of Proposed Algorithm

To comprehensively evaluate the robustness and industrial applicability of the proposed MUFF-TM algorithm, a structured set of validation experiments was conducted. A total of 20 fabric samples were prepared, representing four distinct fabric categories: (i) simple fabrics without printed patterns, (ii) simple patterned fabrics with minimal visual texture, (iii) complex patterned fabrics with dense visual elements, and (iv) complex textured fabrics containing intricate patterns and noticeable physical texture. Figure 6 represents a sample of each fabric category. Five samples were selected from each category, and one sample per category was randomly chosen as the template. Three reference points were manually annotated on each template to ensure consistent landmark definitions across all tests.

For each category, the algorithm was evaluated on both self-detection (template matched against itself) and cross-sample detection (template matched against the remaining four samples). Ground truth coordinates for all reference points were manually recorded through careful inspection of each test image, ensuring pixel-level accuracy without relying on geometric transformation assumptions. This method guaranteed reliable ground truth even under challenging conditions such as fabric deformation, texture variations, illumination changes, and multi-instance scenarios.

To approximate variations commonly encountered in industrial environments, each sample was tested under five controlled conditions: orientation changes, illumination variation, scale changes, fabric

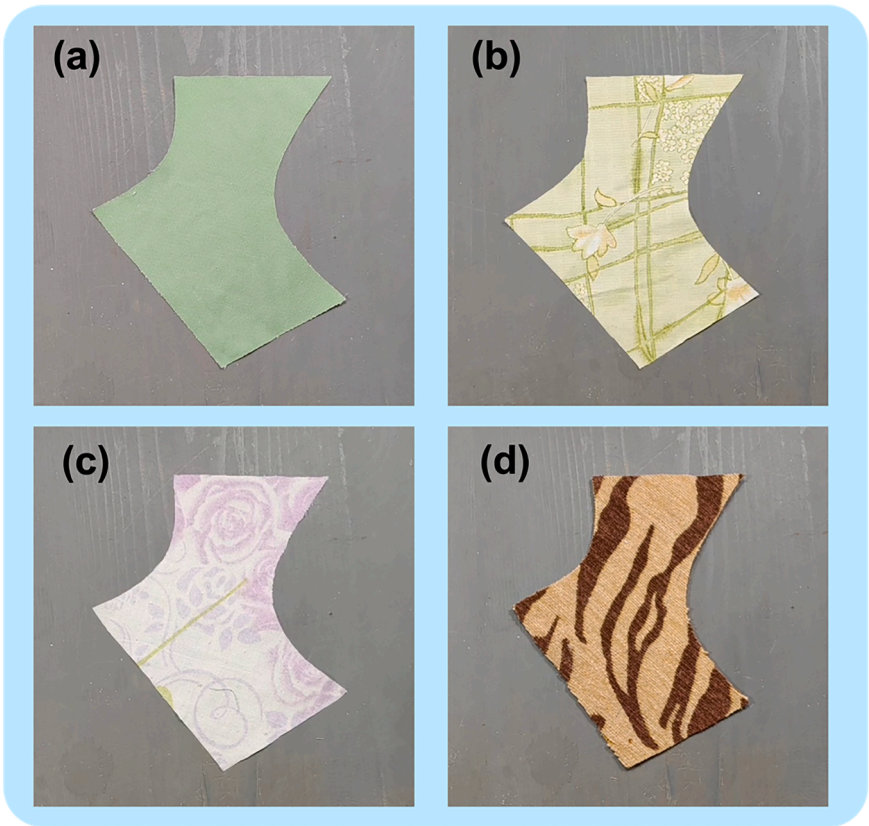


Figure 6. Fabric Samples (a) Simple fabric without printed patterns. (b) Simple patterned fabric with minimal visual texture. (c) Complex patterned fabric with dense visual elements. (d) Complex textured fabric containing intricate patterns and noticeable physical texture.

deformation, and multiple-instance configurations. Across all categories, a total of 220 test cases (input target images) were performed. A detailed breakdown of the experimental conditions and the corresponding number of tests is provided in Table 2. All input target images used in these tests were standardized to a resolution of 1280 × 720 pixels.

2.4.2. Comparative Analysis

To evaluate the performance of the proposed MUFF-TM algorithm against established feature-based methods, validation experiments were conducted using SIFT, SURF, and ORB. The same 20 fabric samples described in Section 2.7.1 were employed, ensuring consistency in sample selection, template assignment, and reference point annotation. Each algorithm was tested under the identical conditions applied to the

MUFF-TM validation, including orientation changes, illumination variation, scale changes, fabric deformation, and multiple-instance configurations. This approach enabled a direct comparison of algorithmic performance across all test cases while maintaining experimental consistency and reproducibility.

3. Results and discussion

The template matching algorithm was systematically evaluated using a diverse set of fabric specimens that varied in orientation, scale, and pattern complexity. The algorithm was tested using specimens acquired through high-resolution imaging. The initial template database consists of representative garment types, encompasses a wide range of weave patterns and fabric textures. Each template was incorporated into the control algorithm, where key points were assigned based on detected corners. This facilitated accurate identification of tacking locations, which were subsequently analyzed for spatial consistency across multiple fabric orientations. The TM algorithm was rigorously tested across fabric specimens exhibiting variations in shape, size, and orientation. The evaluation demonstrated the robustness and adaptability of the algorithm, confirming its capability to maintain accuracy under diverse conditions. These results highlighted the versatility of the proposed template matching algorithm in handling real-world apparel manufacturing scenarios, where fabric variability is inevitable.

Figure 7 presents the data components generated by the algorithm based on user-defined fabric shape and specified tacking locations. Figure 7(a) shows the template, which is a binary square matrix representing the fabric shape, where fabric areas are denoted as ‘1’'s, while the background is represented as ‘0’'s. Figure 7(b) illustrates the data structure containing tacking locations as spatial positions in the coordinate system, where the origin is at the center of the image. These two data components, template, and set of tacking coordinates, are the only

Table 2
Summary of validation experiments.

Condition	Description	Criteria / Levels	Number of Tests
Orientation Variation	Rotation of fabric samples	0°	80
		30°	
		60°	
		180°	
Illumination Variation	Ambient lighting changes	10 lux	60
		20 lux	
		30 lux	
Scale Variation	Camera distance adjustments	Zoom-in (up-scale)	40
		Zoom-out (down -scale)	
Fabric Deformation	Realistic physical distortions	Wrinkles, waves, localized stretching	20
Multiple Instances	Multiple fabric pieces in the same image	2-3 fabrics in a single frame	20
Total number of experiments			220

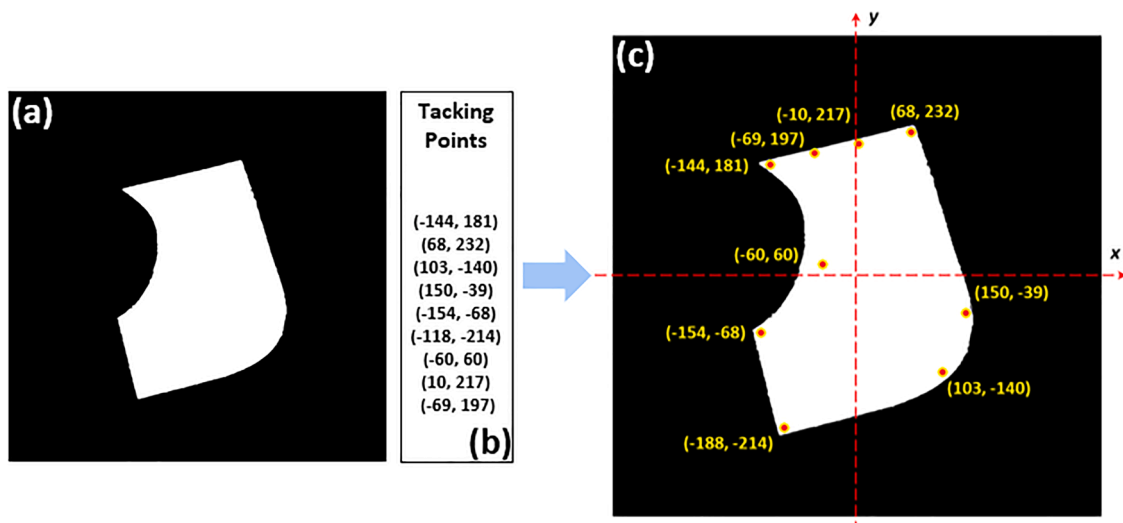


Figure 7. Data component generated by the algorithm (a) Template (b) User Input Tacking Points (c) Illustration of user defined tacking points on template.

requirement from user end. Figure 7(c) provides an illustration of the positioning of user-defined tacking locations within the template. As illustrated, all tacking points lie within the fabric area (white region), since the tacking operation is applied to the fabric. Noteworthy, that the algorithm is capable of handling scenarios, where the coordinates are defined in the background as well.

Figure 8 clearly illustrates the capability of the algorithm to classify fabrics and locate the tacking points through the data components generated from user inputs. Figure 8(a) shows the input image that is fed into the template matching and coordinate identification process. As shown, the algorithm effectively identifies similar shapes to template and rejects irrelevant shapes or objects within the frame. This feature plays a crucial role in handling wrong fabric placements and detecting major defects of fabric specimens. One of the most significant aspects of this method is its ability to precisely locate tacking points on both fabric

specimens, despite variations in orientation and minor shape deformations. The computational performance of the implemented algorithm was evaluated based on its execution time of the algorithm, detection accuracy of the algorithm, and overall performance of the automated ultrasonic tacking machine.

3.1. Computational Efficiency of the algorithm

In industrial computer vision applications, execution time of an algorithm is a critical factor that impacts the real-time processing and overall production workflow [29], since minor inefficiencies of an algorithm can create a bottleneck in the production workflow. There are three main factors that affect the execution time of the proposed algorithm, such as the radius of the Minimum Enclosing Circle (MEC) of the template shape, number of tacking points defined by the user, and the number of fabric pieces in the input target image that match the template shape. An evaluation was conducted to assess the effect of each factor on the algorithm's performance, which analyzed separately, while keeping other two factors constant to ensure a controlled assessment. Figure 8 emphasizes the evaluation results for each factor. Figure 9(a) illustrates the influence of the minimum enclosing circle radius of the template on the execution time of the algorithm. Figure 9(a)(i) presents two instances for shapes with 116-pixel MEC radius and 198-pixel MEC radius, respectively. The algorithm was tested with 9 shapes that varies minimum enclosing circle radius between 100 – 350 pixel and measure the execution time of the algorithm to generate the output. Figure 9(a)(ii) presents the testing results, of the proposed algorithm for the execution time.

The results revealed a proportional relationship with the minimum enclosing circle radius of the template shape. This increase of the execution time was due to increased amount of pixel data caused by increased MEC radius. Further, this relationship is crucial for determining the optimal mounting height of the camera. The camera mounting height refers to the distance between the camera and the machine table, which directly influences the field of view and image resolution. Increasing the mounting height reduces the pixel area of the detected fabric shape, resulting a lower minimum enclosing circle radius that improves the processing speed of the algorithm. However, increasing the height reduces pixel density per unit area, which may compromise the accuracy of coordinate calculations. The optimal height of the camera mounting must be determined by evaluating these factors against accuracy and speed requirements. Figure 9(b) demonstrates the relationship between execution speed and number of tacking locations defined by the user. As illustrated in Figure 9(b)(ii), the algorithm's

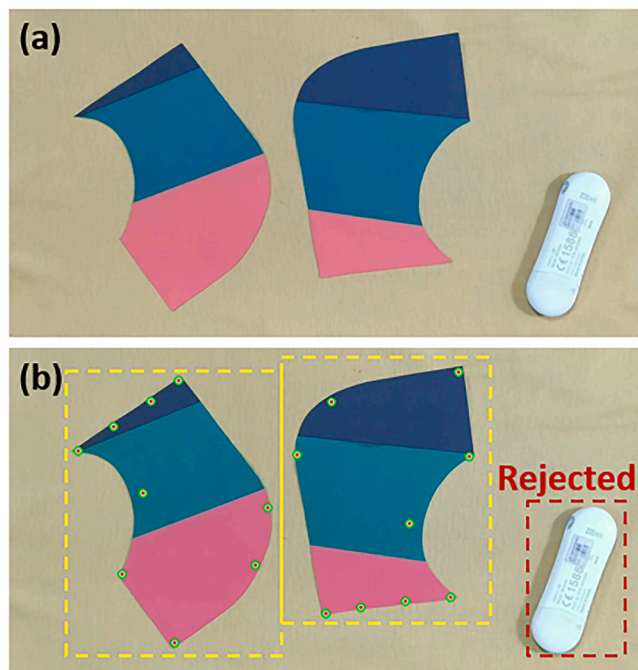


Figure 8. Template matching and coordinate identification process results. (a) input target image. (b) output of the template matching and coordinate identification process.

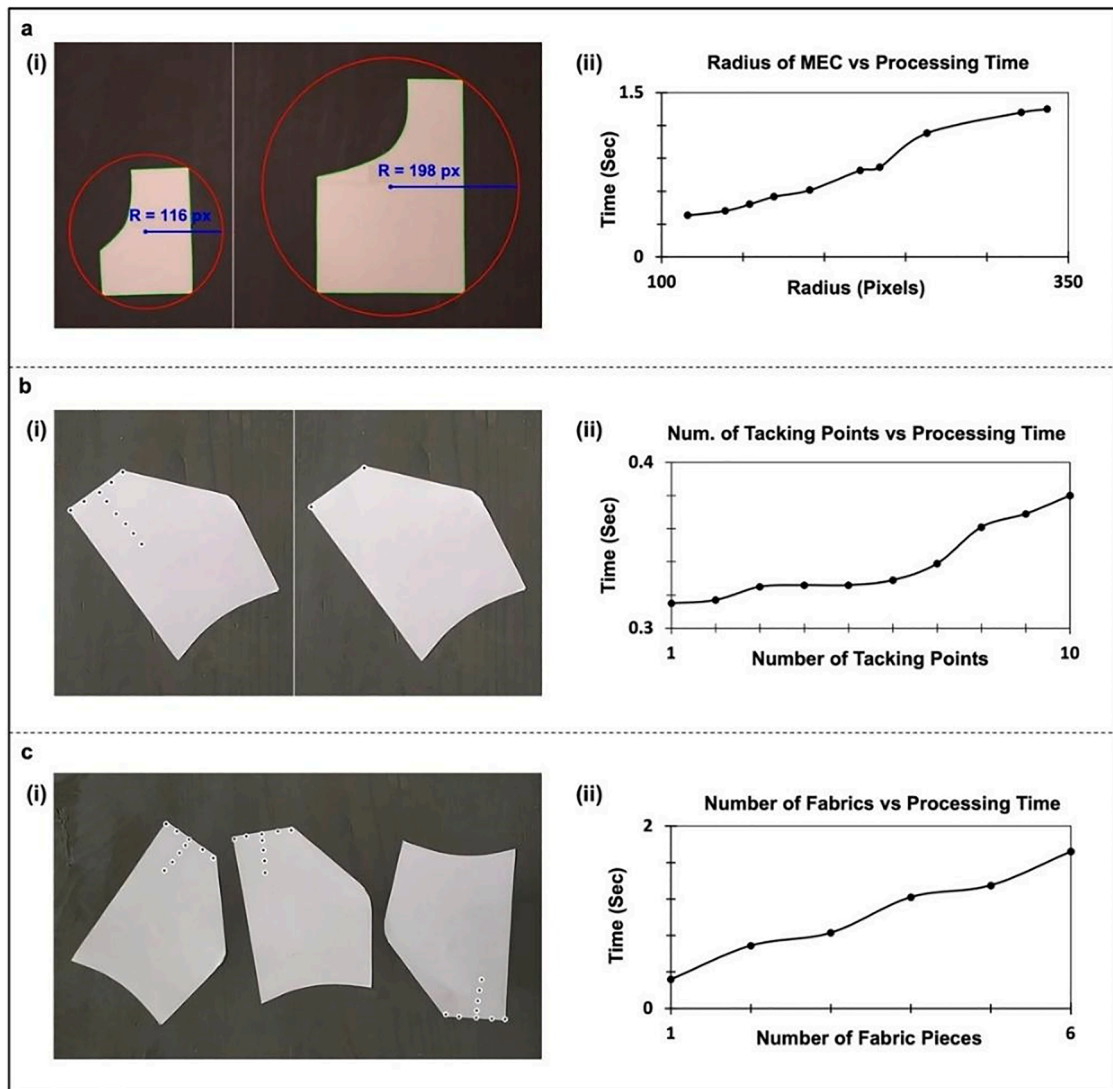


Figure 9. Factors that affect the execution time of the algorithm. (a) Relationship between MEC and execution time. (b) Relationship between number of tacking points and execution time. (c) Relationship between Number of fabrics and execution time.

execution time increases with the number of tacking locations. For each defined tacking point, the algorithm calculates and maps on to the target shape through calculations. While a greater number of tacking points increases computational complexity and results in longer execution times, the underlying calculations involve only basic mathematical operations. As shown in the graph in Figure 9(b)(ii), the derivative is approximately 0.0057 seconds, indicating a relatively slow increase in execution time as the number of tacking locations rises. Hence, the effect of this factor is minimal compared to the other two factors.

Figure 9(c) illustrates the impact of number of fabric pieces in the target image on execution time. Figure 9(c)(i) presents an instance, where the target image includes three similarly shaped fabric pieces along with tacking point detection results. The proposed algorithm is capable of handling multiple fabric shapes within the target image without any specific limitations. Figure 9(c)(ii) illustrates the execution time with increasing number of fabric pieces and showed a gradually increasing relationship. This is an indispensable factor that must be considered when determining the number of fabric pieces to be placed on the table for a single tacking operation.

Considering all these factors, the time complexity of the algorithm can be approximated using Big - 'O' notation as follows:

$$O(n \cdot m \cdot p) \quad (16)$$

where n represents the minimum enclosing circle radius (related to the image size of the template), m denotes the number of tacking points, and p refers to the number of fabric pieces in the input. In all three aforementioned cases, the execution time increased non-linearly with the increase in the respective parameter, which is an expected trend. However, the critical insight is that these parameters exhibit a definitive influence on computational cost and can therefore be considered as threshold variables for industrial applications. Identifying these threshold values is crucial for optimizing productivity and operational efficiency in automated textile processing systems. Furthermore, in industrial information processing tasks, the efficiency of an algorithm is not solely dependent on its computational complexity but also significantly influenced by the performance of the underlying computing hardware [30]. To evaluate this, we conducted a comparative analysis of execution speed using two different processor architectures to assess the impact of hardware capabilities on overall system efficiency as indicated in Table 3.

The experimental results shown in Figure 10 revealed that Computer A significantly outperforms Computer B in all experimented instances with respect to the processing time. As the shape complexity increases from (i) to (vi), computer B demonstrated a higher sensitivity. However, a significantly increased processing time was identified especially for

Table 3

Specifications of Computers Used for Experimental Evaluation.

Feature	Computer A	Computer B
Processor	13th-gen Intel Core i5-1335U	AMD A10-9620P RADEON R5
Cores	10 cores (12 logical processors)	10 cores (4 logical processors)
RAM	4GB	4GB

shapes (iv), (v), and (vi). In contrast, computer A shows a near-linear and minimal increase in processing time, indicating that the factors affecting processing speed of the algorithm can be minimized by the performance of the computer on which the algorithm is running. Thus, the efficient resource usage of Computer A enabled the algorithm to maintain robust and stable performance.

3.2. Analysis of Coordinate Identification Accuracy in the Proposed Algorithm

The accuracy of coordinate identification is crucial, as it directly influences the reliability of the entire system and quality of the apparel manufacturing process. Hence, the algorithm performance was evaluated with ground truth data, which were obtained through precise manual annotations, serving as a reference for measuring positional deviations and identification errors. The detection error was calculated by measuring the Euclidean distance between the detected pixel coordinate and the ground truth coordinate. For the evaluation, two geometrically distinct shapes were deliberately chosen, where one featuring sharp corners and the other with smooth, continuous edges to systematically assess and demonstrate the effectiveness of the algorithm.

Figure 11 depicts the testing results for the fabric shape with sharp corners, which were obtained through two different experimental attempts. In both attempts, the same template was used, but two different target shapes were evaluated. As shown in Figure 11(a)(i) and (b)(i), 11 tacking point coordinates were defined for coordinate identification. The detection error for each tacking point was measured and evaluated separately. Each scenario was tested 100 times by varying the orientation and position of the target and detection errors for each tacking point was recorded separately. The results of these evaluations are illustrated in Figure 10(c), where x-axis represents the defined tacking points, and y-axis represents the detection error (Euclidean Distance) in pixels. As shown in Figure 11(c), Scenario (b) demonstrates a significant increase in the Mean Absolute Error for each tacking point compared to Scenario (a). This increase is due to the deformation in the target relative to the

template in Scenario (a). Deformations in target fabric shape occur when the fabric shape deviates from the predefined template due to stretching, shrinkage, or external forces due to fabric handling.

The shape deformation is clearly illustrated in Figure 11(a) and (b). The final contour alignment shown in Figure 11(a)(iv) provides a comparative visualization of the template and target shapes, where the target shape is nearly identical to the template as it is perfectly aligned with template. However in Figure 11(b)(iv), a noticeable deformation was observed, where the target shape deviates from the template. This deformation introduces discrepancies in coordinate mapping calculations, which then led to increase detection errors in attempt (b). Furthermore as shown in Figure 11(c), tacking points 1, 3, 6, 8, and 11 depicted a smaller increase in Mean Absolute Error due to deformation. This can be attributed to the fact that these tacking points are positioned near the reference points (corners), which are used in coordinate mapping calculations. Hence, the results suggest that the algorithm achieves a higher accuracy for coordinates located in proximity to the corners of the shape.

Similarly, the algorithm was also tested on a shape without corners to evaluate its performance under different geometric conditions. As shown in Figure 12(a)(iv), the template perfectly aligns with the target in attempt (a). However, in attempt (b), a noticeable deformation was observed in the target, as illustrated in Figure 12(b)(iv). Although the Mean Absolute Error of attempt (a) was low, the Mean Absolute Error of attempt (b) was increased due to the presence of deformation. Comparison of results in Figure 11(c) and Figure 12(c) showcases that deformations in shapes with smooth and continuous edges affect the accuracy of algorithm more than the shapes with sharp corners. Table 4 provides a summary of the experimental results, presenting the Mean Squared Error (MSE) and Maximum Error (MAX Error) for each experimental attempt considering the overall performance of the algorithm.

3.3. Validation Results

3.3.1. Validation Results of MUFF-TM algorithm

For each detected point, the Euclidean error between the predicted coordinate and the ground-truth coordinate (x_g, y_g) were computed using equation 17. For each detected reference point, the Euclidean distance between the predicted coordinate (x_d, y_d) and the ground-truth coordinate (x_g, y_g) was computed according to Equation (17)

$$Error = \sqrt{(x_d - x_g)^2 + (y_d - y_g)^2} \quad (17)$$

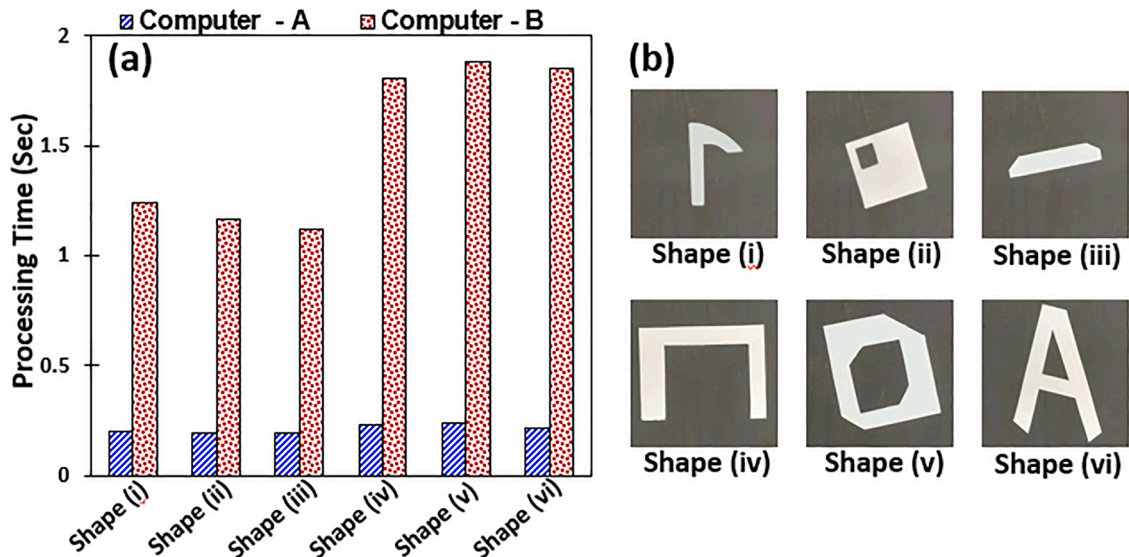


Figure 10. Impact of Computer Performance on Algorithm Speed.

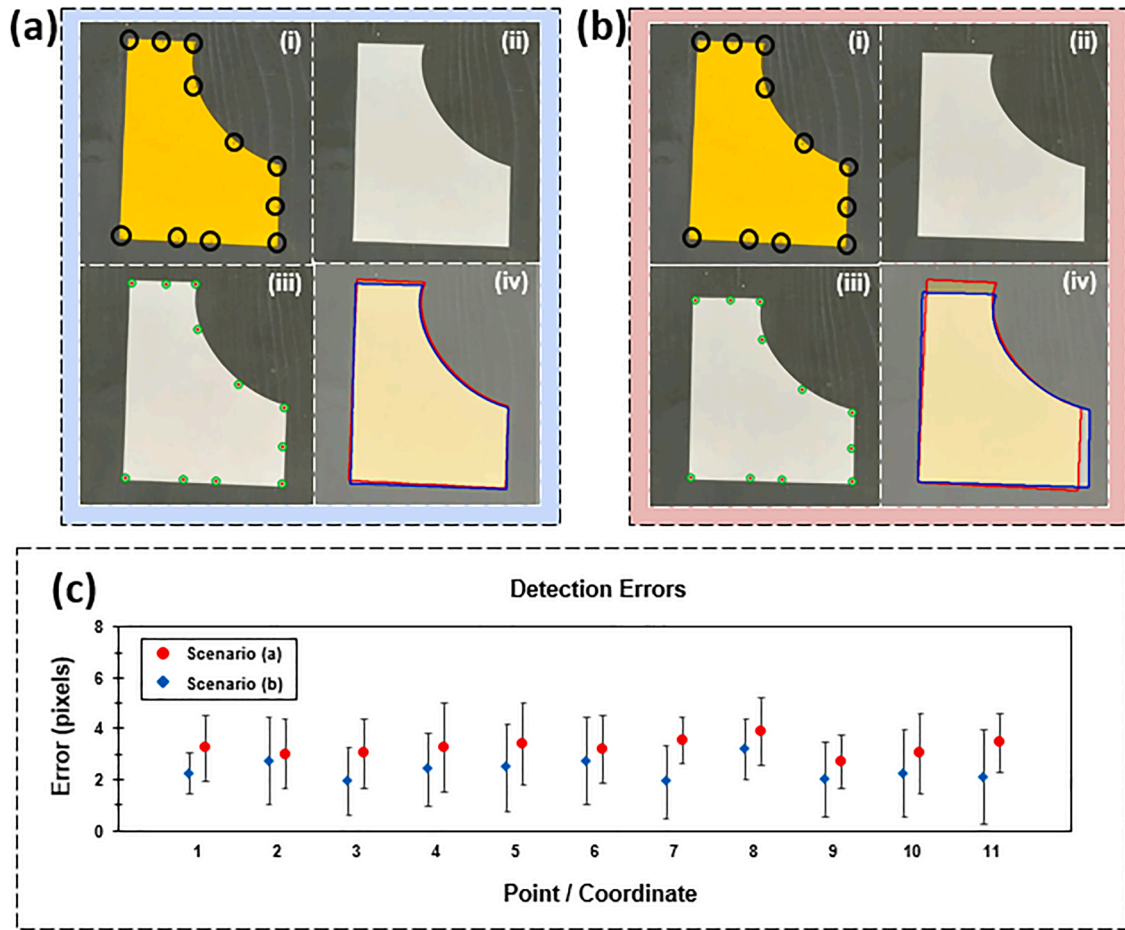


Figure 11. Comparison of coordinate identification accuracy in two experimental attempts for a shape with sharp corners. (a) and (b) illustrate two different detection attempts. (i) template. (ii) target. (iii) outcome of the algorithm. (iv) visualizes the final contour alignment. (c) illustration of detection error.

Based on these point-wise errors, three performance metrics were calculated for every test condition and fabric category, such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Standard Deviation (STD). In addition to these metrics, the Missing Detection Count was calculated. These metrics collectively quantify the accuracy, consistency, and reliability of the proposed MUFF-TM algorithm under varying environmental and structural conditions. Table 5 presents a complete summary of the validation results across all 220 experimental cases, including orientation variation, illumination changes, scale variation, deformation, and multi-instance scenarios.

The algorithm demonstrated strong rotational robustness across all tested orientations, with MAE ranging from 2.14 to 2.99 px and RMSE increasing moderately from 2.38 px at 0° to 3.53 px at 180° . This stability across angles indicates that the contour orientation estimation calculation effectively captures rotational variations, enabling reliable keypoint matching even under significant angular transformations. The slight performance drop at 180° was expected due to structural differences between the template and rotated target. This degradation can be attributed to the inversion of key points, which slightly reduces matching confidence. Since corner features preserve structural information while contour descriptors maintain geometric consistency, the algorithm only exhibits a slight dip in performance. Importantly, no detections were missed across all 80 orientation cases, confirming that MUFF-TM reliably localizes keypoints under substantial angular changes. The algorithm effectively mitigates angular sensitivity and ensures consistent performance under substantial rotation.

Under varying illumination (10, 20, and 30 lux), the algorithm maintained full detectability in all 60 cases. The lowest error occurred at

10 lux (MAE of 1.94 px), indicating that the contour-corner fusion framework remains highly stable even under reduced brightness. This robustness can be attributed to the Harris corner detector's reliance on local intensity variations rather than absolute brightness values, allowing consistent keypoint extraction when global illumination is low but structural contrast is preserved. As illumination increased, errors rose moderately (MAE of 2.80 px at 20 lux and 3.40 px at 30 lux, RMSE of 5.17 px at 30 lux). This performance degradation is likely due to increased surface reflections and specular highlights on fabric textures, which alter local intensity distributions and influence the Harris corner response, thereby slightly reducing keypoint distinctiveness and matching confidence. Nevertheless, because contour features maintain the template geometric structure while corner features are stable key points, the overall matching process remains consistent. Consequently, localization performance remains stable across all lighting scenarios, demonstrating that MUFF-TM is resilient to moderate photometric variations.

In multi-instance tests, the algorithm successfully identified the correct target among multiple visually similar fabrics. For 2-instance cases, MAE and RMSE were 2.37 px and 2.97 px, respectively. For 3-instance cases, errors increased moderately (MAE of 2.96 px and RMSE of 3.94 px), primarily due to increased spatial proximity and partial pattern overlap, which introduce local feature similarity and reduce inter-instance distinctiveness. However, the fusion of contour descriptors with Harris-based corner features enhances discriminative power by combining template boundary structure with locally unique keypoints, thereby reducing false detection in cluttered scenes. Importantly, no detections were missed, indicating that the algorithm

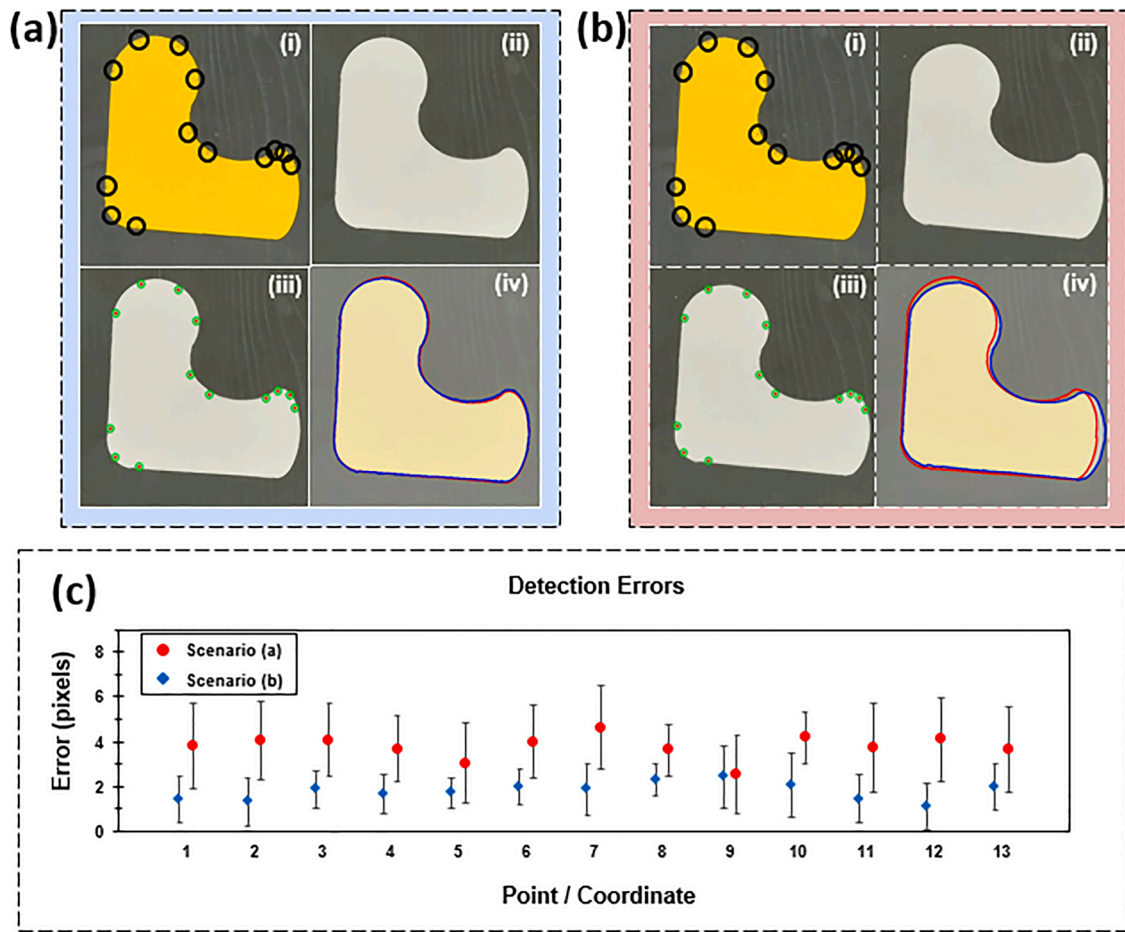


Figure 12. Comparison of coordinate identification accuracy in two different scenarios for a shape with smooth and continuous edges. (a) and (b) illustrate two different detection scenarios. (i) template. (ii) target. (iii) outcome of the algorithm. (iv) visualizes the final contour alignment. (c) illustration of detection error.

Table 4

Mean Squared Error and Maximum Error for each test scenario.

Figure	Scenario	MAE (Pixels)	Max Error (Pixels)
10	Scenario (a)	2.3631	5.83
	Scenario (b)	3.1541	6.40
11	Scenario (a)	1.8563	5.39
	Scenario (b)	3.8771	7.81

maintains robust target discrimination even when similar instances coexist within the same field of view.

Scale variation tests, simulating camera zoom-in and zoom-out revealed that MAE of 2.32 px for 1.2 times (up-scaling) and 2.02 px for 0.85 times (down-scaling). Since the matching process relies on structural configuration rather than absolute pixel dimensions, moderate scale changes do not significantly affect correspondence estimation. Both scenarios produced zero missed detections and low standard deviations, highlighting that MUFF-TM maintains consistent localization

Table 5

Validation Test Results for Proposed Algorithm.

Condition Type	Criteria/Level	Number of Input Samples	Num of Detections	Missing Detections	MAE(px)	RMSE(px)	STD(px)
Orientation	0 degrees	20	60	0	2.1358	2.3840	1.0591
	30 degrees	20	60	0	2.2931	2.5166	1.0369
	60 degrees	20	60	0	2.3709	2.6864	1.2632
	180 degrees	20	60	0	2.9860	3.5261	1.8755
Illumination	10 Lux	20	60	0	1.9423	2.1947	1.0219
	20 Lux	20	60	0	2.8000	3.2634	1.6762
	30 Lux	20	60	0	3.4040	5.1656	3.8854
Multiple Instances	2 Fabric Instances	9	54	0	2.3686	2.9690	1.7902
	3 Fabric Instances	11	99	0	2.9636	3.9376	2.5927
Scale	Zoom In - Up Scale (1.2)	20	60	0	2.3219	2.7019	1.3815
	Zoom Out - Down Scale (0.85)	20	60	0	2.0240	2.3558	1.2055
Deformation	5-10%	13	39	0	4.4863	6.5984	4.8386
	11-15%	7	21	0	5.0462	7.4897	5.5345

accuracy under controlled scale variations.

Fabric deformation caused the highest localization errors among all the tested conditions. Mild deformation (5–10%) produced MAE of 4.49 px and RMSE of 6.60 px, while stronger deformation (11–15%) resulted in MAE of 5.05 px and RMSE of 7.49 px, with STD up to 5.53 px. The 15% deformation limit was selected based on industrial standards. This progressive degradation is expected, as non-planar distortions alter the relative spatial configuration between contour boundaries and corner keypoints, thereby affecting geometric consistency during matching. In particular, wrinkling introduces local curvature changes and slight positional shifts of keypoints, which increase uncertainty. Since the fusion strategy integrates template contour structure with stable corner features, sufficient key point information is preserved to prevent complete matching failure. Therefore, all keypoints were successfully detected

despite deformation, demonstrating that the algorithm tolerates moderate non-rigid variations in fabrics. This behavior is particularly relevant for industrial textile environments, where perfect flatness cannot be guaranteed, and fabric manipulation inherently introduces shape distortions. When considered alongside the results for rotation, illumination, scale variation, and multi-instance scenarios, the findings indicate that MUFF-TM maintains structural matching stability. The experimental validation, therefore, supports its applicability in real-world industrial settings where controlled conditions are rarely achievable.

3.3.2. Results of Comparative Analysis

A comparative analysis was conducted between SIFT, SURF, ORB, and the proposed MUFF-TM algorithm. The total number of input samples was 220, with 3 points to be detected per sample, resulting in a

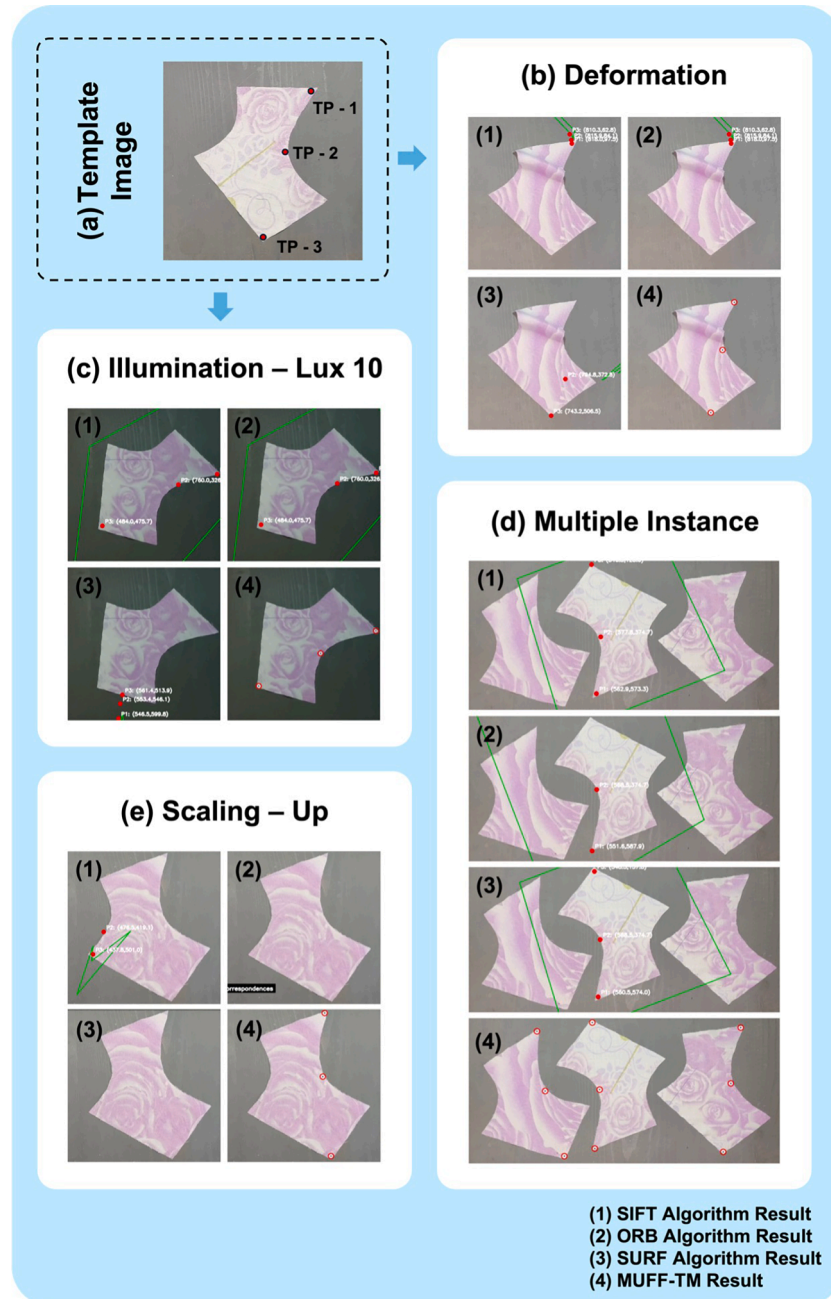


Figure 13. Comparative analysis results of SIFT, SURF, ORB, and the proposed MUFF-TM algorithm. (a) Selected template image with reference points. (b) Algorithm performance under deformation conditions. (c) Performance under low-illumination conditions (10 lux). (d) Detection results for multiple-instance scenarios involving three fabric instances. (e) Algorithm performance under scale-up conditions.

total of 660 points to be detected. The results demonstrate that the proposed MUFF-TM algorithm outperformed the other methods in both fabric detection and coordinate mapping. Figure 13 illustrates an example of the selected template and the corresponding template matching results for each algorithm. Figure 13(a) shows the selected template and reference points, while Figure 13(b)–(e) present examples of deformation, illumination variation, scaling, and multiple-instance conditions respectively. It is evident that SIFT, SURF, and ORB frequently fail to detect points accurately, whereas MUFF-TM consistently detects all points under all tested conditions.

Traditional feature-based algorithms such as SIFT, SURF, and ORB perform poorly on fabrics for several reasons. Unlike rigid printed objects, fabrics undergo non-rigid deformations when pulled, stretched, bent, folded, wrinkled, or slightly curved. These deformations move the texture pattern relative to the shape, causing key points to shift unpredictably, causing matching failures. The tested existing invariant feature-based algorithm assume rigid or near-rigid motion, whereas fabrics are inherently non-rigid, so texture-based key points are unstable. Complex or heavily patterned fabrics exacerbate the problem by generating numerous key points that are unstable, repetitive, ambiguous, and not related to the outer shape, resulting in false matches. For instance, a large floral print may generate over 300 internal key points, while the fabric outline may only require 4–6 important corners.

Fabric appearance also varies under different lighting conditions, such as changes in shadow direction, reflectivity, lighting angle, camera exposure, or folding, which cause local contrast, gradient, and edge intensity variations. SIFT, SURF and ORB rely on consistent gradients and thus fail when these gradients shift. Furthermore, even minor texture distortions, including tiny wrinkles, minor folds, or slight stretching, significantly alter local descriptors, preventing reliable matching. In contrast, MUFF-TM avoids these issues by focusing solely on the silhouette and outer boundary of the fabric, ignoring internal texture and handling deformations through reference points.

In practical applications, the same garment shape may appear in different colors, prints, material blends, or weaving patterns, making feature descriptors meaningless across samples. Feature-based methods rely on appearance consistency, which is rarely present in fabrics. Moreover, traditional algorithms focus on texture rather than shape, while the presented application requires accurate shape detection. Complex, varying, or repetitive textures can cause descriptors to match incorrect regions, resulting in erroneous rotation estimation, misalignment, failed template matching, and incorrect tracking point mapping. Even in textured fabrics, SIFT, SURF, and ORB remain unreliable because fabric textures deform non-rigidly, vary across samples, and generate unstable, repetitive key points that do not correspond to the actual shape geometry.

Table 6 presents the comparison of the number of missed detections and MAE for all 20 input samples under 5 different test conditions in each sample 3 points were to be detected. The results clearly indicate that MUFF-TM achieved perfect detection with zero missed points and maintained a highly stable, sub-5-pixel MAE (2.17 – 4.68 px) across all scenarios. In contrast SIFT, SURF, and ORB exhibited varying degrees of failure, particularly when templates shared the same shape but contained different patterns. Overall, these findings demonstrate that the proposed MUFF-TM algorithm is highly robust, accurate, and suitable for fabric detection and coordinate mapping in practical scenarios where

non-rigid deformation and texture variability are significant challenges. In addition to the missed detection analysis presented in Table 6, detailed quantitative metrics including MAE, RMSE, and STD for MUFF-TM, SIFT, SURF, and ORB are provided in Supplementary Table S1.

The results clearly demonstrate the performance gap between conventional feature-based algorithms (SIFT, SURF, ORB) and the proposed MUFF-TM algorithm. Across all tested conditions orientation, illumination, multiple instances, scale, and deformation the MUFF-TM method achieved 100% accuracy in fabric point detection, confirming its robustness and reliability under diverse real-world constraints. Among the baseline algorithms, SIFT performed the best, achieving 100% accuracy for orientation, but only 65% accuracy under illumination variations and 37%, 85%, and 90% accuracy for multiple instances, scale, and deformation, respectively. SURF followed with detection accuracies of 40%, 35%, 25%, 50%, and 80% under the same conditions. ORB showed the weakest performance, with only 15%, 0%, 29%, 20%, and 55% accuracy for orientation, illumination, multiple instances, scale, and deformation, respectively. Overall, SIFT achieved an average accuracy of 75.5%, SURF 48%, and ORB 23.8%, whereas the proposed MUFF-TM algorithm consistently achieved a 100% detection rate across all evaluated scenarios. Similarly, the MAE results presented in Table 6 highlight a massive disparity in localization accuracy between the algorithms. The proposed MUFF-TM algorithm maintains exceptional stability, with its highest error reaching only 4.68 px under severe fabric deformation. In contrast, classical algorithms like SIFT, SURF, and ORB routinely produce errors exceeding 150 px across all conditions, peaking at an extreme 2076.2 px for ORB under illumination changes, confirming that they are frequently matching completely incorrect fabric features.

In depth analysis of these failure mechanisms reveals the fundamental assumptions underlying classical feature descriptors. SIFT, SURF, and ORB rely on sparse keypoint detection driven by local, high-contrast texture gradients. In textile scenarios, premium smooth fabrics often lack these distinct gradients, while non-rigid deformations (wrinkles, folds) and yarn reorientations create false, highly repetitive local textures. Consequently, when the fabric distorts or lighting changes, algorithms like ORB and SIFT mistakenly match a keypoint to a random fabric fold rather than the true geometric target, resulting in the massive MAE spikes and high miss rates observed in Table 6. In contrast, MUFF-TM employs multichannel feature fusion within a dense template matching framework, aggregating intensity, gradient, and structural similarity over regions rather than isolated keypoints. This region-level similarity integration provides distributed matching evidence that remains stable even when local geometric relationships are altered by stretch, or wrinkling. Consequently, MUFF-TM maintains consistent detection performance across deformation and multi-instance textile scenarios where classical descriptors degrade.

4. Conclusion

This work demonstrates the successful integration of a vision-incorporated single-modal Multichannel Feature Fusion Template Matching (MUFF-TM) algorithm and coordinate localization for real-time 2D textile surface detection in ultrasonic tacking systems. The proposed system is a practical potential for enhancing efficiency and quality in apparel manufacturing workflows. A key insight of the

Table 6
Performance Comparison of Tested Algorithms (Missed Detections (Misses) and MAE).

Algorithm	Orientation		Illumination		Multiple Instances		Scale		Deformation	
	Misses	MAE (px)	Misses	MAE (px)	Misses	MAE (px)	Misses	MAE (px)	Misses	MAE (px)
MUFF-TM (Proposed Algorithm)	0	2.45	0	2.72	0	2.74	0	2.17	0	4.68
SIFT Algorithm	0	244.64	21	187.27	96	384.88	9	365.54	6	168.06
ORB Algorithm	51	339.39	63	2076.2	108	414.79	48	573.35	27	484.02
SURF Algorithm	36	248.85	39	244.74	99	345.90	30	357.26	12	278.24

developed algorithm is its robust capability to consistently match template images with target regions under challenging real-world conditions, including variations in orientation, illumination, scale, deformation, and multiple fabric instances. The comparative evaluation further underscores this strength. While traditional feature-based algorithms such as SIFT, SURF, and ORB exhibited significant performance degradation under these conditions achieving overall accuracies of 75.5%, 48%, and 23.8% respectively. However, the MUFF-TM algorithm achieved 100%-point detection accuracy across all tested scenarios. These results clearly demonstrate the limitations of conventional feature descriptors in textile environments and highlight the superior adaptability of the proposed approach to diverse fabric textures and geometric variations. In addition to accuracy, key threshold parameters identified in the MUFF-TM pipeline were shown to significantly influence processing time, offering valuable insights for workflow and hardware optimization. Performance comparisons across different processors also emphasize the critical role of computational resources in enabling real-time textile automation. Despite the strong performance, the study has several limitations that present opportunities for future research. The number of tested samples can be further increased, and additional environmental variations such as dynamic lighting, motion blur, or high-speed conveyor movement should be incorporated to enhance the statistical robustness of the findings. Although evaluations were conducted under factory-like conditions, full deployment in an active industrial production environment was not feasible due to time and scope constraints. Future work should therefore extend testing to real-time factory settings to verify long-term stability, adaptability, and integration performance. Furthermore, the present study was intentionally constrained to textile materials and tacking operations specifically in the apparel manufacturing. Cross-domain evaluation across unrelated datasets or alternative industrial materials was beyond the scope of this work. Future research should therefore investigate cross-domain robustness and domain adaptation strategies to assess the generalizability of the MUFF-TM framework beyond textile applications. Moreover, integrating machine learning or deep learning techniques represents a promising direction for further enhancing detection robustness, improving generalization to highly deformable or low-texture fabrics, and enabling intelligent decision-making in automated production lines. Extending the algorithm to support more complex apparel designs, multi-layered fabric assemblies, and 3D textile structures may also broaden its applicability. Overall, the advancements presented in this study set the stage for a new era of precision and efficiency in the apparel industry. By demonstrating high accuracy, computational efficiency, and strong real-world adaptability, the MUFF-TM algorithm provides a solid foundation for fully automated textile production systems and opens promising avenues for large-scale industrial adoption.

CRedit authorship contribution statement

Bimsara Sandaruwan Kahandawala: Writing – original draft, Software, Methodology, Formal analysis, Conceptualization. **Rizma Nalmi:** Writing – original draft, Software, Methodology, Formal analysis, Conceptualization. **Buddhika Amila Kumara Sodige:** Validation. **Kasun Subasinghage:** Validation. **Bhagya Nathali Silva:** Supervision, Project administration, Investigation. **Ruchire Eranga Wijesinghe:** Supervision, Project administration, Investigation. **Seong Tak Woo:** Supervision, Project administration, Funding acquisition.

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 funded by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT (No. RS-2023-00273459)). This research was partially supported by Sri Lanka Institute of Information Technology, Sri Lanka, under the grant numbers OR/RG/2026/21, OR/RG/2026/02, OR/RG/2026/07, and OR/RG/2026/11.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.rineng.2026.111491.

Data availability

Data will be made available on request.

References

- [1] Basics of Garment Construction, https://ittc.ac.ke/college-downloads/Basics_of_garment_construction.pdf.
- [2] S. Deaton, *Basting Stitch, Teaching Apparel Production* (2023).
- [3] Xu C, Xu L, Luo K, et al. The essence and applications of machine vision inspection for textile industry: a review. *The Journal of The Textile Institute* n.d.;0:1–25. <https://doi.org/10.1080/00405000.2024.2426257>.
- [4] S. Lee, S.H. Rho, S. Lee, et al., Implementation of an Automated Manufacturing Process for Smart Clothing: The Case Study of a Smart Sports Bra, *Processes* 9 (2021) 289, <https://doi.org/10.3390/pr9020289>.
- [5] A. Elbarbary, HA-F. Al-Fayoumi, ULTRASONIC SEWING OF CLOTHING ACCESSORIES AS AN ENVIRONMENTAL APPROACH TO GREEN MANUFACTURING, *International Design Journal* 13 (2023) 34–45, <https://doi.org/10.21608/idx.2023.233636.1092>.
- [6] N. Seram, Cabon, Darron, Moratuwa, Sri & Lanka, (2008). ULTRASONIC seaming in fashion clothing ULTRASONIC seaming in fashion clothing.
- [7] Y. Hu, Z. Long, AlRegib G.A High-Speed, Real-Time Vision System for Texture Tracking and Thread Counting, *IEEE Signal Process Lett* 25 (2018) 758–762, <https://doi.org/10.1109/LSP.2018.2825309>.
- [8] G. Kaur, B.K. Dey, P. Pandey, et al., A Smart Manufacturing Process for Textile Industry Automation under Uncertainties, *Processes* 12 (2024) 778, <https://doi.org/10.3390/pr12040778>.
- [9] S. Ku, H. Choi, H-Y Kim, et al., Automated Sewing System Enabled by Machine Vision for Smart Garment Manufacturing, *IEEE Robot Autom Lett* 8 (2023) 5680–5687, <https://doi.org/10.1109/LRA.2023.3300284>.
- [10] S.R. Arshad, MK. Shahzad, Deep Learning Based Fabric Defect Detection, *Res Rep Comput Sci* (2024), <https://doi.org/10.37256/rccs.3120244156>.
- [11] M. Raisul Islam, M. Zakir Hossain Zamil, M. Esham Rayed, et al., Deep Learning and Computer Vision Techniques for Enhanced Quality Control in Manufacturing Processes, *IEEE Access* 12 (2024) 121449–121479, <https://doi.org/10.1109/ACCESS.2024.3453664>.
- [12] R. Machado, L.A.M. Barros, V. Vieira, et al., Textile Defect Detection Using Artificial Intelligence and Computer Vision—A Preliminary Deep Learning Approach, *Electronics* 14 (2025) 3692, <https://doi.org/10.3390/electronics14183692>.
- [13] Shital K Dhamal, An Automated Framework for Fabric Defect Detection in Textile Inspection, *Jisem* 10 (2025) 353–367, <https://doi.org/10.52783/jisem.v10i46s.8815>.
- [14] P.R. Jeyaraj, ERS. Nadar, Effective textile quality processing and an accurate inspection system using the advanced deep learning technique, *Textile Research Journal* 90 (2020) 971–980, <https://doi.org/10.1177/0040517519884124>.
- [15] M. Mao, M. Hong, YOLO Object Detection for Real-Time Fabric Defect Inspection in the Textile Industry: A Review of YOLOv1 to YOLOv11, *Sensors* 25 (2025) 2270, <https://doi.org/10.3390/s25072270>.
- [16] P. Swaroop, N. Sharma, An Overview of Various Template Matching Methodologies in Image Processing, *International Journal of Computer Applications* 153 (2016) 8–14, <https://doi.org/10.5120/ijca2016912165>.
- [17] Li C, Li W, Wang Z, et al. Research on Image Feature Extraction and Matching Algorithms for Simultaneous Localization and Mapping. 2021 International Conference on Communications, Information System and Computer Engineering (CISCE), Beijing, China: IEEE; 2021. p. 370–6. <https://doi.org/10.1109/CISCE52179.2021.9445887>.
- [18] E. Karami, S. Prasad, M. Shehata, Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted, Images (2017), <https://doi.org/10.48550/ARXIV.1710.02726>.
- [19] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF. 2011 International Conference on Computer Vision, Barcelona, Spain: IEEE; 2011. p. 2564–71. <https://doi.org/10.1109/ICCV.2011.6126544>.
- [20] Tareen SAK, Saleem Z. A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. 2018 International Conference on Computing, Mathematics and

- Engineering Technologies (iCoMET), Sukkur, Pakistan: IEEE; 2018, p. 1–10. <http://doi.org/10.1109/ICOMET.2018.8346440>.
- [21] Y. Luo, X. Wang, Y. Liao, et al., A Review of Homography Estimation: Advances and Challenges, *Electronics* 12 (2023) 4977, <https://doi.org/10.3390/electronics12244977>.
- [22] H. Spontón, J. Cardelino, A Review of Classic Edge Detectors, *Image Processing On Line* 5 (2015) 90–123, <https://doi.org/10.5201/ipol.2015.35>.
- [23] E. Yun, S. Kim, C. Yun, Development of digitized evaluation methods for fabric shrinkage and damage using image analysis, *Fash Text* 10 (2023) 23, <https://doi.org/10.1186/s40691-023-00346-8>.
- [24] D. Jia, J. Cao, W. Song, et al., Colour FAST (CFAST) match: fast affine template matching for colour images, *Electronics Letters* 52 (2016) 1220–1221, <https://doi.org/10.1049/el.2016.1331>.
- [25] Yzuel MJ, Campos J, Moreno IS, et al. Multichannel pattern recognition of color images. In: Angelsky OV, editor., Chernivtsy, Ukraine: 1999, p. 216. <https://doi.org/10.1117/12.370405>.
- [26] V. Kothari, R. Sreedevi. Cut Resistance of Textile Fabrics — A Theoretical and An Experimental Approach, 2007, p. 32.
- [27] Eltanany AS, Safy Elwan M, Amein AS. Key Point Detection Techniques. In: Hassanien AE, Shaalan K, Tolba MF, editors. Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2019, vol. 1058, Cham: Springer International Publishing; 2020, p. 901–11. https://doi.org/10.1007/978-3-030-31129-2_82.
- [28] J. Sánchez, N. Monzón, A. Salgado, An Analysis and Implementation of the Harris Corner Detector, *Image Processing On Line* 8 (2018) 305–328, <https://doi.org/10.5201/ipol.2018.229>.
- [29] RDSG Campilho, FJG. Silva, Industrial Process Improvement by Automation and Robotics, *Machines* 11 (2023) 1011, <https://doi.org/10.3390/machines1111011>.
- [30] M. Gong, A. Langille, M. Gong, Real-Time Image Processing Using Graphics Hardware: A Performance Study, in: M. Kamel, A. Campilho (Eds.), *Image Analysis and Recognition*, Springer, Berlin, Heidelberg, 2005, pp. 1217–1225, https://doi.org/10.1007/11559573_147.