Repeat Width and Repeat Height Detection in Fabric Image in Fashion

ISBN 978-1-943295-24-1

Jasroop Singh Chadha
Viswanath Reddy
Savita Kumari
Aditya Birla Group
(jasroop.chadha-v@adityabirla.com)
(viswanath.reddy@adityabirla.com)
(savita.kumari@adityabirla.com)

Within the fashion industry, fabric inspection for visual defects such as color, printing-measurement and density is critical for manufacturing apparel. This process is highly dependent on individual capability and leads to production error. Industry solution helps detect piling and tear-based defects but lacks in detecting prints consistency like repeat height and width. In this research article, we propose an algorithmic approach to calculate the repeat width and repeat height from fabric image using computer vision for 3 fabric print categories stripes, prints and checks. This solution helps reduce potential production delays.

Keywords: Fashion Retail Industry, Repeat Pattern, Computer Vision, Fabric Image, Artificial Intelligence

1. Introduction

In the fashion industry, there are multiple categories of fabric patterns. These patterns can broadly be classified, based on motifs such as geometric motifs (including geometric shapes, stripes, and plaids), realistic motifs (replicas or imitations of natural or man-made items) and abstract motifs (color, shape, and size combinations that have no link to natural or man-made items). Designers create these patterns considering multiple factors such as current trends, style specifications, colors, theme and cultural factors. In pattern design, a repeat refers to an identical pattern laid out in a consistent, repetitive layout with horizontal or vertical spacing. A repeat can include a simple motif or an intricate combination of multiple motifs. There are an infinite number of ways to arrange these repeats on fabric, but a few basic layouts are commonly used such as stripes, checks and prints. Stripes are lines arranged in vertical, horizontal, or diagonal directions across the fabric, giving it a simple, linear look. Checks, on the other hand, are created by intersecting lines that form a grid-like pattern resembling plaids. Prints can include various designs, like geometric shapes or abstract motifs, arranged across the fabric to add depth and visual interest. According to [1], quality is a primary concern in the fashion industry. Therefore, quality control and inspections are conducted at multiple stages in apparel manufacturing. Fabric defects can happen due to multiple factors and generally account for 85% of the defects in the garment industry. The detection, identification and prevention of such defects, therefore, becomes essential. In printed fabrics, inconsistencies in repeat patterns can create noticeable flaws, which affect the apparel's quality and appeal.

Various methodologies have been introduced to ensure design precision and alignment across fabrics. For instance, the author describes a technique for detecting repeat patterns using adaptive template matching [2], while in paper [3], another approach leverages the autocorrelation function (ACF) to estimate repeat patterns by identifying correlation peaks in fabric images. Additional methods in this field evaluate texture characteristics such as yarn count, weave repeat, and surface roughness by applying quantitative models like U-Net and WLRL (weighted low-rank decomposition with Laplace regularization). Although these approaches target defect identification and detection in repeat patterns, they exhibit certain limitations such as precise calculation of repeat width and height, which is essential for effective defect detection, especially in designs involving stripes, checks, and prints.

2. Literature Review

In paper [4], the author has utilized fabric images to measure the texture characteristics such as weave repeat, yarn counts and roughness. However, it is limited to detecting defects in weaving and fabric roughness.

The author has utilized adaptive template matching for detecting the repeat patterns [2]. The template is composed of a portion of an image segmented on the principle of maximum edge density. The matching segments that are similar to template, are identified and two displacement vectors are used between adjacent matching segments to estimate size of the repeat pattern. A limitation of this approach is its reliance on the detected edges, which means results can vary if images lack sufficient clarity. However, acquiring industry-standard imaging equipment can be prohibitively expensive.

The author has utilized Unet model (the model consists of two parts encoder and decoder) for detecting and locating the defects in fabric images [5]. For encoding MobileNetV2 has been used to extract features followed by five deconvolution layers as a decoder. However, it only considers defects such as oil stains, tears, etc. which are clearly identifiable visual defects but, it does not cater to identifying defects in printed patterns such as inconsistent repeat width and repeat height.

In paper [6] the author has proposed a WLRL model to detect defects in printed fabrics and received an accuracy of 98%. Again, the defects covered in this model are prominent visual defects such as holes, stains, tears, etc. But it does not encompass the printed pattern defects.

Nasri et al. [3] utilized an auto correlation function (ACF) to estimate periodic Islamic Geometrical Pattern (IGP), by identifying correlation peaks in fabric images. The peak detection is considered as an optimization problem and uses genetic algorithm as a solver.

The author has broken the image into small non-overlapping images of window size N*N, where every small image is classified as defective and non-defective utilizing gaussian markov random field based on likelihood ratio of size alpha [7].

In paper [8], fourier transform is used to detect defects in textured fabrics. They transform fabric images into a spectrum space and then obtain an optimal radius (R). They then set the frequency components outside the circle with radius (R) to zero. Hence, preserving the defects as it is and converting remaining image into uniform gray scale.

3. Proposed Methodology

This section outlines the proposed methodology, offering brief descriptions of all segments involved in the approach, which are essential for calculating repeat width and repeat height. The complete processes for prints, stripes, and checks are illustrated in Fig. 1 (a), (b) and (c).

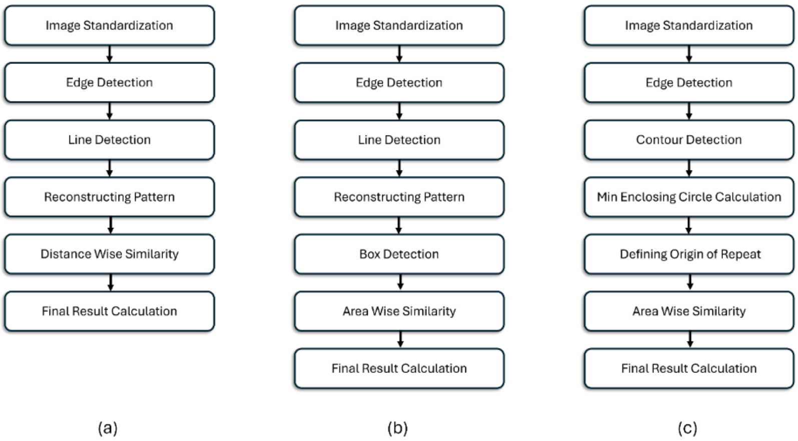

Figure 1 *Process flow for repeat detection: (a) Stripes (b) Checks (c) Prints*

3.1 Standardization of Image

Each image undergone a detailed inspection to identify and address inconsistencies, such as rotations, fabric folds, and visible labels. Following this, extraneous elements like tags are carefully cropped out to maintain focus on the fabric itself. Finally, the processed images are resized to a standardized resolution of 200x200 for consistency in analysis, as illustrated in Fig. 2.

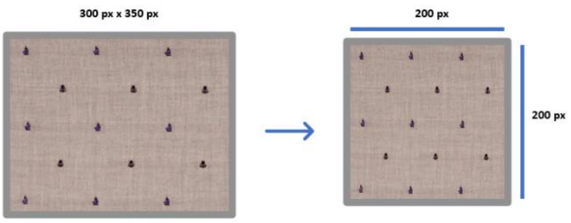

Figure 2 *Image Standardization*

3.2 Edge Detection

After image standardization, edge detection is applied to extract relevant information from the design. The Canny edge detector was utilized with a threshold range of 150 to 255, determined by calculating the minimum and maximum averages of pixel intensity, as shown in eq 1 and eq 2. The visual representation of this stage is illustrated in Fig. 3 and Fig. 4.

$$Threshold_{min} = \mu_{pixel} - k * \sigma_{pixel} \quad (1)$$
$$Threshold_{max} = \mu_{pixel} + k * \sigma_{pixel} \quad (2)$$

where:

- μ_{pixel} is the mean pixel intensity of the standardized image,
- σ_{pixel} is the standard deviation of pixel intensity,
- k is a constant that determines the range around the mean intensity (e.g., for a range of 150 to 255, k can be adjusted to achieve the desired threshold values).

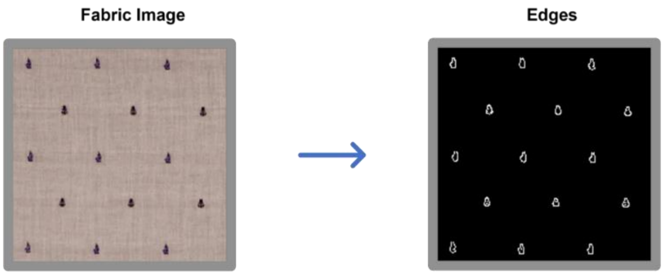

Figure 3 *Edge Detection in Prints*

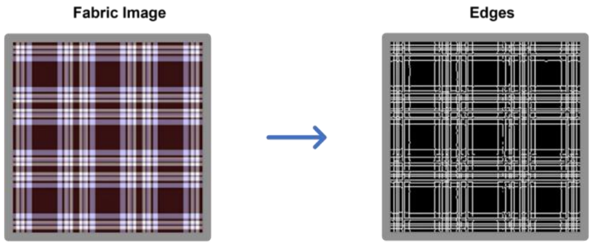

Figure 4 *Edge Detection in Checks*

3.3 Contour Detection

Using OpenCV's chain approximation method, we detect object boundaries within an image which is essential for shape analysis and object recognition. This method optimizes memory by storing only essential contour points, omitting redundant ones along straight lines. As a result, contours are reduced to key points that retain shape accurately, enhancing processing speed and minimizing memory utilization. This technique is widely applied in object detection tasks.

3.4 Line Detector

After obtaining all the edges of an image, for checks and stripes, we iterate row-wise and calculate the sum of each pixel value. This sum is then normalized by dividing by I_{max} and then by N_{py}.

If the normalized value exceeds 0.6 (indicating that 60% of the pixels in a line has intensity of 1), we classify it as a line and reconstruct the pattern on a new image, as shown in Fig. 5.

The line exist condition is mentioned below:

$$I_{AVGy} = \frac{\sum_{x=0}^{n} p_x^y}{I_{max} * N_{py}} \tag{3}$$

$$I_{max} = 255 , N_{py} = 200 \tag{4}$$

Where

- I_{max} is the max pixel intensity,
- N_{py} is the no. of pixels in a row,
- I_{AVGy} average pixel intensity of a vertical line y
- p_x^y pixel intensity at point (x, y)

After all lines have been detected using above mentioned equations, we then reconstruct the pattern by creating a new image and adding lines at same location and with consistent pixel intensity of I_{max} as shown in Fig 5.

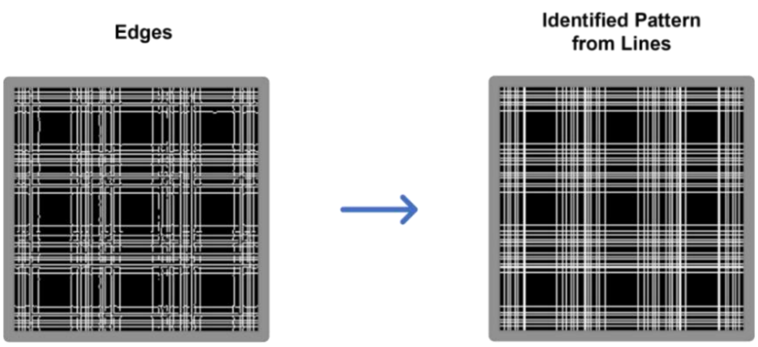

Figure 5 *Reconstructed Pattern*

3.5 Box Detector

Once all lines are detected and the pattern is reconstructed for stripes and checks from section 4.4, we iterate row-wise using pairs of horizontal and vertical lines to identify vertices (or intersection points) of each rectangle. Boxes that fulfill the requirements of eq. 6 are the only ones chosen. Additionally, we calculate metadata, such as the area for each identified box.

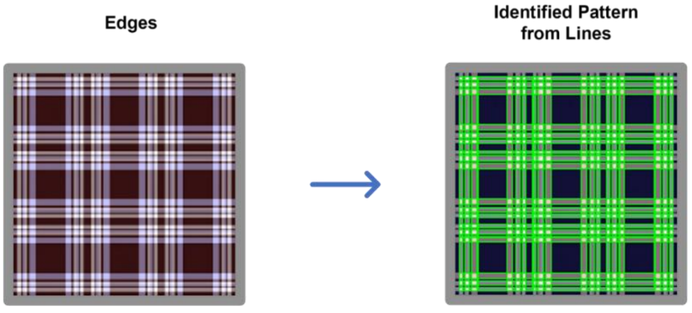

Figure 6 *Detected Lines in Checks*

$$V_{ij} = \{v_{(i,j)}, v_{(i,j+1)}, v_{(i+1,j)}, v_{(i+1,j+1)}\} \quad (5)$$

$$\forall\ v_{(i,j)} = \begin{cases} 0 < x < 200 \\ 0 < y < 200 \end{cases} \quad (6)$$

where:

- $v_{(i,j)}$ denotes vertices of rectangles given their location (i^{th}, j^{th}),

3.6 Min Enclosing Circle for Contours

Once all the contours of an image are obtained from section 4.3, we iterate over them and find an enclosing circle that encompasses that contour with minimum radius as shown in Fig 7. Similarly, we find enclosing circles for all contours and store the meta information such as center of enclosing circle and its radius in a python dictionary-based format.

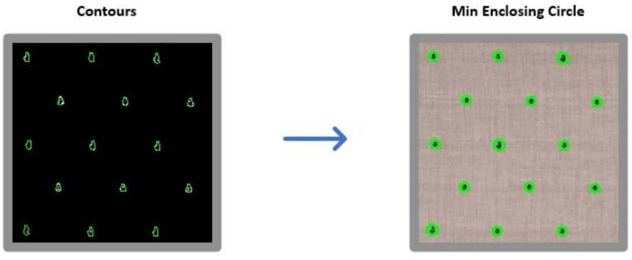

Figure 7 *Min Enclosing circles for Prints*

4.6.1 Center Exist Condition: Radius

After identifying all contour centers from Section 4.6, we calculate the mode of the radii among all enclosing circles. We then retain only the circles with a radius matching this modal value. This constraint ensures uniformity in shape, reflecting the consistency typical of repeating patterns.

$$R_i = \{R_1, R_2, \dots \dots, R_n\} \quad (7)$$

$$f_i = \{ f_1, f_2, \dots \dots, f_n \} \quad (8)$$

$$R_{mode} = f_{max} \quad (9)$$

where:

- R_i denotes radius of a min enclosing given its location (i^{th}),
- f_i denotes frequency of a min enclosing at (i^{th}) location,

4.6.2 Removal of Irrelevant Centers

Once we obtain the centers of enclosing circles, we identify irrelevant centers that arise from the complexity of patterns, such as concentric circles. Additionally, in instances where edges are not fully enclosed in certain fabric prints, multiple contours are detected. To remove those irrelevant centers, we have used gaussian based outlier detection. We calculate the Euclidean distance distribution for each center, representing the distances from all remaining points as shown in the following equations. This results in N distributions from which we extract points that lie at the minima and are considered outliers relative to other distances in the distribution, using the interquartile range.

After obtaining the point combinations, we average them to prevent information loss, which could negatively impact the accuracy of our calculations for repeat width and repeat height.

$$C_{(i,j)} = \{ c_{11}, c_{12}, \dots \dots .., c_{nn} \} \tag{10}$$

$$D_{(1,n)} = \{ d_{(1,2)}, d_{(1,3)}, \dots \dots .., d_{(1,n)} \} \tag{11}$$

$$d_{(i,j)} = \sqrt[2]{(c_{x2} - c_{x1})^2 + (c_{y2} - c_{y1})^2} \tag{12}$$

Irrelevant centers equation:

$$D_{(1,n)} = \{ d_{(1,2)}, d_{(1,3)}, \dots \dots .., d_{(1,n)} \} \begin{cases} d_{(i,j)} < IQR_{lower\ bound} \\ d_{(i,j)} > IQR_{upper\ bound} \end{cases} \tag{13}$$

where:

- $C_{(i,j)}$ denotes centre of a min enclosing circle of a contour given its location (i^{th}, j^{th}),
- $d_{(i,j)}$ denotes distance of between two centres,
- IQR denotes inter quartile range,

3.7 Defining Origin of Repeat for Prints

To calculate the repeat width and height of patterns in prints, it is necessary to take one point of reference and treat it as origin. In this case, it is the centre pixel of image located at (100, 100). We have considered the centre-point $\boldsymbol{C_o}$ as origin in fabric pattern, because the image can be inconsistent at extreme ends such as left, right, up and bottom in terms of clarity and pixel intensity.

4.7.1 Identifying Vertical and Horizontal Points

After, defining the origin for prints we calculate Euclidean distance from all pattern centre points. The centre which is situated at minimum distance from $\boldsymbol{C_o}$ is taken as reference $\boldsymbol{C_{ref}}$. Then, we find the points present in vertical and horizontal direction by calculating difference $\boldsymbol{D_{diff}}$ of co-ordinates to all pattern centres with respect to the reference centre $\boldsymbol{C_{ref}}$ using eq. 15 and eq. 16.

The points which satisfy the constraints given in eq. 17 and eq. 18 are considered.

$$D_{diff} = (D_x , D_y) \tag{14}$$

$$D_x = \| C_{ref\ x} - C_{i\ x} \| \tag{15}$$

$$D_y = \| C_{ref\ y} - C_{i\ y} \| \tag{16}$$

Vertical Points Constraint

$$\forall D_{diff} , \ 0 \le D_x \le 2 \tag{17}$$

Horizontal Points Constraint

$$\forall D_{diff} , 0 \le D_y \le 2 \tag{18}$$

where:

- D_x denotes magnitude of x coordinate distance of centre from centre of reference
- D_y denotes magnitude of y coordinate distance of centre from centre of reference
- $C_{ref\ x}$ denotes x-coordinate value of center of reference
- $C_{ref\ y}$ denotes y-coordinate value of center of reference

The graphical representation is shown in Fig 8.

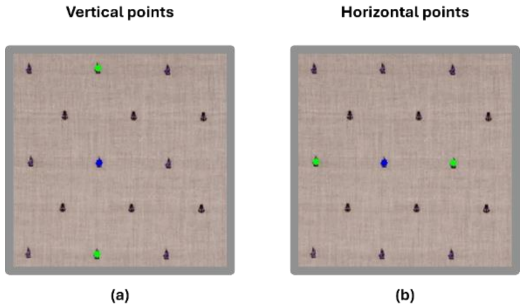

Figure 8 *(A) Vertical Points w.r.t Ref Point (b) Horizontal Points w.r.t Ref Point*

3.8 Calculating Repeat Width & Repeat Height

In case of prints we then calculate Euclidean distance from reference point to all the pattern centers present in vertical and horizontal direction which satisfy the afore-mentioned constraints in section 4.7.1 for prints as shown in Fig 8.

4.8.1 Area Wise Similarity: (Checks)

We find area wise similarity for identifying a repeat. After, receiving results from box detection, we traverse over the boxes row wise using its location. We then take the first box area as reference and calculate the ratio of the area of new box to reference box as shown in equation 19.

$$\rho_i = \frac{A_i}{A_o} \tag{19}$$

where:

- A_o denotes area first box taken as reference,
- A_i denotes area of box at location (i^{th}),
- ρ_i denotes area wise similarity ratio,

We create two sets: in the first, we add each box's coordinates and area until the ratio exceeds 2. In the second set, we continue adding box metadata until the ratio again exceeds 2. This approach helps identify when each small pattern repeats in terms of area. Finally, we extract the first box from each set and calculate the distance between them to determine the repeat width and height.

The value 2 is used because it signifies that the accumulated area has doubled, marking the end of one pattern cycle and the beginning of the next. This makes it an effective threshold for identifying the point at which the pattern repeats.

4.8.2 Distance Wise Similarity: (Stripes)

We find distance wise similarity for identifying repeat. After, receiving results from line-detection we find distance between two consecutive lines and then take ratio of distance of new line with respect to reference distance of reference line as shown in equation 20.

$$\eta_i = \frac{dist_i}{dist_o} \tag{20}$$

where:

- $dist_o$ denotes distance between first two consecutive lines taken as reference,
- $dist_i$ denotes distance between next two consecutive lines at location (i^{th}),
- η_i denotes distance wise similarity ratio

We create two sets: in the first, we add each line's coordinates and thickness until the ratio exceeds 2. In the second set, we continue adding line's metadata until the ratio again exceeds 2. This approach helps identify when each small line repeats in terms of distance. Finally, we extract the first line from each set and calculate the distance between them to determine the repeat width and height.

The value 2 is used because it signifies that the accumulated area has doubled, marking the end of one pattern cycle and the beginning of the next. This makes it an effective threshold for identifying the point at which the pattern repeats.

4.8.3 Final Result: Averaging:

Next, we sort the width and height instances based on area similarity for checks and distance similarity for stripes, while also identifying vertical and horizontal points for prints. We then select the two smallest distance values, calculate their average, and return this average as the repeat width and repeat height, as shown in equation 21.

$$Avg(M_1, M_2) = \frac{dist_{M1} + dist_{M2}}{2} \tag{21}$$

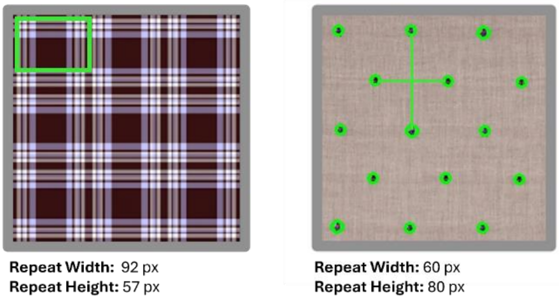

Figure. 9 *Detected Repeat in Checks and Print*

4. Conclusion

In conclusion, our proposed methodology for detecting visual defects by calculating repeat width and height offers a more accurate and adaptable solution for identifying fabric print inconsistencies. While this approach is currently tailored to checks, stripes, and print patterns, it can easily be extended to other fabric patterns across the industry. This methodology not only simplifies the defect detection process but also holds potential for broader applications in quality assurance. With further refinements, such as standardizing image capture distances to convert pixel measurements into centimeters, our approach can achieve even greater precision.

5. Future Scope

There are numerous fabrics patterns available across the fabric industry, apart from checks, stripes and prints. By implementing this approach, addressing discrepancies in repeat width and repeat height for other patterns will become more manageable. Also, the current approach will be improved simultaneously in accordance with factors that can drive better results and overcome any shortcomings in the future. The approach can also be modified according to other prints and their requirements. Apart from simplifying the process of detecting defects, the approach can also be used as an integral part of quality check and quality control. Once the approach gains success by calculating the repeat width and repeat height in the fabric industry, it may transcend boundaries and can be used in other similar industries for similar aspects. Additionally, once we standardize the image capture distance, converting pixel measurements to centimeters will offer further precision.

6. References

1. Amanuel, Lami. 2021. Woven fabric defect control methods in shuttle loom. Journal of Engineered Fibers and Fabrics 16. 15589250211014181.
2. Xiang, Zhong, Ding Zhou, Miao Qian, Miao Ma, Yang Liu, Zhenyu Wu & Xudong Hu. 2020. Repeat pattern segmentation of print fabric based on adaptive template matching. Journal of Engineered Fibers and Fabrics 15. 1558925020973285
3. Nasri, Abdelbar, Rachid Benslimane & Aziza El Ouaazizi. 2014. A genetic based algorithm for automatic motif detection of periodic patterns. In 2014 tenth international conference on signal-image technology and internet-based systems, 112–118. IEEE.
4. Wang, Xin, Nicolas D Georganas & Emil M Petriu. 2010. Fabric texture analysis using computer vision techniques. IEEE transactions on instrumentation and measurement 60(1). 44–56.
5. Jing, Junfeng, Zhen Wang, Matthias Rätsch & Huanhuan Zhang. 2022. Mobile-unet: An efficient convolutional neural network for fabric defect detection. Textile Research Journal 92(1-2). 30–42.
6. Ji, Xuan, Jiuzhen Liang, Lan Di, Yunfei Xia, Zhenjie Hou, Zhan Huan & Yuxi Huan. 2020. Fabric defect fetection via weighted low-rank decomposition and laplacian regularization. Journal of Engineered Fibers and Fabrics 15. 1558925020957654.
7. Cohen, Fernand S., Zhigang Fan & Stephane Attali. 1991. Automated inspection of textile fabrics using textural models. IEEE Transactions on Pattern Analysis & Machine Intelligence 13(08). 803–808.
8. Tsai, Du-Ming & Tse-Yun Huang. 2003. Automated surface inspection for statistical textures. Image and Vision computing 21(4). 307–323.