Efe Ayhan

University of Pennsylvania

Developing an Algorithm for Thread Tension Defect Detection in Textile Weaving

RWTH Aachen University
Institut für Textiltechnik
Arash Razaey

UROP – Undergraduate Research Opportunities Program

Aachen
13/07/2918

Table of Contents:

1. INTRODUCTION & ABSTRACT

1.1 Introduction & Background

Industry 4.0 is the name of the current trend in industrial production to adopt cyber-physical, automated processes, and to extensively use big data and internet of things in order to improve production. It is regarded as the fourth industrial revolution, after mechanization with steam power, the use of electricity through assembly lines/mass production and the integration of computers.

Industry 4.0 includes the development of fully automated production systems, ideally checking for and even correcting potential errors automatically. This relies on the constant gathering and evaluation of data regarding the production through a variety of sensors. For the data gathering methods, machine vision especially shines.

Machine vision is referred to as the "Eyes of Industry 4.0", as it is a very reliable and powerful type of data gathering tool, which can analyse and detect many errors without interfering with the production. It can easily be implemented in existing production lines as all it requires is a camera setup, and with the help of a powerful algorithm, can be immensely valuable.

The textile industry is adopting the concepts of industry 4.0 and transitioning into a fully automated production system. Textile weaving, an important part of textile manufacturing, seems to yield itself very well to a potential application of a machine vision system to check for potential errors in the output weaves (Biche, Gries and Rezaey, 2018).

1.2 Abstract

This paper evaluates the possibility of using machine vision to detect errors in textile weaving, and the potential applications of such a technology in the industry. This paper also aims to propose algorithms and solutions in order to use machine vision algorithms to detect defects in weaves.

Textile weaving can result in a variety of defects, such as fabric texture, roughness, surface resistance and hairiness. This paper will focus on the singular error of banding, which is the direct result of wrong thread tension. The goal is to reliably detect thread tension defects in automated textile weaving with the use machine vision algorithms.

Existing machine vision algorithms will be used and built upon to gather data about textiles and predict defects. Then, the performance of this system will be evaluated, along with if the system is a viable option for companies to implement.

Focus will be given to possibility of this technology being used and developed in the future, what its current shortcomings are and how these problems can be tackled, along with the technical aspects of designing a solution.

2. PROJECT DESCRIPTION

2.1 Textile Production Automation

2.1.1 Context

Industry 4.0, very broadly stated, aims to fully automate processes and limit human interaction. Human workers tend to decrease efficiency and increase cost, especially in countries such as Germany where minimum wages are comparatively high. This has led to the need for quality control processes to be completed by automated systems rather than humans.

Machine vision has been proposed for the main defect detection mechanism for textile weaving for a number of reasons. It is relatively cheap to install and setup on existing systems. It is non-invasive data gathering method, which is preferred in applications such as weaving were small infringements can result in important quality drops. It is reliable and quick enough to detect defects on the whole weave, and existing image recognition algorithms propose good starting point.

Machine vision is already used in many industrial applications to detect defects, and this paper aims to contribute to the automated thread tension defect detection in automated textile weaving processes and systems using machine vision.

2.1.2 Failure Modes Effects Analysis

In an industry application of an automated error checking system, detecting the errors is not enough, but the errors should be given priority scores in order for the automated system to know which errors to tackle first.

For this, the *Failure Modes Effects Analysis (FMEA)* model will be used. FMEA is a step by step analysis method used to identify and account for all possible errors in a system. The model accounts for potential errors, their expected frequencies and their consequences.

FMEA includes not only the identification of different errors but also the management of said errors according to their urgency. An automated system should be able to decide which failures to tackle first, or if to tackle at all, depending on their importance, which can be done by assigning an urgency value on errors. To determine the urgency of each of the errors, Risk Priority Numbers (RPN) should be assigned to every error. RPN is a tool for determining how important an error is, taking into account its severity, occurrence rate and detection rate. All of these numbers are in range 1-5, 1 being almost negligible and 5 being fatal, and the RPN for any error is simply the multiplication of these numbers.

$$RPN = Severity \times Detection \times Occurrence$$

Although the FMEA model is not used in the technical solutions proposed in this paper, it is the next step of implementing a fully automated system, which is why it has been given consideration in this paper. The RPN value for thread tension defects will be given later, along with the discussion of related textile engineering information.

2.2 Current System

2.2.1 Overview

The current machine vision system used for weaving analysis is a system called onLoom. The proposed setup includes a camera mounted on a metal rail system which allows the rapid movement of the camera along the textile in order to capture images of all the weave pattern, as seen in the image 2.2.1 (Schneider, 2013).

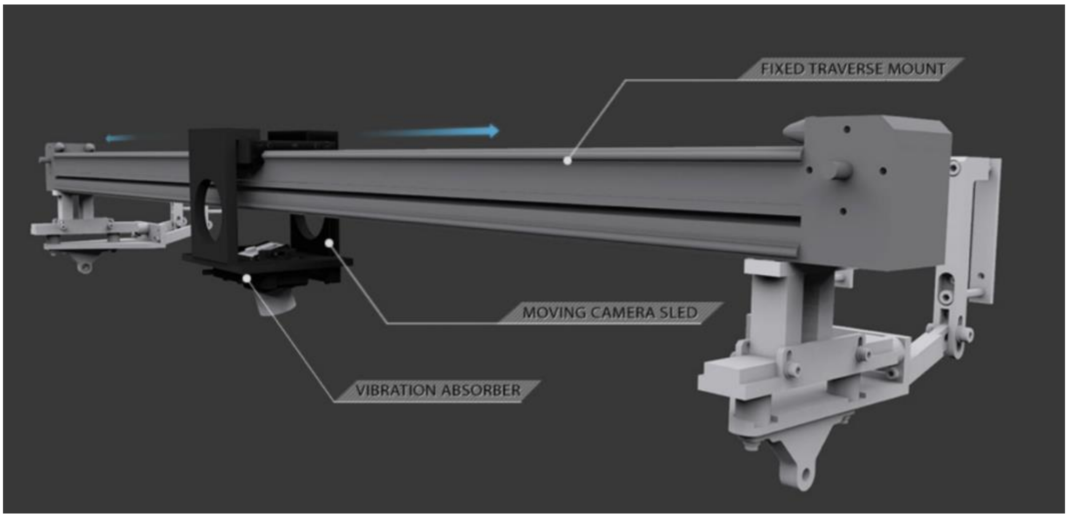

Figure 2.2.1: The implemented system “onLoom”

The weaving machine this setup is mounted on is Picanol NV’s Gamma 8R weaving machine. The camera used to take the pictures of the weave is a JAI BM-500GE monochromatic camera. The images are 5-megapixel quality and are taken at a 15-fps rate. A backlight lighting system synchronized with the camera is used to create consistent lighting by illuminating the weave pattern.

The hardware running the system consists of a I7-950 Intel Processor with 4 cores and hyper threading functionality, Nvidia GTX570 with CUDA integration and 8GB RAM.

2.2.2 Problems and Possible Solutions

There are a few problems with the implemented system that prevent it from working. First of all, the implemented algorithm uses OpenCV for image analysis, and takes advantage of GPU parallel computing using CUDA, to carry out its resource heavy Fourier transform methods in short time. However, the installed OpenCV version (2.1) and CUDA (7.5) do not agree, which makes the Fourier transform methods unusable. Because the program uses legacy libraries, it is not clear which combination of libraries are required to run the program, and even if the correct combination was known, installing would not be straightforward.

Although the program can run without GPU integration, the algorithm is very slow without the GPU resources. In fact, our tests resulted in upwards of 10 minutes for the analysis of a single image, and the system has been designed to analyse many more images to be able to keep up with the speed of the weave being outputted.

Although the problems with the libraries are identified, the ideal solution would be not to download the correct libraries but to update the program to use the current libraries. It is not ideal to depend on legacy libraries, as legacy programs are hard to maintain. This is a future prospect for the implemented system.

There are several hardware problems as well. The initial problem was that the camera was disconnected from the computer onLoom is based on, and no image could be acquired. This problem was solved by updating drivers accordingly and tweaking camera settings. A persisting hardware problem is that the camera does not currently move on the rail automatically and is currently stationary unless moved by external intervention.

The solution to this problem involves mechanical tweaking of the system, which is out of the scope of this project, but nevertheless requires attention to make the system useable.

It is important to note that even though these problems persisted, this system has been used to test the algorithms and solutions presented in this paper, by capturing the example images, and providing the hardware to run the solutions.

2.3 Project Considerations

This project aims to propose different methods of analysing weave patterns for the detection of thread tension errors, in the more general aim to evaluate the possibility of implementing machine vision systems for automated error checking and quality control systems.

This project will extensively draw from previously research, especially by the onLoom system currently implemented but not functioning in the ITA institute of RWTH Aachen, aiming to use both its software and hardware resources.

The design of algorithms or systems which can reliably detect thread tension defects in textile weaving will be considered a success. Implementation could be a future goal, but due to time constraints, full implementation will not be the primary aim. The solutions will be tested however, as extensively as possible.

The conclusion of this research is aimed to not be only about the technical aspect of the research, but an overall conclusion on the possibility of this technology becoming a staple in the industry for the detection of thread tension defects. This is the reason for the discussion of FMEA model and the industry 4.0 generals, as even though they are not directly related to the technical aspect of the research, are still critical to a potential use of this technology to fully automatically detect defects.

3. PROJECT DATA / CONDUCTED RESEARCH

3.1 Textile Analysis

3.1.1 Textile Engineering / Weaving Considerations

To be able to analyse weaved textiles and reach conclusions about the weaving quality from the pictures of the weave, one needs background knowledge on textile engineering and the more specific sub-field of textile weaving technology.

This project focused on one type of textile weaving error; banding (Bandigkeit). This error is most commonly caused by wrong thread tension (Fadenspannung). Wrong thread tension defects happen when one or more of the threads used in weaving doesn’t have the appropriate tension, which tips the balance of the threads and causes defects.

Wrong thread tension results in visible defects in the weave. Most commonly the fault is seen as straight lines going in a single direction, breaking the pattern. In the figure below this phenomenon is visible in the middle right part of the image. This error will cause the distances between the interlocking parts of the weave (nodes) to change. Therefore, defects can be calculated by measuring the length between nodes of the weave or by weft and war densities of the weave and checking for any consistent disturbance of these values.

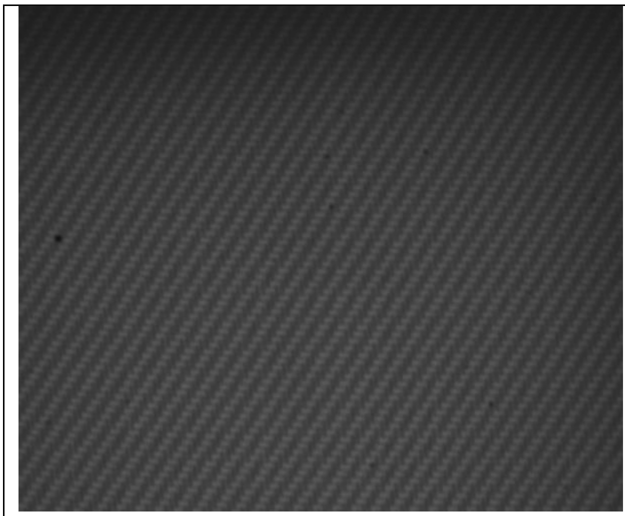	
Image of a regular weaving pattern	Image of the same pattern with thread tension defects

Figure 3.1.1: Comparison of weave patterns with and without thread tension defects.

3.1.2 Error Grading for FMEA

In the industry application of this project, simply identifying errors is not enough, and different errors have to be prioritized in order for the automatized system to tackle them in an appropriate order. For this, as we have mentioned above, FMEA model should be used and different errors should be assigned different RPN values signalling their priority. Higher RPN values represent more critical errors and are more urgent.

Table 3.1 shows the RPN values for the two errors we are focusing in this research project, taken from a bachelor’s thesis on the same subject. (Fuchs, Gries and Razaey, 2016)

	Severity	Occurrence	Detection	RPN
Thread Tension	5	3	3	45

Table 3.1.2: RPN Values for Thread Tension Defects

The practical interpretation of these values if that thread tension defects result in the weave to be completely unusable, these defects occur somewhat regularly, and they can be detected, albeit not reliably. This highlights the need for an automated system for the detection of this error.

In order to be able to record the appropriate RPN values for the errors, the machine vision system has to be able to distinguish between the errors along with determining that there is an error in the first place. This project only focuses on the identification of a single error, wrong thread tension, and therefore the differentiation of errors is not necessary. However, this will become a concern in later applications, and it was seen fitting to be introduced here.

3.2 Existing Resources

This research relies on and aims to build upon several machine vision algorithms used in the project onLoom.

There are two different approaches taken to detecting wrong thread tension in this project. First one was built upon the MATLAB script which can be found on the onLoom website (Schneider and Merhof, 2014). This script was based on analysing the weave as a general pattern, and the weft and warp density values acquired from this script were used to detect thread tension defects.

The second approach used lengths between threads instead of densities, and therefore, an algorithm which detected and identified individual yarn threads was needed. For this, the algorithm described in “Tracking yarns in high resolution fabric images: A real-time approach for online fabric flaw detection paper”, Dorian Schneider (Schneider, 2013) is used.

The setup described under section 2, current system was used to capture weaving images and test the solutions.

3.3 Solutions

3.3.1 Detecting Defects Through Weft and Warp Densities

The onLoom script (Schneider and Merhof, 2014) used for this project achieves the detection of a weaving pattern blindly by creating pairs of weft and warp distances and choosing the best pairs that have an ideal configuration of nodes in order to detect a pattern. If the wefts and warps do not meet an orthogonality criterion, that section of the image is rejected, and a new optimal region is found to detect a pattern. The script uses a cost function dependent on weft and warp distances on the nodes to determine best fit, inside a nested for loop which traverses through the image.

It is expected that thread tension defects are going to result in extra straight lines, which will influence the node placements in certain parts of the yarn. Furthermore, because the influence of thread tension tends to be longer than an instant, there will be numerous nodes which are displaced. If the standard deviations of all of the possible densities from many small sections of the pattern were taken, patterns with thread tension errors would be expected to have high standard deviations and varying densities.

The proposed solution finds the weft and warp densities of small sections of the image by storing all of the densities found by the onLoom script in the internal for loop by brute force to a new matrix. Then, the standard deviations of these densities are found for all of the images. As a final step, standard deviations and other gathered data were compared between example images, some including thread tension errors and some not. From the values of the correct weave patterns, accepted standard deviation and density ranges were decided upon and tested on faulty weaves. All of this is achieved by modifications and augmentations done on the onLoom script.

3.3.2 Detecting Defects Through the Distances Between Nodes

This approach is fairly similar to the first proposed solution, but instead of using weft and warp densities calculated for small sections of the image, the length between adjacent nodes will be used. This is a more direct approach as it looks at every node individually. This approach is based on the same premise that thread tension defects will inevitably be reflected in the distances between nodes.

This algorithm will use the input of **yarnMap,** a very sparsely populated matrix with entries at locations where nodes are, and zeroes where there are no nodes. **yarnMap** is generated by D. Schneider's textile thread analysis algorithm (Schneider, 2013). The first step is to find the distance between each of the nodes and store them in a distance matrix, **distances**. This is simply done by a triple nested for loop through the **yarnMap**. The index of the outer loop is **i,** the middle loop is **j,** and the inner loop is **k.** Whenever used in the loops;

$$0 \le i \le number\ or\ rows\ in\ \boldsymbol{yarnMap}$$
$$0 \le j \le number\ of\ columns\ in\ \boldsymbol{yarnMap}$$
$$0 \le k \le number\ of\ columns\ in\ \boldsymbol{yarnMap}$$

In the middle loop of the following logic takes place:

1) The location of the first non-zero entry is stored in a temporary variable **firstNode.**

Then the algorithm advances to the innermost loop, the **yarnMap** dimensions are bigger than $i + 1 \times j$. If not, the algorithm ends.

2) **yarnMap** is traversed by getting values at **yarnMap[i+1, k]**, incrementing k at every loop.
3) The location of the first non-zero entry is then and stored in the temporary variable **secondNode.**
4) The following 2D Euclidean distance formula is applied to find the distance between the two entries, in up-down, left-right direction.

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

5) This value is stored in a temporary variable **shortestDistance.**
6) **firstNode** is unchanged and the index is incremented until another non-zero entry is reached.
7) This entry is stored under **secondNode.**
8) Distance is again calculated and only if smaller than the value in **shortestDistance,** is stored in **shortestDistance**.
9) This process repeats until the innermost loop ends.

Back to the middle loop;

10) The value **shortestDistance** is mapped to **distances.**
11) The position marker in **distances** is incremented in the +x direction by 1.

In the outer loop, every time after the inner loop runs,

12) The position marker in **distances** is incremented in the -y direction by 1, so that the matrix now stores values in the next row.

This algorithm will output a highly populated matrix **distances,** which stores values for the shortest distances from each node to another adjacent node in strictly downwards (-y direction) from the original node. These distances are very valuable in determining if the pattern of the weave is as desired. This is because they are much more direct than the density values discussed before as the distances are found between all the adjacent nodes of different wefts and warps, and even one out of the ordinary distance can be detected. A wrong thread tension is sure to disturb the distance between adjacent nodes.

With the distance values ready, a similar analysis to section 3.2.1 can be done by finding the standard deviation of the distances. A high standard deviation will mean that there is a lot of variety in the shortest distances between nodes, which means that the weave pattern is disturbed, which would hint for a thread tension defect. Unfortunately, due to time constraints, this algorithm could not be implemented.

4. RESULTS

4.1 Data

Images A, B, C, D, E, F, and G are from one textile without thread tension defects, but possibly with other defects. Images 1, 2, 3, and 4 are from one textile with thread tension defects. The images are given in the appendix.

Table 4.1.2 shows the data collected about the different images. The first four parameters are outputs of the onLoom script. The last column, titled “Nodes Removed from List” is an integer parameter that keeps record of how many clusters are rejected before reaching the best fit for pattern detection. This calculation was detected by augmenting Section 7 of the source code.

Textile	Yarn Frequencies	Weft Density (yarns/cm)	Warp Density (yarns/cm)	Weave Kernel Size	Nodes Removed from List
A	35,0426 | 20,5875	22,2951	13,0984	4 x 4	43
B	35,0426 | 20,5875	22,2951	13,0984	4 x 4	10
C	35,0426 | 20,5875	22,2951	13,0984	4 x 4	69
D	35,0426 | 20,5875	22,2951	13,0984	4 x 4	63
E	35,0426 | 20,5875	22,2951	13,0984	4 x 4	51
F	35,0426 | 20,5875	22,2951	13,0984	4 x 4	46
G	35,0426 | 20,5875	22,2951	13,0984	4 x 4	47
1	19,1875 | 27,8108	16,5044	23,9218	4 x 4	8
2	15,4465 | 27,44	16,7274	29,7154	5 x 5	11
3	19,3386 | 28,1918	16,2813	23,7349	4 x 4	7
4	15,5443 | 28,1918	16,2813	29,5285	5 x 5	9

Table 4.1.1: Data collected using blind weave detection algorithm

Table 4.1.2 shows the calculated standard deviations for the weft and warp densities. As it can be seen, the standard deviations of weft densities are very high for the textiles with the thread tension error. However, this is not seen in the warp densities. Since thread tension results in visible errors only on a single axis, this is logical, and it also shows the importance of calculating the standard deviation for both weft and warp densities.

Textile	Standard Deviation of Warp Densities	Standard Deviation of Weft Densities
A	8.7606, 12.2404	4.2363
B	9.1095	9.9131
C	9.1432	9.0475
D	9.1095, 12.8647	9.5069
E	9.0928, 17.7510	9.9544
F	9.1432, 9.0341	7.1093
G	9.1432, 9.034	7.0117
1	6.6295	52.2249
2	6.2376	56.8893
3	1.0682, 2.0838	47.2694
4	0, 3.4450	53.6598

Table 4.1.2 Standard Deviations of Distances between Warps and Wefts

Unfortunately, due to time constraints, the second approach highlighted has not been able to be tested. This remains an important future prospect for this paper.

4.2 Result of Research

Through the modification of the onLoom script, it was possible to extract weft and warp densities, and the number of discarded clusters about any given textile weave image.

When the weft and warp density values found through the algorithm are compared with the expected values, it can be detected the accuracy of the weft and warp densities can be measured. For textiles A-G, the expected weft density was 22 yarns/cm and expected warp density was 13 yarns/cm. As it can be seen, the highest percent error, on weft density is only 0.9, and from the fact that all the values agree shows that the result for the error is more likely the weaving machine, which shows that the vision algorithm was accurate.

Looking at weft densities, there is a clear difference between weaves with and without thread tension. This is an expected pattern, as thread tension is sure to manipulate node locations and therefore the density calculations. The distances in Textiles 1 – 4 rise and fall quickly and have very high maximums. Spikes in distances in Textiles 1-4 can be linked to higher tension of certain threads. These varying spikes then result in high standard deviations, because thread tension error tends to affect a significant portion of the weave the effect on standard deviations is fairly large, which can then be used to detect the defect itself. On the other hand, the standard deviation calculations of the weft distances of Textiles A-G appear to be fairly low, ranging from 0 to 12.8647 and an outlier of 17.7510 (Textile E), which enforces our conclusion that these textiles do not have thread tension defects.

The interesting result here is that warp densities do not follow this pattern. Warp densities for textiles A-G are considerably higher than textiles 1-4, which in first look, seems like unfortunate. However, the warp density values only work to reinforce the process. Textiles A-G have other faults that are not thread tension, which explains why the standard deviation of the weft and warp densities are not very small. This also explains the reason why textiles A-G dropped so many nodes from the list when finding the optimal ones.

Thread tension only causes defect in one direction, shown as straight lines, it would be logical if the densities of only weft is affected, while warp is unaffected. The fact that the standard deviation of weft densities for textiles 1-4 is so dramatically high, while the standard deviation for the warp densities is very low, caused from the thread tension defect only affecting the weft threads visibly reinforces the idea that the anomaly spotted by this analysis is a thread tension defect.

This last analysis depends on the layout of the specific system, direction of the weave and the pattern of the weave. However, with consideration of the direction of the weave, the analysis should be applicable to other systems easily.

As shown, this data is invaluable in detecting any thread tension defects present on the weave. From the small number of test images this paper had access to, it has been shown that using weft and warp densities and their standard deviations can be used not only to predict when there is an error, but to predict when there is a thread tension defect specifically.

5. EVALUATION

This paper has been about the potential application of machine vision to textile weaving in order to detect thread tension errors. Initial considerations about the need for automation and the economic benefits gives the idea that incorporating automatic quality control and error detection systems is very beneficial. In terms of using machine vision to do so, the current progress and overall analysis of the system strongly shows us that machine vision is an ideal and efficient way to carry out quality control and error checking tasks, especially for textile weaving. Automated error checking systems, with their synergy with existing error classification models such as FMEA, are the future of the textile weaving process, and this research shows that machine vision-based algorithms and analysis techniques yield themselves well to these kinds of applications.

The more technical aspects of this paper are promising as well. Even though the more sophisticated approach to error checking proposed has not been implemented and tested due to time considerations, even a relatively rudimentary analysis of weft and warp densities can give very promising results, allowing a system to detect, very reliably thread tension defects, and in most cases, identify the defect correctly as thread tension. However, although the system was implemented, again due to time considerations, it has not been tested thoroughly, which is an important consideration to keep in mind when evaluating these results. This paper concludes that the future is definitely bright for the implementation of machine vision systems to carry out automated error checking in textile weaving, and other industry 4.0 applications in this industry.

6. REFERENCES

Schneider, D. (2013). *Tracking yarns in high resolution fabric images: A real-time approach for online fabric flaw detection.* Institute of Imaging & Computer Vision RWTH Aachen University.

Schneider, D. and Merhof, D. (2014). *Blind weave detection for woven fabrics.* [online] OnLoom Imaging. Available at: http://onloom.lfb.rwth-aachen.de/ [Accessed 13 Jul. 2018].

Fuchs, J., Gries, T. and Razaey, A. (2016). *Entwicklung eines automatisierten Qualitätssicherungskonzepts für eine Weberei*. Undergraduate. RWTH Aachen.

Biche, W., Gries, T. and Rezaey, A. (2018). *Entwicklung eines 4.0 Qualitätssicherungskonzeptes zur Minimierung der Ausschussrate in der Produktionskette beim Preforming*. Graduate. RWTH Aachen.

D. Schneider, T. Holtermann, F. Neumann, A. Hehl, T. Aach and T. Gries, "A vision based system for high precision online fabric defect detection," *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Singapore, 2012, pp. 1494-1499.
doi: 10.1109/ICIEA.2012.6360960

7. APPENDIX

Appendix A

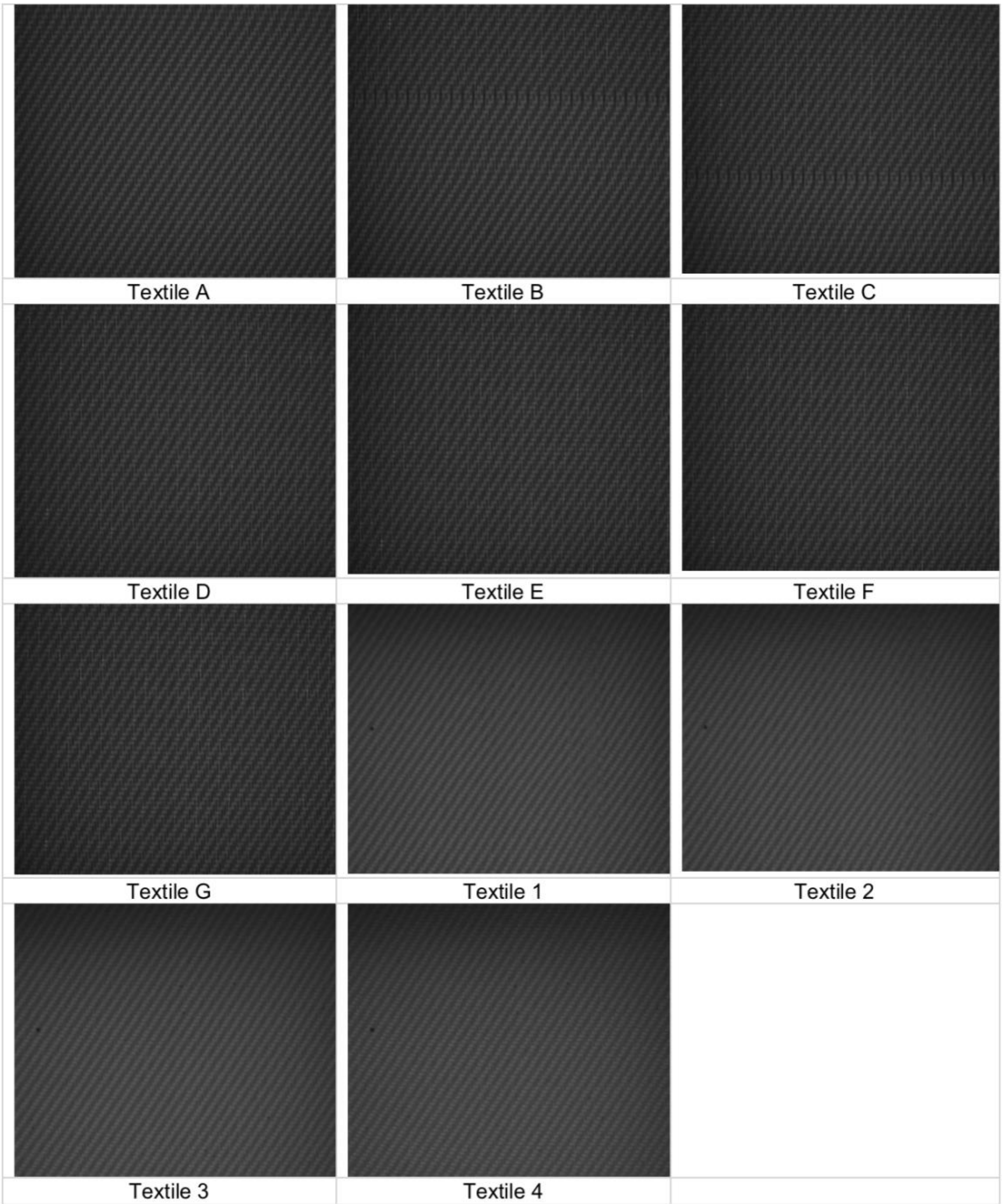

Figure 4.2: Textile images captured using OnLoom’s camera