Video-Based Wetting Detection For Blended Fabrics

Xianpeng Liu
Electrical and Computer Engineering
North Carolina State University
Raleigh, USA
xliu59@ncsu.edu

Chau-Wai Wong
Electrical and Computer Engineering
North Carolina State University
Raleigh, USA
chauwai.wong@ncsu.edu

***Abstract*—Textile scientists are seeking for automated ways to understand the wicking phenomenon of blended fabrics from recorded videos at the pixel level. In response to such need, we design a video-based method for detecting pixels that will become wet and for estimating the timestamps of wetting events, which is the first step toward characterizing the wicking phenomenon. Since the wicking behaviors of the blended fabrics can be very different from one yarn to another within a small spatial region, simple frame-level thresholding with morphological preprocessing steps does not fit this application scenario. In this paper, we analyze for each pixel the color variation along the time for the wetting event detection. We develop an iterative merging algorithm rooted from the likelihood ratio test to obtain a coarse-level timestamp. The timestamp is then refined using a parametric curve fitted to a small neighborhood. Experimental results show that our automated method can achieve satisfactory wetting detection performance when the generated binary wetting-event video is compared with the raw wicking video.**

***Index Terms*—Change-point detection, wetting, wicking, blended fabric**

I. Introduction

Understanding the wicking phenomenon of fabrics is very important to textile scientists as it affects the physiological comfort of the person whose skin is in touch with the fabric [1]–[3]. Improving the wicking-performance in fabrics has been the main goal of many segments of the textile industry such as sportswear, military apparel, and textile printing.

To develop better-performed wicking fabrics, textile scientists need a deeper understanding of the wicking mechanism of fabrics, e.g., how liquid transports within yarns and between yarns. However, the state-of-the-art theories on wicking are lacking [4], [5]. The experimental investigations are therefore needed into the yarn-level wicking behaviors. In the wicking-performance experiments conducted by our textile colleagues, blended fabrics made up of hydrophobic and hydrophilic yarns are used. In each experiment, a fabric is kept static, and colored water will be injected into one hydrophilic yarn using a needle. The colored water will propagate both i) along the fibers of the yarns, and ii) from one yarn to another. As time goes on, more and more locations that corresponds to the hydrophilic yarns of the fabric will change color due to wetting. To quantitatively measure the wicking process within the fabric, the key stage is to analyze whether and when the yarns get wet in the recorded wicking-performance videos.

In this work, we propose a video analysis method for detecting pixels that will become wet and estimating timestamps of wetting events. For each pixel in the video, its color with respect to time can be treated as a time series, and a wetting event can be defined as an abrupt change of the color in time. Detecting abrupt changes in time series can be cast into a change-point detection problem [6]–[8]. We develop a wetting event detection method by searching for the timestamp that corresponds to the fastest change in color. Instead of directly working on a change-point detection problem of a vector-valued time series, i.e., using intensity measurements from all three color channels, we first target at finding a color direction that best distinguishes the pixels that are dry and wet. Then, we develop a coarse wet event detection method to obtain a map of timestamps in quantized values. Those quantized timestamps are then refined using a parametric curve fitted to a small neighborhood of the coarse-level timestamp.

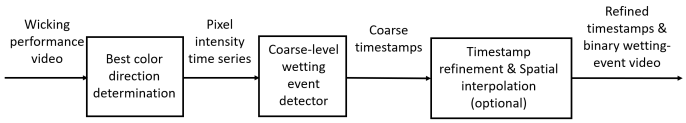

Fig. 1. Block diagram for video-based wetting timestamp detection algorithm.

II. Proposed Video-Based Wetting Event Detection Method

Our textile colleagues provided us with four typical wicking-performance videos of fabrics made through knitting or waving from different types of yarns that they recorded for wicking phenomenon studies. The experiments were recorded using consumer-grade mobile cameras. Artifacts may appear in videos, such as slight vibration of the experimental platform, camera focus change, ambient light change, etc. These artifacts will eventually become noise of various characteristics in the time series of color. A simple method of thresholding in color is not likely to work well because more than one timestamp may be generated for a pixel due to noisy time series and determining the optimal thresholds for different pixels can be challenging. Our proposed algorithm is designed to generate exactly one timestamp per pixel and resist a reasonable amount of the aforementioned noise. Fig. 1 shows the block diagram of the proposed method.

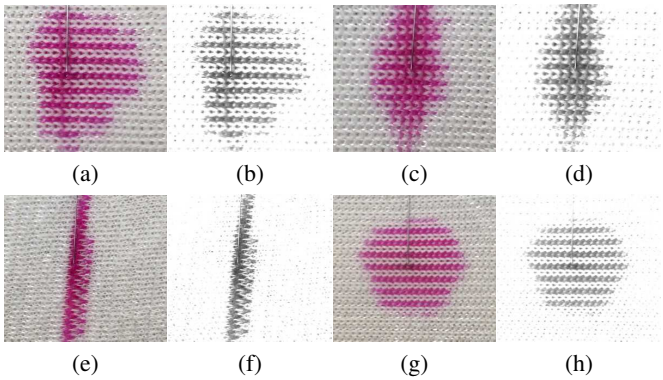

Fig. 2. Representative frames from four wicking-performance videos before and after projecting to the color channel that maximizes the contrast between dry and wet pixels: (a), (c), (e), (g) raw frames; (b), (d), (f), (h) frames after projecting to the color channel of highest contrast. (Best viewed in color)

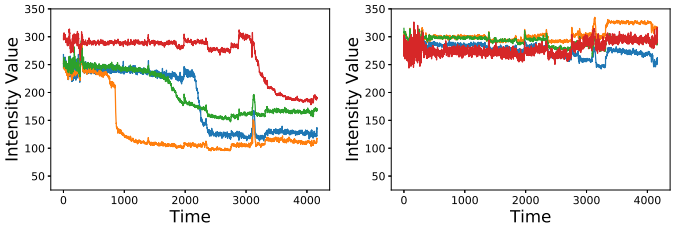

Fig. 3. Intensity of pixels as a function of time: (a) curves for pixels that will eventually get wet, and (b) curves for pixels that will never get wet. Different colors correspond to pixels at different locations.

A. Selection of Most Discriminative Color Direction

To take the most advantage of all three channels in videos and to reduce the dimensionality of the sample point of the input time series, we apply the principal component analysis with the manual inspection to determine the best color direction for the subsequent video analysis. The results of our preprocessed frames in the four test videos are shown in Fig. 2. We visualize in Fig. 3 some representative color time series projected to the selected color direction.

We downsample the video spatially to boost the signal-to-noise ratio against sensor noise and to reduce the computational load for the wetting event detector. The desired frame resolution is determined by the smallest dimension of the fabric structure, i.e., the thinnest yarn, which is about 32 pixels wide. The resolution should be at least $2\times$ the Nyquist rate to avoid aliasing. We choose $4\times$ the Nyquist rate, which leads to 1/4 downsampling factor for all videos.

In order to detect the wetting event timestamp for a pixel, we first need to investigate the difference between dry and wet pixels. For videos captured by consumer-grade mobile cameras, each pixel has red, green, and blue color values that correspond to the strength of the captured light around frequency bands of red, green, and blue light, respectively. Each channel may have different values depending on the color of the wicking liquid and the intrinsic color of the fabric. To take the most advantage of all three channels and to reduce the dimensionality of the sample point of the input time series from three to one, we can apply any dimensionality reduction method to maximize the color contrast between the dry and wet pixels.

In this very first exploration, we apply the principal component analysis with manual inspection to determine the best color direction for the subsequent video analysis. Here, manual inspection is needed to pick the best color direction because the first principal component corresponding to the largest color variation of the video data may not be due to the wicking effect. Other sources of color variation including camera motion, out-of-focus blur, camera sensor noise may cause stronger variations in color. For our four test videos of whitish fabric and pink wicking fluid, we found that the first principal component with RGB weights 0.35, 0.77, and 0.52 can capture the most of color variation due to wicking. The results of our preprocessed frame in the four test videos are shown in Fig. 2. It reveals a high contrast between dry and wet pixels. The vector-valued time series after projecting to the best color direction will be referred to as the intensity time series hereafter.

We visualize in Fig. 3 some representative time series of intensity values. It reveals that when a pixel is dry, its intensity value keeps oscillating around a nominal value. When it starts to get wet, its intensity value drops quickly. When it is completely wet, its intensity will keep oscillating around a lower nominal value. At times, some sharp peaks are observed for all pixels, which may be caused by some global noise.

B. Coarse-Level Wetting Event Detection

In the case that the fabric is perfectly still in the video and there is no noise disturbing colors of the yarns, the timestamp of the wetting event can be determined by finding the time with the largest intensity change. We define the time of the quickest drop in an uncorrupted intensity time series $u_0(x, y, t)$ as the timestamp of a wetting event, namely,

$$t_{\text{wet}} = \text{argmax}_{\text{t}}(|\partial u_0(x, y, t)/\partial t|), \tag{1}$$

where (x, y) is the spatial location of a pixel in the video, and t is frame index.

For the test videos obtained using a consumer-grade mobile camera and in conditions with unavoidable shakes of the camera and minor deformation of the fabric during the wicking process, the intensity time series, $u(x, y, t)$, can be noisy. We propose using the cumulative intensity $U(x, y, t) = \int_0^t u(x, y, \tau)d\tau$ to reduce the impact of the noise, and subsequently work on the less noisy cumulative intensity curve that has a much better linearity for the wetting event detection. Other possible solutions include lowpass filtering or fitting a parametric smooth curve to the full-length signal $u(x, y, t)$, and then detect the time with the largest slope. However, the former approach needs to design/select a filter and is more computational intensive; the latter approach is complicated because it requires multiple parameters precisely to describe the full-length curve with heterogeneity.

Algorithm 1: Coarse-level wetting event detection.

input : Cumulative intensity for a particular pixel
output: Coarse-level timestamp of the detected wetting event, $t_{\text{wet\_coarse}}$

Step 1. Initialization

Segment the time series into N equal-length pieces.
Fit a straight line to each segment. Obtain SSE R_k.
Fit a straight line to each pair of neighboring segments, i.e., kth and $(k+1)$st segments. Obtain SSE $R_{k,k+1}$.

Step 2. Merging

repeat
  Merge the two segments with smallest $\Lambda^{k,k+1}$.
until $\min_k(\{\Lambda^{k,k+1}\}) > \eta$ is true, or only 2 segments remain

Step 3. Coarse-level timestamp localization

if More than 2 segments remain
  $t_{\text{wet\_coarse}} = L \cdot \text{argmax}_k |\arg[(1+j\hat{\beta}_{k,1})/(1+j\hat{\beta}_{k-1,1})]|$
else if angle of two remaining segments $< \theta$ **then**
  The pixel did not get wet
else
  $t_{\text{wet\_coarse}}$ = index at the junction of the two lines
end

We propose to first detect a coarse timestamp using iterative merging of segments and then narrow it down using a parametric approach to find the precise timestamp of the wetting event. For iterative merging, it is motivated by a statistical test for assessing the linearity of a line segment consisting of two subsegments. We formulate a hypothesis testing problem such that $H_0 : \beta_k = \beta_{k+1}$ vs. $H_1 : \beta_k \neq \beta_{k+1}$, where $\beta_k = (\beta_{k,0}, \beta_{k,1})$ contains coefficients of the kth line segment. We can use a likelihood ratio test (LRT) to determine whether two line segments can be considered to belong to the same line. We reject the null hypothesis H_0 if $\Lambda^{k,k+1} = \frac{R_k + R_{k+1}}{R_{k,k+1}} > \eta$, where R_k is the sum of squared error (SSE) for fitting the kth segment, and $R_{k,k+1}$ is the SSE for the combined segment. We show below that we can use a constant threshold η to achieve high detection power for segments k and $k+1$ of arbitrary lengths. The LRT is equivalent to an F-test whose degree of freedoms is 2 and $n_k + n_{k+1}$ [9], where n_k and n_{k+1} are the length of two segments, respectively. Since $n_k + n_{k+1}$ is usually very large, the F-test statistic is not a function of n_k and n_{k+1}, and therefore, a constant η can achieve high detection power. Note that the test statistic $\Lambda^{k,k+1}$ characterizes the degree of linearity of the combined segment: $\Lambda^{k,k+1}$ will be large if combined segment has bad linearity.

Iterative Merging of Short Segments The coarse-level wetting event detection is to find the range of the wetting event at low computational complexity. The pseudo code for our coarse-level wetting event detection is shown in Algorithm 1. We first cut the whole time series uniformly into N segments. Each segment is fitted by a straight line, and we calculate SSE for each fitted line, denoted as $R_1, R_2, ..., R_N$. Then, we also fit every two neighboring segments with a straight line, and calculate SSE for each fitted line, denoted as $R_{1,2}, R_{2,3}, ..., R_{N-1,N}$.

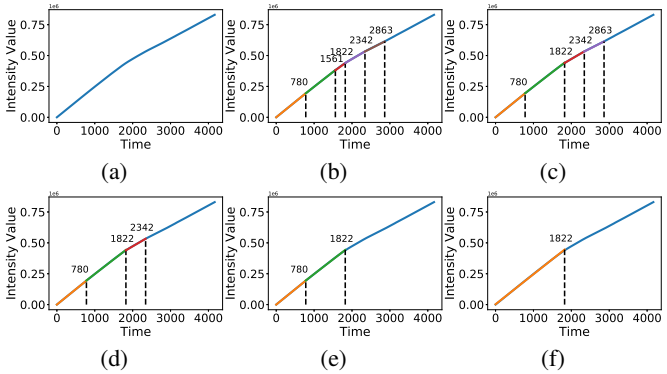

Fig. 4. Demonstration of the merging process for the segmented cumulative intensity curve of a pixel: (a) the raw curve, (b)–(f) last five merging steps from six to two segments. This example shows that by successively merging segments of strongest linearity, the region of the largest convexity will stand out, and the location of the final junction will be considered to be coarse-level wetting timestamp. (Best viewed in color.)

At every merging step, we merge only the two segments that has the smallest $\Lambda^{k,k+1}$. This ensures that we only merge the segments that have the best linearity. We use the hyper-parameter η to stop merging if all of the segments left have bad linearity. Fig. 4 shows an example of the merging process. It reveals that by successively merging segments of strongest linearity, the region of the largest convexity will stand out, and the location of the final junction will be considered to be coarse-level wetting timestamp.

Note that the correct use of the LRT requires each segment to be an approximated straight line. This requires us to merge the segments in a bottom-up approach instead of a top-down approach. If we are to find the wetting timestamp in a top-down approach, it is highly likely that the segments being tested are not approximated straight lines. This will violate the LRT's assumption and may render the test less powerful.

Coarse-Level Timestamp Localization We decide that a pixel does not get wet if the merging process stops at two remaining segments and the angle of two segments is smaller than a threshold θ. We determine the coarse-level timestamp by searching for the point that has the largest change in the angle of the fitted lines, namely, $t_{\text{wet\_coarse}} = L \cdot \text{argmax}_k |\arg[(1+j\hat{\beta}_{k,1})/(1+j\hat{\beta}_{k-1,1})]|$, where $j^2 = -1$, $\hat{\beta}_{k,1}$ is the estimated slope of the kth segment, and $L =$ Time Series Length$/N$.

C. Timestamp Refinement Using Parametric Curves

We refine the coarse timestamps for those pixels that are detected to be wet. By the design of the coarse-level merging based detection method, the timestamp of the quickest change lies within a small range around the coarsely detected timestamp. We should search for the time that corresponds to the fastest drop in intensity, or the time that corresponds to the most negative convexity in the cumulative intensity curve. To reduce the impact of the noise, we can parameterize the cumulative intensity and find the analytic expression for time of the most negative convexity. To ensure there exists a smallest value in the convexity function within an open

interval, we may design the convexity function to be a 2nd-order polynomial concave upward with its vertex below x-axis, and in turn, to model the cumulative intensity as a 4th-order polynomial.

Specifically, we use the time series in the range of $2L$ around the coarsely detected point as input, fit a 4th-order polynomial curve $y(t) = \beta_0+\beta_1 t+\beta_2 t^2+\beta_3 t^3+\beta_4 t^4$, and treat the timestamp of largest convexity of the parameterized cumulative curve or the largest slope of the parameterized intensity curve, namely, $t = \beta_3/(4\beta_4)$, as the final timestamp of the detected wetting event. We choose the 4th-order polynomial because we need its convexity (or its 2nd-order derivative) to potentially have an extreme value within an open interval.

It should be noted that in most cases, the aforementioned approach is effective in producing refined timestamps. However, it cannot deal with the cases that the raw intensity drops fastest at either end of the refinement interval. In some rare cases, it may also fail to generate a parametric convexity curve that is concave upward due to strong noise. In this case, no most negative convexity can be detected hence no refinement output is available.

For a small percentage of pixels that the refinement fails, we obtain the interpolated timestamp using those timestamps in its spatial neighborhood. Noted that interpolated results may be wrong if the neighboring pixel used for interpolation is from another yarn which is not in the same wicking status. Depending on the needs of textile scientists, both the fine-tuned results with missing points or the fine-tuned results with interpolated points can be useful.

The wetting event timestamp matrix can be visualized using a binary wetting-event video that for each pixel, at most one intensity-flip event from 0 to 1 can occur: $\mathrm{BV}(x, y, t) = \mathbb{1}[t(x, y) > t_{\mathrm{wet}}(x, y)]$, where $\mathbb{1}(\cdot)$ is the indicator function.

III. Experimental Results

As discussed in Section II, the performance of the whole detection system is determined by various steps, such as the choice of the color direction for projection, parameters of coarse-level detection, the parameterization at the fine-tune stage, etc. Among these factors, the performance of the coarse-level detection which infers from the cumulative intensity curve the rough location of the wetting event is the most important.

For the coarse segmentation part, the number of segments, N, is set to be 16. The threshold η for the LRT is set to be 0.005. Using these parameters, the results of coarse-level wetting timestamp map are shown in the first column of Fig. 5. Compared to the frames toward the end of each video showing snapshots of relatively stable wetting patterns, the coarse-level wetting timestamp map is dominantly correct on the locations of the wetting pixels. However, since the coarse timestamps are quantized, the binary wetting-event videos do not look smooth.

The results of the refined timestamp map are shown in the second column of Fig. 5, in which the quantizer timestamps are refined to take values in $\mathbb{R}_+$. Compared to the coarse-level detection method, a small number of points fail to refine. The reason is discussed in Section II-C. In fact, this is acceptable as wicking happens continuously in space, textile scientists are able to exploit information from neighboring locations. The binary wetting-event videos for the refined timestamps are smooth but they miss a few locations.

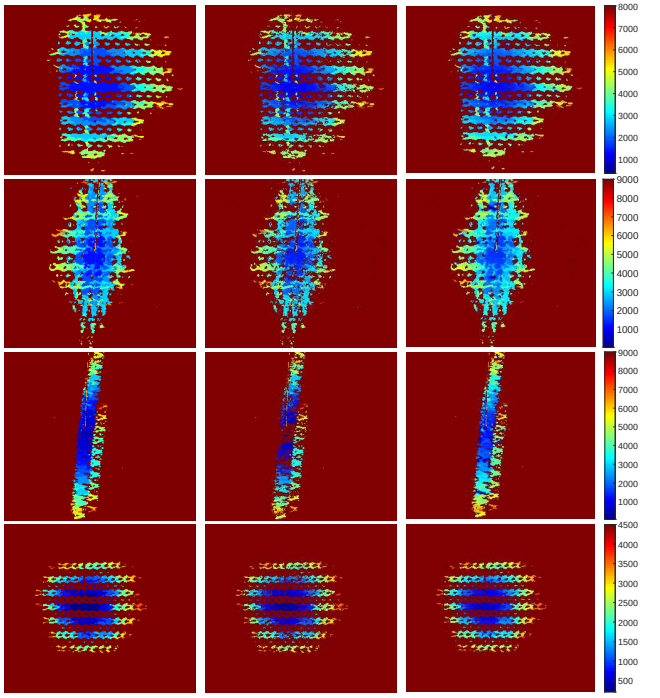

Fig. 5. Wetting events detection results for four test videos. First column: coarse-level wetting timestamp map. Second column: refined timestamp map with missing points. Third column: refined timestamp map with interpolation. Binary wetting-event video demo: goo.gl/SLLDuC. (Best viewed in color.)

The results of the interpolated timestamp map are shown in the third column of Fig. 5. The failed-to-detect locations in the second column of Fig. 5 are filled with values calculated using neighboring timestamps. It is found that most of the interpolated timestamps are consistent with the wetting time of the videos. At this initial stage, we compare our results with raw videos in a qualitative manner and textile scientists are satisfied with the results. In our future work, we plan to collect ground-truth labels for assessing the quantitative performance of the proposed algorithm.

Below we examine the processing time of our proposed method. Our method is inherently parallelizable: when detecting wetting timestamps, the program can run in parallel, which greatly reduces processing time. For a one-minute video, it takes about 1.5 hours for coarse-level wetting detection. For each pixel, it takes 30–50 ms to complete coarse-level wetting detection. The time needed for fine-tuning and interpolation is negligible compared to that of the coarse-level detection.

IV. Conclusion

In this paper, we proposed a wetting detection method for blended fabric using wicking-performance videos. We have analyzed the color variation along the time to detect the wetting

event for each pixel location. We have developed a coarse-level wet event detection method to obtain a map of timestamps in quantized values. Those quantized timestamps were then refined using a parametric curve fitted to a small neighborhood. Experimental results showed that our automated algorithm can achieve satisfactory wetting detection performance when the generated binary wetting-event video is qualitatively compared with the raw wicking video.

ACKNOWLEDGMENT

We thank Ms. Hey-sang Kim, Dr. Stephen Michielsen, and Dr. Emiel DenHartog for providing the background knowledge and the sample videos.

REFERENCES

[1] J.-K. Davis and P. A. Bishop, "Impact of clothing on exercise in the heat," *Sports Medicine*, vol. 43, no. 8, pp. 695–706, Aug. 2013.

[2] L. G. Berglund and R. R. Gonzalez, "Evaporation of sweat from sedentary man in humid environments," *Journal of Applied Physiology*, vol. 42, no. 5, pp. 767–772, May 1977.

[3] D. Pascoe, T. Bellingar, and B. McCluskey, "Clothing and exercise. II. Influence of clothing during exercise/work in environmental extremes," *Sports medicine*, vol. 18, no. 2, pp. 94–108, Aug. 1994.

[4] T. L. Owens, J. Leisen, H. W. Beckham, and V. Breedveld, "Control of microfluidic flow in amphiphilic fabrics," *ACS Applied Materials & Interfaces*, vol. 3, no. 10, pp. 3796–3803, Oct. 2011.

[5] B. Das, A. Das, V. K. Kothari, and R. Fangueiro, "Mathematical model to predict vertical wicking behaviour. Part II: Flow through woven fabric," *The Journal of The Textile Institute*, vol. 102, no. 11, pp. 971–981, Mar. 2011.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 15:1–58, Jul. 2009.

[7] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 5, pp. 363–387, Oct. 2012.

[8] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, Jun. 2014.

[9] H. Scheffe, *The Analysis of Variance*. New York: Wiley, 1959, ch. 2.