Video-Based Wetting Detection For Blended Fabrics

Xianpeng Liu and Chau-Wai Wong

***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

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 [1]–[3]. 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. 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.

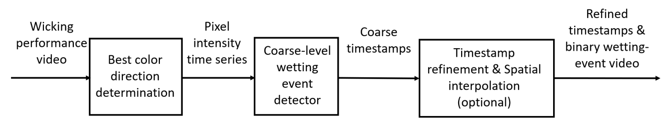

Fig. 1: Block diagram for the video-based wetting timestamp detection algorithm.

II. Summary of the Proposed Approach

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. 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.

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 frame 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.

B. Coarse-Level Wetting Event Detection

Based on the observations of the time series in Fig. 3, we define the time of the quickest drop in an uncorrupted version intensity time series $u(x, y, t)$ as the timestamp of a wetting event, namely,

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

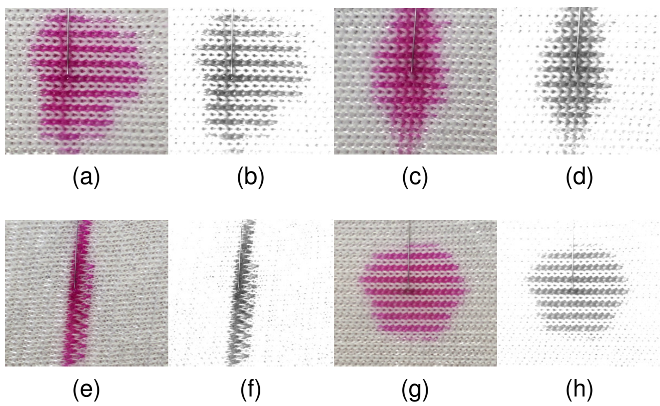

Fig. 2: Representative frames from four wicking-performance videos before and after projecting to the color channel that maximizes the contrast between dry and wet pixels: (a), (c), (e), (g) raw frames; (b), (d), (f), (h) frames after projecting to the color channel of highest contrast. (Best viewed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

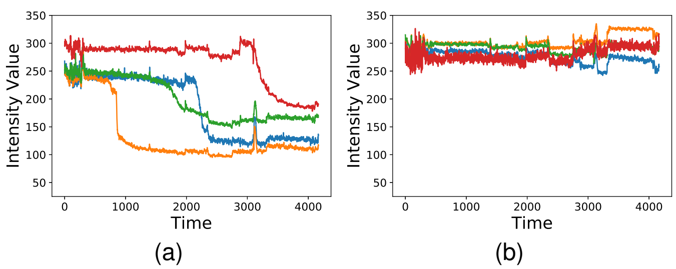

Fig. 3: Intensity of pixels as a function of time: (a) curves for pixels that will eventually get wet, and (b) curves for pixels that will never get wet. Different colors correspond to pixels at different locations.

where (x, y) is the spatial location of a pixel in the video, and t is frame index. Since the curves shown in Fig. 3 are noisy and (1) performs poorly in noisy scenarios, we propose to use the cumulative curve for reducing the impact of the noise, and subsequently to work on the less noisy cumulative intensity curve that has a much better linearity for the wetting event detection.

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. It is easy to show that one can use a constant threshold of η to achieve high detection power for segments k and $k+1$ of lengths that are not too short. Note that the test statistic

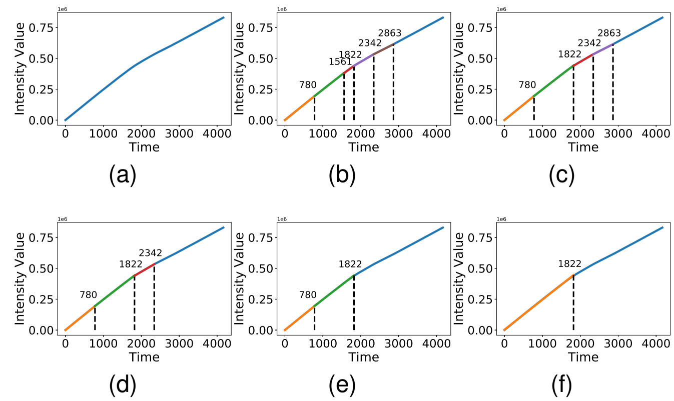

Fig. 4: Demonstration of the merging process for the segmented cumulative intensity curve of a pixel: (a) the raw curve, (b)–(f) last five merging steps from six to two segments. This example shows that by successively merging segments of strong linearity, the region of the largest convexity will stand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

$\Lambda^{k,k+1}$ characterizes the degree of linearity of the combined segment: $\Lambda^{k,k+1}$ will be large if the 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. At every merging step, we merge only the two segments that have the smallest $\Lambda^{k,k+1}$. This ensures that we only merge the segments that have the best linearity. We use the hyperparameter 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 strong linearity, the region of the largest convexity will stand out, and the location of the final junction will be considered to be the coarse-level wetting timestamp.

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, $t_{\text{wet-coarse}} = L \cdot \text{argmax}_i |\arg[(1 + j\hat{\beta}_{i,1})/(1 + j\hat{\beta}_{i-1,1})]|$, where $j^2 = -1$, $\hat{\beta}_{i,1}$ is the estimated slope of the ith segment, and $L = \text{Time Series Length}/N$.

C. Timestamp Refinement Using Parametric Curves

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. Specifically, we use the cumulative curve 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

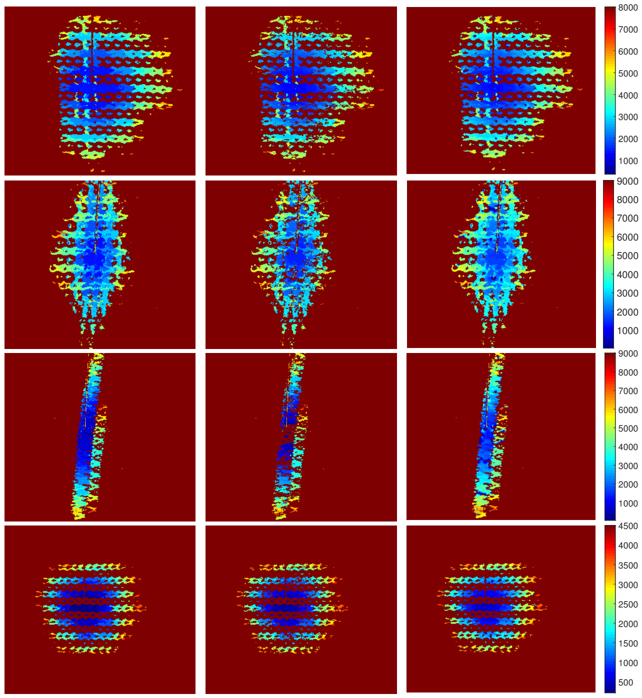

Fig. 5: Wetting events detection results for four test videos. First column: coarse-level wetting timestamp map. Second column: refined timestamp map with missing points. Third column: refined timestamp map with interpolation. Binary wetting-event video demo: goo.gl/SLLDuC. (Best viewed in color.)

because we need its convexity (or its 2nd-order derivative) to potentially have an extreme value within an open interval. For a small percentage of pixels that the refinement fails, we obtain the interpolated time stamp using those timestamps in its spatial neighborhood.

III. Experimental Results

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. 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.

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 ongoing work, we will 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 compared with the raw wicking video.

References

[1] J.-K. Davis and P. A. Bishop, Impact of clothing on exercise in the heat, Sports Medicine, vol. 43, no. 8, pp. 695706, 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. 767772, 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, p. 94108, 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. 37963803, 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. 971981, 2011.

[6] V. Chandola, A. Banerjee, and V. Kumar, Anomaly detection: A survey, ACM Computing Surveys, vol. 41, no. 3, pp. 15:158, 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. 363387, 2012.

[8] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, A review of novelty detection, Signal Processing, vol. 99, pp. 215249, 2014.