Information Processing and Management 52 (2016) 989–1003

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Strokes of insight: User intent detection and kinematic compression of mouse cursor trails

Daniel Martín-Albo[a], Luis A. Leiva[b,1,*], Jeff Huang[c], Réjean Plamondon[d]

[a] Universitat Politècnica de València, 46022 Valencia, Spain
[b] Sciling, 46022 Valencia, Spain
[c] Brown University, Providence, RI 02912, United States
[d] École Polytechnique de Montréal, QC 06079, Canada

ARTICLE INFO

Article history:
Received 28 December 2015
Revised 6 April 2016
Accepted 13 April 2016
Available online 6 May 2016

Keywords:
Mouse cursor analysis
User intent
Segmentation
Compression
Synthesis
Kinematics
Sigma-Lognormal

ABSTRACT

Web users often have a specific goal in mind comprising various stages that are reflected, as executed, by their mouse cursor movements. Therefore, is it possible to detect automatically which parts of those movements bear any intent and discard the parts that have no intent? Can we estimate the intent degree of the non-discarded parts? To achieve this goal, we tap into the Kinematic Theory and its associated Sigma-Lognormal model ($\Sigma\Lambda$M). According to this theory, the production of a mouse cursor movement requires beforehand the instantiation of an action plan. The $\Sigma\Lambda$M models such an action plan as a sequence of strokes' velocity profiles, one stroke at a time, providing thus a reconstruction of the original mouse cursor movement. When a user intent is clear, the pointing movement is faster and the cursor movement is reconstructed almost perfectly, while the reverse is observed when the user intent is unclear.

We analyzed more than 10,000 browsing sessions comprising about 5 million of data points, and compared different segmentation techniques to detect discrete cursor chunks that were then reconstructed with the $\Sigma\Lambda$M. Our main contribution is thus a novel methodology to automatically tell chunks with and without intention apart. We also contribute with kinematic compression, a novel application to compress mouse cursor data while preserving most of the original information. Ultimately, this work enables a deeper understanding of mouse cursor movements production, providing an informed means to gain additional insight about users' browsing behavior.

1. Introduction

A website can easily find out where a user has been on their pages through server access logs, but this yields an incomplete picture of what their users were actually doing. In contrast, page-level interactions, derived from recorded client-side mouse cursor data, provide fine-grained information about users' browsing behavior. More concretely, to find out *what* content is consumed and *how* it was consumed at the page level, the website can record richer interactions such as cursor movements and hovering, scrolling activity, and text highlighting. These interactions can then be interpreted

* Corresponding author. Tel.: +34963878172.
E-mail addresses: damarsi1@upv.es (D. Martín-Albo), llt@acm.org (L.A. Leiva), ipm@jeffhuang.com (J. Huang), rejean.plamondon@polymtl.ca (R. Plamondon).

[1] Work partially conducted while affiliated with the Universitat Politècnica de València.

http://dx.doi.org/10.1016/j.ipm.2016.04.005
0306-4573/

into higher-level behaviors like reading and marking interesting text with the cursor, quickly skimming the entire page, or moving the cursor out of the way to the side (Leiva & Huang, 2015).

While browsing a website, users often have a specific goal in mind (e.g. purchasing an item), which arises from a decision process comprising various stages (Karimi, Papamichail, & Holland, 2015; MacKay & Watters, 2012; Sellen, Murphy, & Shaw, 2002). The execution of each stage is ultimately reflected by different mouse cursor movements, since they are initiated unconsciously at first and enter consciousness afterward (Haggard, 2011). In other words, "users first focus and then execute actions" (Leiva & Vivó, 2007). How to leverage this type of movements is still a subject of debate in the research community. For example, search engines can re-rank search results using what people click as implicit feedback (either personalizing results for individuals from their click history, or using aggregated data from past searches to improve the overall ranking), e-commerce sites can learn what parts of the page deter potential customers, and social networking sites can use aggregated usage metrics to improve the usability of their application. However, not all parts of a cursor movement have an explicit goal or intention, nor does every cursor coordinate have equal value to website developers or data scientists. Therefore, is it possible to automatically detect which parts of those movements have a high probability of bearing any intent[2] and discard the parts that have no intent? Furthermore, can we estimate the intent degree of the non-discarded parts?

The Kinematic Theory (Plamondon, 1995a) provides a well-established and solid framework for the study of the production of human movements. The Sigma-Lognormal Model ($\Sigma\Lambda$M) (Plamondon & Djioua, 2006) is the latest instantiation of this framework and has been used, among others, in graphonomics (Galbally, Plamondon, Fierrez, & Garcia, 2012; O'Reilly & Plamondon, 2009), handwriting (Djioua & Plamondon, 2009; Martín-Albo, Plamondon, & Vidal, 2014), gestures (Leiva, Martín-Albo, & Plamondon, 2015; Plamondon, O'Reilly, Galbally, Almaksour, & Anquetil, 2014), or reproducing wrist movement and eye saccades (Plamondon, 1995b). Furthermore, it has been shown that a computer mouse is reliable enough to be considered as a velocity acquisition device (O'Reilly & Plamondon, 2011). Yet to date, $\Sigma\Lambda$M has not been used to study mouse cursor movements on websites. There has never been any study into treating cursor trails as the result of complex human motor control behaviors. After all, given that a mouse cursor movement is a specific class of human pointing movement, it makes sense to use the $\Sigma\Lambda$M to study user intent on websites from mouse cursor trails. Our work is the first to model the movements themselves using the Kinematic Theory.

Moreover, transforming the cursor trail into a 6-parameter model reduces the dimensionality to something that is more easily interpretable. This simpler feature set can be applied e.g. to supervised learning algorithms that train classifiers of user intent or relevance based on cursor activity.

In this article, we contribute with a novel methodology to automatically tell mouse cursor trails with and without intention apart. We also contribute with kinematic compression, a novel technique to compress mouse cursor data while preserving most of the original information. Ultimately, this work enables a deeper understanding of mouse cursor movements production, providing an informed means to gain additional insight about users' browsing behavior.

2. Related work

In this section we first review previous work on mouse cursor tracking, in order to provide the necessary background for the reader. Then we discuss previous work on modeling human movements, which laid the foundations of the Kinematic Theory. In the next section we elaborate more on such theory and show its value to determine intent from mouse cursor trails.

2.1. Mouse cursor tracking

There has been an enormous body of research investigating user interactions from mouse cursor data. The first research projects modified the Web browser to collect cursor data, mainly to identify user interest in the page. In early work, Goecks and Shavlik (2000) modified a Web browser to record themselves browsing hundreds of Web pages. They found that a neural network could predict variables such as the amount of cursor activity, which they considered surrogate measurements of user interest. Claypool, Le, Wased, and Brown (2001) developed the "curious browser", a custom Web browser that recorded activity from 75 students browsing over 2500 pages. They found that cursor travel time was a positive indicator of a Web page's relevance, but could only differentiate highly irrelevant pages. They also found that the number of clicks on a page did not correlate with its relevance, despite the intuition that clicks represent links that users found appealing. Shapira, Taieb-Maimon, and Moskowitz (2006) developed a special Web browser and recorded cursor activity from a small number of company employees browsing the Web. They found that the ratio of cursor movements to reading time was a better indicator of page quality than cursor travel distance and overall length of time that users spent on a page.

There are several services, including commercial products (e.g., ClickTale, LuckyOrange, MouseFlow) that have offered cursor tracking analytics for website operators, typically presenting the data as heatmaps and replays. These services offer cursor interaction data to website developers to apply in usability analysis as they see fit. Leiva and Vivó released smt2ϵ (Leiva & Vivó, 2013), an open source cursor tracking toolkit that allows website owners to set up their own cursor-based analytics on their website. Their toolkit has been used by numerous website developers to track the cursor interactions from their own users.

[2] We refer to "intent" as "the user has a target position in mind".

In usability settings, there has been work to analyze the cursor to learn about engagement on Web pages. Hijikata (2004) used client-side logging to monitor five subjects browsing a total of 120 pages. They recorded actions such as text tracing and link pointing using the cursor. The findings showed that these behaviors were good indicators for interesting regions of the page, around 1.5 times more effective than rudimentary term matching between the query and regions of the page. Atterer, Wnuk, and Schmidt (2006) investigated the usability of an online form through cursor analytics, Arroyo, Selker, and Wei (2006) presented visualizations of cursor trails to students who “proposed and prototyped redesigns reorganizing information where it could be easily found, and simpler to navigate,” and Arapakis, Lalmas, Cambazoglu, Marcos, and Jose (2014a); Arapakis, Lalmas, and Valkanas (2014b); Arapakis and Leiva (2016) investigated user engagement through mouse cursor movements and overall interaction activity. Leiva and Vidal (2010) have used page-level cursor interactions to cluster Web documents, and Leiva and Vivó (2013) have applied cursor tracking to predict time spent on a page, number of clicks, scroll reach, and amount of cursor movements.

In the search domain, Guo and Agichtein (2008) captured cursor movements using a modified browser toolbar and found differences in cursor travel distances between informational and navigational queries, as defined by human labelers. Furthermore, a decision tree could classify the query type using cursor movements more accurately than using clicks. Guo and Agichtein (2010) also used interactions such as cursor movement, hovers, and scrolling to accurately infer search intent and interest in search results. They focused on automatically identifying a searcher’s research or purchase intent based on features of the interaction. In a more recent paper, Guo and Agichtein (2012) look at cursor interactions after the click onto the landing page and find that these post-click interactions (e.g., cursor movements, dwell time) correlate with document relevance. They showed that a post-click behavior model is more effective than simply using dwell time for computing document relevance scores. Huang, White, and Dumais (2011) sought to understand cursor behavior on search engine results pages to understand result-level relevance and search abandonment. This work was extended by Diriye, White, Buscher, and Dumais (2012) to investigate the use of cursor interactions for classifying the reason why a user abandoned a query, whether it was because they were satisfied because they found the information they were seeking, or dissatisfied at the point of abandonment. Finally, Huang, White, Buscher, and Wang (2012) and Speicher, Both, and Gaedke (2013) modeled user cursor interactions on the search engine results pages by extending click models to compute more accurate relevance judgments for the search results.

User satisfaction prediction is another strand of research that has benefited from mouse cursor analysis. For example, a recent concern in search performance evaluation is how to derive implicit but useful indicators of user interest, as many browsing sessions have no click-through data. Inspired by recent studies in predicting result relevance based on mouse movement patterns, also known as *motifs* (Lagun, Ageev, Guo, & Agichtein, 2014), Liu et al. (2015) propose to estimate the utility of search results and the efforts in search sessions with motifs extracted from mouse movement data on search result pages.

In each of the above investigations, cursor trails have been treated as discrete points, such as the number and location of the movements or the time spent pointing. But there has never been any study into treating the cursor trail as a sequence of movements, produced by complex human motor control behaviors. As previously introduced, our work is the first to model the movements themselves using the Kinematic Theory, which can be applied to many of the goals above.

2.2. Modeling human movements

Many models have been proposed to study human movement production; e.g., models relying on neural networks (Bullock & Grossberg, 1988), equilibrium point models (Feldman, 1966), behavioral models (Thomassen, Keuss, & van Galen, 1983), coupled oscillator models (Hollerbach, 1981), kinematic models (Meyer, Smith, Kornblum, Abrams, & Wright, 1990; Plamondon, 1995a), or models exploiting minimization principles (Flash & Hogan, 1985; Neilson, 1993). Some models exploit the properties of various functions to reproduce human movements; e.g., exponentials (Plamondon & Lamarche, 1986), second order systems (Denier Van Der Gon & Thuring, 1965), gaussians (Leclerc, Plamondon, & Lorette, 1992), beta functions (Alimi, 2003), splines (Morasso, Mussa, & Ruggiero, 1983), Viviani’s curves (Viviani & Flash, 1995), and trigonometrical functions (Maarse, 1987). Among these, the $\Sigma\Lambda$M, which takes into account different psychophysiological features such as the neuromuscular response time, provides the most solid framework to date for the study of the production of human movements. In fact, Plamondon, Alimi, Yergeau, and Leclerc (1993) compared 23 different models to describe human movements and found that the lognormal approach outperformed all of the other approaches, which can be considered as successive approximations (Djioua & Plamondon, 2009).

According to the motor control theory literature, for short-duration movements, practically the entire movement will be executed as planned by the user (Glover, 2004). This makes the $\Sigma\Lambda$M an ideal approach worth of study. As illustrated in Fig. 1 and described in the next section, the $\Sigma\Lambda$M approaches complex human movements as a vectorial overlap of consecutive strokes, where each stroke has a lognormal-shaped velocity profile and is described by a set of parameters $\{t_0, \mu, \sigma, D, \theta_s, \theta_e\}$.

3. Kinematic theory

When a user executes an action, she thinks ahead where to move the mouse cursor (e.g., go to the scroll bar, click on a link, etc.). This may happen either consciously or unconsciously (Soon, Brass, Heinze, & Haynes, 2008). In the context of the

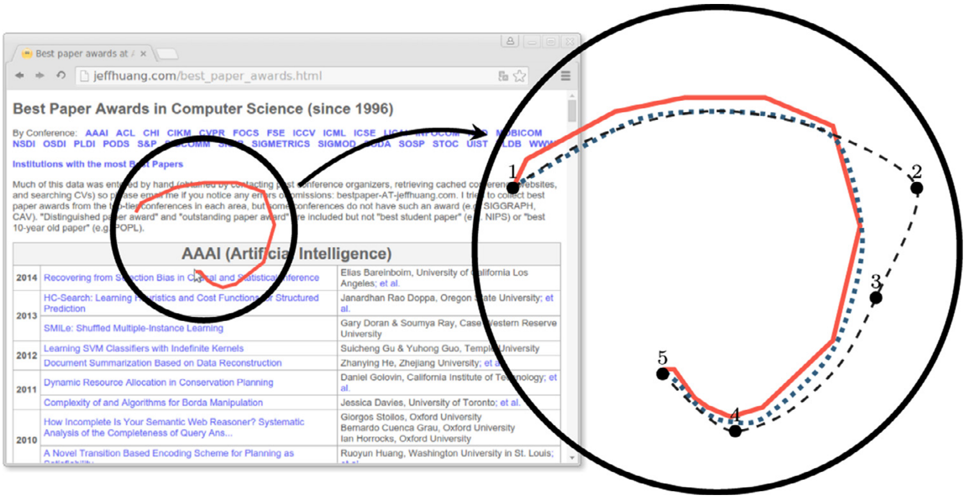

Fig. 1. Example of a cursor trail (red solid line) and its reconstruction (blue dotted line) according to the action plan of the user (dashed gray line), connecting a sequence of virtual targets (black dots). The reconstructed cursor trail is modeled by the summation of a series of lognormal strokes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Kinematic Theory, this execution is known as the "action plan" of the user. The action plan is made up of virtual targets, which are linked with circle arcs (see Fig. 1) representing a sequence of discontinuous trajectories, hereinafter referred to as *strokes*. Each stroke triggers a large number of coupled subsystems, and produces an elementary movement unit as a result, characterized by a lognormal impulse response (Plamondon, 1995b).

Mathematically, the magnitude of the velocity of the ith stroke is described by a lognormal-shaped function scaled in amplitude by a command parameter D_i and time-shifted by the time occurrence t_{0_i} of this command:

$$\begin{aligned} \|\vec{v}_i(t)\| &= D_i \Lambda(t; t_{0_i}, \mu_i, \sigma_i^2) \\ &= \frac{D_i}{\sigma_i \sqrt{2\pi}(t - t_{0_i})} \exp\left(\frac{-[\ln(t - t_{0_i}) - \mu_i]^2}{2\sigma_i^2} \right) \end{aligned} \tag{1}$$

where μ_i and σ_i define the variability of the neuromuscular execution.

The $\Sigma\Lambda$M assumes that a complex movement (e.g. a handwritten word, a signature, a gesture, or a mouse cursor trail) is generated by the temporal overlap of a series of strokes. The velocity of the human movement $\vec{v}(t)$ is computed as the vectorial summation of each stroke's velocity $\vec{v}_i(t)$:

$$\vec{v}(t) = \sum_{i=1}^{N} \vec{v}_i(t) = \sum_{i=1}^{N} \begin{bmatrix} \cos\phi_i(t) \\ \sin\phi_i(t) \end{bmatrix} D_i \Lambda(t; t_{0_i}, \mu_i, \sigma_i^2) \tag{2}$$

where the angular position $\phi_i(t)$ is obtained by:

$$\phi_i(t) = \theta_{s_i} + \frac{\theta_{e_i} - \theta_{s_i}}{2} \left[1 + \operatorname{erf}\left(\frac{\ln(t - t_{0_i}) - \mu_i}{\sigma_i \sqrt{2}} \right) \right] \tag{3}$$

being θ_{s_i} and θ_{e_i} the starting angle and the end angle of a given stroke, respectively.

Given that strokes are "hidden" in the mouse cursor movements, a $\Sigma\Lambda$M parameter extractor (Martín-Albo, Plamondon, & Vidal, 2015) is necessary to automatically uncover them. This extractor provides a set of lognormal equations (represented by their parameters) that approximates the original trajectory, described as follows.

The reconstruction of the original cursor trajectory can be computed using the following compact notation (O'Reilly & Plamondon, 2009):

$$\begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{bmatrix} = \sum_{i=1}^{N} \frac{D_i}{\theta_{e_i} - \theta_{s_i}} \begin{bmatrix} \sin\phi_i(t) - \sin\theta_{s_i} \\ -\cos\phi_i(t) + \cos\theta_{s_i} \end{bmatrix} \tag{4}$$

which indicates the evolution of the mouse cursor coordinates x, y over time.

The $\Sigma\Lambda$M reconstruction quality is usually evaluated using the signal to noise ratio (SNR) between the original and the reconstructed velocity profile:

$$\mathrm{SNR}(\mathbf{v}, \tilde{\mathbf{v}}) = 10 \log \left(\frac{\sum_1^t \|\mathbf{v}\|^2}{\sum_1^t \|\mathbf{v} - \tilde{\mathbf{v}}\|^2} \right) \tag{5}$$

Table 1
Dataset overview. The '# Events' column refers to the number of mouse cursor events selected for analysis.

# Sessions	# Events	# Pre-click logs	# Post-click logs
10,471	5,406,546	50,116	10,360

with $\mathbf{v} = \vec{v}(t) = [v_x(t)\, v_y(t)]^T$ denoting the velocity of the original signal and $\tilde{\mathbf{v}} = [\tilde{v}_x(t)\, \tilde{v}_y(t)]^T$ the velocity of the reconstruction. Previous works suggest that different SNR thresholds can be used to quantify what is considered as a good reconstruction (Martín-Albo et al., 2015), although in general values higher than 15 dB are considered to be a reference baseline (Leiva et al., 2015). In this article, we will use this baseline to determine if there is intent in a mouse cursor trail. Furthermore, we will take into account the number of lognormal strokes used in the reconstruction to measure the degree of user intent, motivated by the following discussion.

According to the Kinematic Theory, the generation of a human movement obeys the lognormality principle (Plamondon, O'Reilly, Rémi, & Duval, 2013). This principle states that a user in perfect control of his movements produces the minimum number of "perfect" lognormal strokes in order to generate the intended trajectory. In contrast, when she is not in full control, the produced strokes will not be ideal lognormals or she will use a large number of these to produce the movement. Therefore, the lognormality of velocity profiles can be interpreted as reflecting the behavior of users who are ideal motion planners, in perfect control of their movements. Taking into account these observations, the core idea in this article is to analyze mouse cursor movements by reproducing these with the $\Sigma\Lambda$M and interpret these in term of action intent.

4. Intent in cursor trails

As previously introduced, we are interested in studying sub-movement intents, so it is necessary to apply some kind of segmentation of the original mouse cursor movements. To this end, we formulated the following research questions:

1. How can a cursor trail be divided into smaller, independent chunks?
2. How well can we tell chunks with and without intention apart?

To answer the first question, we will test different trajectory segmentation techniques, to be described in Section 4.2. To answer the second question, we will apply the $\Sigma\Lambda$M.

4.1. Dataset

We used the Live dataset (Leiva & Huang, 2015) to conduct our experiments. This dataset, summarized in Table 1, is publicly available at http://hci.cs.brown.edu/mousetrap/.

The dataset consists of around 10K sessions on a web page about best paper awards in Computer Science, depicted in Fig. 1. Browsing sessions from 7K unique users were mostly of relatively short duration (one minute and a half on average per cursor trail), comprising near 5.5 M of cursor coordinates in total. Browser events that relate to user's activity (e.g. `click`, `mousemove`, `scroll`) were recorded by a background script together with timestamps and the DOM element that relates to each event. All events were captured via event listeners, which provide the maximum sampling resolution one can get from the web browser. The resulting interaction logs were stored in CSV files with companion metadata files in XML format. We should remark that users were browsing in situ in their natural environments, and the mouse cursor data were recorded in the wild. Therefore, we consider this dataset to be representative enough as to conduct our experiments.

4.1.1. Data preprocessing

We created two partitions for analysis, one with logs ending in a click event and other without, motivated by previous works that suggest that cursor points are presumably more important just before an action occurs (Huang et al., 2012; Leiva & Huang, 2015). In Section 7 we comment on the impossibility of collecting more detailed ground truth data. Fig. 2 shows how we proceeded.

Each log file contains a sequence of browser events, from which we are interested in `mousemove` and `click` events for analysis. As observed in the figure, if a log file has a `click` event, all `mousemove` events prior to the click are saved as an independent, "pre-click" log file. The remaining sequence(s) of `mousemove` events in the log are saved as a "post-click" log file.

4.2. Methods

It is possible to segment a cursor trail into a number of smaller chunks by using a series of techniques that exploit different properties of the data. The most common approach involves looking for change points in the mouse cursor trail (Keogh, Chu, Hart, & Pazzani, 2004). For example, one may assign a segment boundary whenever there is a large jump in the average value of the trail. This can be easily accomplished by setting up a threshold.

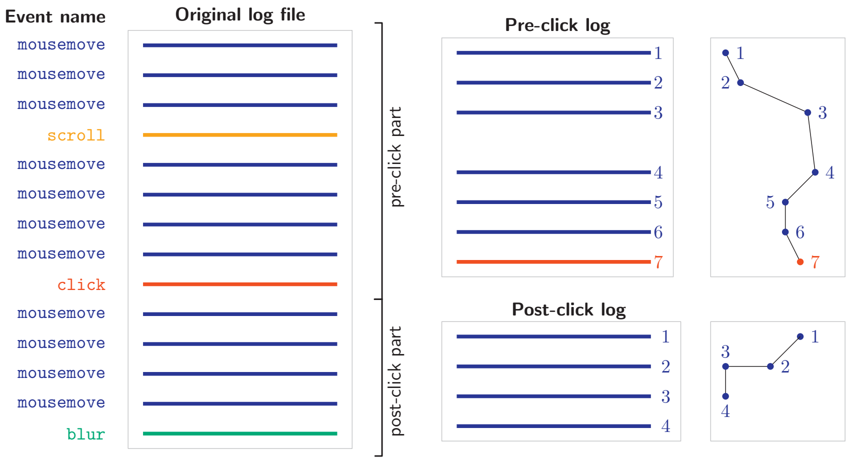

Fig. 2. Data preprocessing: from log files to cursor trails. If a log had any click events, it was divided into pre- and post-click parts, generating thus sub-log files that were then used as input trails.

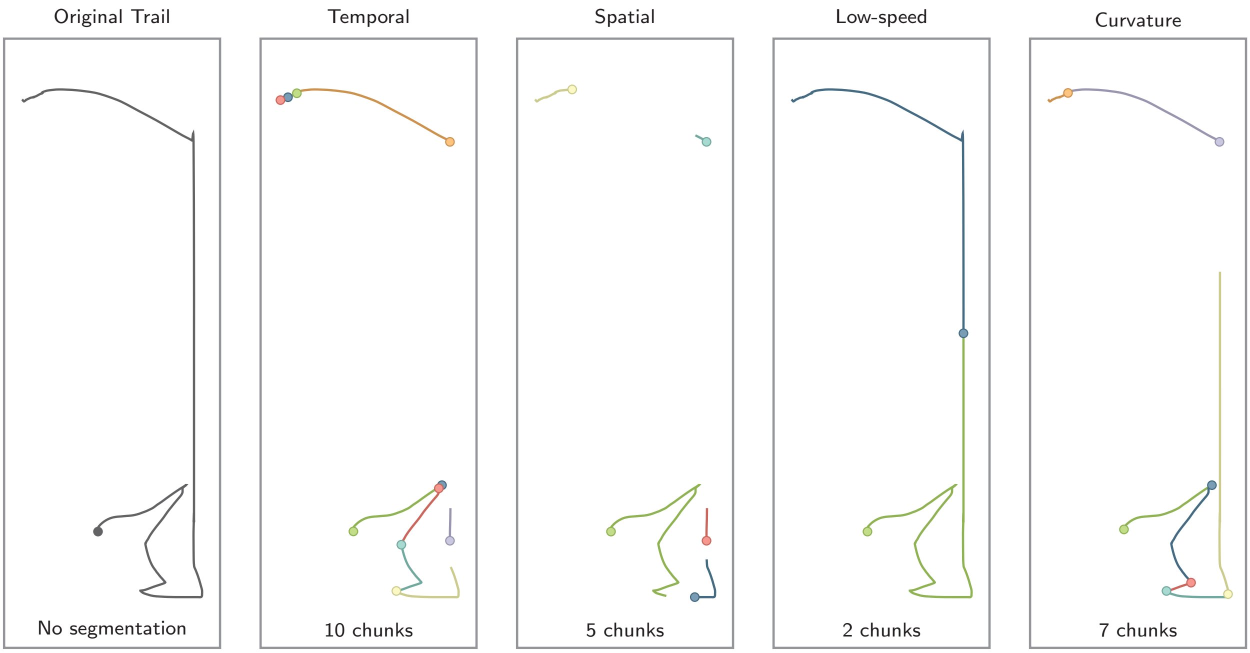

Fig. 3. Segmentation examples. A dot indicates the beginning of a cursor chunk.

Following this observation, we tested the following threshold-based segmentation techniques, because of their popularity and simplicity—they are unsupervised approaches that can be obtained in linear time, $\mathcal{O}(n)$. Fig. 3 provides an example of each technique.

Temporal Compute the time difference between consecutive trail points.
At time i, the time difference is computed as follows:

$$\Delta t_i = t_i - t_{i-1} \tag{6}$$

This technique is based on the idea that long time intervals might correspond to different actions.

Spatial Compute the Euclidean distance between consecutive trail points.
The distance for the ith point is computed as follows:

$$\Delta d_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{7}$$

This technique is based on the idea that big differences in the distance traveled by the cursor might indicate different actions.

Low-speed Compute the speed differences between consecutive trail points.
The speed for the for the ith point is calculated as follows:

$$\Delta v_i = \sqrt{\left(x'_i - x'_{i-1}\right)^2 + \left(y'_i - y'_{i-1}\right)^2} \tag{8}$$

where x' and y' refer to the first derivatives of x and y with respect to t:

$$x'_i = \frac{x_{i+1} - x_{i-1}}{t_{i+1} - t_{i-1}}$$
$$y'_i = \frac{y_{i+1} - y_{i-1}}{t_{i+1} - t_{i-1}} \tag{9}$$

This technique is based on the idea that actions start and end with close-to-zero speed.

Curvature Compute the (unsigned) curvature of each point.
The unsigned curvature for the ith point is computed as follows:

$$\kappa_i = \frac{\|x'_i \cdot y''_i - y'_i \cdot x''_i\|}{\left(x'^2_i + y'^2_i\right)^{3/2}} \tag{10}$$

where x'' and y'' are the second derivatives of x and y with respect to t:

$$x''_i = \frac{x'_{i+1} - x'_{i-1}}{t_{i+1} - t_{i-1}}$$
$$y''_i = \frac{y'_{i+1} - y'_{i-1}}{t_{i+1} - t_{i-1}} \tag{11}$$

This technique is based on the idea that important changes in direction might indicate different actions.

It is worth noting that each of these techniques produces time series with a lot of jitter. Thus, in order to avoid unreasonably segmented chunks, we apply a Savitzky-Golay filter (Savitzky & Golay, 1964) to smooth the values.

The thresholds for each of the four techniques were estimated by means of the interquartile range (IQR) technique (Tukey, 1977), a statistic method for outliers detection: it computes the 99.3% confidence interval (CI) for all values in a given metric's distribution and discards all values that fall outside the CI. We leverage this notion to choose as segmentation boundaries those points that fall outside the CI. This way, we represent a cursor trail as a sequence of chunks delimited by these boundaries. Notice that time, distance, and curvature segmentations look for values that surpass the established threshold; however low-speed looks for values that are lower than the threshold.

Once all chunks are identified, they are reconstructed with the $\Sigma\Lambda$M, obtaining a model per chunk as a result. Given that the $\Sigma\Lambda$M needs at least five points for reconstruction (Martín-Albo et al., 2015), segmented chunks with less than five points are discarded.

As discussed in Section 3, the SNR not only measures the $\Sigma\Lambda$M reconstruction quality, but also can be used to estimate if there is intent in a user movement. Concretely, when there is a clear intent, the mouse cursor trail will be better reconstructed, with high SNR; however a low SNR value will be observed when the user intent is unclear; for example, due to an involuntary or noisy movement. Overall, a chunk is considered to have no intent if the SNR of its reconstruction is less than 15 dB (Leiva et al., 2015).

Further, in order to measure the user intent degree, i.e., the *size* of the intent, we report the SNR/nbLog ratio, where nbLog is the number of lognormal strokes used in the reconstruction; i.e., N in Eqs. (2) and (4). According to the lognormality principle (Plamondon et al., 2013), a trail with high intent will have a high SNR/nbLog ratio. This is so to compensate for apparently high-quality reconstructions that however required a high number of lognormals. This outcome (high SNR *and* nbLog) is not desirable since theoretically we can approach any SNR value with infinite lognormals, thus degrading the model quality.

Nevertheless, we have noticed that there is a fundamental problem if we use this ratio to compare different segmentation techniques. Since each segmentation technique may generate trails of very different sizes, the longer the duration, the smaller the SNR/nbLog ratio will be. (The number of lognormals required to reconstruct a human movement is directly proportional to the duration of that movement.) Therefore, because SNR/nbLog has been historically adopted to quantify the user intent, it is necessary to normalize nbLog by the temporal duration of the chunk (hereinafter nbLog*) produced by each segmentation technique, making the ratio independent of the chunk duration. This way, SNR/nbLog* becomes a more general measure of user intent (the higher the better).

4.3. Evaluation

Fig. 4 provides statistics about the result of the different segmentation techniques.

As shown in the leftmost subfigure, spatial, low-speed, and curvature segmentations achieve very similar results. Notice that, after segmentation, a pre-click log file generates a number of post-click chunks but at most one pre-click chunk.

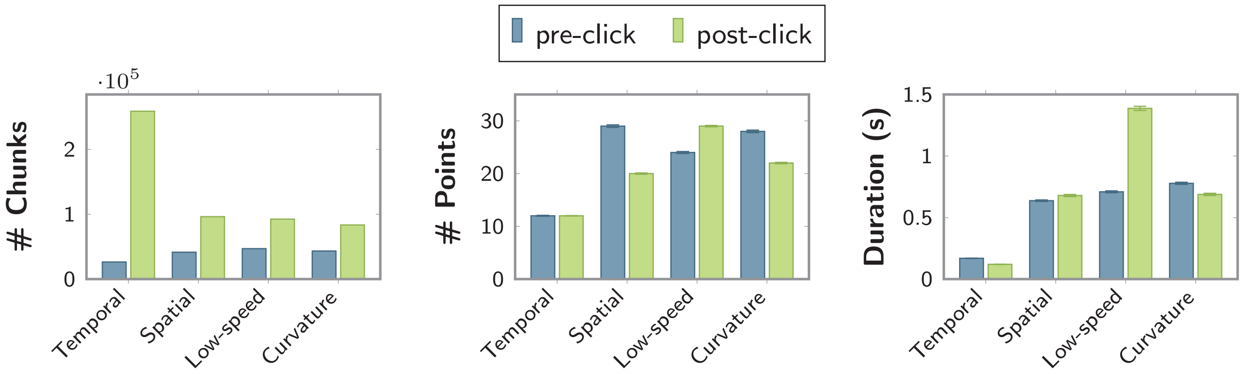

Fig. 4. Segmentation results overview. Left: Total number of segmented chunks. Middle: Average number of points per chunk. Right: Average chunk duration, in seconds. Error bars denote 95% confidence intervals.

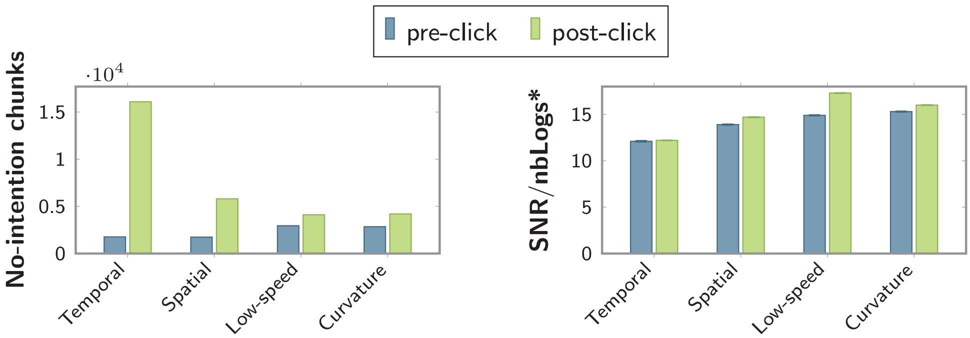

Fig. 5. Left: Number of no-intention chunks (cursor trails reconstructed with SNR $<$ 15 dB) for each segmentation technique. Right: Comparison of user intent degree. Error bars denote 95% confidence intervals.

Therefore, these techniques discarded some pre-click chunks; concretely, those that were segmented with less than five cursor points.

As observed in the middle and rightmost subfigures, temporal segmentation tends to generate very short segments (both in size and duration): around 10 points and 200 ms per chunk on average. This not only results in discarding a significantly higher number of pre-click chunks than the other techniques, but also oversegmenting post-click chunks.

Also worth mentioning, in the rightmost subfigure we can see that low-speed segmentation generates longer post-click chunks. Contrary to the other techniques, low-speed segmentation does not skip chunks with less than 5 points, and so preserves all points. This behavior can be appreciated in Fig. 3 and is explained by the fact that this technique looks for close-to-zero speed regions (local minima) so it tends to merge short-duration chunks. The other techniques, on the contrary, look for singularities in the cursor trail (local maxima).

Fig. 5 shows the experiment results pertaining user intent detection (left) and user intent degree (right). As previously stated, a chunk is considered with no intent (and therefore discarded) if its SNR is less than 15 dB. In the leftmost subfigure, it can be noticed that all techniques tend to discard more post-click chunks than pre-click chunks. This was expected, since the formers usually have less intention. Further, both temporal and spatial segmentation tend to discard less pre-click chunks than low-speed and curvature segmentation, but also detect more low-intention chunks. Concretely, temporal segmentation produced significantly more low-intention chunks than the other techniques. This is explained by the fact that this technique tends to oversegment the mouse cursor trails, creating thus many chunks of very short duration (Fig. 3), and so this technique is eventually penalized according to the SNR/nbLog* measure.

In relative terms, the temporal technique discards 0.03% of post-click chunks, whereas the spatial technique discards 0.5% of them. The low-speed technique in turn discards 2.7% of post-click chunks, and the curvature technique discards 3.1% of them. With respect to pre-click chunks, the temporal technique discards 0.04%; the spatial technique discards 0.47%; the low-speed technique discards 2.69%; and finally, the curvature technique discards 4.11%.

As shown in the rightmost subfigure, both temporal and spatial segmentations produce pre-click chunks with less intent. As previously discussed, these techniques are usually prone to oversegmentations, which causes the $\Sigma\Lambda$M to include a higher (than usual) number of lognormal strokes to properly model the cursor trail, which is penalized by the lognormality principle as a result.

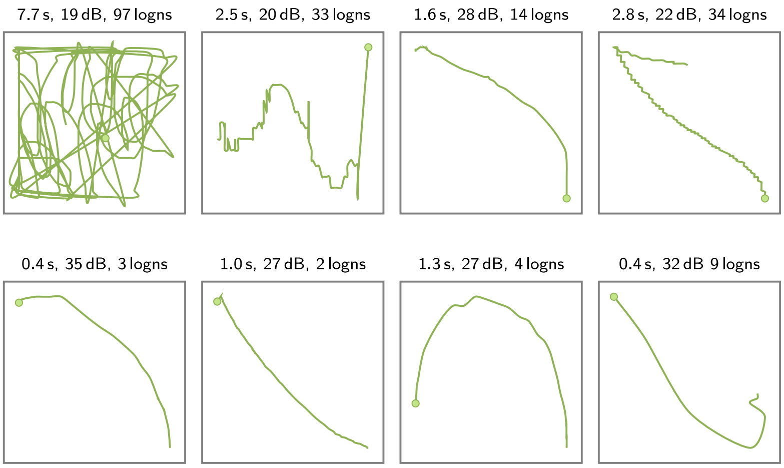

Fig. 6. Examples of cursor chunks with no intent (top row, first 2 subfigures), low intent (top row, last 2 subfigures) and cursor chunks with high intent (bottom row). A dot indicates the beginning of a cursor chunk. `logns` indicates the number of lognormals that were required to reconstruct each chunk.

Finally, Fig. 6 shows some examples of chunks with low and high intent. It can observed that low-intent chunks have higher SNR values together with higher number of lognormals. In general, we also observed that the detected cursor chunks often have longer duration when they are classified as having low-intent.

Based on these experiments, we conclude that low-speed and curvature segmentations provide the most competitive results in terms of detecting user intent. This is a reasonable outcome, since speed and curvature are anticorrelated and, in general, high curvature values are achieved at low speed values (Lacquaniti, Terzuolo, and Viviani, 1983; Plamondon, 1995a; Viviani & Flash, 1995). Although speed computation is simpler and less noisy than curvature computation, we recommend using any of them both as a preprocessor prior to reconstructing cursor trails and as a technique to reliably capture user intent.

5. Kinematic compression

In this section we provide a practical application of this work, beyond the scope of the previous experiments. Because after modeling each mouse cursor chunk we get a set of equations that approximates the original trajectory, we can transmit the model parameters over the network or store them in a database, instead of dealing with the raw cursor data, saving thus bandwidth and improving storage space. We have named this approach “kinematic compression”, and can be considered a lossy compression technique.

Furthermore, given that the reconstruction of a cursor chunk is computed as a summation of lognormal strokes, some of those strokes could be discarded if they do not contribute strongly to the reconstruction of the cursor chunk. In fact, it has been found that some lognormals are actually modeling signal noise introduced by the computer mouse (Martín-Albo et al., 2015). To this end, we formulated the following research questions:

1. To which extent can a cursor trail be summarized by their $\Sigma\Lambda$M parameters?
2. How much information can be preserved after compression?

To answer the first question, it seems intuitive to use a reconstruction quality measure such as the SNR. However, in order to better frame our results with previous work, we will look at the “trail replication” measure (Leiva & Huang, 2015), which is a per-pixel similarity metric between a cursor trail $\mathbf{c} = [x(t)\, y(t)]^T$ and its reconstruction $\tilde{\mathbf{c}} = [\tilde{x}(t)\, \tilde{y}(t)]^T$:

$$\tau(\mathbf{c}, \tilde{\mathbf{c}}) = \frac{1}{N}\sqrt{\sum_{i=1}^{t} (\mathbf{c} - \tilde{\mathbf{c}})^2} \tag{12}$$

Nevertheless, in the appendix we prove that SNR and τ are closely related: maximizing SNR is equivalent to minimizing τ (Theorem 1).

To answer the second question, we will look at the “compression ratio” measure.[3] Following previous work (Leiva & Huang, 2015), a compression ratio of 90% means that 100 bytes of cursor data are reduced to 90 bytes. Taking into account

[3] Defined as the ratio between the bytes of the “stringified” $\Sigma\Lambda$M parameters (a list of $\{t_0, \mu, \sigma, D, \theta_s, \theta_e\}$ objects) and the “stringified” original cursor trail (a list of x, y, t tuples). In signal processing, this ratio is called “space saving”, although in this article we will stick to the previous definition in order to compare with state-of-the-art lossy compression techniques of cursor trails (Leiva & Huang, 2015).

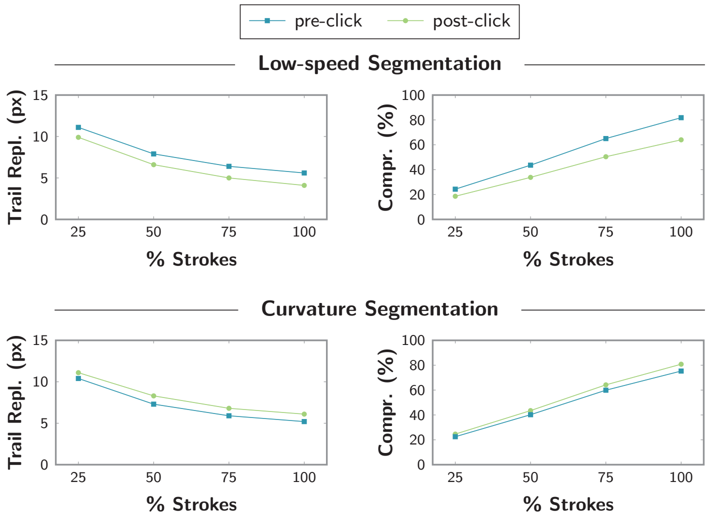

Fig. 7. Kinematic compression results with all chunks processed, no matter their reconstruction quality. 95% confidence intervals are all below 0.1%, so they are omitted.

these definitions, a good compression technique should achieve low trail replication values at low compression ratios. In both cases, the lower these measures the better.

On the other hand, different compression ratios might be useful in different situations; the situations can reflect a desire for consuming less bandwidth, more accurate replication of the original data, or a combination of both. With kinematic compression, different compression ratios can be achieved by removing the less important strokes, so we should be able to reconstruct the original cursor chunks with similar quality. Of course, as more strokes are used for reconstruction, the smaller the trail replication will be, because the reconstruction would approach better the original cursor chunk, at the cost of a higher compression ratio. Thus, we should seek a good balance between reconstruction quality and compressed data size.

5.1. Dataset

We used the Live dataset (Section 4.1) as well as the segmented chunks produced by the best segmentation techniques from the previous evaluation; c.f. low-speed and curvature segmentation.

5.2. Methods

We conducted two experiments to show the value of kinematic compression. In the first one, all cursor chunks are compressed, no matter their reconstruction quality. In the second one, only chunks with high reconstruction quality (SNR $>$ 15 dB) are compressed. In addition, we test different compression ratios by removing the less important strokes. The contribution of each stroke is estimated by taking into account the values of its command parameter D_i, which correlates to the lognormal amplitude. More precisely, D_i is the area under the lognormal curve (Section 3). Thus, for each chunk, lognormals are sorted in descending order by D_i and the top N% is kept, $N \in [25, 50, 75, 100]$.

Notice that the trail replication measure should take as input normalized cursor chunks, since computing a cursor distance of, say, 100 px in a 1600 px wide viewport is not comparable to computing the same distance in a 640 px wide viewport. Therefore, to perform such normalization we scale down the biggest dimension of a cursor chunk's bounding box to a size of 100 px, while keeping the aspect ratio for the other dimension.

5.3. Evaluation

The results of the first experiment are shown in Fig. 7, where all chunks are compressed, no matter their reconstruction quality. The "% Strokes" axis indicates the percentage of lognormals used to reconstruct each chunk. As can be observed, both techniques compress cursor chunks with similar accuracy: from 10 to 5 px of trail replication with 25% and 100% of the strokes, respectively. In relative terms, this represents an error margin comprised between 1% and 0.5%.

In the second experiment we examined the behavior of kinematic compression when only chunks with high reconstruction quality (SNR $>$ 15 dB) are considered. Fig. 8 summarizes the results. In this case, poorly reconstructed chunks are replaced by their original cursor chunk, since they would not be considered for analysis in a real-world scenario, contributing therefore with trail replication of 0 px and 100% compression ratio. This explains why the trail replication is much

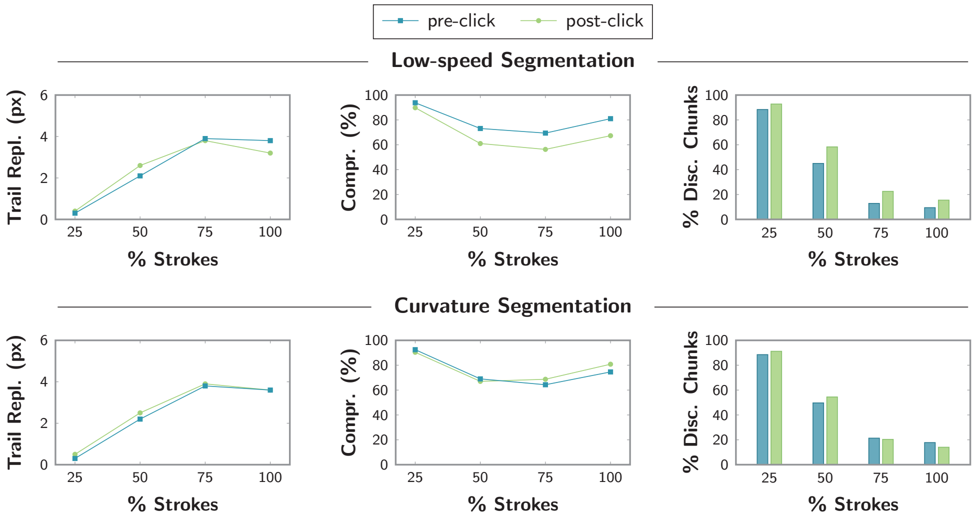

Fig. 8. Kinematic compression results where only those chunks with SNR > 15 dB are processed. 95% confidence intervals are all below 0.1%, so they are omitted.

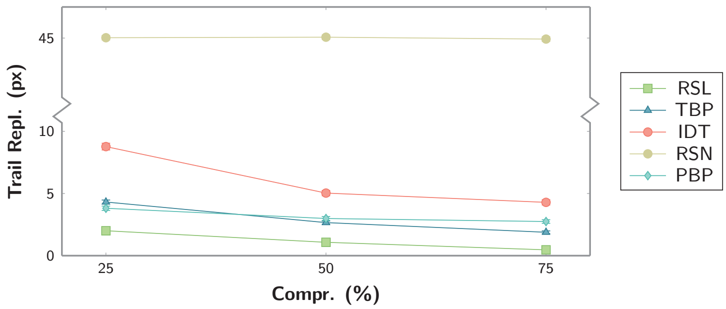

Fig. 9. Reproducing the trail replication results from (Leiva & Huang, 2015), normalized as in this article.

lower for compression ratios below 75%. In general, as observed in the figure, a significant number of strokes is discarded for compression ratios below 75%, which artificially lowers the trail replication results. Taken together, these results suggest that there exists an interesting quality-compression tradeoff when about 75% of the original strokes are preserved.

By way of comparison, Fig. 9 reproduces five state-of-the-art lossy compression techniques from previous work (Leiva & Huang, 2015): linear resampling (RSL), time-based polling (TBP), distance thresholding (IDT), non-linear resampling (RSN), and pause-based polling (PBP). We should remark that our technique cannot be plotted together in this figure because the processed information is different. On the one hand, the kinematic compression technique uses as compression ratio the percentage of *strokes*, since it is what $\Sigma\Lambda$M can reproduce. On the other hand, the lossy compression techniques use as compression ratio the percentage of *points*, since these techniques aim to simplify cursor trails by removing redundancy and non-significant information, selecting thus a subset of the original cursor points.

As shown in Fig. 9, the lossy compression techniques achieve good trail replication results, excepting RSN which performed worse than kinematic compression itself. However, we should point out that there is more to these results than a mere comparison of compression ratio vs. trail replication vs. percentage of preserved strokes and points. For example, $\Sigma\Lambda$M not only allows to reproduce the original cursor trail, but also to generate any number synthetic samples from the modeled data. In Section 7 we elaborate more on this topic and discuss further applications.

6. Limitations

The $\Sigma\Lambda$M can be used to discard mouse cursor movements that have not been planned by the user, which presumably do not have a clear intent. Overall, the tested segmentation techniques are able to detect more intent in the pre-click chunks. This is expected, since a user click has been typically planned beforehand. However, some of these clicks could have been accidental. Unfortunately, the dataset we used in our experiments does not have such fine-grained ground truth labels and thus this experiment is left as an opportunity for future work. We elaborate more on this issue in the next section.

Our experiments are based on a single dataset, which was collected on a page designed to attract academic researchers. Therefore, it could not reflect the behavior patterns of “typical” Web users. However, we have tested other general-purpose

Table 2
Perfomance overview of kinematic compression (seconds to compress) using the low-speed segmentation technique on 135 710 cursor trails.

Mdn	Mean	SD	Min	Max
80.00	665.1	2693.2	0.0	35919.0

mouse cursor datasets in previous work (Leiva & Huang, 2015) and have not noticed significant differences in comparison to the Live dataset, research-wise. That said, we find it interesting to analyze other public datasets and see whether our methodology for user intent detection achieves a similar outcome.

Finally, we should mention a limitation of the applicability of this work. Our current implementation of kinematic compression is still a prototype and so it is rather slow. For example, using a regular computer (i5 CPU, 4GB RAM) it takes a minute and a half to compress a cursor trail of about 50 points. Table 2 provides a summary of kinematic compression performace using the low-speed segmentation technique. Therefore, a development effort is needed in order to deploy this technique on the browser. However, the segmentation techniques are performance-friendly (Section 4.2), and relatively simple to implement as a client-side script. In fact we plan to make it so with the $\Sigma\Lambda$M in the future.

7. General discussion

This work is based on the solid foundations of the Kinematic Theory, which states that the generation of a human movement obeys the lognormality principle: a user in perfect control of his movements produces a minimum number of strokes in order to generate the intended trajectory. The $\Sigma\Lambda$M takes into account different psychophysiological features to model these strokes by means of their lognormal velocity profiles. Our experiments show that the temporal overlap of these velocity profiles allows for a fairly accurate reconstruction of the original human movement, in this case as reflected by their mouse cursor trails.

Based on the reconstruction quality of a mouse cursor trail, it is possible to discriminate among intended and unintended movements. However, to prove this hypothesis in the most stringent way, two studies should be conducted. First, a reliable and effective trajectory segmentation and reconstruction methodology has to be implemented and validated. Second, the results of this methodology should be validated with more detailed ground truth data. This article has focused on the first part and succeeded. The second part, as previously mentioned, would required fine-grained ground truth labels which cannot be practically collected, even with sophisticated techniques such as electroencephalography (Libet, Gleason, Wright, & Pearl, 1983). For instance, we should have each cursor trail annotated at different submovement levels with millisecond precision, each level having an “intended” or “unintended” labels. Nevertheless, from an operational point of view, the lognormality principle provides a nice operational approach to analyze this problem. The principle having been confirmed e.g. through children learning, adult aging, and getting affected by health problems (Plamondon et al., 2013), there is no reason to believe that it will not apply to the context of this work. Therefore, we can skip the second requirement and immediately use our methodology in various practical applications where an estimate of the user’s intent is important.

Regarding kinematic compression, we should point out the fact that $\Sigma\Lambda$M not only allows to reproduce the original cursor trail, but also to generate any number synthetic samples from the modeled data (Galbally et al., 2012; Leiva et al., 2015; Martín-Albo et al., 2015). Therefore, kinematic compression provides a fundamental advantage over today’s cursor lossy compression techniques. For instance, instead of collecting every user’s behaviors, only a fraction of these could be sent to the web server for later analysis, saving thus bandwidth and improving client-side performance (since not all interactions would be captured, only those that bear some intent).

This article is the first to introduce a method to transform the feature space of cursor trails; prior cursor-based applications analyzed the raw trail directly. Being able to transform a cursor trail with hundreds or thousands of `{x,y,t}` points into a 6-parameter feature set is a useful first step for training supervised learning algorithms. Furthermore, our method can be used to automatically label the intent of small cursor submovements, which has proven difficult to perform in practice, as previously discussed.

As we are now able to capture realistic motions into a few parameters due to the dimensionality reduction, there are a few potential applications for developers. For example, it makes it easier to detect user intent, which can be combined with existing work in endpoint prediction (Lank, Cheng, & Ruiz, 2007) to determine the next pages that need to be loaded. This allows browser developers to do predictive loading of pages that will be clicked and prerender them in the browser. The result to the end user is a reduced time for pages to load when the prediction is correct. Another application is the ability to cluster user intents together, since trails originally composed of hundreds of coordinates are instead represented by a few $\Sigma\Lambda$M parameters. This allows developers to group together trails and users in order to perform behavioral analytics across similar groups.

In sum, we leverage a verified theory that makes practical predictions about user intent, the problem being how to implement this in an effective methodology, which is the core contribution of this article.

8. Conclusion and future work

This article provides a solid operational methodology to detect user intent, being able to tell mouse cursor chunks with and without intention apart by means of the Kinematic Theory and its associated Sigma-Lognormal model. As a practical application, we have shown that this model can also be used to compress mouse cursor data while preserving most of the original information. Ultimately, this work enables a deeper understanding of mouse cursor movements production, providing an informed means to gain additional insight about users' browsing behavior.

As discussed in the previous section, an interesting research avenue for future work is related to cursor logs preprocessing. Right now the creation of pre- and post-click logs is automatic though rather naïve (Fig. 2), so a more sophisticated technique should be devised, possibly combining eye-gaze and/or electroencephalography data, if possible. We also believe that future research effort should be directed at improving the cursor segmentation techniques themselves.

Finally, we believe that sharing our evaluations may benefit researchers and commercial analytics services. The log files derived from our experiments, 10K pre-click and 50K post-click trails worth of cursor data, are available upon email request.

Acknowledgments

We thank Joan Puigcerver for contributing to the mathematical proof presented in the appendix. This work is part of the Valorization and I+D+i Resources program of VLC/CAMPUS and has been funded by the Spanish MECD as part of the International Excellence Campus program. This work has also been supported by the Spanish MINECO under FPU scholarship AP2010-0575, and by NSERC-Canada grant RGPIN-2014-04946.

Appendix

Theorem 1. *Maximizing the signal to noise ratio (SNR) of a cursor trail velocity and its reconstruction is equivalent to minimizing the trail replication (τ) between a cursor trail and its reconstruction.*

Proof. We begin with the SNR definition (5):

$$\arg\max_{\pi} \mathrm{SNR}(\mathbf{v}, \tilde{\mathbf{v}}_{\pi}) = \arg\max_{\pi} 10 \log \left(\frac{\sum_1^t \|\mathbf{v}\|^2}{\sum_1^t \|\mathbf{v} - \tilde{\mathbf{v}}_{\pi}\|^2} \right) \tag{13}$$

where $\pi = \{t_0, \mu, \sigma, D, \theta_s, \theta_e\}$ is the set of $\Sigma\Lambda$M parameters.

The arg max operator does not change if the function is multiplied by a constant (10, in this case), since it does not depend on π, therefore we can drop that constant:

$$\arg\max_{\pi} \mathrm{SNR}(\mathbf{v}, \tilde{\mathbf{v}}_{\pi}) = \arg\max_{\pi} \log \left(\frac{\sum_1^t \|\mathbf{v}\|^2}{\sum_1^t \|\mathbf{v} - \tilde{\mathbf{v}}_{\pi}\|^2} \right) \tag{14}$$

we can also drop the log function, since $\log(\cdot)$ is a monotonically increasing function:

$$= \arg\max_{\pi} \left(\frac{\sum_1^t \|\mathbf{v}\|^2}{\sum_1^t \|\mathbf{v} - \tilde{\mathbf{v}}_{\pi}\|^2} \right) \tag{15}$$

the numerator does not depend on π either, thus

$$= \arg\max_{\pi} \left(\frac{1}{\sum_1^t \|\mathbf{v} - \tilde{\mathbf{v}}_{\pi}\|^2} \right) \tag{16}$$

given that $\arg\max_{\pi} f(\pi) = \arg\min_{\pi} \frac{1}{f(\pi)}$,

$$= \arg\min_{\pi} \left(\sum_{i=1}^{t} \|\mathbf{v} - \tilde{\mathbf{v}}_{\pi}\|^2 \right) \tag{17}$$

now, as in Eqs. (14) and (16), we can multiply by a constant that does not depend on π (in this case, $1/N$):

$$= \arg\min_{\pi} \left(\frac{1}{N} \sum_{i=1}^{t} \|\mathbf{v} - \tilde{\mathbf{v}}_{\pi}\|^2 \right) \tag{18}$$

again, the result of the arg min operator does not change if we take the root square:

$$= \arg\min_{\pi} \left(\frac{1}{N} \sqrt{\sum_{i=1}^{t} \|\mathbf{v} - \tilde{\mathbf{v}}_{\pi}\|^2} \right) \tag{19}$$

which is actually the definition of trail replication (12):

$$= \arg\min_{\pi} \tau(\mathbf{v}, \tilde{\mathbf{v}}_\pi) \tag{20}$$

and since $\mathbf{v} \propto [x(t)\ y(t)]^T$, it proves the theorem. □

References

Alimi, M. A. (2003). Beta neuro-fuzzy systems. In W. Duch, & D. Rutkowska (Eds.), *TASK quarterly journal, special issue on neural networks: 7* (pp. 23–41).

Arapakis, I., Lalmas, M., Cambazoglu, B. B., Marcos, M.-C., & Jose, J. M. (2014a). User engagement in online news: Under the scope of sentiment, interest, affect, and gaze. *JASIST, 65*(10), 1988–2005.

Arapakis, I., Lalmas, M., & Valkanas, G. (2014b). Understanding within-content engagement through pattern analysis of mouse gestures. In *Proceedings ACM international conference on information and knowledge management (CIKM)* (pp. 1439–1448).

Arapakis, I., & Leiva, L. A. (2016). Predicting user engagement with direct displays using mouse cursor information. In *Proceedings of ACM international conference on information retrieval (SIGIR)*.

Arroyo, E., Selker, T., & Wei, W. (2006). Usability tool for analysis of web designs using mouse tracks. In *Proceedings of extended abstracts on human factors in computing systems (CHI EA)* (pp. 484–489).

Atterer, R., Wnuk, M., & Schmidt, A. (2006). Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In *Proceedings of WWW* (pp. 203–212).

Bullock, D., & Grossberg, S. (1988). The VITE model: A neural command circuit for generating arm and articulator trajectories. In *Dynamic patterns in complex systems* (pp. 305–326).

Claypool, M., Le, P., Wased, M., & Brown, D. (2001). Implicit interest indicators. In *Proceedings of international conference on intelligent user interfaces (IUI)* (pp. 33–40).

Denier Van Der Gon, J., & Thuring, J. (1965). The guiding of human movements. *Kybernetik, 14*, 145–148.

Diriye, A., White, R. W., Buscher, G., & Dumais, S. (2012). Leaving so soon? understanding and predicting web search abandonment rationales. In *Proceedings of ACM international conference on information and knowledge management (CIKM)* (pp. 1025–1034).

Djioua, M., & Plamondon, R. (2009). Studying the variability of handwriting patterns using the Kinematic Theory. *Human Movement Science, 28*(5), 588–601.

Feldman, A. (1966). Functional tuning of the nervous system with control of movement or maintenance of a steady posture. *Biophysics, 11*(1), 565–578.

Flash, T., & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience, 5*(7), 1688–1703.

Galbally, J., Plamondon, R., Fierrez, J., & Ortega-Garcia, J. (2012). Synthetic on-line signature generation. Part I: Methodology and algorithms. *Pattern Recognition, 45*(7), 2610–2621.

Glover, S. (2004). Separate visual representations in the planning and control of action. *Behavioral and Brain Sciences, 27*(1), 3–24.

Goecks, J., & Shavlik, J. (2000). Learning users' interests by unobtrusively observing their normal behavior. In *Proceedings of international conference on intelligent user interfaces (IUI)* (pp. 129–132).

Guo, Q., & Agichtein, E. (2008). Exploring mouse movements for inferring query intent. In *Proceedings of ACM internartional conference on information retrieval (SIGIR)* (pp. 707–708).

Guo, Q., & Agichtein, E. (2010). Ready to buy or just browsing? Detecting web searcher goals from interaction data. In *Proceedings of ACM international conference on information retrieval (SIGIR)* (pp. 130–137).

Guo, Q., & Agichtein, E. (2012). Beyond dwell time: Estimating document relevance from cursor movements and other post-click searcher behavior. In *Proceedings of WWW* (pp. 569–578).

Haggard, P. (2011). Decision time for free will. *Neuron, 69*(3), 404–406.

Hijikata, Y. (2004). Implicit user profiling for on demand relevance feedback. In *Proceedings of international conference on intelligent user interfaces (IUI)* (pp. 198–205).

Hollerbach, J. (1981). An oscillation theory of handwriting. *Biological Cybernetics, 39*(2), 139–156.

Huang, J., White, R. W., Buscher, G., & Wang, K. (2012). Improving searcher models using mouse cursor activity. In *Proceedings of ACM international conference on information retrieval (SIGIR)* (pp. 195–204).

Huang, J., White, R. W., & Dumais, S. (2011). No clicks, no problem: using cursor movements to understand and improve search. In *Proceedings of SIGCHI conference on human factors in computing systems (CHI)* (pp. 1225–1234).

Karimi, S., Papamichail, K. N., & Holland, C. P. (2015). The effect of prior knowledge and decision-making style on the online purchase decision-making process: A typology of consumer shopping behaviour. *Decision Support Systems, 77*, 137–147.

Keogh, E., Chu, S., Hart, D., & Pazzani, M. (2004). Segmenting time series: a survey and novel approach. In *Data mining time series databases. vol. 57. ch. 1* (pp. 1–21).

Lacquaniti, F., Terzuolo, C., & Viviani, P. (1983). The law relating the kinematic and figural aspects of drawing movements. *Acta psychologica, 54*(1), 115–130.

Lagun, D., Ageev, M., Guo, Q., & Agichtein, E. (2014). Discovering common motifs in cursor movement data for improving web search. In *Proceedings of international conference on web search and data mining (WSDM)* (pp. 183–192).

Lank, E., Cheng, Y.-C. N., & Ruiz, J. (2007). Endpoint prediction using motion kinematics. In *Proceedings of SIGCHI conference on human factors in computing systems (CHI)* (pp. 637–646).

Leclerc, F., Plamondon, R., & Lorette, G. (1992). Des gaussiennes pour la modélisation des signatures et la segmentation des tracés manuscrits. *Traitement du Signal, 9*(4), 347–358.

Leiva, L. A., & Huang, J. (2015). Building a better mousetrap: Compressing mouse cursor activity for web analytics. *Information Processing and Management, 51*(2), 114–129.

Leiva, L. A., Martín-Albo, D., & Plamondon, R. (2015). Gestures à go go: Authoring synthetic human-like stroke gestures using the kinematic theory of rapid movements. *ACM Transactions on Intelligent Systems and Technology, 7*(2).

Leiva, L. A., & Vidal, E. (2010). Assessing user's interactions for clustering web documents: a pragmatic approach. In *Proceedings of hypertext (HT)* (pp. 277–278).

Leiva, L. A., & Vivó, R. (2007). A gesture inference methodology for user evaluation based on mouse activity tracking. In *Proceedings of IADIS international conference on interfaces and human computer interaction (IHCI)* (pp. 58–67).

Leiva, L. A., & Vivó, R. (2013). Web browsing behavior analysis and interactive hypervideo. *ACM Transactions on the Web, 7*(4).

Libet, B., Gleason, C. A., Wright, E. W., & Pearl, D. K. (1983). Time of conscious intention to act in relation to onset of cerebral activity (readiness-potential). the unconscious initiation of a freely voluntary act. *Brain, 106*(3), 623–642.

Liu, Y., Chen, Y., Tang, J., Sun, J., Zhang, M., Ma, S., & Zhu, X. (2015). Different users, different opinions: Predicting search satisfaction with mouse movement information. In *Proceedings of ACM international conference on information retrieval (SIGIR)* (pp. 493–502).

Maarse, F. J. (1987). *The study of handwriting movement: Peripheral models and signal processing techniques*. Swets & Zeitlinger. http://repository.ubn.ru.nl/bitstream/handle/2066/113343/mmubn000001_038860821.pdf?sequence=1.

MacKay, B., & Watters, C. (2012). An examination of multisession web tasks. *Journal of the American Society for Information Science and Technology, 63*(6), 1183–1197.

Martín-Albo, D., Plamondon, R., & Vidal, E. (2014). Training of on-line handwriting text recognizers with synthetic text generated using the kinematic theory of rapid human movements. In *Proceedings of international conference on frontiers in handwriting recognition (ICFHR)* (pp. 543–548).

Martín-Albo, D., Plamondon, R., & Vidal, E. (2015). Improving sigma-lognormal parameter extraction. In *Proceedings of international conference on document analysis and recognition (ICDAR)*.

Meyer, D. E., Smith, J. E. K., Kornblum, S., Abrams, R. A., & Wright, C. E. (1990). Speed-accuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action. *Attention and Performance, 13*(23), 173–226.

Morasso, P., Mussa Ivaldi, F. A., & Ruggiero, C. (1983). How a discontinuous mechanism can produce continuous patterns in trajectory formation and handwriting. *Acta Psychologica, 54*(1), 83–98.

Neilson, P. (1993). The problem of redundancy in movement control: the adaptive model theory approach. *Psychological Research, 55*(1), 99–106.

O'Reilly, C., & Plamondon, R. (2009). Development of a Sigma-Lognormal representation for on-line signatures. *Pattern Recognition, 42*(12), 3324–3337.

O'Reilly, C., & Plamondon, R. (2011). Can computer mice be used as low-cost devices for the acquisition of planar human movement velocity signals? *Behavior Research Methods, 43*(1), 229–238.

Plamondon, R. (1995a). A kinematic theory of rapid human movements. Part I: Movement representation and control. *Biological Cybernetics, 72*(4), 295–307.

Plamondon, R. (1995b). A kinematic theory of rapid human movements. Part II: Movement time and control. *Biological Cybernetics, 72*(4), 309–320.

Plamondon, R., Alimi, A. M., Yergeau, P., & Leclerc, F. (1993). Modelling velocity profiles of rapid movements: a comparative study. *Biological Cybernetics, 69*(1), 119–128.

Plamondon, R., & Djioua, M. (2006). A multi-level representation paradigm for handwriting stroke generation. *Human Movement Science, 25*(4–5), 586–607.

Plamondon, R., & Lamarche, F. (1986). Modelization of handwriting: A system approach. In H. S. R. Kao, G. P. van Galen, & R. Hoosain (Eds.), *Graphonomics: Contemporary research in handwriting* (pp. 169–183).

Plamondon, R., O'Reilly, C., Galbally, J., Almaksour, A., & Anquetil, E. (2014). Recent developments in the study of rapid human movements with the Kinematic Theory: Applications to handwriting and signature synthesis. *Pattern Recognition Letters, 35*, 225–235.

Plamondon, R., O'Reilly, C., Rémi, C., & Duval, T. (2013). The lognormal handwriter: learning, performing, and declining. *Frontiers in Psychology, 4*(1), 945:1–945:14.

Savitzky, A., & Golay, M. J. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry, 36*(8), 1627–1639.

Sellen, A., Murphy, R., & Shaw, K. (2002). How knowledge workers use the web. In *Proceedings of SIGCHI conference on human factors in computing systems (CHI)* (pp. 227–234).

Shapira, B., Taieb-Maimon, M., & Moskowitz, A. (2006). Study of the usefulness of known and new implicit indicators and their optimal combination for accurate inference of users interests. In *ACM symposium on applied computing (SAC)* (pp. 1118–1119).

Soon, C. S., Brass, M., Heinze, H.-J., & Haynes, J.-D. (2008). Unconscious determinants of free decisions in the human brain. *Nature Neuroscience, 11*, 543–545.

Speicher, M., Both, A., & Gaedke, M. (2013). Tellmyrelevance! predicting the relevance of web search results from cursor interactions. In *Proceedings of ACM international conference on information and knowledge management (CIKM)* (pp. 1281–1290).

Thomassen, A. J. W. M., Keuss, P. J. G., & van Galen, G. (1983). Motor aspects of handwriting. *Acta Psychologica, 54*(1–3), 354.

Tukey, J. W. (1977). *Exploratory data analysis*. Boston, MA (USA): Addison-Wesley Publishing Company.

Viviani, P., & Flash, T. (1995). Minimum-jerk, two-thirds power law, and isochrony: Converging approaches to movement planning. *Journal of Experimental Psychology, 21*(1), 32–53.