Gesture Heatmaps: Understanding Gesture Performance with Colorful Visualizations

Radu-Daniel Vatavu
University Stefan cel Mare
of Suceava
Suceava 720229, Romania
vatavu@eed.usv.ro

Lisa Anthony
Department of CISE
University of Florida
Gainesville, FL 32611 USA
lanthony@cise.ufl.edu

Jacob O. Wobbrock
Information School | DUB Group
University of Washington
Seattle, WA 98195-2840 USA
wobbrock@uw.edu

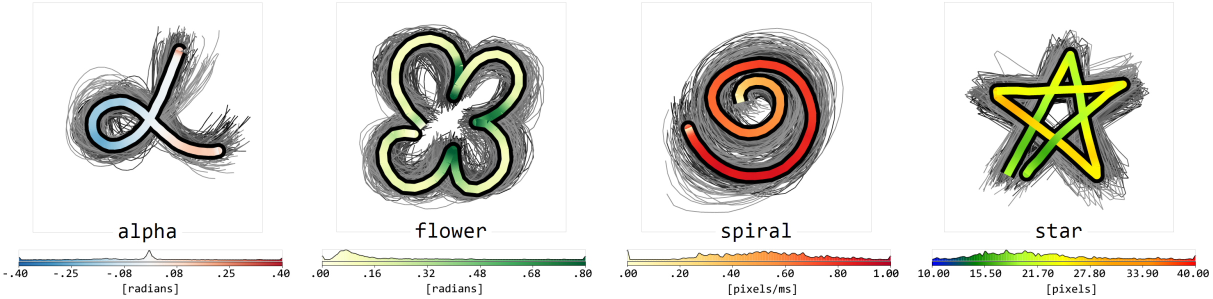


Figure 1. Gesture heatmap examples illustrating user variation during gesture articulation measured as *localized* absolute and relative turning angles ("alpha" and "flower"), articulation speed ("spiral"), and shape distance error from a template ("star"). NOTE: Heatmaps were generated with our tool, GHoST (Gesture HeatmapS Toolkit).

ABSTRACT

We introduce *gesture heatmaps*, a novel gesture analysis technique that employs color maps to visualize the variation of local features along the gesture path. Beyond current gesture analysis practices that characterize gesture articulations with single-value descriptors, *e.g.*, size, path length, or speed, gesture heatmaps are able to show with colorful visualizations how the value of any such descriptors vary along the gesture path. We evaluate gesture heatmaps on three public datasets comprising 15,840 gesture samples of 70 gesture types from 45 participants, on which we demonstrate heatmaps' capabilities to (1) explain causes for recognition errors, (2) characterize users' gesture articulation patterns under various conditions, *e.g.*, finger versus pen gestures, and (3) help understand users' subjective perceptions of gesture commands, such as why some gestures are perceived easier to execute than others. We also introduce *chromatic confusion matrices* that employ gesture heatmaps to extend the expressiveness of standard confusion matrices to better understand gesture classification performance. We believe that gesture heatmaps will prove useful to researchers and practitioners doing gesture analysis, and consequently, they will inform the design of better gesture sets and development of more accurate recognizers.

Keywords

Gesture heatmaps; features; recognition; user study; toolkits.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces & Presentation**]: User Interfaces.

ICMI '14, November 12 - 16 2014, Istanbul, Turkey
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2885-2/14/11...$15.00.
http://dx.doi.org/10.1145/2663204.2663256.

1. INTRODUCTION

As touch-screen interfaces continue to be the dominant form of interaction with mobile devices, it is important to consider best practices for supporting smooth and accurate touch and gesture-based interfaces. One can consider the problem space in two parts: that of the best ways to support user needs, expectations, and abilities in the design of the gestures themselves [19,30,31,33], and that of the best ways to recognize gestures through the design of recognition algorithms [3,28,34]. Both components would benefit from a broader understanding of *how users actually articulate gestures*. Previous work has examined this question from several angles, such as users' consensus for multi-stroke gesture production [2], analysis of the articulation characteristics of specific user groups [14], the effect of gesture implementer on articulation [27], and explorations of how gestures vary relative to each other [29]. Several tools have been developed to enable designers to assess recognition performance and analyze users' gestures [2,5,18,29]. However, researchers and practitioners still lack adequate tools to readily visualize and explore gesture articulation patterns.

We introduce *gesture heatmaps* (see Figure 1), a novel gesture analysis technique that employs color maps to visualize the variation of local features along the gesture path. Gesture heatmaps go beyond today's gesture analysis practices that employ single-value descriptors to characterize gesture articulation, *e.g.*, size, path length, or speed [2,14,27,29,31] by providing rich visualizations of how such descriptors vary along the gesture path. We demonstrate the use of gesture heatmaps with three case studies involving public datasets comprising 15,840 samples of 70 gesture types from 45 participants. Specifically, we show how gesture heatmaps are able to (1) explain causes for recognition errors, (2) characterize users' gesture performance under various articulation conditions, *e.g.*, finger versus pen gestures, and (3) help designers understand users' subjective perceptions of gestures, such as why some gestures are perceived easier to execute than others. To this end, we also introduce new concepts, such as the *chromatic confusion matrix* that extends the standard confusion matrix with gesture heatmaps.

The contributions of this paper are: (1) the introduction of a novel gesture visualization technique, *gesture heatmaps*, that focuses on exposing variations in local features along the gesture path; (2) an exploration of gesture heatmaps with public datasets, uncovering new findings related to the relationship between users' gesture articulation patterns, recognizers' classification performance, and users' subjective perceptions about gesture commands; and (3) a tool to compute gesture heatmaps and chromatic confusion matrices, the Gesture HeatmapS Toolkit (GHoST). Our work enables a deeper understanding of users' gesture articulation patterns, which subsequently will be useful to researchers and practitioners for improving touch and gesture interaction by designing better gesture sets and more accurate gesture recognizers.

2. RELATED WORK

We review in this section techniques and tools for analyzing users' gesture articulation patterns. We also point to relevant literature employing heatmaps for visualizing scientific data.

2.1 Techniques to study gesture articulation

Understanding the way users articulate stroke gestures is essential to designing good gesture sets and developing robust recognizers. To this end, researchers have proposed several techniques and, in many cases, delivered companion toolkits to support analysis of users' gesture articulation patterns [2,14,18,27,29]. These previous techniques employed geometric and kinematic features to characterize gesture articulations [2,14,29], used gesture descriptors to group gestures into perceptually-similar classes [18,30,31], and analyzed articulation consensus across users [2,33]. For example, several tools are available to assist in designing highly-recognizable gesture sets. The Gesture Design Tool [18] informs designers about classification errors by computing and presenting distance and confusion matrices. MAGIC [5] is another tool designed to assist practitioners to record motion gestures and visualize recognition performance measures on those gestures.

Researchers have employed feature analysis to characterize users' gesture articulation patterns and to uncover particular aspects of stroke gesture input under various articulation conditions. For example, Tu et al. [27] compared finger and pen gestures using geometric and kinematic features and found that finger-drawn gestures were larger and faster than pen gestures, but similar in articulation time and proportional shape distance. Kane et al. [14] compared gesture articulations of blind and sighted users, and reported significant differences in gesture speed, size, and shape. Anthony et al. [2] analyzed user consensus for multi-stroke articulation in terms of preferred number of strokes, stroke ordering, and stroke direction, which they measured as agreement rates [33]. They reported high agreement within users (.91), lower agreement between users (.55), and pointed to several connections between gesture complexity and consensus of gesture articulation. Agreement rates can now be computed automatically with the GEsture Clustering toolKit (GECKo) [2]. Vatavu et al. [29] introduced relative accuracy measures to characterize the degree by which single and multi-stroke gesture articulations vary from templates stored in the training sets of gesture recognizers. The authors delivered the Gesture RElative Accuracy Toolkit (GREAT) that computes geometric, kinematic, and articulation relative accuracy measures [29]. The gesture recognition literature also contains many features [6,24,32] that can be used to further characterize users' gesture articulations in more depth. For example, Rubine's statistical classifier [24] employs a set of 11 geometric and kinematic features, Blagojevic et al. [6] described 114 features, and Willems et al. [32] employed the g-48 gesture set to recognize multi-strokes using SVM and MLP classifiers.

Researchers have also looked at people's perceptions about gesture commands in order to identify perceptually-similar classes [19, 22,30,31]. For example, Long et al. [19] investigated the visual similarity of pen gestures and derived a computable model for perceptual similarity using gesture features, such as curviness, that correlated $R^2 = .56$ with user-reported similarity. Vatavu et al. [31] found that subjects were highly consistent in estimating the execution difficulty of single-stroke gestures, leading to two estimation rules based on production time that approximated absolute and relative perceived difficulty with 75% and 90% accuracy, respectively. Recently, Rekik et al. [22] extended the study to include multi-stroke gestures. Subjects were also consistent in their perception of gesture scale (*e.g.*, large versus small gestures), which is predictable with 90% accuracy using a rule based on the area size of the gesture bounding box [30].

2.2 Using heatmaps to inform design

Heatmaps are a common technique to visualize users' interactive behavior, most frequently in terms of mouse cursor movement and eye gaze [12,21], but also for touch patterns on mobile devices [16,25]. The ultimate goal of heatmap analysis in HCI is to pinpoint usability problems, to allow designers to understand their users' interaction patterns and improve their designs. For example, Huang et al. [12] showed that mouse cursor position closely relates to eye gaze, and used heatmaps of click and cursor movement positions to understand and improve the way people use search engines. Navalpakkam and Churchill [21] employed both cursor and eye gaze heatmaps in a study devoted to measuring and predicting aspects of users' web experience, such as frustration and reading struggles. Heatmaps have also been used to visualize touch patterns on mobile devices. For instance, Lettner and Holzmann [16] suggested the use of heatmaps to visualize users' touch paths on mobile devices in order to detect usability issues, *e.g.*, significantly more touches in the vicinity of a target may indicate users having problems acquiring that target from the first attempt. Schaefers et al. [25] employed heatmaps of touch points to uncover users' preferred patterns for articulating swipe gestures on mobile devices.

3. GESTURE HEATMAPS

Pseudo-coloring numerical data represents a standard information visualization technique to display scalar values in a non-numerical way [8,20]. The goal is to present the viewer with an overall image of the variation within the data and, consequently, to allow understanding of complex numerical information with the use of appropriate colors. A color map (*e.g.*, the rainbow map), is employed to compute the color of each value point in the data by means of interpolation. Color maps usually exploit perceptual cues with their selected range of colors. For example, "warm" orange and red colors usually indicate more magnitude in the data, while "cold" colors, such as green and blue, are used to encode data values with smaller magnitudes. We build on the large literature of heatmap visualization and its applications to different areas of study [8,21,23,25,26] to introduce gesture heatmaps to provide thorough characterization of users' gesture articulation patterns with the use of appropriately colored rendering of localized gesture features. In this section, we present technical details of how to construct a colorful heatmap for stroke gestures in order to render values of localized gesture features, and we discuss several color scheme choices that we believe are best suited for this goal.

3.1 Computing gesture heatmaps

We use gesture heatmaps to visualize the variation that is present during articulation of stroke gestures, such as the shape error of gesture candidates with respect to predefined templates [28,34], differences in articulation speed within and between users [29], or the amount of shape deformation that users naturally apply to the geometry of the gesture shape during articulation under various conditions [2,29]. Consequently, gesture heatmaps are computed from gesture samples captured from users, such as those used to train recognizers [3,28,34]. Let T denote such a dataset composed of gestures represented as a series of 2-D points with timestamps:

$$\{p_i = (x_i, y_i, t_i) \in \mathbb{R}^2 \times \mathbb{R}^+ \mid i = 1..n\} \quad (1)$$

We assume that all the gestures in the set have been normalized with respect to their sampling rate and their location in the plane, *i.e.*, all the gestures have the same number of points n, and were translated to origin so that their centroids are now $(0, 0)$. The gesture re-sampling step is required by the gesture task axis extraction algorithm that aligns individual points belonging to different gestures [29], as we explain further in the paper. In this work, we use n=64 sampling points to represent gestures, a value that was used in the $1 recognizer [34]. The translation-to-origin pre-processing step is required so that the geometrical features that we compute are invariant to the specific location where gestures have actually been produced on the interactive touch-screen area, *i.e.*, we make sure that the feature values we work with are translation-invariant. These two pre-processing steps are straightforward to implement, and pseudo-code for implementing them has already been made available in the gesture recognition literature [3,28,34]. Note that we do not normalize gestures with respect to scale or rotation (which are other common gesture pre-processing techniques, usually executed before gesture recognition [3,28,34]), because we want to capture the full amount of shape deformation that users naturally produce during articulation.[1]

Let f be a gesture feature defined at each point p_i of the gesture path. For example, $f(p_i)$ may be the local turning angle or the local articulation speed at point p_i. We discuss in detail the features that we use in the Case Studies section of the paper. Let $\mathcal{C}$ be a color map represented as an array of colors, $\{\mathcal{C}_i \mid i = 1..|\mathcal{C}|\}$, that we assume already sorted in ascending order in terms of users' perceptions of the relative differences between different colors [1,11,17]. Perceptual ordering of colors means that the first entries in the color map are intuitively perceived as having "less" amount of magnitude (*e.g.*, usually expressed with brightness or saturation) when compared with subsequent colors in the map. We discuss our choices for color maps in the next section. Using interpolation between consecutive colors in the array, we can create fine-resolution transitions between these colors and, consequently, we can assume that $\mathcal{C}$ contains a sufficient number of entries. Each feature value is then mapped to a color from $\mathcal{C}$ using a linear interpolation technique:

1. Normalize the value of the feature $f(p_i)$ in [0..1]:

$$f_{norm}(p_i) = \frac{f(p_i) - \min\limits_{i=1..n} f(p_i)}{\max\limits_{i=1..n} f(p_i) - \min\limits_{i=1..n} f(p_i)} \quad (2)$$

2. Use the normalized feature value to index the color map, $\mathcal{C}_{\lfloor f_{norm} \cdot |\mathcal{C}| \rfloor}$. For example, if the normalized feature value is .721 and the color map contains 600 distinct colors, the corresponding color will be $\mathcal{C}_{\lfloor .721 \times 600 \rfloor} = \mathcal{C}_{\lfloor 432.6 \rfloor} = \mathcal{C}_{432}$.

The colors corresponding to feature values for consecutive points p_i and p_{i+1} on the gesture path are used to render the stroke segment $[p_i p_{i+1}]$ with a linear color gradient interpolating the colors of its two extremities using arc-length as the interpolation parameter. The result is a smooth color gradient going from the first to the last point of the gesture path reflecting changes in the feature values. Please note that heatmap colors need to be rendered on top of an actual gesture shape, which would ideally be the *representative* articulation of that gesture type. The literature offers several options for selecting this representative articulation, such as using a predefined template [27] or the gesture task axis [29] in one of its forms: designer-defined, centroid of all gestures, or the sample closest to the centroid [29] (p. 282). In this work, we adopt the latter approach, and choose the shape of the gesture heatmap to be the sample from the dataset that is closest to the centroid of all gestures of the same type. Figure 2 illustrates the representative gesture shape of a set of samples with speed values mapped to colors.

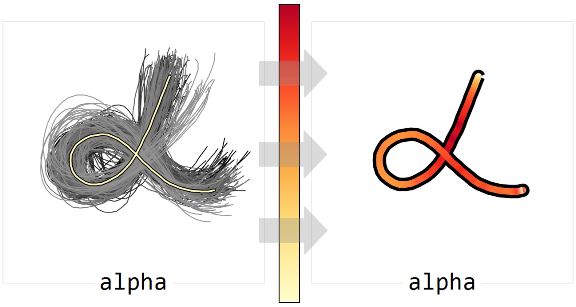


Figure 2: A representative shape of a gesture set (in yellow in the left image) is rendered with colors to highlight variation in a gesture feature (right). In this example, the heatmap shows variation in speed for the "alpha" symbol, measured from 220 samples from 11 participants [31]. Note how speed increases during the straight parts, and decreases during the loop.

3.2 Color schemes

Traditionally, heatmap visualizations have employed the *rainbow color scheme*, which has become the prevalent color map in the information visualization community [8]. At the same time, some work has shown that the pseudo-colors of the rainbow map do not represent the optimal choice to visualize data effectively [8,17,23]. For example, Borland and Taylor II [8] pointed to several usability problems for the rainbow color map, such as the lack of perceptual ordering of the colors of the light spectrum, the uncontrolled variation in luminance for these colors, and potential misleads that may occur during data interpretation. Despite these problems, the rainbow color map is still heavily used to display data in the scientific community, even by information visualization researchers [8].

Beyond rainbow colors, alternative color schemes have been proposed to better exploit people's capacity to perceive differences between hues and luminance. Moreland [20] lists criteria for good color maps, such as the ability of the heatmap to maximize perceptual resolution, to provide intuitive perception of color order, and to produce aesthetically pleasing images. One simple coloring scheme is represented by *graylevel scales* that use shades of gray to encode levels of amplitude in the data. The graylevel scale is effective because the human visual system is sensitive to changes in luminance that can be interpreted accordingly as changes in the amplitude of data values. However, brightness perception has been found to be dependent on context, such as the brightness of the surrounding area, making the same color appear differently in different contexts, which is known as the simultaneous contrast effect [26]. Moreland [20] explored *diverging color maps* that transition from one color to another by passing through an unsaturated color, such as white, and showed the advantages of these color maps compared to the rainbow color scheme. Light and Bartlein [17] suggested alternatives for the rainbow colors, such as a modified spectral color scheme, colors pro-

[1] However, depending on the purpose of the study, gestures could undergo normalization by scale and/or rotation [3,28,34].

duced by single-hue progressions (*e.g.*, from white to purplish-blue), and diverging progression between two hues (*e.g.*, from blue to gray, or the orange-white-purple diverging scheme). Finally, the ColorBrewer web site and application [1] provides hand-crafted color scheme suggestions for visualizing sequential (grayscale), diverging, and qualitative data.

Informed by the aforementioned literature, we adopted in this work the following color schemes to visualize our gesture heatmaps:

❶ **Diverging color map.** Two different colors are used to indicate the minimum and maximum magnitude of the values present in the data. The leftmost color of the scale, corresponding to the minimum magnitude, changes gradually into a neutral color (*e.g.*, white) positioned at the center of the scale, and then gradually turns into the second, rightmost color, corresponding to the maximum magnitude in the data. This type of double-ended color scheme is useful to visualize ratio data (*i.e.*, data that has a clear zero point), by indicating on which side of zero a given value lies. This color scheme represents our default choice for gesture heatmap visualizations of features that have values located on both sides of an explicit zero point, such as turning angle, curvature, acceleration, etc. Because diverging color maps lack a natural order of colors, it is common practice to choose extremity colors that are usually associated with low/high and cold/hot connotations [20]. Examples are red and yellow colors associated to warm and blue and green to cold, which seems to be a perception invariant across subjects and cultures [11]. We adopted one of the hand-crafted color schemes available on ColorBrewer [1] that employs blue and red diverging colors. (See Figure 1 on the first page showing local turning angles for the "alpha" symbol.)

❷ **Sequential (grayscale) color map.** This color map interpolates colors gradually from light to dark, with dark denoting the maximum magnitude in the data. It is our default choice for visualizing features whose values do not contain a significant midpoint, such as features represented by values that are all located on one side of the zero point. Examples from our case study features would be speed or Euclidean shape distance values that are always positive. In this case, the visualization literature recommends using a sequence of lightness steps combined with a single hue to visualize such data [17]. We used again ColorBrewer [1] to select two hand-crafted sequential schemes with green and orange hues. (See the "flower" and "spiral" gesture heatmaps of Figure 1 showing the variation in relative turning angle and articulation speed along the gesture path.)

❸ **Rainbow color map.** Despite being criticized in the literature [8,17,23], we decided to implement this color scheme as well due to its popularity and wide adoption even in the information visualization community [8]. Cold colors (blue and green) encode low magnitude in the data, while warm colors (orange and red) are mapped to high-magnitude values. (See the "star" gesture heatmap of Figure 1 computed to show the Euclidean shape distance.)

4. CASE STUDIES

Gesture heatmaps are general visualizers of any feature that is ultimately the designer's choice. Here we illustrate the use of gesture heatmaps with three case studies that cover important aspects of gesture interaction design: (1) we reveal causes of erroneous classification by using gesture heatmaps to visualize the gesture articulation shape distance used by the $1 gesture recognizer [34]; (2) we show how gesture heatmaps can be used to understand people's subjective perceptions about gesture commands by visualizing the articulation speed for the execution difficulty datasets [31]; (3) we employ gesture heatmaps to characterize users' gesture articulation differences between finger and pen gestures on the MMG multistroke gesture dataset [3,4]. Overall, we provide results from 15,840 gesture samples of 70 gesture types from 45 participants.

4.1 Understanding gesture recognition errors

We show in the following how gesture heatmaps can point to causes leading to recognition errors that cannot otherwise be revealed with today's standard measures for assessing recognition performance, such as error rates and confusion matrices. Being highly visual in nature, gesture heatmaps highlight "hot" parts of a gesture shape that exhibit high variance, helping practitioners to optimize their gesture shapes and maximize recognition performance. For this case study, we employ the dataset of Wobbrock et al. [34] on which the performance of the $1 gesture recognizer was evaluated[2]. We only report results for the subset of gestures articulated at fast speed, as they had the highest rate of recognition errors [34] (p. 165). This dataset is composed of 1,600 samples of 16 distinct gesture types articulated by 10 participants for 10 times each.

The performance of a gesture recognizer is evaluated today using accuracy or error rates [3,18,24,28,34]. High recognition accuracy (and, implicitly, low recognition errors) signal a high-performing gesture recognizer. These measures are useful to understand recognition performance overall and to compare recognition performance between experimental conditions, *e.g.*, the $1 recognizer was found 7% more accurate than Rubine's recognizer, and equally accurate as the Dynamic Time Warping cost function [34] (p. 165). To find out more about recognition performance, practitioners can compute recognition rates per gesture type that indicate poorly-recognized gestures, *e.g.*, see the performance of the "circle" and "question mark" symbols in Figure 3 with 75% and 92% recognition rates[3], significantly lower than the performance of the other gesture types ($\chi^2(15)=196.184$, $p<.01$). Practitioners can also resort to confusion matrices that characterize error rates in more depth by showing how often one gesture type is misrecognized for another. For example, Figure 4 reveals that the 8% error rate of "question mark" is mostly caused by the "right square bracket" and "right curly brace" symbols. Long et al. [18] showed that practitioners can successfully employ these numbers to select highly-recognizable gestures for their applications. However, numbers only are not informative enough to explain the *cause* of recognition errors. Recognition percentages are valuable means to sum up the level of performance,

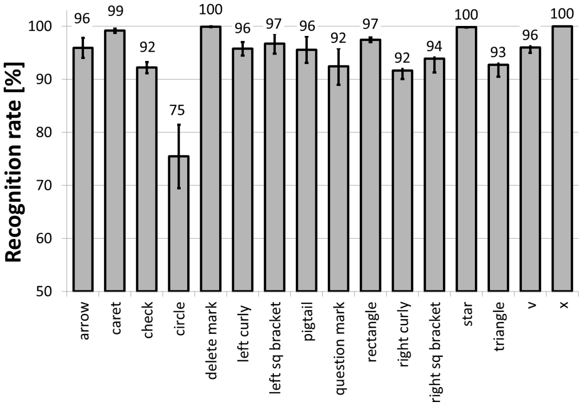
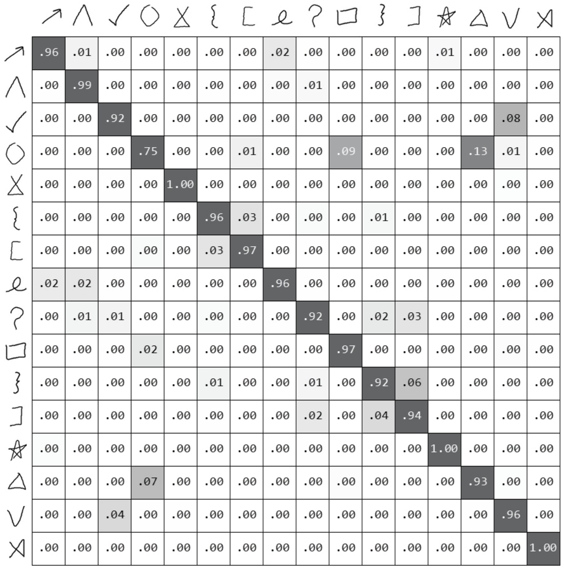


Figure 3: User-independent recognition rates for the gestures of the $1 dataset computed with $1's Euclidean cost function [34].

[2] http://depts.washington.edu/aimgroup/proj/dollar/

[3] We compute and report average recognition rates for the Euclidean cost function and the Nearest-Neighbor classification technique (*i.e.*, the $1 recognizer) in the user-independent scenario using the methodology from [28] (p. 275). We vary the number of training participants P and training samples per gesture type T in a geometric progression from 1 to 8 (*i.e.*, $P = 1, 2, 4, 8$, and $T = 1, 2, 4, 8$). All gestures were re-sampled to n=64 points, uniformly scaled, and translated to origin, as in [3,28,34].



	↗	∧	✓	○	X	{	[	ℓ	?	▭	}	]	☆	△	V	⋊
↗	.96	.01	.00	.00	.00	.00	.00	.02	.00	.00	.00	.00	.01	.00	.00	.00
∧	.00	.99	.00	.00	.00	.00	.00	.00	.01	.00	.00	.00	.00	.00	.00	.00
✓	.00	.00	.92	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.08	.00
○	.00	.00	.00	.75	.00	.00	.01	.00	.00	.09	.00	.00	.00	.13	.01	.00
X	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
{	.00	.00	.00	.00	.00	.96	.03	.00	.00	.00	.01	.00	.00	.00	.00	.00
[	.00	.00	.00	.00	.00	.03	.97	.00	.00	.00	.00	.00	.00	.00	.00	.00
ℓ	.02	.02	.00	.00	.00	.00	.00	.96	.00	.00	.00	.00	.00	.00	.00	.00
?	.00	.01	.01	.00	.00	.00	.00	.00	.92	.00	.02	.03	.00	.00	.00	.00
▭	.00	.00	.00	.02	.00	.00	.00	.00	.00	.97	.00	.00	.00	.00	.00	.00
}	.00	.00	.00	.00	.00	.01	.00	.00	.01	.00	.92	.06	.00	.00	.00	.00
]	.00	.00	.00	.00	.00	.00	.00	.00	.02	.00	.04	.94	.00	.00	.00	.00
☆	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	1.00	.00	.00	.00
△	.00	.00	.00	.07	.00	.00	.00	.00	.00	.00	.00	.00	.00	.93	.00	.00
V	.00	.00	.04	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.96	.00
⋊	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	1.00

Figure 4: Standard confusion matrix (user-independent) for the $1 gesture set and $1's Euclidean cost function [34].

but they cannot tell us why a recognizer has troubles with a certain gesture. Knowing more about the *why* of recognition errors will help us to more intelligently design gesture sets and choose recognizers not likely to encounter the same types of recognition errors. Next, we show how gesture heatmaps uncover causes of recognition errors, while we introduce the chromatic confusion matrix.

The $1 recognizer computes the sum of Euclidean distances between points to discriminate between gesture types [34] (p. 162). This cost function can also be interpreted as a feature, *i.e.*, the relative shape error between candidate p and template q [29] (p. 280):

$$ShE(p, q) = \sum_{i=1}^{n} \|p_i - q_i\| \qquad (3)$$

Because shape error is computed at each point on the gesture path, we can visualize it using gesture heatmaps for which the color of each point is in direct correspondence with the shape error between the two gestures at that point precisely. We generated heatmaps for all the gesture types in the $1 dataset and arranged them in the form of a confusion matrix so that heatmap colors reflect the amount of shape error between a candidate and a template; see Figure 5. The candidate shown for each gesture type is actually the user-independent gesture task axis of Vatavu et al. [29], computed from all the samples available in the set for that gesture type (in our case, 100 articulations = 10 participants × 10 executions).

Note how gesture heatmaps provide more information than the standard confusion matrix about the cause of recognition errors. For instance, the diagonal of the chromatic matrix shows candidates compared to gestures of the same type for which shape errors are low (an expected result) and, therefore, all color hues are mostly blue. However, color intensity increases toward "hot" hues, such as orange and red, for those gesture parts of the candidate that are most dissimilar from their corresponding parts on the template, and decrease toward "cold" green and blue hues for similar gesture parts. For instance, the cause of the "question mark" symbol being misrecognized as right curly and square braces appears more clear now: the shape of the question mark superimposes almost perfectly on the shapes of these other gestures, which causes low shape error values between their corresponding points, shown by the high proportion of blue segments in that pairing. The same can be said for other gesture pairs, such as "circle" and "triangle", left and right "curly braces", "curly braces" and "square brackets", "v" and "check", etc. Shape errors are low for these gesture types because they follow the same articulation direction (*e.g.*, going from top to bottom for braces, brackets, and the question mark), while their specific shape characteristics (*e.g.*, the upper curl of the question mark or the middle curly point of the curly braces) are sometimes overtaken in magnitude by the variation induced by different users articulating these shapes (which can be seen in the background gestures shown for each candidate heatmap). On the other hand, we can easily identify the most dissimilar gestures in the set, such as "delete mark", "star", or x", that present large shape errors for almost all their parts (reflected with orange and red hues).

When we know the features that might impact the performance of a recognizer, such as shape error being critical to the way $1 recognizes gestures, gesture heatmaps enable us to visually compare those features quickly for candidate gesture sets, and re-design the shape of gesture pairs which will be too similar for this recognizer. For example, once practitioners have identified the causes leading to recognition errors, they can proceed to rectify these causes, without necessarily having to remove a particular gesture from the set [18]. For example, the articulation direction of the "circle" can be reversed so that it won't conflict any longer with that of the "triangle", knowing (and seeing now concretely) that the $1 recognizer matches points in their order of input. Also, the shape design of the curly braces can specifically include a more pointed curl, which will slightly shift the centroid of these shapes toward the left, increasing therefore the shape error with respect to the "question mark" and "right square bracket" symbols. The same approach could be adopted for the second part of the "check" symbol, making it even longer to better disambiguate it from the "v" shape. Such changes acting only on some parts of the gesture shape can improve recognition accuracy, without removing that gesture from the set, as was the only option with previous approaches [18].

4.2 User perception of execution difficulty

We next show how gesture heatmaps are valuable in uncovering connections between objective articulation performance (*e.g.*, execution speed), and people's subjective perceptions about the gestures they articulate, usually collected as Likert ratings [31,33]. Prior work by Vatavu et al. [31] found that people's perceptions about the execution difficulty of stroke gestures can be reliably estimated using kinematic measurements on the gesture path. The authors reported 96% positive correlation between perceived difficulty and articulation time, and 87% negative correlation with articulation speed. The longer it takes people to articulate a gesture shape or the slower they are during articulation, the more likely people will rate that shape as more difficult to produce. However, there is no explanation in that work of the causes of these high correlations. In the following, we point to possible explanations for this phenomenon by visualizing articulation speed values[4] for the gesture types used in the execution difficulty work. There are two gesture datasets reported in [31] composed of 5,040 and 4,400 samples of 38 distinct gesture types collected from 25 participants with 20 executions each[5]. We employ one dataset to derive possible causes for the difficulty perception phenomenon, which we validate on the gestures belonging to the second set.

Figure 6 shows gesture heatmaps generated for localized articulation speed measured at each point p_i as the ratio of the path length

[4] We focus on the articulation speed because time heatmaps are not interesting for this analysis, as time increases monotonically along the gesture path. However, local speed values change according to the gesture shape.

[5] http://www.eed.usv.ro/~vatavu/index.php?menuItem=downloads

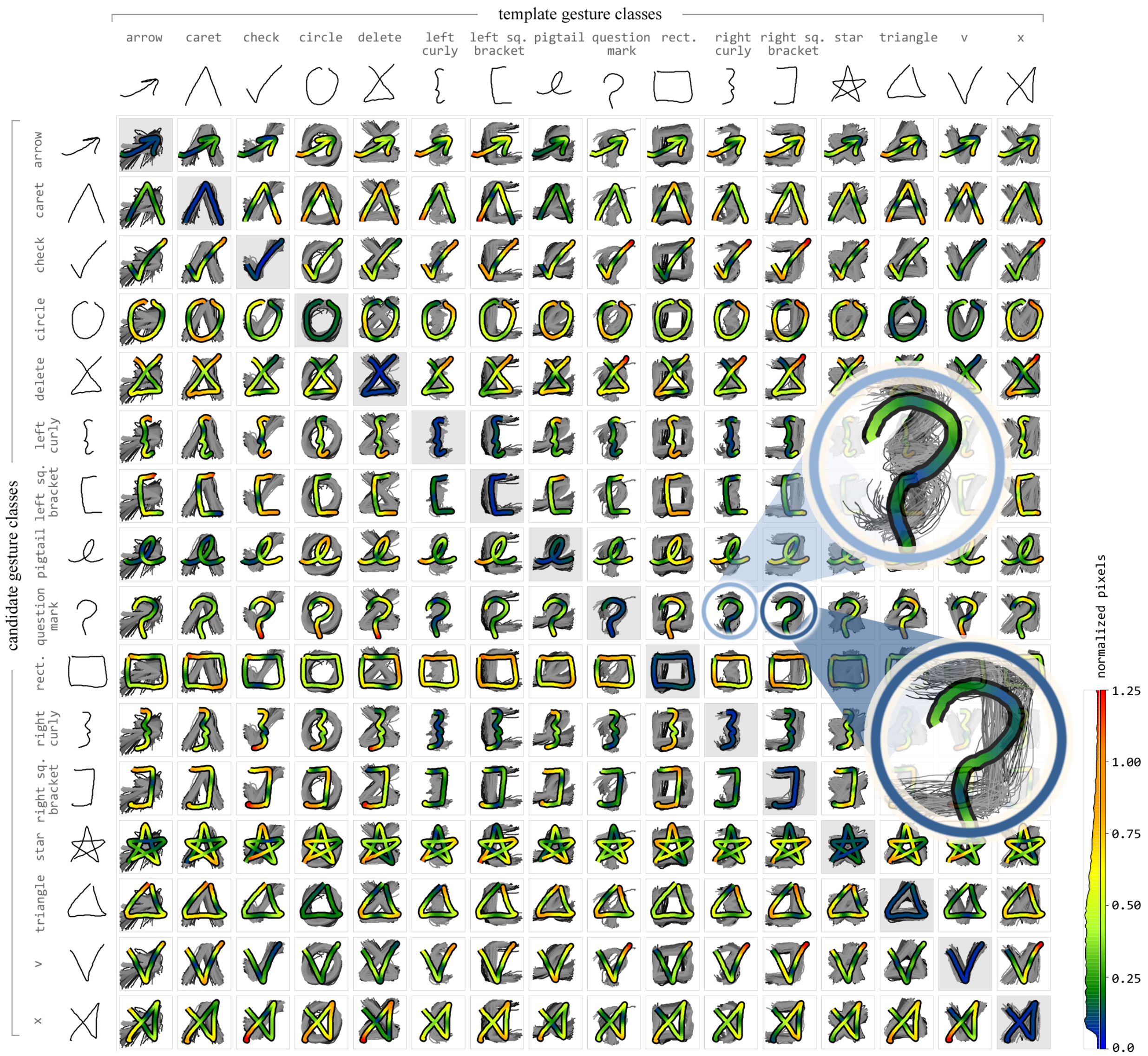


Figure 5: Gesture heatmaps for the 16 gestures of the $1 dataset [34] showing the Shape Error [29,34] between candidates and templates in the form of a *chromatic confusion matrix*. NOTES: Cold colors (*e.g.*, blue and green) show small shape errors between candidates and templates, which may cause recognition errors. The matrix was generated with our toolkit, GHoST, in the user-independent scenario. The scale shows values in normalized pixel values (*i.e.*, all gestures were scaled down to [0,1]×[0,1]).

between points p_{i-1} and p_{i+1} and the corresponding time duration:

$$s_i = \frac{\|p_{i-1} - p_i\| + \|p_i - p_{i+1}\|}{t_{i+1} - t_{i-1}} \qquad (4)$$

Each gesture has a ranking number showing its position in the list of all gestures in ascending order by perceived execution difficulty. These rankings are median values computed from the ratings collected from participants [31] (p. 96). Note how our speed gesture heatmaps actually show with colors the bell-shaped velocity profiles observed in motor control studies [10], with speed increasing along the stroke path until it reaches its midpoint, after which speed starts to decrease. Please note however that motor control theory's definition of a "stroke" is different from that commonly accepted in HCI as the path between two consecutive touch-down and touch-up events (see also [13]). For example, the "strike" gesture (index 16, Figure 6, left), although being a unistroke, is actually decomposable into six distinct ballistic movements, according to models in motor control theory. Looking at the heatmaps of the first dataset (Figure 6, left) we see that shapes with few such fundamental movements are rated less difficult to execute, such as "circle" (one ballistic movement), digit "3" (two movements), "6" (one movement), letter "m" (three movements), "sail" (four movements), and so on. The heatmaps of these gestures use hot orange and red colors showing fast articulation speeds. As the gesture shape becomes more complex with more turning points, relative articulation speed decreases as our gesture heatmaps correctly reflect with brighter colors. When we look at the gestures of the second dataset (Figure 6), we see that these observations hold true: complex trajectories that need to be articulated with more ballistic strokes are shown in brighter colors corresponding to slower relative speeds, as users have to continuously accelerate and decelerate along the gesture path. These speed patterns may be one reason why people perceive these gestures as more difficult to execute. For example, the "spiral" shape (index 8, Figure 6, right) allows continuous acceleration, while the "star" requires five decelerations along its path.

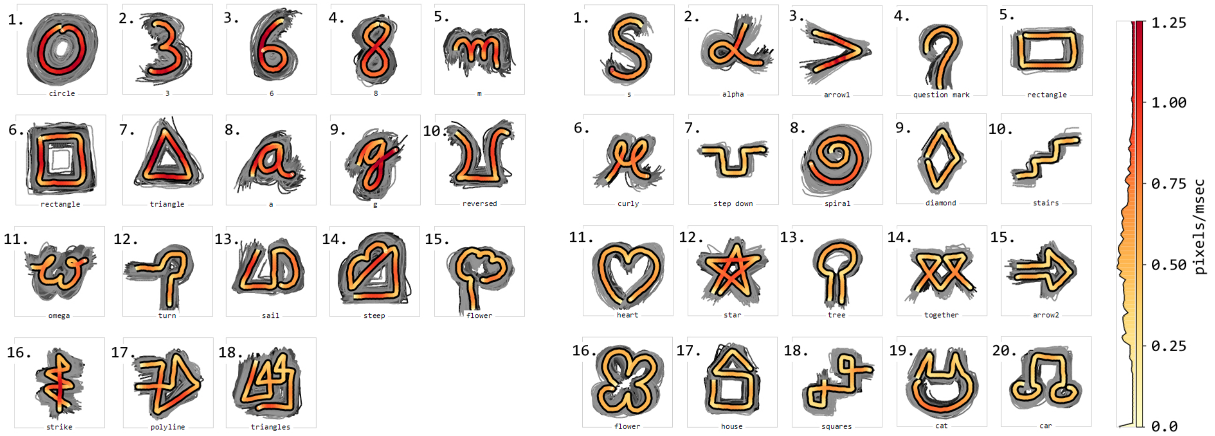


Figure 6: Gesture heatmaps showing articulation speed values for the two execution difficulty datasets of Vatavu et al. [31]. NOTES: Brighter colors show slow speeds, which correlated negatively with perceived difficulty [31], and thus indicate gesture parts that make gestures difficult to articulate. Numbers in the top-left corner show gesture ranking in ascending order of perceived difficulty.

We believe these observations are useful to explain, at least partially, the complexity of the difficulty perception phenomenon reported in [31]. While we do not provide the decisive answer to this question in this work (nor is it our goal) because of the complexity of this phenomenon (*e.g.*, the effect of practice, familiarity, writer speed, etc.), we believe we open new paths toward new gesture discoveries. For example, we think that gesture heatmaps may be used to spot ballistic strokes, which we believe to be in connection with Isokoski's approach for measuring geometric complexity [13], hopefully leading to advances in assessing gesture complexity. Also, ballistic movements could lead to a more suitable segmentation of a gesture shape into primitives, which may be used to increase the performance of the Curves Lines Corners (CLC) model [9] for estimating gesture production time by considering actual user-generated interruptions in the gesture path instead of decomposition of that path into standard geometric primitives. While we only point at these fruitful lines of work, we are eager to see how researchers will use our gesture heatmaps to explain more subtle connections between user perception and gesture articulation.

4.3 Finger versus stylus gestures

We next show how gesture heatmaps can reveal more discoveries about users' gesture articulation patterns and how these discoveries connect to previous work, as we look at the effect of gesture implementer (*i.e.*, finger or stylus) on articulation. Previous work [27] found that gesture implementer has no influence on the values of the proportional shape distance for single-stroke gestures. Another study [4] found recognition differences between multi-stroke gestures articulated with a finger versus a stylus, *i.e.*, higher recognition accuracy with the $N recognizer was seen on gestures made with a finger. Because recognizers like $1, $N, and $P [3,4,28,34] employ the shape distance as the recognizer cost function, variations in this feature have the potential to impact recognition accuracy significantly. Therefore, in this case study, we use Shape Error gesture heatmaps (eq. 3) to understand more about the articulation differences caused when employing the finger versus the stylus. We employ the MMG dataset [4] composed of 9,600 samples of 16 distinct multi-stroke gestures collected from 20 participants using either the finger or the stylus[6]. Because gestures in this set vary in terms of stroke ordering and stroke direction [4,29], we computed Shape Error with the $P alignment technique that minimizes the sum of Euclidean distances between pairs of points [28].

Gesture heatmaps showed in Figure 7 indicate differences in the accuracy (in terms of Shape Error) at which users produce gestures with the finger versus the pen. For example, finger articulation of "D" and "P" show the most concentrated regions of high shape error (Figure 7, left, orange and red), while the articulations of the same shapes using the stylus are displayed in green and yellow, indicating much lower shape error with this implementer. In contrast, "H" and "N" articulated by a stylus show the highest shape errors, but again do not exhibit high shape error when articulated by the finger. These gesture types are also among the most highly-confused pairs by the $N-Protractor recognizer [4] (p. 120). Mann-Whitney U tests confirmed significant differences (at $p<.01$) between the mean Shape Errors for 12 out of all 16 gesture types (exceptions were "arrowhead", "asterisk", "five point star", and "X").

Beyond pointing to overall differences between articulations (*e.g.*, there is more shape error when articulating "P" with the finger than the pen), gesture heatmaps can also point to localized "hot" spots on the gesture shape that exhibit large shape errors. For example, strokes' start and end points are shown with more intense colors, *e.g.*, see "arrowhead", "line", "I", "asterisk", etc., showing that large shape errors occur frequently at these locations. This observation connects to the concern in the sketching community to remove hooks that occur at the end of strokes (caused by users lifting the stylus off of the surface) in order to increase recognition accuracy of sketch input [15] (p. 11). This finding may inform the designer to replace a multi-stroke gesture with the single-stroke equivalent where applicable, *e.g.*, "N" can be articulated with three strokes (as it was in the MMG set), but also as a single-stroke. We believe that "hot" point scrutiny will help practitioners understand better how users articulate gestures, and thus, improve their gesture set designs.

5. CONCLUSION

We introduced in this work gesture heatmaps as a practical visualization technique to aid researchers during gesture analysis. To this end, we release the Gesture HeatmapS Toolkit (GHoST) as open source software[7] that computes heatmap visualizations for both user-dependent and independent scenarios, and exports results as `.csv` and `.bmp` files. Future work will explore animations and motion features [7] for stroke gestures. It is one of the goals of this work to encourage researchers to explore gesture heatmaps in order to uncover new findings about users' gesture articulation behavior, to design better gesture sets, and to develop more accurate recognizers.

[6] https://depts.washington.edu/aimgroup/proj/dollar/ndollar.html

[7] http://depts.washington.edu/aimgroup/proj/dollar/ghost.html

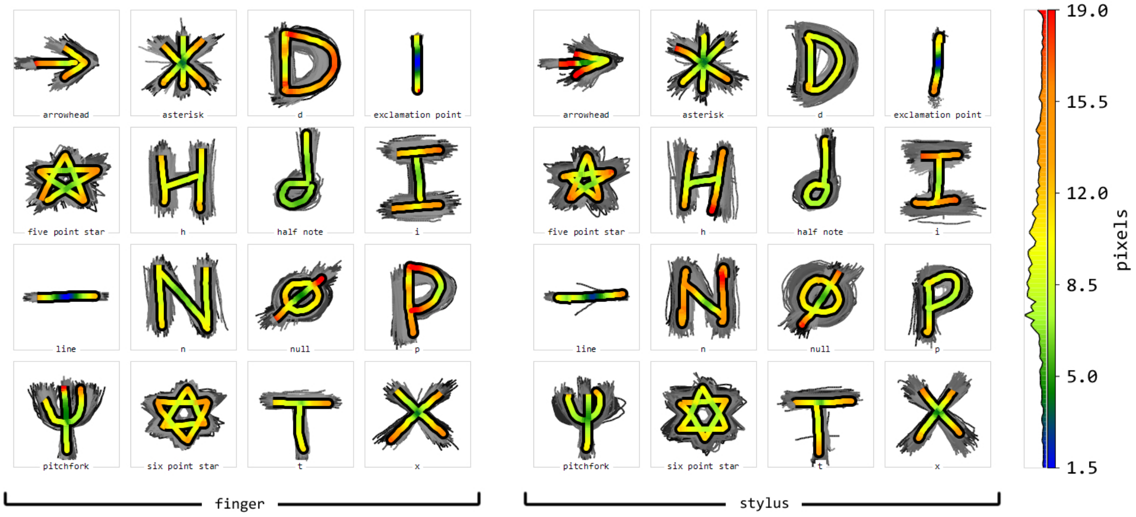


Figure 7: Shape error heatmaps for gestures articulated with the finger and the stylus [3,4]. Note how gesture heatmaps point to localized "hot" spots on the gesture shape that exhibit large shape errors, such as strokes' start and end points (orange and red).

6. REFERENCES

[1] ColorBrewer: Color advice for maps. http://colorbrewer2.org/.

[2] Anthony, L., Vatavu, R.-D., and Wobbrock, J. O. Understanding the consistency of users' pen and finger stroke gesture articulation. In *GI'13* (2013), 87–94.

[3] Anthony, L., and Wobbrock, J. O. A lightweight multistroke recognizer for user interface prototypes. In *GI'10* (2010), 245–252.

[4] Anthony, L., and Wobbrock, J. O. $N-Protractor: A fast and accurate multistroke recognizer. In *GI'12* (2012), 117–120.

[5] Ashbrook, D., and Starner, T. MAGIC: A motion gesture design tool. In *CHI '10*, ACM (2010), 2159–2168.

[6] Blagojevic, R., Chang, S. H.-H., and Plimmer, B. The power of automatic feature selection: Rubine on steroids. In *SBIM'10*, Eurographics Association (2010), 79–86.

[7] Bobick, A. F., and Davis, J. W. The recognition of human movement using temporal templates. *IEEE TPAMI 23*, 3 (March 2001), 257–267.

[8] Borland, D., and Taylor II, R. M. Rainbow color map (still) considered harmful. *IEEE CGA 27*, 2 (March 2007), 14–17.

[9] Cao, X., and Zhai, S. Modeling human performance of pen stroke gestures. In *CHI '07*, ACM (2007), 1495–1504.

[10] Djioua, M., and Plamondon, R. Studying the variability of handwriting patterns using the kinematic theory. *Hum Mov Sci. 28*, 5 (2009), 588–601.

[11] Hardin, C., and Maffi, L. (Eds.) *Color Categories in Thought and Language*. Cambridge University Press, 1997.

[12] Huang, J., White, R. W., and Dumais, S. No clicks, no problem: Using cursor movements to understand and improve search. In *CHI '11*, ACM (2011), 1225–1234.

[13] Isokoski, P. Model for unistroke writing time. In *CHI '01*, ACM (2001), 357–364.

[14] Kane, S. K., Wobbrock, J. O., and Ladner, R. E. Usable gestures for blind people: understanding preference and performance. In *CHI '11*, ACM (2011), 413–422.

[15] LaViola, Jr., J. J. Sketching and gestures 101. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, ACM (2006).

[16] Lettner, F., and Holzmann, C. Heat maps as a usability tool for multi-touch interaction in mobile applications. In *MUM '12*, ACM (2012), 49:1–49:2.

[17] Light, A., and Bartlein, P. J. The end of the rainbow? color schemes for improved data graphics. *EOS Transactions of the American Geophysical Union 85*, 40 (2004), 385–391.

[18] Long, Jr., A. C., Landay, J. A., and Rowe, L. A. Implications for a gesture design tool. In *CHI '99*, ACM (1999), 40–47.

[19] Long, Jr., A. C., Landay, J. A., Rowe, L. A., and Michiels, J. Visual similarity of pen gestures. In *CHI '00* (2000), 360–367.

[20] Moreland, K. Diverging color maps for scientific visualization. In *ISVC '09*, Springer (2009), 92–103.

[21] Navalpakkam, V., and Churchill, E. Mouse tracking: Measuring and predicting users' experience of web-based content. In *CHI '12*, ACM (2012), 2963–2972.

[22] Rekik, Y., Vatavu, R.-D., and Grisoni, L. Understanding users' perceived difficulty of multi-touch gesture articulation. In *ICMI '12*, ACM (2014).

[23] Rogowitz, B. E., and Treinish, L. A. Data visualization: The end of the rainbow. *IEEE Spectr. 35*, 12 (Dec. 1998), 52–59.

[24] Rubine, D. Specifying gestures by example. *SIGGRAPH Comput. Graph. 25*, 4 (July 1991), 329–337.

[25] Schaefers, K., Ribeiro, D., and de Barros, A. C. Beyond heat maps: Mining common swipe gestures. In *MUM'13* (2013).

[26] Stone, M. Representing colors as three numbers. *IEEE Comp. Graph. Appl. 25*, 4 (2005), 78–85.

[27] Tu, H., Ren, X., and Zhai, S. A comparative evaluation of finger and pen stroke gestures. In *CHI '12*, ACM (2012), 1287–1296.

[28] Vatavu, R.-D., Anthony, L., and Wobbrock, J. O. Gestures as point clouds: A $P recognizer for user interface prototypes. In *ICMI '12*, ACM (2012), 273–280.

[29] Vatavu, R.-D., Anthony, L., and Wobbrock, J. O. Relative accuracy measures for stroke gestures. In *ICMI'13*, ACM (2013), 279–286.

[30] Vatavu, R.-D., Casiez, G., and Grisoni, L. Small, medium, or large?: Estimating the user-perceived scale of stroke gestures. In *CHI '13*, ACM (2013), 277–280.

[31] Vatavu, R.-D., Vogel, D., Casiez, G., and Grisoni, L. Estimating the perceived difficulty of pen gestures. In *INTERACT'11*, Springer (2011), 89–106.

[32] Willems, D., Niels, R., van Gerven, M., and Vuurpijl, L. Iconic and multi-stroke gesture recognition. *Pattern Recognition 42*, 12 (2009), 3303–3312.

[33] Wobbrock, J. O., Morris, M. R., and Wilson, A. D. User-defined gestures for surface computing. In *CHI '09*, ACM (2009), 1083–1092.

[34] Wobbrock, J. O., Wilson, A. D., and Li, Y. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In *UIST '07*, ACM (2007), 159–168.