Improving Gesture Recognition Accuracy on Touch Screens for Users with Low Vision

Radu-Daniel Vatavu
MintViz Lab | MANSiD Research Center
University Stefan cel Mare of Suceava
Suceava 720229, Romania
vatavu@eed.usv.ro

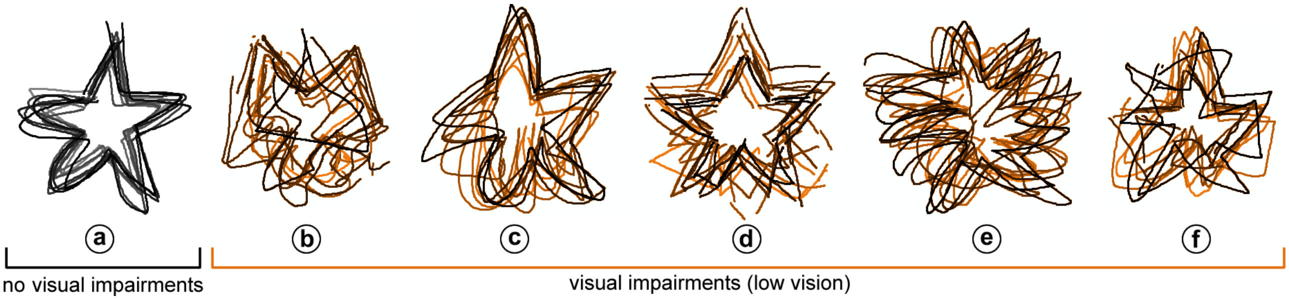

Figure 1. Gesture articulations produced by people with low vision present more variation than gestures produced by people without visual impairments, which negatively affects recognizers' accuracy rates. Ten (10) superimposed executions of a "star" gesture are shown for six people: a person without visual impairments (a); three people with congenital nystagmus and high myopia (b), (c), (d); and two people with chorioretinal degeneration (e), (f).

ABSTRACT

We contribute in this work on gesture recognition to improve the accessibility of touch screens for people with low vision. We examine the accuracy of popular recognizers for gestures produced by people with and without visual impairments, and we show that the user-independent accuracy of $P, the best recognizer among those evaluated, is small for people with low vision (83.8%), despite $P being very effective for gestures produced by people without visual impairments (95.9%). By carefully analyzing the gesture articulations produced by people with low vision, we inform key algorithmic revisions for the $P recognizer, which we call $P+. We show significant accuracy improvements of $P+ for gestures produced by people with low vision, from 83.8% to 94.7% on average and up to 98.2%, and 3× faster execution times compared to $P.

ACM Classification Keywords

H.5.2. [Information Interfaces and Presentation (e.g., HCI)] User Interfaces: *Input devices and strategies*; K.4.2. [Computers and Society] Social Issues: *Assistive technologies for persons with disabilities*.

Author Keywords

Gesture recognition; Touch screens; Touch gestures; Visual impairments; Low vision; $1; $P; $P+; Recognition accuracy; Evaluation; Point clouds; Algorithms; Recognition.

CHI 2017, May 06–11, 2017, Denver, CO, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-4655-9/17/054...$15.00
DOI: http://dx.doi.org/10.1145/3025453.3025941

INTRODUCTION

Today's touch screen devices are little accessible to people with visual impairments, who need to employ workaround strategies to be able to use them effectively and independently [17,18,42]. Because smart devices expose touch screens not adapted to non-visual input, touch and gesture interaction pose many challenges to people with visual impairments [11,12,20,30], which can only be addressed with a thorough understanding of their gesture articulation performance.

The term "visual impairments" includes a broad range of visual abilities, from moderate to severe impairments and blindness. Moderate and severe visual impairments are grouped under the term "low vision" in the International Classification of Diseases [35]. According to a 2014 fact sheet of the World Health Organization [34], out of an estimated 285 million people with visual impairments worldwide, 246 million have low vision (86.3%). However, there has been very little research to understand how people with low vision use gestures on touch screen devices, while the majority of efforts on designing accessible touch interfaces were directed towards blind people [7,12,17,20]. Unlike blind people, however, people with low vision do rely on their visual abilities during their everyday activities, including operating computers and mobile devices, but they face challenges caused by vision disturbances. Common disturbances include blurred vision (caused by refractive errors, 43% incidence worldwide), faded colors or glare (such as in cataracts, 33% incidence), blind spots in the visual field (as in diabetic rethinopathy), etc., and are usually accompanied by physiological discomfort (*e.g.*, eye burning, stinging, dryness, itching, tiredness, etc.) [6,26,34].

In this work, we focus on touch screen gestures produced by people with low vision (see Figure 1) and we report, for the first time in the literature, differences in gesture articulation between people with low vision and people without visual impairments. We show that these differences impact negatively the recognition performance of today's popular gesture recognizers, such as $1, DTW, $P, etc. [2,3,21,27,32,46,59], which deliver user-independent recognition rates between 70.4% and 83.8%, with an average error rate of 22.5%. Even $P, the best recognizer among those evaluated in this work, delivered an error rate of 16.2% (83.8% accuracy) for gestures articulated by people with low vision yet only 4.1% (95.9% accuracy) for people without visual impairments. To reduce this unnecessary gap in accuracy, we propose algorithmic improvements for $P that increase recognition rates for people with low vision from 83.8% to 94.7% in average and up to 98.2% (1.8% error rate) with training data from 8 participants $\times$ 8 samples.

The contributions of this work are: (1) we examine touch gestures produced by people with low vision and we report differences in recognition accuracy and articulation performance compared to gestures produced by people without visual impairments; (2) we introduce $P+, an algorithmic improvement over the $P gesture recognizer; and (3) we conduct an evaluation of $P+ and show that $P+ increases accuracy for people with low vision with +17.2% on average (all recognizers considered), adds +10.9% accuracy to $P, and is 3× faster than $P, with only minimum code changes. Our results will help make touch interfaces more accessible to people with low vision, *enabling them to use gestures that are recognized as accurately as gestures produced by people without visual impairments*, removing thus the unnecessary gap in recognition accuracy.

RELATED WORK

We review in this section previous work on accessible touch input and discuss gesture recognition and analysis techniques.

Touch interfaces for people with visual impairments

The vast majority of previous work on designing accessible touch interfaces for people with visual impairments has focused on blind people. For instance, Kane *et al.* [17] proposed "Slide Rule," a set of audio-based interaction techniques that enable blind users to access multitouch devices, *e.g.*, Slide Rule speaks the first and last names of contacts in the phone book when the finger touches the screen. Azenkot *et al.* [7] developed "PassChords," a multitouch authentication technique for blind users that detects consecutive taps performed with one or more fingers. Oh *et al.* [33] were interested in techniques to teach touch gestures to users with visual impairments and introduced gesture sonification (*i.e.*, finger touches produce sounds, which create an audio representation of a gesture shape) and corrective verbal feedback (*i.e.*, speech feedback provided by analyzing the characteristics of the produced gesture). Buzzi *et al.* [10] and Brock *et al.* [9] investigated interaction modalities to make visual maps accessible to blind people. Kane *et al.* [19] introduced "touchplates," which are tactile physical guides of various shapes overlaid on the touch screen. In the context of ability-based design [58], Gajos *et al.* [15] developed SUPPLE++, a tool that automatically generates user interfaces that can accommodate varying vision abilities.

Despite the strong focus on applications for blind people, only few studies have examined how people with visual impairments use touch gestures [11,12,20]. Kane *et al.* [20] analyzed blind people's gestures and reported preferences for gestures that use an edge or a corner of the screen. They also found that blind people produce gestures that are larger in size and take twice as long to produce than the same gestures articulated by people without visual impairments. Buzzi *et al.* [11,12] reported preferences for one-finger one-stroke input and short gesture trajectories. However, we found no studies to address touch gestures for people with low vision. Also, there is little information in the literature regarding the recognition accuracy of gestures produced by people with visual impairments. The only data available (for blind people) comes from Kane *et al.* [20], who employed the $N recognizer [2] and reported recognition rates between 44.9% and 78.7%, depending on the training condition. Therefore, more work is needed to understand and, consequently, improve the gesture input performance of people with visual impairments on touch screens.

Gesture recognition and analysis

Gesture recognition has been successfully implemented with the Nearest-Neighbor (NN) classification approach in the context of supervised pattern matching [2,3,27,37,43,45,46,59]. The NN approach is easy to understand, straightforward to implement, and works with any gesture dissimilarity measure [38,44,50,51,59]. Moreover, training NN recognizers only requires adding and/or removing samples from the training set, making the training process effortless for user interface designers. Popular gesture recognizers implementing the NN approach include the "$-family of recognizers" composed of $1, $N, Protractor, $N-Protractor, and $P [2,3,27,46,59]. Other approaches to multitouch gesture classification, such as Gesture Coder [28], Gesture Studio [29], and Proton [22], enable developers with tools to implement gesture recognition by means of demonstration and declaration. More sophisticated recognizers are also available, such as statistical classifiers [8,40], Hidden Markov Models (HMMs) [41] and their extensions, such as Parametric HMMs [57] and Conversive Hidden non-Markovian Models [13], advanced machine learning techniques, such as Support Vector Machines, Multilayer Perceptrons [56], Hidden Conditional Random Fields [55] and, recently, deep learning [54]. However, such sophisticated approaches require advanced knowledge in machine learning or at least access to third party libraries, not always available for the latest languages or platforms. Of all these techniques, we choose to focus in this work on Nearest-Neighbor approaches due to their ease of use and implementation in any language and on any platform, as our goal is to address effective touch input for all current, but also future touch-sensitive devices.

Several tools and measures have been proposed for gesture analysis [1,39,47,48,49,52,53]. For example, GECKo [1] is a tool that computes and reports users' consistency in articulating stroke gestures. GHoST [48] generates colorful "gesture heatmaps" for practitioners to visualize users' geometric and kinematic variations in gesture articulation. Vatavu *et al.* [47] introduced a set of gesture measures with the GREAT tool to evaluate gesture accuracy relative to stored templates. In this work, we rely on this existing body of knowledge on gesture analysis to understand differences between gestures produced by people with and without visual impairments.

Participant (age, gender)	Eye condition	Diopters (right eye)†	Diopters (left eye)†
P_1 (52.1 yrs., male)	congenital nystagmus, high myopia	−16.00	−18.00
P_2 (49.2 yrs., male)	congenital nystagmus, high myopia	−12.50	−13.00
P_3 (23.9 yrs., male)	congenital nystagmus, macular dysplasia, microphthalmus, moderate myopia	−4.00	−4.50
P_4 (54.3 yrs., female)	hyperopia, macular degeneration, optic nerve atrophy	+4.00	+4.50
P_5 (31.1 yrs., male)	astigmatism, congenital nystagmus, high myopia	−16.00	−16.25
P_6 (33.5 yrs., female)	congenital nystagmus, hyperopia	+6.00	+6.25
P_7 (32.4 yrs., male)	chorioretinal degeneration, high myopia	−16.00	−14.00
P_8 (33.5 yrs., female)	chorioretinal degeneration, moderate hyperopia	+3.50	+4.00
P_9 (46.8 yrs., female)	macular choroiditis, degenerative high myopia, strabismus	−10.00	−11.00
P_{10} (19.1 yrs., male)	chorioretinal degeneration, cataract, high myopia	−6.00	−6.25

† Myopia is classified by degree of refractive error into: *low* (−3.00 diopters (D) or less), *moderate* (between −3.00 and −6.00 D), and *high* (−6.00 D or more) [5]. Hyperopia is classified into: *low* (+2.00 D or less), *moderate* (between +2.25 and +5.00 D), and *high* (refractive error over +5.00 D) [4].

Table 1. Demographic description of participants with low vision for our gesture study on touch screens.

EXPERIMENT

We conducted an experiment to collect touch gesture data from people with low vision and without visual impairments.

Participants

Twenty (20) participants (12 male, 8 female) aged between 19 and 54 years ($M = 35.7$, $SD = 11.7$ years) took part in the experiment. Ten participants had various visual impairments (see Table 1 for a description) and they were recruited from a local organization for people with disabilities. Age distributions were balanced for the two groups ($M = 37.7$ and $M = 33.7$ years, $t_{(18)} = .752$, $p > .05$, *n.s.*). Gender was distributed equally as well (6 males and 4 females in each group).

Apparatus

Gestures were collected on a Samsung Galaxy Tab 4 with a display of 10.1 inches and resolution of 1280 × 800 pixels (149 dpi), running Android OS *v*4.4.2. A custom software application was developed to implement the experiment design and to collect and log participants' touch gesture articulations.

Design

The experiment design included two independent variables:

1. Visual-Impairment, nominal variable, with two conditions (yes/no), allocated between participants.
2. Gesture, nominal variable, with twelve (12) conditions: *letter "A"*, *arrow right*, *circle*, *letter "M"*, *stickman*, *question mark*, *letter "R"*, *letter "S"*, *sol key*, *spiral*, *square*, and *star* (see Figure 2), allocated within participants.

These gestures were chosen to be representative of letters, symbols, and generic geometric shapes; see [1,2,52,59] for studies looking at similar gesture types. They were also chosen for their different complexities, between 2 and 12, which we evaluated using Isokoski's definition of shape complexity[1] [16], as well as for their different difficulty levels, 1 to 12, which we estimated using the rule[2] of Vatavu *et al.* [52] (p. 101).

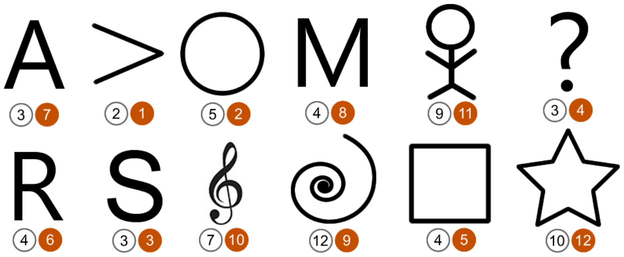

Figure 2. Gesture types used in the experiment. From top to bottom and left to right: *letter "A"*, *arrow right*, *circle*, *letter "M"*, *stickman*, *question mark*, *letter "R"*, *letter "S"*, *sol key*, *spiral*, *square*, and *star*. Numbers in circles show shape complexity [16] (white) and estimated difficulty [52] (orange); larger values denote more complexity/difficulty.

Task

Participants were instructed to perform gestures as fast and accurately as they could. Each gesture was displayed on the lower side of the screen and participants could draw on the upper side (tablet in portrait mode). The gesture to perform was shown as a large image of about 5×5 cm. A familiarization session before the experiment confirmed that all participants were able to see and identify gestures correctly. Participants were allowed to repeat a trial (up to 3 times) if they felt the gesture they had performed did not match the expected result. In total, there were 10 repetitions for each gesture and, consequently, a minimum of 120 samples were recorded per participant. The order of Gesture trials was randomized across participants. A training phase took place before the actual experiment so that participants would familiarize themselves with the device and the task. In average, gesture collection took 11.6 minutes ($SD = 4.4$) for participants without visual impairments and 19.5 minutes ($SD = 11.1$) for participants with low vision.

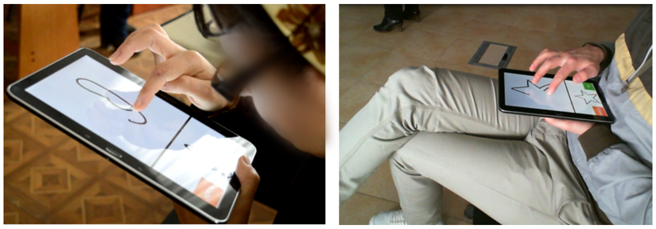

Figure 3. Two participants completing the gesture task. Left: a participant with low vision. Right: a participant without visual impairments.

[1] The Isokoski complexity of a shape represents the minimum number of linear strokes to which the shape can be reduced, yet still be recognized by a human observer [16]. For example, the Isokoski complexity of a line is 1, while the shape complexity of letter "A" is 3.

[2] Gesture A is likely to be perceived more difficult to execute than gesture B if the production time of A is greater than the time required to produce B [52].

GESTURE RECOGNITION ACCURACY

We evaluated the recognition accuracy of our participants' gesture articulations using gesture dissimilarity measures commonly employed in the literature, such as the Euclidean, Dynamic Time Warping (DTW), Hausdorff, Angular Cosine, and point-cloud matching. These measures constitute the basis of popular gesture and sketch recognizers, such as \$1, \$N, \$P, Protractor, \$N-Protractor, and SHARK^2 [2,3,21,23,27,46,59]. We decided to consider all these dissimilarity measures in our evaluation, because they each possess their own strengths in how they match gestures, *e.g.*, by mapping points directly (\$1, \$N, and SHARK^2 [2,23,59]), matching point clouds (\$P [46]), computing minimum distances between sets of points (Hausdorff [21]), working with gestures as vectors in hyperspace (Protractor and \$N-Protractor [3,27]), while DTW has been extremely popular for generic time series classification, including gestures [37,59]. To develop a thorough understanding of the recognition performance of each of these popular dissimilarity measures for gestures produced by people with low vision, we ran three distinct evaluation procedures, as follows:

1. **User-dependent training**. For this evaluation, recognizers were trained and tested with data from each participant individually. This experiment had two independent variables:
   (a) RECOGNIZER, nominal variable, with 5 conditions: Euclidean, DTW, Angular Cosine, Hausdorff, \$P.
   (b) T, ordinal variable, representing the number of training samples per gesture type available in the training set, with 4 conditions: 1, 2, 4, and 8.

   We computed user-dependent recognition rates by following methodology from the gesture recognition literature [2,3,46, 59]. For each participant, T training samples were randomly selected for each gesture type and one additional gesture sample (different from the first T) was randomly selected for testing. This procedure was repeated for 100 times for each gesture and each participant. In total, there were 20 (participants) × 5 (recognizers) × 4 (values for T) × 100 (repetitions) = 40,000 recognition trials for this evaluation.
2. **User-independent training**. In this case, recognizers were trained and tested with gesture samples from different participants from the same group. This evaluation experiment had the following independent variables:
   (a) VISUAL-IMPAIRMENT, nominal variable, with 2 conditions (yes/no), allocated between participants.
   (b) RECOGNIZER, nominal variable, with 5 conditions: Euclidean, DTW, Angular Cosine, Hausdorff, \$P.
   (c) P, ordinal variable, representing the number of training participants from which gesture samples are selected for the training set, with 4 conditions: 1, 2, 4, and 8.
   (d) T, ordinal variable, representing the number of training samples per gesture type available in the training set, with 4 conditions: 1, 2, 4, and 8.

   Again, recognition rates were computed by following methodology from the literature [46], as follows: P participants were randomly selected for training and one additional participant (different from the first P) was randomly selected for testing. This selection procedure was repeated for 100 times. T samples were randomly selected for each gesture type from each of the P training participants and one gesture sample was randomly selected for each gesture type from the testing participant. The selection of the T+1 gestures was repeated for 100 times for each P. In total, there were 2 (visual impairment conditions) × 5 (recognizers) × 4 (values for P) × 100 (repetitions for selecting P participants) × 4 (values for T) × 100 (repetitions for selecting T samples) = 1,600,000 recognition trials for this evaluation.
3. **User-independent training (variant)**. Recognizers were trained and tested with gesture samples from different participants. However, for this evaluation, participants *without* visual impairments were exclusively used for training and we evaluated recognition performance on gestures produced by participants *with low vision*. In total, there were 5 (recognizers) × 4 (values for P) × 100 (repetitions) × 4 (values for T) × 100 (repetitions) = 800,000 recognition trials.

All gestures were scaled to the unit box, resampled into $n = 32$ points, and translated to origin, according to preprocessing procedures recommended in the literature [2,3,43,45,46,59].

User-dependent recognition results

Figure 4 shows the user-dependent recognition rates for each RECOGNIZER as a function of the number of training samples T per gesture type. Overall, recognition rates were lower for participants with low vision than for participants without visual impairments (average 91.2% versus 96.5%, Mann-Whitney's $U = 2159623.500$, $Z_{(N=4800)} = -17.124$, $p < .001$, $r = .247$). This result was confirmed for each RECOGNIZER individually ($p < .001$, Bonferroni correction of .05/5=.01).

We found a significant effect of RECOGNIZER on accuracy for each VISUAL-IMPAIRMENT condition ($\chi^2_{(4,N=480)} = 233.666$ and 215.504, respectively, both $p < .001$). The \$P recognizer showed the best performance with an average accuracy of 94.0% for people with low vision and 99.0% for people without impairments; see Figure 4b. Follow-up Wilcoxon signed-rank post-hoc tests (Bonferroni corrected at $.05/4 = .0125$) showed significant differences between \$P and all the other RECOGNIZERs, except DTW (all $p < .001$, effect sizes between .189 and .468). The lowest recognition rates were exhibited by the Angular Cosine (87.7% and 94.5%) and the Euclidean dissimilarity measures (88.2% and 94.5%). The improvement in recognition accuracy determined by more training samples T was larger for participants with low vision (from 84.5% to 96.0%, gain +11.5%) than for participants without visual impairments (from 92.8% to 99.0%, gain +6.2%).

User-independent recognition results

Figure 5 shows the user-independent recognition accuracy rates for each RECOGNIZER as a function of the number of participants P providing training samples. Overall, recognition rates were lower for participants with low vision than for participants without visual impairments (average 77.5% versus 88.1%, Mann-Whitney's $U = 296071.500$, $Z_{(N=1920)} = -13.562$, $p < .001$, $r = .310$). This result was confirmed for each RECOGNIZER individually (all $p < .001$ with a Bonferroni correction of $.05/5 = .01$).

We found a significant effect of RECOGNIZER on recognition accuracy for each VISUAL-IMPAIRMENT condition

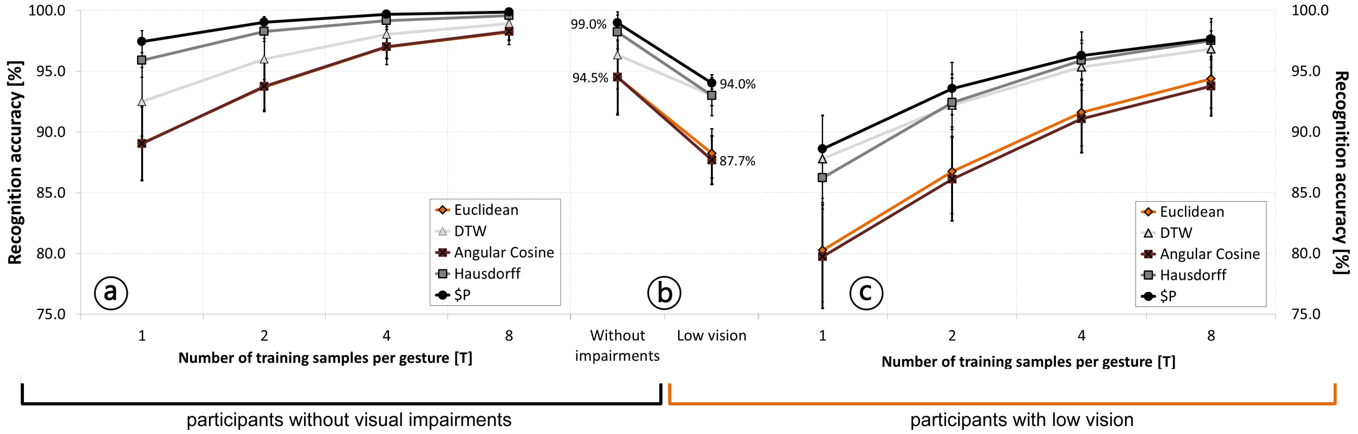

Figure 4. Recognition accuracy of several popular gesture dissimilarity measures in the context of user-dependent training for (a) participants without visual impairments, (c) participants with low vision, and (b) average accuracy for both groups.

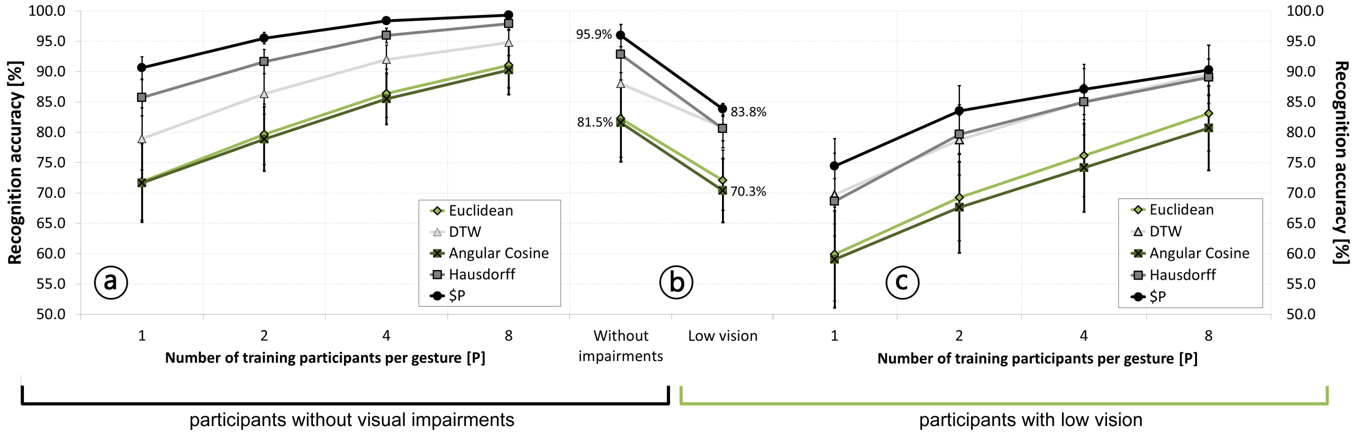

Figure 5. Recognition accuracy of several popular gesture dissimilarity measures in the context of user-independent training for (a) participants without visual impairments, (c) participants with low vision, and (b) average accuracy for both groups. NOTE: the horizontal axis shows the number of participants (P) employed for training; accuracy rates represent average values for T (training samples per gesture type) ranging from 1 to 8.

($\chi^2_{(4,N=192)}$=247.723 and 199.408, respectively, p<.001). The $P recognizer showed again the best performance with an average accuracy of 83.8% for people with low vision and 95.9% for people without visual impairments; see Figure 5b. Follow-up Wilcoxon signed-rank post-hoc tests (Bonferroni corrected at $.05/4 = .0125$) showed significant differences in recognition accuracy between $P and all the other RECOGNIZERs, except DTW. The improvement in recognition accuracy determined by higher P values (more training participants) was larger for participants with low vision (from 72.2% to 82.5%, gain +10.3%) than for participants without visual impairments (from 84.6% to 91.4%, gain +6.8%). Compared to user-dependent tests, user-independent rates were much lower for people with low vision (77.5% versus 91.8%), corresponding to an average error rate of 22.5%.

User-independent recognition results (variant)

We also computed user-independent recognition rates by training recognizers exclusively with gesture samples from participants without visual impairments and testing on gestures produced by participants with low vision. Figure 6a shows the recognition accuracy rates that we obtained for each RECOGNIZER as a function of the number of participants P (without visual impairments) that provided training samples. Overall, recognition rates were lower than when recognizers were trained with data from participants with low vision (average accuracy 75.5% and 79.0%, respectively, difference −3.5%, $Z_{(N=192)} = -3.987$, $p < .001$, $r = .288$); see Figure 6b.

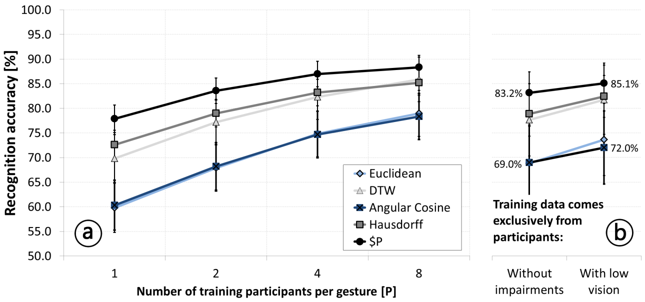

Figure 6. Recognition accuracy for participants with low vision using training samples from participants without visual impairments.

	Participants with low vision					Participants without visual impairments				
	AR	SkE	SkOE	ShE	BE	AR	SkE	SkOE	ShE	BE
Recognition rate (user-dependent)	.766**	-.599*	-.838**	-.822**	-.639*	.666*	-.638*	-.486	-.283	-.366
Recognition rate (user-independent)	.845**	-.793**	-.881**	-.524	-.572†	.849**	-.518	-.625*	-.053	-.423

** Correlation is significant at the .01 level. * Correlation is significant at the .05 level. † Correlation is marginally significant, .052.

Table 2. Pearson correlations ($N = 12$, data aggregated by gesture type) between recognition rates, consistency of gesture articulation (AR, SkE, and SkOE), and relative accuracy measures (ShE and BE) for participants with low vision (left) and without visual impairments (right).

GESTURE ARTICULATION ANALYSIS

To understand our participants' gesture articulation performance and, thus, to explain differences in recognition results between gestures produced by people with and without visual impairments, we examined the consistency [1,53] and relative accuracy [47,48] of our participants' gestures.

Consistency of gesture articulations

Gesture consistency measures how much users' articulations of a given gesture type are consistent with each other. For instance, a "square" gesture may be produced in many different ways, such as by starting from the top-left corner and drawing one continuous stroke, by making two 90-degree angles, or by using four strokes to draw each of the square's sides, etc. Consistency has been measured using agreement rates (AR) [1,53]. For instance, if out of 20 articulations of a "square" gesture, a user chooses to produce the first variant for 6 times, the second for 10 times, and the third for 4 times, then the consistency of that user's articulations for the "square" gesture is:

$$AR_{square} = \frac{6\times 5+10\times 9+4\times 3}{20\times 19} = .347 \tag{1}$$

which means that, overall, 34.7% of all the pairs of that user's gesture articulations are alike; see [53] (p. 1327) for the agreement rate formula, more discussion and examples of how to compute agreement rates. Gesture consistency, as an agreement rate, varies between 0 and 1, where 1 means absolute consistency, *i.e.*, all gestures were articulated exactly in the same way in terms of number of strokes, stroke ordering, and stroke directions. These types of variation in how users produce gestures make a recognizer's task more difficult, especially for recognizers that rely on features which expect a predefined order of strokes and points within a stroke [27,40,59] or for recognizers that cannot evaluate all the possible articulation variations of a given gesture type [2,3].

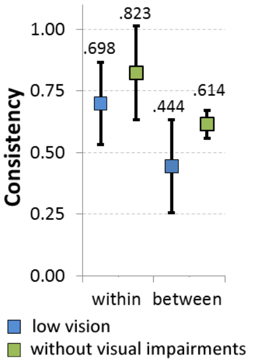

Figure 7. Consistency of gesture articulation.

We computed gesture consistency values per participant (*i.e.*, within-participant consistency) and also overall for all participants' gestures (*i.e.*, between-participants consistency), according to the methodology of Anthony *et al.* [1]; see Figure 7 for results. We found that participants with low vision were significantly less consistent in their articulations than participants without visual impairments (average consistency .698 versus .823, $U = 5622.000$, $Z_{(N=240)}=-3.164$, $p<.001$, $r=.204$). Between-participants consistency was about 30% lower, but the difference between groups was no longer significant (.444 versus .614, $U = 51.000$, $Z_{(N=24)}=-1.213$, $p>.05$, *n.s.*); see Figure 7.

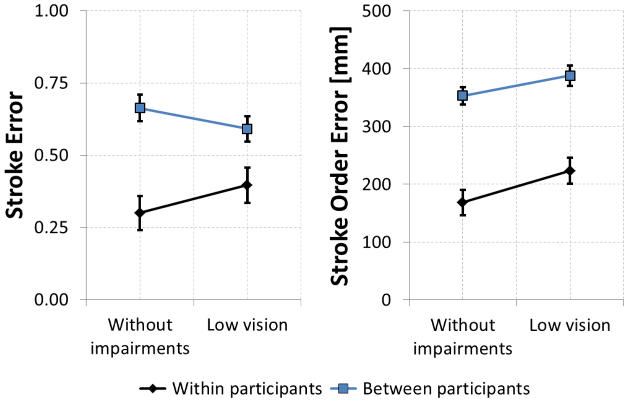

Figure 8. Gesture articulation consistency measured with the Stroke Error and Stroke Order Error accuracy measures of Vatavu *et al.* [47].

Pearson correlation analysis (Table 2) showed that gesture consistency (measured as agreement rate, AR) was significantly related to recognition accuracy: greater the consistency, higher the recognition rates for both groups ($p<.01$ and $p<.05$).

Accuracy of gesture articulation

To extend our results on gesture consistency, we employed the Stroke Error (SkE) and Stroke Order Error (SkOE) measures of Vatavu *et al.* [47]. Stroke Error computes the difference in the number of strokes of two articulations of the same gesture type, while Stroke Order Error is an indicator of stroke ordering accuracy, computed as the absolute difference between the \$1 and the \$P cost measures of matching points [47] (p. 281). Figure 8 shows the results. Participants with low vision were less consistent than participants without visual impairments in terms of the number of strokes they produced individually (within-participants SkEs were 0.40 and 0.30, Mann-Whitney's U=676428.000, $Z_{(N=2445)}$=5.491, $p<.001$, r=.111). Differences in SkE were no longer significant for group-wise analysis (between-participants SkEs were 0.66 and 0.59, U=728013.000, $Z_{(N=2445)}$=1.262, $p>.05$, *n.s.*). The order of strokes varied more for participants with low vision, both at the individual level (within-participants SkOEs 222.8 mm and 168.3 mm, Mann-Whitney's U=649872.500, $Z_{(N=2445)}$=5.601, $p<.001$, r=.113) and also per group (between-participants SkOEs were 387.6 mm and 352.8 mm, U=700426.500, $Z_{(N=2445)}$=2.670, $p<.001$, r=.054). Pearson correlation analysis (see Table 2) showed that Stroke Error and Stroke Order Error were significantly related to recog-

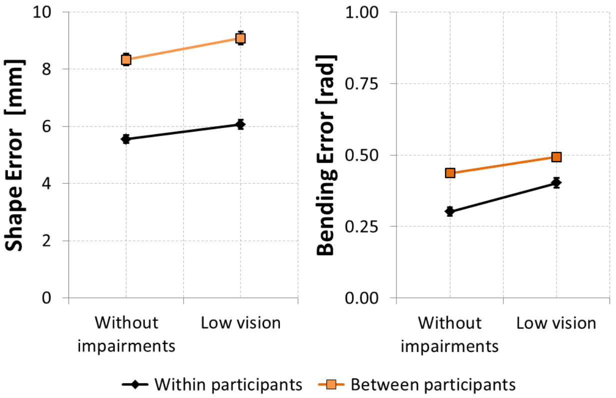

Figure 9. Geometric variation in gesture articulation measured with the Shape Error and Bending Error measures of Vatavu *et al.* [47].

nition accuracy: less variation in the number of strokes and stroke order, higher the recognition rates for both groups.

To understand variations in the geometry of the gesture shapes produced by our participants, we employed the Shape Error (ShE) and Bending Error (BE) measures of Vatavu *et al.* [47]. Shape Error computes the average absolute deviation of one gesture articulation from another in terms of the Euclidean distance. Bending Error measures users' tendencies to "bend" the strokes of a gesture with respect to a reference gesture shape; see [47] (pp. 280-281) for exact formulas. Participants with low vision varied the geometry of their gesture articulations significantly more than participants without visual impairments, both at individual level (within-participants ShEs were 6.1 mm and 5.5 mm, Mann-Whitney U=688475.000, $Z_{(N=2445)}$=3.354, p<.001, r=.068) as well as per group (between-participants ShEs 9.1 mm and 8.3 mm, Mann-Whitney U=678211.000, $Z_{(N=2445)}$=3.942, p<.001, r=.080); see Figure 9, left. Also, participants with low vision bended more their gesture strokes than participants without visual impairments, both at individual level (within-participants BEs 0.40 rad and 0.30 rad, Mann-Whitney U=569982.500, $Z_{(N=2445)}$=10.144, p<.001, r=.205) and per group (within-participants BEs were 0.40 rad and 0.30 rad, Mann-Whitney U=678485.500, $Z_{(N=2445)}$=3.926, p<.001, r=.079); see Figure 9, right. Pearson correlation analysis (Table 2) showed that ShE and BE values were significantly related to the recognition rates of gestures produced by participants with low vision (p<.01 and p<.05), but not to the recognition rates of participants without visual impairments (*n.s.* at p=.05).

Summary

We found that people with low vision are less consistent in their gesture articulations than people without visual impairments (consistency measured with AR, SkE, and SkOE), and that they produce more geometric variations for gesture shapes (captured by the ShE and BE measures). The results on gesture consistency help explain the low accuracy of the Euclidean and Angular Cosine recognizers, which match gesture points in their chronological order of input and, consequently, cannot handle variations in stroke ordering and stroke direction. These results show the need of a gesture recognizer that does not rely on the chronological order in which strokes are entered, such as $P or Hausdorff [21,46]. The lower rates delivered by $P and Hausdorff for gestures produced by people with low vision compared to gestures produced by people without visual impairments are explained by the geometric variations captured by the ShE measure. Larger deviations in the x and y coordinates for the points making up the gesture shape result in suboptimal point matchings produced by these recognizers. This result highlights the need for a more flexible point matching strategy for $P, which we present in the next section. Furthermore, the geometric variations captured in the BE measure and the correlations between BE and recognition rates for participants with low vision point to the importance of local gesture shape information (such as the turning angles employed by the BE measure) for matching points on gestures produced by people with low vision. We build on these findings in the next section and introduce $P+, an algorithmic improvement of the $P recognizer for people with low vision.

THE $P+ GESTURE RECOGNIZER

In this section, we use our gesture analysis results to inform key changes in how the $P recognizer matches points for gestures produced by people with low vision. First, we briefly review how $P operates. Then, we introduce $P+, an updated algorithmic design for the $P recognizer, that improves recognition accuracy and execution speed for gestures produced by people with low vision.

Overview of the $P point-cloud gesture recognizer

The $P recognizer was introduced by Vatavu *et al.* [46] as an *articulation-independent gesture recognizer*. The independence of $P from the articulation details of a gesture (*e.g.*, number of strokes, stroke ordering, or stroke direction) is a direct consequence of $P representing gestures as clouds of points with no timestamps. Given two point clouds, a candidate C and a template T, which have been previously resampled into the same number of n points, $P computes an (approximate) optimum alignment between C and T and returns the sum of Euclidean distances between the aligned points as the dissimilarity between gestures C and T:

$$\delta(C,T) = \sum_{i=1}^{n} \left(1 - \frac{i-1}{n}\right) \cdot \left\|C_i - T_j\right\| \qquad (2)$$

where $\left\|C_i - T_j\right\|$ is the Euclidean distance between points C_i and T_j, with T_j being an unmatched point from cloud T that is closest to C_i. The $P dissimilarity measure is defined as $\min\{\delta(C,T), \delta(T,C)\}$; see Vatavu *et al.* [46] (p. 276).

The $P+ gesture recognizer

In the following, we introduce two key changes in how $P matches the points of the candidate and the template gestures.

❶ **Optimal alignment between points.** $P computes an approximate optimal alignment between two point clouds by relying on a strategy that compromises true optimal alignment [36] (p. 248) and execution time [46] (p. 276). In this process, each point C_i from the first cloud is matched to point T_j from the second cloud if T_j is closest to C_i and T_j has not been matched before. Unfortunately, this constraint leads to suboptimal matchings between points (see Figure 10), which

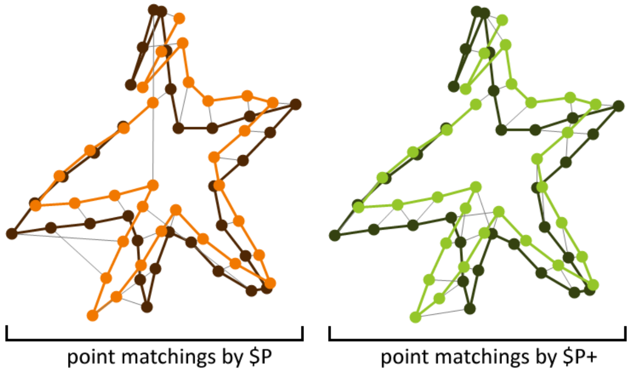

Figure 10. Point matchings produced by $P (left) and $P+ (right) for two "star" gestures produced by a participant with low vision. Note how the one-to-many matchings of $P+ help reduce undesirable point matchings of $P (see the long connecting lines in the figure on the left).

are matched by $P simply because they represent the "best" option when considering all the unmatched points available at matching time. The result is an artificial increase in the magnitude of the sum of Euclidean distances (eq. 2), which we spotted in the previous section with the ShE and SkOE measures. Our alternative is to give each point C_i the chance to match its closest point from the second cloud T, even if that closest point has been used before in other matchings; see Figure 10. Once all the points from the first cloud have been matched, it is possible that some points from the second cloud are still unmatched. These points will be matched with their closest points from the first cloud. Equation 2 becomes:

$$\delta^{+}(C,T) = \sum_{i=1}^{n} \min_{j=1,n} \left\| C_i - T_j \right\| + \sum_{j} \min_{i=1,n} \left\| C_i - T_j \right\| \qquad (3)$$

where the first sum goes through all the points C_i of the first cloud and the second sum goes through all the points T_j from the second cloud that were not matched in the first sum. The weights $(1 - \frac{i-1}{n})$ from eq. 2 are now superfluous, since the one-to-many matchings alleviate the need to restart the cloud matching procedure with different starting points [46] (p. 276). The new $P+ dissimilarity measure is:

$$\$P{+}(C,T) = \min\left\{\delta^{+}(C,T), \delta^{+}(T,C)\right\} \qquad (4)$$

The modified point cloud distance algorithm is illustrated by the CLOUD-DISTANCE procedure in the pseudocode at the end of this paper. Note that one-to-many matchings for $P have been proposed before by Martez *et al.* [31] in the context of classifying touch patterns for people with motor impairments. In that version, the authors implemented a recursive approach that switches periodically the order of the two clouds until all the points are matched (p. 1941). The version that we propose considers a strategy that requires no recursive calls, matches the two point clouds in maximum two runs only, and is faster (from $O(n^{2.5})$ to $O(n^2)$, see our discussion next in the paper).

❷ Exploiting the shape structure of the point cloud. To achieve articulation independence, the $P recognizer dismisses the order in which strokes and points within a stroke are produced. Thus, a gesture point cloud is actually a set of 2-D points with no particular ordering. $P produces matchings between points by solely considering their x and y coordinates and disregards any connections between consecutive points. However, it is those connections that actually make up the shape of a gesture and $P misses that aspect entirely. A connection between consecutive points C_{i-1}, C_i, and C_{i+1} is completely characterized by the curvature at point C_i. (It is a known fact in differential geometry that the curvature signature function of a planar curve parameterized by arc-length fully prescribes the original curve up to a rigid motion transformation [14].) The curvature at a given point is defined as the change in tangential angle at that point divided by the amount of change in arc-length ($d\theta/ds$). However, because we resample gestures uniformly, the change in arc-length (ds) is constant between any two consecutive points on the gesture path, which allows us to simplify the definition of curvature to the turning angle θ between vectors $\overrightarrow{C_iC_{i+1}}$ and $\overrightarrow{C_{i-1}C_i}$. Based on these observations, we propose the following updated formula to evaluate the distance between two points C_i and T_j for the purpose of more efficient point matching:

$$\left\| C_i - T_j \right\| = \sqrt{(C_{i.x} - T_{j.x})^2 + (C_{i.y} - T_{j.y})^2 + \left(C_{i.\theta} - T_{j.\theta}\right)^2} \qquad (5)$$

in which we added to the Euclidean distance the difference in turning angles $\left(C_{i.\theta} - T_{j.\theta}\right)^2$. The idea to use angles for matching points is supported by our results on the Bending Error measure (see the previous section), which showed that variations in turning angles across the gesture path are important enough to determine significant differences between the gesture articulations of our two groups of participants (Figure 9) and that BE correlates negatively with recognition rates for gestures produced by participants with low vision (Table 2).

Because we still want our recognizer to be independent of the directions of gesture strokes (note that turning angles are equal in magnitude, but with opposite signs when the order of the points changes, *i.e.*, C_{i-1}, C_i, C_{i+1} versus C_{i+1}, C_i, C_{i-1}), we compute θ as the absolute shortest angle between vectors $\overrightarrow{C_iC_{i+1}}$ and $\overrightarrow{C_{i-1}C_i}$, with values in $[0..\pi]$ (see Figure 11). Because θ is now part of the Euclidean distance, we also need to make sure that differences in θ are of similar magnitude as differences in the x and y coordinates or, otherwise, the distance from eq. 5 will be biased towards the variable with the largest magnitude. As gestures already go through preprocessing and are scaled in the unit box [2,46,59], we also normalize θ in the interval $[0..1]$ by dividing it by π. $C_{i.\theta}$ and $T_{j.\theta}$ from eq. 5 are thus real values between 0 and 1. The modified point distance is illustrated in the POINT-DISTANCE procedure in the pseudocode provided at the end of this paper.

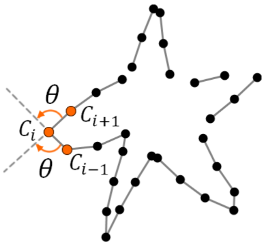

Figure 11. Turning angle at point C_i on the gesture path.

EVALUATION OF THE $P+ GESTURE RECOGNIZER

We evaluated the $P+ gesture recognizer using the same methodology as before. In this section, we focus primarily on the user-independent recognition performance of $P+ for gestures produced by people with low vision, for which we observed low recognition performance delivered by all the recognizers that we evaluated, *i.e.*, 77.5% average accuracy overall and 83.8% accuracy for $P; see the previous sections.

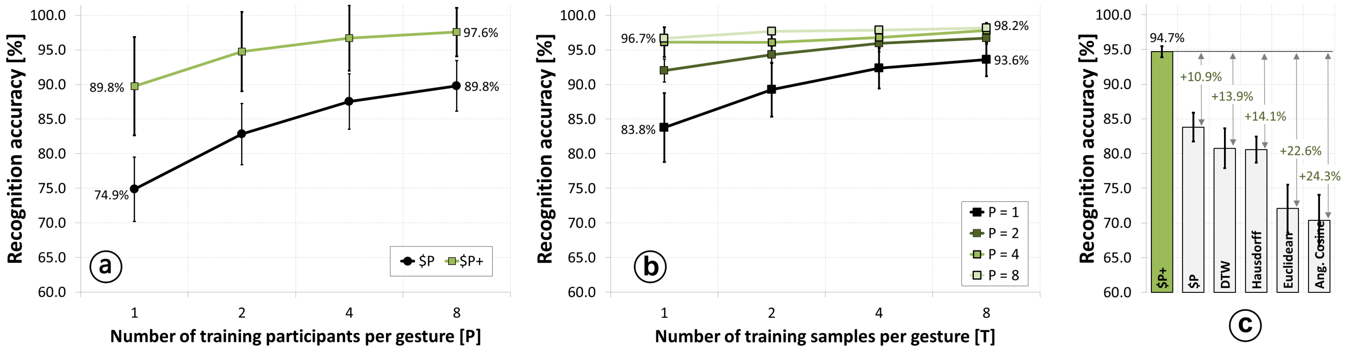

Figure 12. Recognition accuracy rates (user-independent) for gestures articulated by people with low vision: (a) direct comparison between $P+ and $P; (b) combined effect of the number of training participants P and the number of training samples per gesture type T on the recognition accuracy of $P+; (c) gain in recognition accuracy enabled by $P+ over the other gesture dissimilarity measures.

Recognition rates for people with low vision

We report recognition results from a total number of 320,000 trials by evaluating 2 recognizers ($P and $P+) × 4 (conditions for the number of training participants P) × 100 (repetitions for P) × 4 (conditions for the number of samples per gesture type T) × 100 (repetitions for T). Figure 12a shows the recognition performance of $P+ compared to $P for gestures produced by people with low vision. A Wilcoxon signed-rank test showed a statistically significant difference of +10.9% in recognition performance between $P+ and $P (94.7% versus 83.8%, $Z_{(N=192)}$=11.915, p<.001) with a large effect size (r=.860). Recognition rates delivered by $P+ for gestures produced by people with low vision are now comparable to the rates delivered by the original $P recognizer for people without visual impairments; see the previous section. Originally, we found a difference in accuracy between the two groups of 12.1% (83.8% versus 95.9%; see Figure 5b). Now, $P+ reached 94.7% recognition accuracy on average, reducing the difference to merely 1.2%. Having reached this important recognition accuracy milestone, we can now examine the performance of the $P+ gesture recognizer in more detail.

Effect of the size of the training set on recognition rates

Figure 12b illustrates the effect of the number of training participants P and the number of samples per gesture type T on the recognition accuracy of $P+. Overall, training data from more participants improved the accuracy of $P+ from 89.8% to 97.6% ($\chi^2_{(3,N=48)}$=125.623, p<.001). More training samples per gesture type increased recognition accuracy as well ($\chi^2_{(3,N=48)}$=82.911, p<.001). $P+ delivered a recognition accuracy over 92.0% with just one training sample per gesture type from two participants with low vision, reached +96.0% when four participants provided one gesture sample, and reached 98.2% with training data from eight participants, each providing eight samples per gesture type; see Figure 12b. The error rate for the maximum size of the training set that we evaluated (8 participants × 8 samples) was just 1.8%.

Figure 12c illustrates the gain in recognition accuracy (averaged across all P and T conditions) brought by $P+ compared to all the other recognizers evaluated in this work. For example, $P+ adds +10.9% accuracy to $P and +14.1% accuracy to the Hausdorff dissimilarity measure. These results show that $P+ performs a much better alignment between gesture points than its peers, $P and Hausdorff, which also work with point cloud representations. Also, $P+ outperforms gesture recognizers that are not flexible to the multitude of variations possible in gesture articulation: +22.6% extra accuracy over the Euclidean distance (implemented by the $1 recognizer [59]) and +24.3% difference with respect to the Angular Cosine dissimilarity measure (implemented by Protractor [27]). While DTW does implement some flexibility in point matching, it still cannot deal properly with variations in articulations that deviate substantially from predefined ways to produce a gesture (the difference in accuracy compared to $P+ was 13.9%).

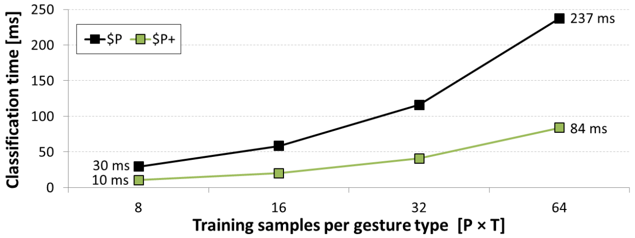

Figure 13. Average classification times of $P and $P+, function of the number of training samples per gesture type (P × T × 12 gesture types). Times were measured on a Intel® Core™2 Quad CPU Q6600 @2.4GHz.

Classification speed

$P+ has lower time complexity and, consequently, classification times measured in practice decrease substantially. Figure 13 shows classification times for $P and $P+ function of the number of samples per gesture type (P × T) in the training set. We found that $P+ was 3 times faster than the original $P recognizer (38.9 ms versus 110.3 ms, $Z_{(N=400)}$=−17.331, p<.001). This speed-up is achieved because $P+ does not need to perform repetitive point matchings with various starting points as $P does, which reduces its time complexity from $O(n^{2.5})$ to $O(n^2)$, where n is the number of points on the gesture path. While we only report classification times for n=32 points, greater speed-ups are expected for larger n values.

Extended applicability of $P+

We informed the algorithmic improvements of the $P+ recognizer by key observations that we derived by analyzing gestures articulated by people with low vision. Our main goal in this work was to improve the classification performance of state-of-the-art gesture recognizers for touch gestures produced by people with low vision. However, it is interesting to evaluate how $P+ performs on other gestures. To find that out, we repeated our recognition experiment for gestures articulated by people *without* visual impairments using the same conditions as before (320,000 recognition trials). We found that $P+ delivered higher recognition accuracy than $P (98.3% versus 95.9%, $Z_{(N=192)}$=8.990, p<.001) with an overall gain of +2.4% and a medium effect size (r=.459). Friedman tests showed statistically significant effects of the number of training participants ($\chi^2_{(3,N=48)}$=88.841, p<.001) and the number of training samples per gesture type ($\chi^2_{(3,N=48)}$=81.377, p<.001) on the recognition rates delivered by $P+ for gestures produced by people without visual impairments. These results suggest that our algorithmic improvements in the design of the $P+ recognizer are likely to find applications for other user populations as well, but we leave the confirmation of this hypothesis as well as follow-up explorations for future work.

GESTURE DATASET

The lack of public gesture datasets for people with visual impairments motivated us to release our gesture dataset in the community. Our dataset, composed of 2445 gesture samples of 12 distinct gesture types collected from 20 participants (10 participants with low vision), can be downloaded from `http://www.eed.usv.ro/~vatavu` and used for free for research purposes. By publicly releasing this dataset, we hope to foster new developments in gesture recognition and assistive touch gesture input techniques for people with low vision, advancing scientific knowledge in the community.

CONCLUSION AND FUTURE WORK

We evaluated in this work the recognition accuracy of state-of-the-art gesture recognizers based on the Nearest-Neighbor approach for gestures produced by people with low vision, for which we observed an average error rate of 22.5%. By carefully analyzing gesture articulations, we made key observations that informed algorithmic improvements in the core design of the $P point-cloud gesture recognizer. The new recognizer iteration, $P+, increased accuracy for gestures produced by people with low vision up to 98.2% (error rate 1.8%). Our contributions will help make touch interfaces more accessible to people with low vision, enabling them to use gestures that are recognized as accurately as gestures produced by people without visual impairments, removing thus the unnecessary gap in recognition accuracy between the two groups.

Future work will evaluate the performance of other, more sophisticated machine learning approaches for touch gesture recognition. A comparison between the "$-family of gesture recognizers" [2,3,27,46,59], invented for their ease of implementation and portability across languages and platforms, and standard machine learning approaches, such as statistical classifiers, HMMs, SVMs, neural networks, and deep learning [13,41,54,55], has been long overdue in the community. Running such comparison evaluations can be easily facilitated by large gesture training sets, readily available with gesture synthesis [24,25]. Future work will explore such interesting directions in order to extend our knowledge about designing robust gesture recognizers for efficient touch and gesture input for people with all abilities.

ACKNOWLEDGMENTS

This work was supported from the project PN-II-RU-TE-2014-4-1187 financed by UEFISCDI, Romania.

APPENDIX A: PSEUDOCODE

We present in this section minimal pseudocode for $P+ based on the $P pseudocode from Vatavu *et al.* [46] (p. 280). For space concerns, we only list the parts of $P that we updated. Complete pseudocode for $P+ as well as C# and JavaScript implementations are available at `http://www.eed.usv.ro/~vatavu`. In the following, POINT is a structure that exposes x and y coordinates, the stroke ID, and the normalized turning angle θ. POINTS is a list of points and TEMPLATES a list of POINTS with gesture class data.

```
$P+RECOGNIZER (POINTS $C$, TEMPLATES $templates$)
$n \leftarrow 32$
NORMALIZE($C$, n)
$score \leftarrow \infty$
for all $T$ in $templates$ do
    NORMALIZE($T$, n) // should be pre-processed
    $d \leftarrow \min(\text{CLOUD-DISTANCE}(C,T), \text{CLOUD-DISTANCE}(T,C))$
    if $score > d$ then
        $score \leftarrow d$
        $result \leftarrow T$
return $\langle result, score \rangle$
```

```
CLOUD-DISTANCE (POINTS $C$, POINTS $T$, int $n$)
$matched \leftarrow$ new bool[$n$]
$sum \leftarrow 0$
// match points from cloud $C$ with points from $T$; one-to-many matchings allowed
for $i = 1$ to $n$ do
    $min \leftarrow \infty$
    for $j = 1$ to $n$ do
        $d \leftarrow$ POINT-DISTANCE($C_i$, $T_j$)
        if $d < min$ then
            $min \leftarrow d$
            $index \leftarrow j$
    $matched[index] \leftarrow true$
    $sum \leftarrow sum + min$
// match remaining points $T$ with points from $C$; one-to-many matchings allowed
for all $j$ such that not $matched[j]$ do
    $min \leftarrow \infty$
    for $i = 1$ to $n$ do
        $d \leftarrow$ POINT-DISTANCE($C_i$, $T_j$)
        if $d < min$ then $min \leftarrow d$
    $sum \leftarrow sum + min$
return $sum$
```

```
POINT-DISTANCE (POINT $a$, POINT $b$)
return $\left((a.x - b.x)^2 + (a.y - b.y)^2 + (a.\theta - b.\theta)^2\right)^{\frac{1}{2}}$
```

```
NORMALIZE (POINTS $points$, int $n$)
$points \leftarrow$ RESAMPLE($points$, $n$)
SCALE($points$)
TRANSLATE-TO-ORIGIN($points$, $n$)
COMPUTE-NORMALIZED-TURNING-ANGLES($points$, n)
```

```
COMPUTE-NORMALIZED-TURNING-ANGLES (POINT $C$, int $n$)
$C_{1.\theta} \leftarrow 0, C_{n.\theta} \leftarrow 0$
for $i = 2$ to $n - 1$ do
    $C_{i.\theta} \leftarrow \frac{1}{\pi} \arccos\left(\frac{(C_{i+1.x} - C_{i.x}) \cdot (C_{i.x} - C_{i-1.x}) + (C_{i+1.y} - C_{i.y}) \cdot (C_{i.y} - C_{i-1.y})}{\|C_{i+1} - C_i\| \cdot \|C_i - C_{i-1}\|}\right)$
return
```

REFERENCES

1. Lisa Anthony, Radu-Daniel Vatavu, and Jacob O. Wobbrock. 2013. Understanding the Consistency of Users' Pen and Finger Stroke Gesture Articulation. In *Proc. of Graphics Interface 2013 (GI '13)*. Canadian Inf. Processing Society, 87–94. http://dl.acm.org/citation.cfm?id=2532129.2532145

2. Lisa Anthony and Jacob O. Wobbrock. 2010. A Lightweight Multistroke Recognizer for User Interface Prototypes. In *Proc. of Graphics Interface 2010 (GI '10)*. Canadian Inf. Processing Society, 245–252. http://dl.acm.org/citation.cfm?id=1839214.1839258

3. Lisa Anthony and Jacob O. Wobbrock. 2012. $N-protractor: A Fast and Accurate Multistroke Recognizer. In *Proc. of Graphics Interface 2012 (GI '12)*. Canadian Inf. Processing Society, 117–120. http://dl.acm.org/citation.cfm?id=2305276.2305296

4. American Optometric Association. 1997a. Optometric Clinical Practice Guideline: Care of the Patient with Hyperopia. (1997). http://www.aoa.org/documents/optometrists/CPG-16.pdf

5. American Optometric Association. 1997b. Optometric Clinical Practice Guideline: Care of the Patient with Myopia. (1997). http://www.aoa.org/documents/optometrists/CPG-15.pdf

6. American Optometric Association. 1997c. Optometric Clinical Practice Guideline: Care of the Patient with Visual Impairment (Low Vision Rehabilitation). (1997). http://www.aoa.org/documents/optometrists/CPG-14.pdf

7. Shiri Azenkot, Kyle Rector, Richard Ladner, and Jacob Wobbrock. 2012. PassChords: Secure Multi-touch Authentication for Blind People. In *Proc. of the 14th Int. ACM Conf. on Computers and Accessibility (ASSETS '12)*. ACM, New York, NY, USA, 159–166. DOI: http://dx.doi.org/10.1145/2384916.2384945

8. Rachel Blagojevic, Samuel Hsiao-Heng Chang, and Beryl Plimmer. 2010. The Power of Automatic Feature Selection: Rubine on Steroids. In *Proc. of the Seventh Sketch-Based Interfaces and Modeling Symposium (SBIM '10)*. Eurographics Association, 79–86. http://dl.acm.org/citation.cfm?id=1923363.1923377

9. Anke Brock, Philippe Truillet, Bernard Oriola, and Christophe Jouffrais. 2014. *Making Gestural Interaction Accessible to Visually Impaired People*. Springer Berlin Heidelberg, Berlin, Heidelberg, 41–48. DOI:http://dx.doi.org/10.1007/978-3-662-44196-1_6

10. Maria Claudia Buzzi, Marina Buzzi, Barbara Leporini, and Loredana Martusciello. 2011. Making Visual Maps Accessible to the Blind. In *Proc. of the 6th Int. Conf. on Universal Access in Human-Computer Interaction: Users Diversity (UAHCI'11)*. Springer, 271–280. http://dl.acm.org/citation.cfm?id=2027376.2027408

11. Maria Claudia Buzzi, Marina Buzzi, Barbara Leporini, and Amaury Trujillo. 2015. Exploring Visually Impaired People's Gesture Preferences for Smartphones. In *Proc. of the 11th Conf. on Italian SIGCHI Chapter (CHItaly 2015)*. ACM, New York, NY, USA, 94–101. DOI: http://dx.doi.org/10.1145/2808435.2808448

12. Maria Claudia Buzzi, Marina Buzzi, Barbara Leporini, and Amaury Trujillo. 2016. Analyzing visually impaired people's touch gestures on smartphones. *Multimedia Tools and Applications* (2016), 1–29. DOI:http://dx.doi.org/10.1007/s11042-016-3594-9

13. Tim Dittmar, Claudia Krull, and Graham Horton. 2015. A new approach for touch gesture recognition: Conversive Hidden non-Markovian Models. *Joural of Computational Science* 10 (2015), 66–76. http://dx.doi.org/10.1016/j.jocs.2015.03.002

14. Manfredo Do Carmo. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, New Jersey, USA.

15. Krzysztof Z. Gajos, Jacob O. Wobbrock, and Daniel S. Weld. 2007. Automatically Generating User Interfaces Adapted to Users' Motor and Vision Capabilities. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. ACM, New York, NY, USA, 231–240. DOI: http://dx.doi.org/10.1145/1294211.1294253

16. Poika Isokoski. 2001. Model for Unistroke Writing Time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. ACM, New York, NY, USA, 357–364. DOI: http://dx.doi.org/10.1145/365024.365299

17. Shaun K. Kane, Jeffrey P. Bigham, and Jacob O. Wobbrock. 2008. Slide Rule: Making Mobile Touch Screens Accessible to Blind People Using Multi-touch Interaction Techniques. In *Proc. of the 10th Int. ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '08)*. ACM, New York, NY, USA, 73–80. DOI: http://dx.doi.org/10.1145/1414471.1414487

18. Shaun K. Kane, Chandrika Jayant, Jacob O. Wobbrock, and Richard E. Ladner. 2009. Freedom to Roam: A Study of Mobile Device Adoption and Accessibility for People with Visual and Motor Disabilities. In *Proc. of the 11th Int. ACM Conf. on Computers and Accessibility (ASSETS '09)*. ACM, New York, NY, USA, 115–122. DOI: http://dx.doi.org/10.1145/1639642.1639663

19. Shaun K. Kane, Meredith Ringel Morris, and Jacob O. Wobbrock. 2013. Touchplates: Low-cost Tactile Overlays for Visually Impaired Touch Screen Users. In *Proc. of the 15th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '13)*. ACM, New York, NY, USA, Article 22, 8 pages. DOI: http://dx.doi.org/10.1145/2513383.2513442

20. Shaun K. Kane, Jacob O. Wobbrock, and Richard E. Ladner. 2011. Usable Gestures for Blind People: Understanding Preference and Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 413–422. DOI: http://dx.doi.org/10.1145/1978942.1979001

21. Levent Burak Kara and Thomas F. Stahovich. 2004. Hierarchical Parsing and Recognition of Hand-sketched Diagrams. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, NY, USA, 13–22. DOI: http://dx.doi.org/10.1145/1029632.1029636

22. Kenrick Kin, Björn Hartmann, Tony DeRose, and Maneesh Agrawala. 2012. Proton: Multitouch Gestures As Regular Expressions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2885–2894. DOI: http://dx.doi.org/10.1145/2207676.2208694

23. Per-Ola Kristensson and Shumin Zhai. 2004. SHARK2: A Large Vocabulary Shorthand Writing System for Pen-based Computers. In *Proc. of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, NY, USA, 43–52. DOI: http://dx.doi.org/10.1145/1029632.1029640

24. Luis A. Leiva, Daniel Martín-Albo, and Réjean Plamondon. 2015. Gestures À Go Go: Authoring Synthetic Human-Like Stroke Gestures Using the Kinematic Theory of Rapid Movements. *ACM Trans. Intell. Syst. Technol.* 7, 2, Article 15 (Nov. 2015), 29 pages. DOI:http://dx.doi.org/10.1145/2799648

25. Luis A. Leiva, Daniel Martín-Albo, and Radu-Daniel Vatavu. 2017. Synthesizing Stroke Gestures Across User Populations: A Case for Users with Visual Impairments. In *Proc. of the 35th ACM Conf. on Human Factors in Computing Systems (CHI '17)*. ACM. http://dx.doi.org/10.1145/3025453.3025906

26. L.A. Levin and D.M. Albert. 2010. *Ocular Disease: Mechanisms and Management*. Elsevier.

27. Yang Li. 2010. Protractor: A Fast and Accurate Gesture Recognizer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 2169–2172. DOI: http://dx.doi.org/10.1145/1753326.1753654

28. Hao Lü and Yang Li. 2012. Gesture Coder: A Tool for Programming Multi-touch Gestures by Demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2875–2884. DOI: http://dx.doi.org/10.1145/2207676.2208693

29. Hao Lü and Yang Li. 2013. Gesture Studio: Authoring Multi-touch Interactions Through Demonstration and Declaration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 257–266. DOI: http://dx.doi.org/10.1145/2470654.2470690

30. David McGookin, Stephen Brewster, and WeiWei Jiang. 2008. Investigating Touchscreen Accessibility for People with Visual Impairments. In *Proc. of the 5th Nordic Conf. on Human-Computer Interaction (NordiCHI '08)*. ACM, New York, NY, USA, 298–307. DOI: http://dx.doi.org/10.1145/1463160.1463193

31. Martez E. Mott, Radu-Daniel Vatavu, Shaun K. Kane, and Jacob O. Wobbrock. 2016. Smart Touch: Improving Touch Accuracy for People with Motor Impairments with Template Matching. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1934–1946. DOI: http://dx.doi.org/10.1145/2858036.2858390

32. C.S. Myers and L.R. Rabiner. 1981. A comparative study of several dynamic time-warping algorithms for connected word recognition. *The Bell System Technical Journal* 60, 7 (1981), 1389–1409. http://dx.doi.org/10.1002/j.1538-7305.1981.tb00272.x

33. Uran Oh, Stacy Branham, Leah Findlater, and Shaun K. Kane. 2015. Audio-Based Feedback Techniques for Teaching Touchscreen Gestures. *ACM Trans. Access. Comput.* 7, 3, Article 9 (Nov. 2015), 29 pages. DOI: http://dx.doi.org/10.1145/2764917

34. World Health Organization. 2014. Visual impairment and blindness. Fact Sheet N282. (2014). http://www.who.int/mediacentre/factsheets/fs282/en/

35. World Health Organization. 2016. ICD-10, Visual disturbances and blindness. (2016). http://apps.who.int/classifications/icd10/browse/2016/en#/H53-H54

36. C. H. Papadimitriou and K. Steiglitz. 1998. *Combinatorial optimization: algorithms and complexity*. Dover Publications, Mineola, New York, USA.

37. Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2012. Searching and Mining Trillions of Time Series Subsequences Under Dynamic Time Warping. In *Proc. of the 18th ACM Int. Conf. on Knowledge Discovery and Data Mining (KDD '12)*. ACM, New York, NY, USA, 262–270. DOI: http://dx.doi.org/10.1145/2339530.2339576

38. Yosra Rekik, Radu-Daniel Vatavu, and Laurent Grisoni. 2014a. Match-up & Conquer: A Two-Step Technique for Recognizing Unconstrained Bimanual and Multi-finger Touch Input. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*. ACM, New York, NY, USA, 201–208. DOI: http://dx.doi.org/10.1145/2598153.2598167

39. Yosra Rekik, Radu-Daniel Vatavu, and Laurent Grisoni. 2014b. Understanding Users' Perceived Difficulty of Multi-Touch Gesture Articulation. In *Proceedings of the 16th International Conference on Multimodal Interaction (ICMI '14)*. ACM, New York, NY, USA, 232–239. DOI: http://dx.doi.org/10.1145/2663204.2663273

40. Dean Rubine. 1991. Specifying Gestures by Example. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '91)*. ACM, New York, NY, USA, 329–337. DOI: http://dx.doi.org/10.1145/122718.122753

41. Tevfik Metin Sezgin and Randall Davis. 2005. HMM-based Efficient Sketch Recognition. In *Proc. of the 10th Int. Conf. on Intelligent User Interfaces (IUI '05)*. ACM, New York, NY, USA, 281–283. DOI: http://dx.doi.org/10.1145/1040830.1040899

42. Kristen Shinohara and Josh Tenenberg. 2007. Observing Sara: A Case Study of a Blind Person's Interactions with Technology. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility (Assets '07)*. ACM, New York, NY, USA, 171–178. DOI: http://dx.doi.org/10.1145/1296843.1296873

43. Radu-Daniel Vatavu. 2011. The Effect of Sampling Rate on the Performance of Template-based Gesture Recognizers. In *Proc. of the 13th Int. Conf. on Multimodal Interfaces (ICMI '11)*. ACM, 271–278. DOI: http://dx.doi.org/10.1145/2070481.2070531

44. Radu-Daniel Vatavu. 2012. 1F: One Accessory Feature Design for Gesture Recognizers. In *Proc. of the 2012 ACM Int. Conf. on Intelligent User Interfaces (IUI '12)*. ACM, New York, NY, USA, 297–300. DOI: http://dx.doi.org/10.1145/2166966.2167022

45. Radu-Daniel Vatavu. 2013. The Impact of Motion Dimensionality and Bit Cardinality on the Design of 3D Gesture Recognizers. *Int. J. Hum.-Comput. Stud.* 71, 4 (April 2013), 387–409. DOI:http://dx.doi.org/10.1016/j.ijhcs.2012.11.005

46. Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2012. Gestures As Point Clouds: A $P Recognizer for User Interface Prototypes. In *Proc. of the 14th ACM Int. Conference on Multimodal Interaction (ICMI '12)*. ACM, New York, NY, USA, 273–280. DOI: http://dx.doi.org/10.1145/2388676.2388732

47. Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2013. Relative Accuracy Measures for Stroke Gestures. In *Proc. of the 15th ACM Int. Conf. on Multimodal Interaction (ICMI '13)*. ACM, 279–286. DOI: http://dx.doi.org/10.1145/2522848.2522875

48. Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2014. Gesture Heatmaps: Understanding Gesture Performance with Colorful Visualizations. In *Proc. of the 16th Int. Conf. on Multimodal Interaction (ICMI '14)*. ACM, New York, NY, USA, 172–179. DOI: http://dx.doi.org/10.1145/2663204.2663256

49. Radu-Daniel Vatavu, Gabriel Cramariuc, and Doina Maria Schipor. 2015. Touch Interaction for Children Aged 3 to 6 Years: Experimental Findings and Relationship to Motor Skills. *International Journal of Human-Computer Studies* 74 (2015), 54–76. http://dx.doi.org/10.1016/j.ijhcs.2014.10.007

50. Radu-Daniel Vatavu, Laurent Grisoni, and Stefan-Gheorghe Pentiuc. 2009. Gesture-Based Human-Computer Interaction and Simulation. Springer-Verlag, Chapter Gesture Recognition Based on Elastic Deformation Energies, 1–12. DOI:http://dx.doi.org/10.1007/978-3-540-92865-2_1

51. Radu-Daniel Vatavu, Laurent Grisoni, and Stefan-Gheorghe Pentiuc. 2010. Multiscale Detection of Gesture Patterns in Continuous Motion Trajectories. In *Proc. of the 8th Int. Conf. on Gesture in Embodied Communication and Human-Computer Interaction (GW'09)*. Springer-Verlag, 85–97. DOI:http://dx.doi.org/10.1007/978-3-642-12553-9_8

52. Radu-Daniel Vatavu, Daniel Vogel, Géry Casiez, and Laurent Grisoni. 2011. Estimating the Perceived Difficulty of Pen Gestures. In *Proc. of the 13th IFIP TC 13 Int. Conf. on Human-computer Interaction (INTERACT'11)*. Springer, 89–106. http://dl.acm.org/citation.cfm?id=2042118.2042130

53. Radu-Daniel Vatavu and Jacob O. Wobbrock. 2015. Formalizing Agreement Analysis for Elicitation Studies: New Measures, Significance Test, and Toolkit. In *Proc. of the 33rd Annual ACM Conf. on Human Factors in Computing Systems (CHI '15)*. ACM, 1325–1334. DOI: http://dx.doi.org/10.1145/2702123.2702223

54. Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. 2016. Interacting with Soli: Exploring Fine-Grained Dynamic Gesture Recognition in the Radio-Frequency Spectrum. In *Proc. of the 29th Annual Symp. on User Interface Software and Technology (UIST '16)*. ACM, 851–860. DOI: http://dx.doi.org/10.1145/2984511.2984565

55. Sy Bor Wang, Ariadna Quattoni, Louis-Philippe Morency, and David Demirdjian. 2006. Hidden Conditional Random Fields for Gesture Recognition. In *Proc. of the 2006 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR '06)*. 1521–1527. DOI: http://dx.doi.org/10.1109/CVPR.2006.132

56. Don Willems, Ralph Niels, Marcel van Gerven, and Louis Vuurpijl. 2009. Iconic and Multi-stroke Gesture Recognition. *Pattern Recognition* 42, 12 (2009), 3303–3312. DOI:http://dx.doi.org/10.1016/j.patcog.2009.01.030

57. Andrew D. Wilson and Aaron F. Bobick. 1999. Parametric Hidden Markov Models for Gesture Recognition. *IEEE TPAMI* 21, 9 (1999), 884–900. DOI: http://dx.doi.org/10.1109/34.790429

58. Jacob O. Wobbrock, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon Froehlich. 2011. Ability-Based Design: Concept, Principles and Examples. *ACM Trans. Access. Comput.* 3, 3, Article 9 (2011), 27 pages. DOI: http://dx.doi.org/10.1145/1952383.1952384

59. Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures Without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes. In *Proc. of the 20th Annual ACM Symp. on User Interface Software and Technology (UIST '07)*. ACM, 159–168. DOI: http://dx.doi.org/10.1145/1294211.1294238