EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling
Ergun Akleman (Editor)

EXPRESSIVE 2015

Real-Time Activity Prediction: A Gaze-Based Approach for Early Recognition of Pen-Based Interaction Tasks

Çağla Çığ[†1] and Tevfik Metin Sezgin[‡1]

[1]Koç University, Istanbul, Turkey

Abstract

Recently there has been a growing interest in sketch recognition technologies for facilitating human-computer interaction. Existing sketch recognition studies mainly focus on recognizing pre-defined symbols and gestures. However, just as there is a need for systems that can automatically recognize symbols and gestures, there is also a pressing need for systems that can automatically recognize pen-based manipulation activities (e.g. dragging, maximizing, minimizing, scrolling). There are two main challenges in classifying manipulation activities. First is the inherent lack of characteristic visual appearances of pen inputs that correspond to manipulation activities. Second is the necessity of real-time classification based upon the principle that users must receive immediate and appropriate visual feedback about the effects of their actions. In this paper (1) an existing activity prediction system for pen-based devices is modified for real-time activity prediction and (2) an alternative time-based activity prediction system is introduced. Both systems use eye gaze movements that naturally accompany pen-based user interaction for activity classification. The results of our comprehensive experiments demonstrate that the newly developed alternative system is a more successful candidate (in terms of prediction accuracy and early prediction speed) than the existing system for real-time activity prediction. More specifically, midway through an activity, the alternative system reaches 66% of its maximum accuracy value (i.e. 66% of 70.34%) whereas the existing system reaches only 36% of its maximum accuracy value (i.e. 36% of 55.69%).

Categories and Subject Descriptors (according to ACM CCS): H.1.2 [Models and Principles]: User/Machine Systems—Human information processing H.5.2 [Information Interfaces and Presentation (e.g., HCI)]: User Interfaces—Input devices and strategies (e.g., mouse, touchscreen)

Keywords: eager activity recognition, sketch recognition, proactive interfaces, multimodal interaction, sketch-based interaction, gaze-based interaction, feature extraction

1. Introduction

Typical pen-based interaction consists of *stylized* and *non-stylized* pen inputs. Stylized pen inputs correspond to pre-defined symbols and gestures. They have characteristic visual appearances, hence they can be classified with conventional image-based recognition algorithms (Figure 1a). On the other hand, non-stylized pen inputs correspond to pen inputs that lack a characteristic visual appearance. Accordingly, for non-stylized pen inputs, appearance alone does not carry sufficient information for classification purposes. For example, the pen trajectories for pen-based manipulation activities such as dragging, resizing, and scrolling have roughly the same shape (Figure 1b).

There are many approaches in the literature for classifying stylized pen inputs such as symbols and gestures [Rub91, KS04, HD05]. However, just as there is a need for systems that can automatically recognize symbols and gestures, there is also a pressing need for systems that can automatically recognize pen-based manipulation activities that yield non-stylized pen input. Automatic recognition of manipulation activities is desirable since it potentially eliminates the need for unnatural and imposed behaviors that must accompany these activities. For example, when a user wants to *drag* or *resize* an object, s/he must explicitly communicate which ac-

† e-mail: ccig@ku.edu.tr
‡ e-mail: mtsezgin@ku.edu.tr

DOI: 10.2312/exp.20151179

tivity s/he intends to perform via selecting the intended activity from a context menu or via carefully locating predefined regions dictated by the user interface (such as a four-headed arrow in the middle of an object for *dragging*, or a double-headed arrow around the edges of an object for *resizing*). These auxiliary behaviors are repeatedly and somehow habitually performed by users during daily pen-based interaction, but in fact go against the philosophy of pen-based interfaces as a more intuitive interaction alternative.

(a) NicIcon iconic pen gestures [NWV08] serve as examples of stylized pen input. Each gesture has an easily distinguishable characteristic visual appearance.

(b) Pen trajectories for virtual interaction tasks [ÇS15] serve as examples of non-stylized pen input that do not have characteristic visual appearances and do not lend themselves well to conventional image-based recognition algorithms.

Figure 1: *Stylized* and *non-stylized* pen inputs.

The task of classifying manipulation activities is of a more challenging nature both due to the inherent lack of characteristic visual appearances, and more importantly due to the necessity of real-time classification. Manipulation activities must be recognized in real-time in order for the pen-based interface to actively detect and switch to the currently intended mode of manipulation and provide immediate and appropriate visual feedback about the effects of user's actions [Nor02]. For instance, when the user places the stylus pointer on an object and starts dragging the object, s/he must be able to see the change in the object's position in real-time.

Eye tracking technology has greatly improved in the last few years, and it is now possible to embed gaze detection functionalities into portable devices such as tablets and smart phones [Ble13]. We propose to use eye gaze movements that naturally accompany pen-based user interaction for real-time classification of non-stylized pen inputs. To illustrate our approach, we have adapted an existing gaze-based activity prediction system [ÇS15] to the needs of real-time activity prediction. In the rest of the paper, this system will be referred to as the **static system**. Furthermore, we have developed an alternative time-based **dynamic system** specifically tailored for real-time activity prediction. We comparatively evaluate these two systems with respect to prediction accuracy and early prediction speed. Our evaluation is focused on a number of frequently employed pen-based interaction tasks. These tasks are: *drag*, *maximize*, *minimize*, *scroll*, and *free-form drawing*. Our results show that the dynamic approach that we propose based on Hidden Markov Models (HMMs) is more suitable than a static approach based on Dynamic Time Warping (DTW) for real-time gaze-based activity prediction in pen-based devices.

2. Related work

We have presented a gaze-based real-time activity prediction system for pen-based devices. State-of-the-art related work falls under two main categories: *real-time sketch recognizers* and *gaze-based activity predictors*. In summary, existing real-time sketch recognizers only work on stylized pen inputs and existing gaze-based activity predictors are able to detect the performed activity only after the activity ends.

There are many approaches in the literature for classifying stylized pen inputs such as symbols and gestures [Rub91, KS04, HD05]. All these approaches focus on classifying fully completed sketches. A more challenging task is auto-completion, i.e. classifying sketches in real-time before they are fully completed. Auto-completion of stylized pen inputs has also been tackled to some extent. Prominent examples deal with recognizing primitive geometric shapes (e.g. circles and squares) [AN00], complex Chinese characters [LMS08], Course of Action Diagram symbols [TYS12], and multi-touch gestures [SW14] before the drawings are fully completed. We focus on the even more challenging task of classifying partially completed non-stylized pen inputs.

One active line of research on gaze-based interaction aims to predict user activities during interaction with computerized systems. Prominent examples deal with predicting office activities [BRT11], Google Analytics tasks [CAD*11], graph-based information visualization tasks [SCC13], and pen-based virtual interaction tasks [ÇS15]. All of the existing studies, however, are able to detect the performed activity only after the activity ends. Therefore, it is not possible to employ these systems in real-time proactive user interfaces.

3. After-the-fact activity prediction

In this section, the existing static system [ÇS15] and the newly developed alternative dynamic system are described. We compare the two methods with respect to after-the-fact (as opposed to early) activity prediction accuracy. For all experiments, we use the multimodal database detailed in [ÇS15]. This database consists of sketch and gaze data collected for 5 different activities (*drag*, *maximize*, *minimize*, *scroll*, and *free-form drawing*) from 10 participants (6 males, 4 females) over 10 randomized repeats across 3 scales. The scale variable determines the length of the desired pen motion and was set to 21 cm, 10.5 cm, and 5.25 cm for the *large*, *medium*, and *small* scales, respectively. The *free-form drawing* activity differs from the remaining activities in a special way. If our prediction system is to be employed in a proactive user interface, the ability to distinguish between

the intention to sketch and the intention to interact becomes vital. Accordingly, the *free-form drawing* activity is included in our study to avoid unsolicited task activation. For collecting this database of synchronized sketch and gaze data, the authors used a tablet and a Tobii X120 stand-alone eye tracker for the sketch and gaze modalities, respectively. Tobii X120 operates with a data rate of 120 Hz, tracking accuracy of 0.5°, and drift of less than 0.3°. The tracker allows free head movement inside a virtual box with dimensions 30×22×30 cm.

The existing static system utilizes three kinds of features for gaze-based task prediction: (1) evolution of instantaneous sketch-gaze distance over time, (2) spatial distribution of gaze points collected throughout an activity, and (3) IDM visual sketch features [OD09]. Among these kinds of features, only the first one takes time element into consideration. For that reason, when designing the alternative dynamic system, we primarily focused on different approaches for computing this feature. For all experiments, we report the mean prediction accuracy obtained via 5-fold cross validation.

3.1. Static system

In the static system, the authors use a time-series signal to represent the time-wise evolution of the instantaneous distance between pen tip and gaze direction over time. Initially, they compute one or multiple characteristic signals per activity (Figure 2). When it comes to determining which activity a new signal belongs to, they measure the similarity of the new signal to each of the characteristic signals and use an SVM model previously trained with these similarity values to determine the label of the new signal (Figure 3). The authors use an open-source DTW library detailed in [ÇS15] for computing the similarity of two given signals.

3.2. Dynamic system

Similarly in the dynamic system, we use a time-series signal to represent the time-wise evolution of the instantaneous distance between pen tip and gaze direction. We observe a rise in this signal when the sketch-gaze distance increases, a fall when the sketch-gaze distance decreases, and no change when the sketch-gaze distance is constant over a period of time. Based on this observation, we train an HMM for each activity. Using HMMs gives us the ability to learn compact models of how hand-eye coordination behaviors change over the course of an activity and allows us to obtain a likelihood value from each HMM for classifying a given sketch-gaze distance signal.

When training the HMMs, we assume that (1) there are 3 different states as rising, falling, and steady and (2) the observations come from a Gaussian Mixture Model (Figure 4). When it comes to determining which activity a new signal belongs to, we initially apply a simple preprocessing step to the signal. During this preprocessing step, the original signal is first differentiated and then downsampled to decrease the noise in the original signal and highlight state changes.

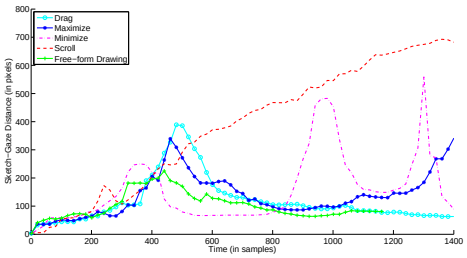

Figure 2: Characteristic signals obtained from sketch-gaze distance signals of each activity.

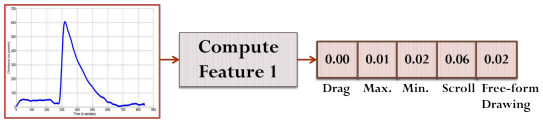

Figure 3: Extraction of the sketch-gaze distance feature in the static system. For a given signal, its similarity to each of the characteristic signals is measured and the degree of matching is used as an informative feature for classifying activities.

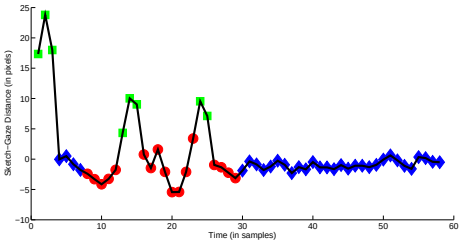

Figure 4: Visualization of the HMM state assignments for the data points of a sample preprocessed signal. The data points are assumed to come from a normal distribution. The active state is represented with green squares when the sketch-gaze distance increases; red circles when it decreases; and blue diamonds when it is constant over a period of time.

Figure 5: Extraction of the sketch-gaze distance feature in the dynamic system. For a given signal, its log probability of being generated by each of the HMMs is calculated and the degree of likelihood is used an informative feature for classifying activities.

Afterwards, we calculate the likelihood of the preprocessed signal being generated by each of the HMMs and use an SVM model previously trained with these likelihood values to determine the label of the new signal (Figure 5). We use an open-source HMM library [DM10] for all HMM-related calculations.

3.3. Experiment results

We conducted a one-way ANOVA to examine the effect of system type on prediction accuracy across the *static system* and *dynamic system* conditions. There was no significant effect of system type on prediction accuracy at the $p < 0.05$ level for all *large* ($p = 0.304$), *medium* ($p = 0.266$), and *small* ($p = 0.536$) scales (Figure 6). Nevertheless, the newly developed alternative dynamic system (83.77 ± 5.13) was found to be better on average than the existing static system (82.14 ± 3.82) in terms of activity prediction accuracy (although firm conclusions cannot be reached due to the limited amount of data available).

To examine the effect of system type on prediction accuracy when only the sketch-gaze distance feature is used, we conducted a one-way ANOVA. The sketch-gaze distance feature is important for real-time activity prediction since it is the only one that takes time element into consideration and attempts to capture the dynamic aspects of human hand-eye coordination behavior. The dynamic system (73.53 ± 1.13) was found to be significantly better than the static system (70.88 ± 2.18) in terms of capturing the sketch-gaze distance feature [$F(1,8) = 5.832, p = 0.042$]. ANOVA results hint that the newly developed alternative dynamic system may be a better candidate than the existing static system for real-time activity prediction (Figure 7).

4. Real-time activity prediction

Existing gaze-based activity prediction systems are able to detect the performed activity only after the activity ends. However, in line with the feedback principle of design [Nor02], users must be informed in real-time about the effects of their actions via immediate and appropriate visual feedback. For instance, when the user places the stylus pointer on an object and starts dragging the object, s/he must be able to see the change in the object's position in real-time. If we take this one step further, early prediction speed becomes even more important in a proactive user interface (that actively monitors the user, and switches to the intended mode of interaction on behalf of the user). If we continue with the same example, when the user places the stylus pointer on an object and starts moving the cursor away from the object, the proactive interface actively detects that the user wants to *drag* the object, and saves the user time and energy by automatically switching to the *drag* mode of interaction.

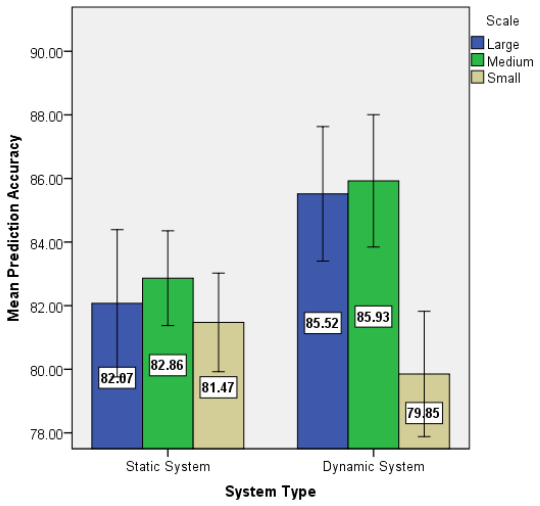

Figure 6: Mean prediction accuracy values obtained for each system type and scale using all three kinds of features (sketch-gaze distance, spatial distribution, IDM). Models trained with different kinds of features are combined via classifier-level fusion. Error bars indicate ± 1 standard error.

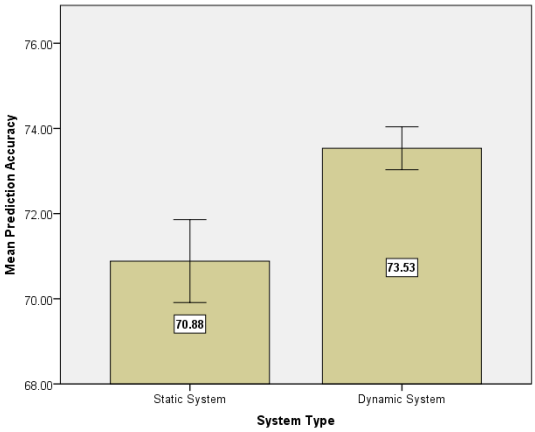

Figure 7: Mean prediction accuracy values obtained for each system type irrespective of scale using only the sketch-gaze distance feature. Error bars indicate ± 1 standard error.

4.1. Baseline

Initially, we analyze the real-time activity prediction performance of the existing static and dynamic systems without any specialized training for real-time prediction, hence the title *naive approach*. The fundamental difference between the static and dynamic systems lies in the approach each system adopts for computing the sketch-gaze distance feature. The remaining two kinds of features are computed identically for the two systems and models trained with different kinds of features are combined via classifier-level fusion. For

this reason, real-time activity prediction performance of the two systems is measured only on the basis of the sketch-gaze distance feature.

For the experiments reported in this section, we first generate 10 different test signals from each individual test signal. These 10 signals respectively correspond to the first 10%, 20%, ..., 100% of the original test signal. The sub-signals created from the set of all test signals are then fed to the real-time prediction systems as test data. Mean prediction accuracy at the start of an activity is assumed to be 20%, i.e. the random baseline accuracy for recognizing 5 different activities. The experiment is repeated for the *large*, *medium*, and *small* scales, as well as for the *all scales* case, which corresponds to the entire database. Experiment results show that the dynamic system is able to accurately predict the currently performed activity earlier than the static system (Figure 8). For instance, if we consider the entire database, at the point when only 50% of the data is observable, the dynamic system reaches 60% of its maximum accuracy value (i.e. 60% of 74.14%) whereas the static system reaches only 43% of its maximum accuracy value (i.e. 43% of 70.88%).

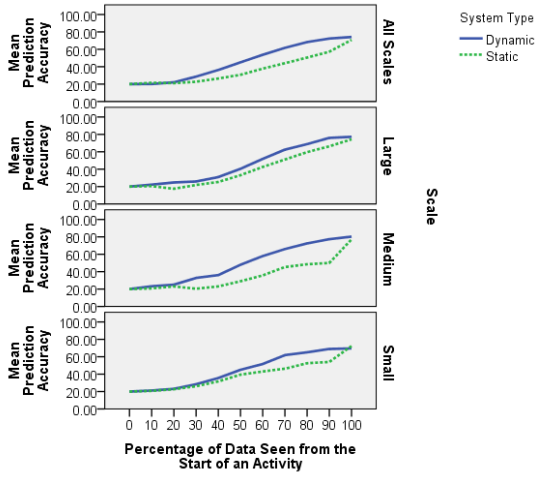

Figure 8: Mean real-time prediction accuracy values obtained for each system type and scale as a function of percentage of data seen from the start of the activity. For these experiments, existing static and dynamic systems are used without any specialized training for real-time prediction.

4.2. Specialized training for real-time activity prediction

There exists some problems associated with the naive approach. First of all, in the naive approach, real-time activity prediction systems are not purposely trained for real-time activity prediction. For this, in line with the test scenario of the naive approach, we generate 10 different signals from each signal used in training the prediction models. The sub-signals created from the set of all training signals are then separated into 5 different groups as follows (note that a typical signal lasts about 2 seconds):

- First group consists of sub-signals that last shorter than 500 milliseconds ($0 \leq duration \leq 500$),
- Second group consists of sub-signals that last shorter than 1000 milliseconds ($500 < duration \leq 1000$),
- Third group consists of sub-signals that last shorter than 1500 milliseconds ($1000 < duration \leq 1500$),
- Fourth group consists of sub-signals that last shorter than 2000 milliseconds ($1500 < duration \leq 2000$), and
- Fifth group consists of sub-signals that last longer than 2000 milliseconds ($duration > 2000$).

After the groups are formed, we train a separate SVM model for each group using the sub-signals comprising each group. Accordingly, the first model captures the characteristics of signals that last shorter than 500 milliseconds while the second model captures the characteristics of signals that last longer than 500 and shorter than 1000 milliseconds.

Second, response time of a pen-based user interface utilizing either of the real-time activity prediction systems will inevitably be affected by the computations necessary for inferring the activity. Hence, when calculating the early prediction speed of a system, we must take into account the computational complexity of the algorithm used for activity prediction. In order to determine the label of a given signal, the static system measures the similarity of the given signal to each of the characteristic signals using the DTW algorithm. This process takes an average of 1.125 seconds for a single signal. On the other hand, the dynamic system initially applies a simple preprocessing step to the given signal and then calculates the likelihood of the preprocessed signal being generated by each of the HMMs to determine the label of a given signal. This process takes an average of 0.0064 seconds for a single signal. According to these computational time measurements, the static system is not able to give any feedback for the first 1.125 seconds of an activity while the dynamic is not able to give any feedback for the first 0.0064 seconds.

And finally, in a real user interface, there is no way of knowing the percentage of activity completed by the user at a random point during an activity; one can only know the amount of time passed from the start of an activity. For this reason, the experiments should measure how real-time prediction accuracy values change over time instead of over percentage of data seen.

In consideration of the factors listed above, we repeated the experiments and conducted a two-way ANOVA to examine the effect of system type and elapsed time on real-time prediction accuracy. ANOVA revealed (1) a main effect of system type on prediction accuracy [$F(1,36) = 107.067, p = 0.000$], (2) a main effect of elapsed time on prediction accuracy [$F(5,36) = 93.634, p = 0.000$], and (3) a significant interaction between system type and elapsed time [$F(5,36) = 6.333, p = 0.000$]. Experiment results again show that the

dynamic system is able to accurately predict the currently performed activity earlier than the static system (Figure 9). More specifically, if we consider the entire database, midway through an activity, the dynamic system reaches 66% of its maximum accuracy value (i.e. 66% of 70.34%) whereas the static system reaches only 36% of its maximum accuracy value (i.e. 36% of 55.69%).

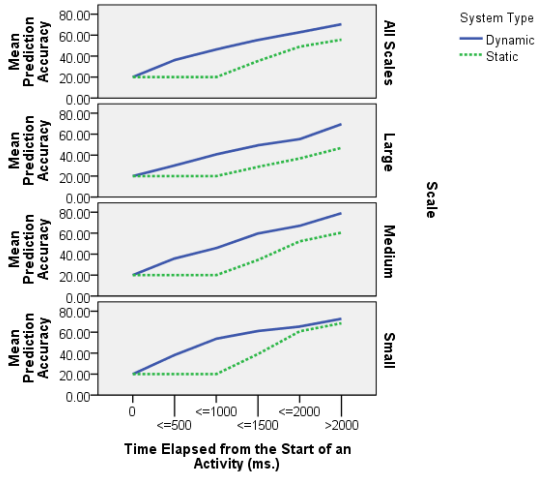

Figure 9: Mean real-time prediction accuracy values obtained for each system type and scale as a function of time elapsed from the start of the activity. For these experiments, purpose-trained static and dynamic systems are used.

5. Conclusions and future work

In this paper, we have presented our work on gaze-based real-time activity prediction in pen-based devices. We have proposed a dynamic approach based on Hidden Markov Models (HMMs) and compared it with an existing static approach based on Dynamic Time Warping (DTW). Through a set of carefully designed experiments and accompanying comprehensive statistical analysis, we have demonstrated that the dynamic approach is a more successful candidate (in terms of prediction accuracy and early prediction speed) than the static approach for real-time activity prediction. We believe that our novel activity prediction system will open the way for unprecedented gaze-based proactive user interfaces for pen-based devices.

On the basis of the promising findings presented in this paper, our ongoing work aims to develop an improved real-time activity prediction system based on Dynamic Bayesian Networks (DBNs). The fundamental difference between the HMM- and DBN-based systems will lie in our ability to explicitly model high-level processes that occur during human hand-eye coordination behavior (e.g. gazing at the object to be manipulated, gazing at the intended final position of the object). Another substantial extension might explore the feasibility of using our real-time activity prediction system to build a proactive user interface. When the user performs a pen action (demarcated by a pen-down and a pen-up event), the planned proactive user interface will actively detect and switch to the currently intended mode of interaction based on user's synchronized pen trajectory and eye gaze information during pen-based interaction. Intention predictions will be carried out by the previously trained HMM-based model and the features extracted from the corresponding sketch-gaze data of the user.

Acknowledgements

Authors gratefully acknowledge the grants from TÜBİTAK (Grant No: 110E175 and Grant No: 113E325).

References

[AN00] ARVO J., NOVINS K.: Fluid sketches: continuous recognition and morphing of simple hand-drawn shapes. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology* (2000), pp. 73–80. doi:10.1145/354401.354413. 2

[Ble13] BLEICHER A.: Eye-tracking software goes mobile. Website, 2013. http://spectrum.ieee.org/computing/software/eyetracking-software-goes-mobile/. 2

[BRT11] BULLING A., ROGGEN D., TRÖSTER G.: What's in the eyes for context-awareness? *IEEE Pervasive Computing 10*, 2 (2011), 48–57. doi:10.1109/MPRV.2010.49. 2

[CAD*11] COURTEMANCHE F., AÏMEUR E., DUFRESNE A., NAJJAR M., MPONDO F.: Activity recognition using eye-gaze movements and traditional interactions. *Interacting with Computers 23*, 3 (2011), 202–213. doi:10.1016/j.intcom.2011.02.008. 2

[ÇS15] ÇIĞ Ç., SEZGIN T. M.: Gaze-based prediction of pen-based virtual interaction tasks. *International Journal of Human-Computer Studies 73* (2015), 91–106. doi:10.1016/j.ijhcs.2014.09.005. 2, 3

[DM10] DUNHAM M., MURPHY K.: Probabilistic modeling toolkit for matlab/octave, version 3. Website, 2010. https://github.com/probml/pmtk3/. 4

[HD05] HAMMOND T., DAVIS R.: Ladder, a sketching language for user interface developers. *Computers & Graphics 29*, 4 (2005), 518–532. doi:10.1016/j.cag.2005.05.005. 1, 2

[KS04] KARA L. B., STAHOVICH T. F.: Hierarchical parsing and recognition of hand-sketched diagrams. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (2004), pp. 13–22. doi:10.1145/1029632.1029636. 1, 2

[LMS08] LIU P., MA L., SOONG F. K.: Prefix tree based auto-completion for convenient bi-modal chinese character input. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (2008), pp. 4465–4468. doi:10.1109/ICASSP.2008.4518647. 2

[Nor02] NORMAN D. A.: *The design of everyday things*. Basic Books, 2002. 2, 4

[NWV08] NIELS R. M. J., WILLEMS D. J. M., VUURPIJL L. G.: The nicicon database of handwritten icons. In *Proceedings of the 11th International Conference on the Frontiers of Handwriting Recognition* (2008), pp. 296–301. 2

[OD09] OUYANG T. Y., DAVIS R.: A visual approach to sketched symbol recognition. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence* (2009), pp. 1463–1468. 3

[Rub91] RUBINE D.: Specifying gestures by example. *SIGGRAPH Computer Graphics 25*, 4 (1991), 329–337. doi:10.1145/122718.122753. 1, 2

[SCC13] STEICHEN B., CARENINI G., CONATI C.: User-adaptive information visualization: using eye gaze data to infer visualization tasks and user cognitive abilities. In *Proceedings of the 18th International Conference on Intelligent User Interfaces* (2013), pp. 317–328. doi:10.1145/2449396.2449439. 2

[SW14] SCHMIDT M., WEBER G.: Prediction of multi-touch gestures during input. In *Human-Computer Interaction. Advanced Interaction Modalities and Techniques*, vol. 8511 of *Lecture Notes in Computer Science*. Springer International Publishing, 2014, pp. 158–169. doi:10.1007/978-3-319-07230-2_16. 2

[TYS12] TIRKAZ Ç., YANIKOĞLU B., SEZGIN T. M.: Sketched symbol recognition with auto-completion. *Pattern Recognition 45*, 11 (2012), 3926–3937. doi:10.1016/j.patcog.2012.04.026. 2