

BopoNoto: An Intelligent Sketch Education Application for Learning Zhuyin Phonetic Script

Paul Tael and Tracy Hammond

Sketch Recognition Lab
Texas A&M University
College Station, TX 77843 USA
{ptael,hammond}@cse.tamu.edu

Abstract— The zhuyin phonetic script not only provides greater learning benefits compared to romanization systems for Chinese Mandarin language students with existing English fluency, but also allows students to take advantage of its use in Taiwan where it exists in both official and popular usage. However, while pen-enabled computing educational applications can assist traditional pedagogical approaches for accelerating mastery of the script, existing approaches are either constrained in providing writing assessment, catered to native language users, or are less flexible in recognizing more natural writing. We present BopoNoto, an intelligent sketch education application for language students to learn zhuyin. Our application provides a sketching interface for practicing the symbols, and a sketch recognition system for assessing the visual and technical correctness of their input. From our evaluations, BopoNoto successfully demonstrates strong results in understanding and assessing students’ written input.

Keywords— sketch recognition; intelligent user interface; intelligent tutoring system; Chinese Mandarin; zhuyin; bopomofo

I. INTRODUCTION

The written Chinese language differs from its writing counterparts in western languages such as English in that Chinese consists of characters that are inherently non-phonetic [2,33], and so mastering the pronunciation of characters in written Chinese is highly challenging for language students with primarily English language fluency [33]. There however exists different phonetic systems for written Chinese that map the non-phonetic Chinese characters to their equivalent pronunciations, including the popular hanyu pinyin that is a romanization system conventionally taught in American Chinese language classes due to students’ familiar with alphabet letters already in English [12]. However, hanyu pinyin has its own disadvantages that can be challenging for native English users learning Chinese Mandarin, such as hanyu pinyin’s orthographic rules and pronunciations not allowed or existing, respectively, in English [10].

One alternative to romanized phonetic scripts such as hanyu pinyin is the zhuyin phonetic system (Figure 1), known more formally as zhuyin fuhao, colloquially as bopomofo, and officially as Mandarin Phonetic Symbols I (MPS1). Like other native non-romanized East Asian language phonetic scripts such as hiragana and katakana in Japanese and hangul in Korean [2,16], zhuyin was designed specifically to map directly and uniquely to the sounds in Chinese Mandarin [6,11]. Furthermore, zhuyin is officially and commonly used by people in Taiwan [1,24], and is continually taught to both domestic

elementary students [17] and foreign university students [21] alike, so students familiar with the script – especially those living and studying already in Taiwan – can also take advantage of being able to understand them from their surroundings and with locals there.

B: ㄅ	P: ㄆ	M: ㄇ	F: ㄈ	D: ㄉ
T: ㄊ	N: ㄋ	L: ㄌ	G: ㄍ	K: ㄎ
H: ㄏ	J: ㄐ	Q: ㄑ	X: ㄒ	ZH: ㄓ
CH: ㄔ	SH: ㄕ	R: ㄖ	Z: ㄗ	C: ㄘ
S: ㄙ	A: ㄚ	O: ㄛ	E: ㄜ	EH: ㄝ
AI: ㄞ	EI: ㄟ	AU: ㄠ	OU: ㄡ	AN: ㄢ
EN: ㄣ	ANG: ㄤ	ENG: ㄥ	ER: ㄦ	Y: ㄩ
	U: ㄨ		YU: ㄩ	

Figure 1. The thirty-seven symbols that represent the phonetic system of zhuyin (i.e., zhuyin fuhao, bopomofo, Mandarin Phonetic Symbols I).

While zhuyin provides several important advantages to such language students with primarily existing English fluency, mastering the script involves an initially steeper learning curve, a relatively larger set of symbols, completely unfamiliar symbols, and more diverse visual complexity compared to the English letters found in its romanized phonetic script counterparts [1], where the last two features stem from the symbols’ historic origins in Chinese characters [24]. Traditional pedagogical approaches for learning zhuyin thus has parallels with related non-phonetic scripts in East Asian languages, which rely on instructors and course materials introducing rote memorization and written techniques of stroke count, order, and technique (e.g., [20]).

The importance of writing technique practice for written East Asian scripts such as zhuyin includes enabling students to improve the aesthetics of their writing [13], but unsupervised writing practice outside of the classroom – using traditional pedagogical approaches such as rote memorization – may hinder students in memorizing the symbols effectively if they unintentionally develop bad writing habits [19]. Computer-assisted educational tools have strong potential in aiding instructors in addressing such concerns by providing emulated supervision outside of the classroom when instructors are less accessible outside of class, but available educational apps

provide minimal or lacking writing feedback, relevant language handwriting interfaces focus more on best interpreting what the user wrote, and related current sketch research systems require that students write their input in a specific way to receive accurate assessment.

In this paper, we therefore introduce an intelligent sketch application called BopoNoto, which is a portmanteau of the script name ‘bopomofo’ and a common East Asian pronunciation of the word ‘note’. Our application enables students to practice learning the zhuyin phonetic script through direct writing practice and accompanying automated feedback on the visual and technical correctness of their written symbols. Our application provides students with different writing interfaces to practice their knowledge of individual symbols and the correct phonetics mapping of prompted Chinese characters, and allows them to automatically receive and review mistakes in their symbol writing.

II. RELATED WORK

As an intelligent sketch application for the zhuyin phonetic script, we discuss related work from language education apps for learning zhuyin and other related written East Asian language scripts, handwriting input systems for inputting written East Asian languages, and sketch recognition interfaces for educational domains including written East Asian languages.

A. Language Education Apps

With the growing ubiquity of mobile devices, developers have tapped into the technology to create digital educational interfaces including learning zhuyin (e.g., [4]). However, many of these accessible mobile app interfaces are constrained to traditional flash card and binary feedback learning, and also lack direct writing input and assessment. Other interfaces for learning symbols in other written East Asian language scripts have taken more unique approaches in creating engaging interactive group learning experiences with mobile devices [9,25,30], but do not address individual learning practice that incorporates a writing practice modality with automated assessment. More recent educational mobile apps (e.g., [23]) have moved beyond the limitations of prior flash card-based apps to include writing with assessment, but are still constrained to tracing practice and binary feedback.

B. Handwriting Input Systems

Peripherally related to educational writing interfaces for East Asian languages such as zhuyin are intelligent handwriting input systems that enable users to naturally write out the symbols on touch- and stylus-enabled computing devices. Advances from the machine learning research community in recognizing written East Asian language symbols have been explored for decades [15], and the lessons from these research efforts have led to the development of robust handwriting input systems for written East Asian language symbols that anticipated the rising popularity of touchscreens. These works include Synaptics [18] for early-generation stylus-driven touchscreens, Microsoft [22] for tablet notebooks, and Google [7] for smartphones. However, the focus of these handwriting input systems involves best interpreting the intentions of users’ written input such as messy and incomplete input. Such systems therefore cater to input from users with existing fluency of written East Asian symbols such

as zhuyin, and may not be appropriate to novice students who have not yet properly established good writing habits.

C. Sketch Recognition Interfaces

Two particularly related systems include Hashigo [27], an intelligent sketch education interface for helping students practice Japanese kanji; and LAMPS [28], an analogous interface for Chinese zhuyin. The strengths of these systems are that students can write on these interfaces and immediately receive feedback on the visual and technical correctness of their input. However, these systems primarily rely on geometric sketch recognition approaches, and thus perform less optimally for students’ written input that are considered still visually correct but cannot be cleanly segmented into smaller geometric components for recognition and assessment. In the case of LAMPS for zhuyin symbols, the authors highlight limitations in recognizing symbols that cannot be geometrically approximated by straight polylines, such as those containing arcs, curves, and dots.

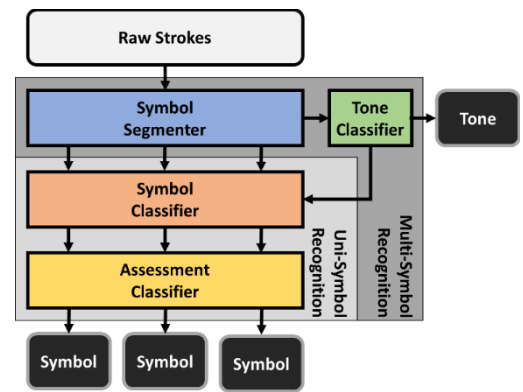


Figure 2. An overview of BopoNoto’s recognition system for classifying both individual and multiple zhuyin phonetic symbols.

III. RECOGNITION SYSTEM

Designing the intelligent sketching interface for the BopoNoto application to robustly recognize students’ zhuyin phonetic symbols that are appropriate for classroom submission motivated us to develop a recognition system consisting of an optimized collection of gesture and sketch recognition techniques and heuristics, as shown by the system overview in Figure 2.

A. Symbol Segmentation

The first layer of BopoNoto’s recognition system is symbol segmentation (Figure 3), which first takes the raw strokes from the students’ collection of one or more written symbols and then segments them into their individual symbols. The design of our symbol segmenter takes lessons from both the drawing behavior cues from cognitive theory research [32] and Asian calligraphy [29] for the symbol segmentation process, which involves windowing the strokes left to right and assuming an approximate square block boundary for candidate symbols to segment.

Based on these assumptions from the literature, our algorithm first determines the candidate symbol’s potential boundary size by calculating the length threshold of either the height of the entire raw strokes’ bounding box height or the width of the largest raw stroke. We desired the latter to handle

cases when the input consists of symbols with very short heights. We next iterate through and temporarily store each raw stroke from left to right based on the order that the user wrote them, and then calculate the bounding box width of all the iteratively stored strokes. If this width exceeds 125% of the length threshold, we then segment the collected strokes as a candidate symbol, reset the stored strokes, and shift the process to the next unadded strokes. Otherwise, the process continues for the next stroke until all symbols are segmented into a list of symbols for the next layer of the recognition system.

One special case involves symbols where a stroke is added to the bottom-left of the written symbol, which causes the segmenter to possibly prematurely segment the symbols in the middle of the collected strokes, since the width may exceed the boundary width from the left side of the candidate symbol. In order to alleviate this special case, we automatically add all strokes that are left of the previous added stroke, since we assume that such strokes will not form a distinct symbol due to drawing assumptions.

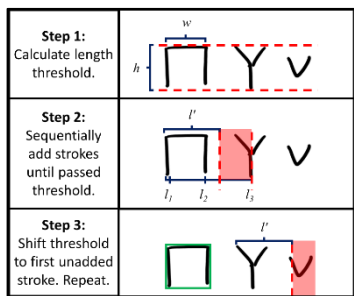


Figure 3. The general three-step process for segmenting raw strokes into a list of zhuyin symbols.

B. Tone Symbol Recognition

Immediately following the symbol segmenter, we then take the last symbol from the collected list and test whether the symbol is instead one of the four explicit tone marks found in the zhuyin phonetic system to represent tonal sounds in Chinese Mandarin (Figure 4).

Since the four explicit zhuyin tone marks visually consist of either segmented lines or dots, we apply geometric shape tests from relevant sketch recognition algorithms. For the straight lines that form the rising second tone and the falling fourth tone, we utilize line tests from ShortStraw [35]. The process first involves removing potential garbage points from the stroke that occurs from users pressing down and lifting up their finger or stylus from the touch surface (i.e., 5% from the beginning and end of the stroke). Next, we calculate the distance of the ideal line formed from the endpoints and the path length of the stroke, then take the ratio of the lesser value to the greater value. If this ratio exceeds 0.8, we then measure the angle formed from the stroke’s endpoints. If the angle is within a threshold angle range of 45% or 120%, we classify the stroke as either a rising second or a falling fourth tone, respectively.

Our classifier then proceeds to test the stroke as a low third tone, which visually consists of a line that is initially angled down and then angled back up. One potential solution is to utilize a number of corner segmentation algorithms to detect the three corners (i.e., bottom left, top, and bottom right) in the

stroke. However, we alternatively applied geometric heuristic tests on the stroke to avoid the overhead associated with incorporating a separate corner segmenter by simply trimming the garbage points, locating the furthest left, top, and right points, and then applying similar line tests on these substrokes for angles of 120% and 60%, respectively. If the stroke passes these conditions, we classify it as a low third tone.

Finally, we tested the stroke for its potential as a neutral fifth tone – which is visually represented as a dot – by using insights from the Dahmen scribble recognizer [5]. Specifically, we first calculate the width and height of the stroke’s bounding box, then calculate the path length of the stroke, and lastly calculate the density of the stroke by taking the quotient of the stroke path length and stroke bounding box area. If this density ratio exceeds 0.3, the stroke is then classified as a neutral fifth tone. Otherwise, the stroke remains unclassified and the process proceeds to the next layer of the recognition system.



Figure 4. The four tone marks in the zhuyin phonetic script (L-R): the rising second tone, the low third tone, the falling fourth tone, and the neutral fifth tone. The high first tone is implicitly implied if the phonetic symbols lacks a tone mark in zhuyin.

C. Symbol Classification

The crux of the recognition system lies in the individual zhuyin symbols, where segmented symbols from the symbol segmenter and possibly the tone classifier subsequently proceed through the symbol classifier. The aim of the symbol classification for our BopoNoto application is to ensure that it can classify students’ symbols that are visually similar to model written symbols. As a result, we first recruited a university Chinese Mandarin language instructor from Taiwan with several years of classroom experience to provide us with model written symbols. The data collection of these model written symbols involved prompting the instructor to write five iterations of each zhuyin symbol for a total of $5 \times 185 = 377$ writing samples.

From the language instructor’s model written symbols, we group these sketches as training data, particularly as templates for the template matching process. Our reasoning is that template matching potentially enables us to more easily identify written input that is pedagogically visually correct instead of simply determining the input’s best interpretation found in other machine learning techniques that may not penalize for students’ sloppy writing. We therefore initially experimented with various naive and modified Euclidean and Hausdorff distance-based template matching algorithms on the individual symbols for their classification feasibility, and empirically observed that existing approaches in the literature that were successful for smaller sets of symbols and were appropriate as touchscreen gestures had performed less optimally for a larger zhuyin phonetic symbol set of nearly forty symbols and of relatively greater visual complexity.

We eventually developed a two-part template matching algorithm designed to perform well in classifying zhuyin symbols by relying on two different metrics: a specialized

Hausdorff distance-based score and a stroke points coverage ratio (Figure 5).

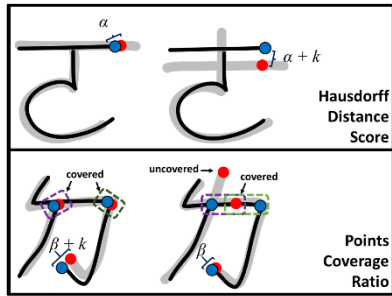


Figure 5. The individual symbol classification process that relies on both a Hausdorff distance score and a points coverage ratio.

1) Hausdorff Distance Score

The first metric is the Hausdorff distance score, and we calculate this by normalizing (i.e., resample, scale, and translate) the individual symbol’s raw strokes and classifying them using a specialized Hausdorff distance-based template matcher [26] derived from existing template matchers (e.g., [14]) and that was adopted in this paper for relatively more complex symbols such as those found in zhuyin. One such special case is for a zhuyin phonetic symbol consisting of a single horizontal line, which has issues for general template matching algorithms due to rescaling it into a square bounding box [34]. To compensate for this special case, we do a pre-test by first determining if the symbol is single-stroke, a line, and horizontal. If so, we automatically classify this stroke as that particular zhuyin phonetic symbol.

The next step involves the actual score calculation conventionally performed in Hausdorff distance template matching, which involves iterating through each resampled point in the written symbol, locating the closest corresponding point in the candidate model template, calculating the Euclidean distance, and then taking a running sum of that distance into the combined sum of distances. We then calculate the score from the equation below [26], which is a scaled variation of the score employed in [34] but for relatively more complex symbols.

2) Points Coverage Ratio

The second metric that we use in the symbol classification process specifically addresses issues of visually similar symbols, which occurs for a non-trivial number of zhuyin symbols. We initially observed that misclassification cases occurred for symbols that were visually similar due to the fact that the subset of points from model template of an incorrect symbol type overlaps more closely than the entire set of points from the closest model template of the correct symbol type. Due to this observation, we set forth to calculate the ratio of points from the model template that was paired as the shortest distance to the points in the student’s written input.

With the information from the second metric, we then go back to the sorted metric from the first metric and locate the model template with the highest coverage ratio in the top 10% score-ranked model templates. The zhuyin symbol type of the model template that satisfies the conditions from these two metrics is eventually used to classify the student’s written input.

D. Assessment Classification

Once the symbol is visually classified from the previous layer of the recognition system through the symbol classifier, we finally assess the technical correctness of the written symbol for visually correct symbols. This assessment consists of three technical correctness tests for the following: stroke count, stroke order, and stroke direction.

1) Stroke Count Test

The first test is in determining the correct number of strokes. To perform this, we simply count the number of strokes from the students’ written input to those from the model template. If there is a mismatch in the count, the written input is assessed as having an incorrect stroke count.

2) Stroke Order Test

For symbols that pass the stroke count test, we then perform a stroke order test. This process first involves pairing the strokes from the students’ written input to their equivalent strokes from the model template. In order to do so, we approximate optimal pairings of stroke pairs between the students written input and the model template by taking the start, middle, and end point of the written input and model template, and calculate the summed distances from all three pair of points. Whichever stroke’s trio of points from the model template has the shortest distance from that written input stroke is therefore paired to that model template’s stroke. The process continues until all strokes from the written input are paired to their equivalent strokes in the model template.

The next step then relies on sorting the pairs of strokes from between the written input and model template in ascending temporal order of the model templates strokes, since we assume that the model template will have the correct temporal ordering. From this sorting information, we also check to see if the written input’s strokes are also correctly sorted in ascending time. If not, we record the index of the first instance of the temporally misplaced stroke and assess the written input as having incorrect stroke order.

3) Stroke Direction Test

The last assessment check is in the correctness of the stroke direction. For cases where the written input is assessed as having correct stroke order, we rely on the list of temporally-ordered paired strokes by iteratively comparing the endpoints of the model template’s stroke with those of the student’s written input. We first try to find the corresponding endpoints of the model template to the written input by first determining whether the endpoints of the written input’s stroke approximates an ideal vertical line. If so, we then pair the topmost point of the written input’s stroke to the topmost point of the model template’s stroke and likewise for the bottommost point. Otherwise, we pair the corresponding endpoints by the leftmost and rightmost points.

Once the endpoints of the corresponding strokes from the written input and model template are mutually paired with each other, we then perform a Boolean check on the temporal ordering of the endpoints of the model template stroke and then on the written input. That is, we check if the top or left endpoint temporally occurs before the bottom or right endpoint, respectively, for the model template stroke, and likewise for the written input stroke. If the Boolean check is equivalent for both

the model template and the written input for that particular stroke, we then consider that stroke to have correct stroke direction. We iteratively apply this stroke direction test on each stroke of the written input, and classify the entire symbol as having correct stroke direction. Otherwise, we record the index of the first instance of the incorrect stroke direction and label the symbol as such.

IV. SKETCHING INTERFACE

We subsequently incorporate our recognition system that automatically assesses students’ visual and technical correctness on their written zhuyin symbols input into two intelligent sketch user interfaces: one for writing practice on prompted individual symbols and one for writing practice on the phonetic symbol mapping of prompted Chinese characters.

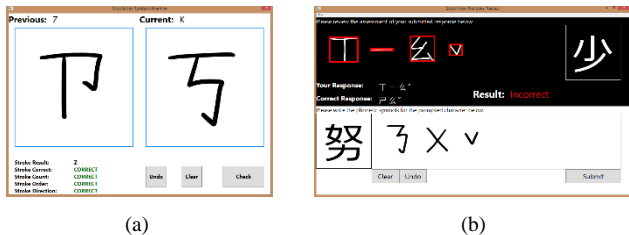


Figure 6. BopoNoto’s different writing interfaces: (a) symbol writing practice interface and (b) phonetics writing interface.

A. Symbol Writing Practice Interface

The first intelligent sketch user interface in the BopoNoto application allows students to practice writing individual symbols and subsequently receive visual and technical correctness, which is appropriate for students who are initially learning about the zhuyin phonetic script or who wish for a refresher of the zhuyin phonetic symbols directly (Figure 6a). This particular interface consists of randomly displaying the approximate English transliteration of the symbol for the user to write; the user then provides their solution in a sketching canvas on the right side of the interface. The user has access to conventional interaction buttons to undo or clear their strokes, and can submit their solution for the interface.

Following the submission of their written input, the left side of the interface subsequently displays a visual representation of any mistakes in their sketch for review (see Figure 7), while also displaying textual information listing the technical correctness of their submitted written input.

B. Phonetics Writing Practice Interface

The second intelligent sketch user interface allows students to practice writing the phonetics of different Chinese characters (Figure 6b). That is, for each prompted Chinese character, the user must write the phonetic symbols associated with the pronunciation of that character. This particular interface is well-suited for students who have become more confident in their familiarity of the zhuyin phonetic symbols.

The specifics of this particular interface follows similarly to the symbols writing practice interface, where users first write their written multi-symbol input and then submit their answer for feedback. Their submitted answer is then highlighted to visually demonstrate how closely proportioned their individual

symbols are, as well as indicating how their input compares to the expected model solution.

C. Assessment and Feedback Display

For the symbols writing practice interface, we provide two types of feedback for communicating the assessment of the students’ written zhuyin phonetic symbol input. The first way is through text that lists the different written technique assessments and corresponding correctness level. The second way is through visual cues that overlap the original written input with either highlighted strokes and points or examples of the expected model template solution.

Examples demonstrating the specific visualizations of the four assessment cases of incorrect visual structure or written technique can be found in Figure 7. For incorrect stroke count, the user is shown both an example of the expected model template solution, as well as highlighted stroke endpoints from both the written input and model template, where the blue endpoints on the model template indicate the correct stroke. For incorrect stroke direction, the first temporally incorrect stroke is highlighted in red. For incorrect stroke direction, the temporally incorrect starting endpoint of the stroke and the stroke itself is highlighted as an orange point and a red stroke, respectively. For incorrect symbols, the submitted written input is overlaid with the strokes from a model template solution as orange strokes. Lastly, correct submitted written input – not shown in Figure 7 – does not provide any visual cues as a way to notify the user that there are no errors within their solution.

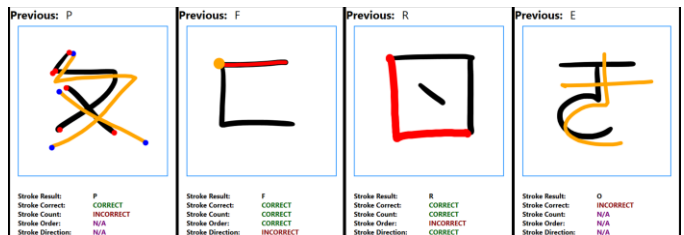


Figure 7. Visualizations of the different assessment cases of incorrect visual structure or written technique (L-R): incorrect stroke count, incorrect stroke direction, incorrect stroke direction, and incorrect symbol.

V. RESULTS AND DISCUSSION

Since our BopoNoto application is an intelligent sketch user interface, we approach evaluating our application from its recognition performance, where we evaluate how well it classifies users’ written symbols and compare it to related zhuyin sketch education application LAMPS [28]; and also from its interaction performance, where we evaluate how well it assesses written symbols and whether users understood how to use the writing practice interfaces.

A. Recognition Evaluation

1) Overall Symbol Recognition Performance

For evaluating the performance of BopoNoto’s recognition system, we recruited ten participants – three females – who consisted of one former Chinese as a Second Language (CSL) instructor with native zhuyin fluency, four Taiwanese international engineering graduate students with native zhuyin fluency, and two American university students with no prior exposure with zhuyin, and three American university students

visual and technical correct of students' written input. From our evaluations, we discover that our application succeeds in robust recognition and understandable interaction of zhuyin phonetic symbols.

REFERENCES

- [1] Ager, S. Zhuyin fuhao / bopomofo. <http://bit.ly/1I2OSL4>, 2015. [Online; accessed 31-May-2015].
- [2] Campbell, L. *Historical Linguistics: An Introduction*, 3 ed. The MIT Press, Cambridge, 2013.
- [3] Cheema, S., and LaViola, J. Physicsbook: A sketch-based interface for animating physics diagrams. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces, IUI '12*, ACM (New York, NY, USA, 2012), 51–60.
- [4] Chiu, C. Learn bopomofo! <http://bit.ly/1SRzTXw>, 2011. [Online; accessed 31-May-2015].
- [5] Dahmen, K., and Hammond, T. Distinguishing between sketched scribble look alike. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3, AAAI'08*, AAAI Press (2008), 1790–1791.
- [6] DeFrancis, J. *Chinese Language: Fact and Fantasy*, 1 ed. University of Hawaii Press, Honolulu, 1984.
- [7] Deselaers, T., Keysers, D., Rowley, H., Wang, L.-L., Carbone, V., Popat, A., and Narayanan, D. Google handwriting input in 82 languages on your android mobile device. <http://bit.ly/1FM4Gvg>, 2015. [Online; accessed 31-May-2015].
- [8] Dixon, D., Prasad, M., and Hammond, T. icandraw: Using sketch recognition and corrective feedback to assist a user in drawing human faces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10*, ACM (New York, NY, USA, 2010), 897–906.
- [9] Edge, D., Searle, E., Chiu, K., Zhao, J., and Landay, J. A. Micromandarin: Mobile language learning in context. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11*, ACM (New York, NY, USA, 2011), 3169–3178.
- [10] Everson, M. E. Issues in chinese literacy learning and implications for teacher development. In *Issues in Chinese Language Education and Teacher Development*, P. Duff and P. Lester, Eds. University of British Columbia: UBC Centre for Research in Chinese Language and Literacy Education, 2008, 70–78.
- [11] Government Information Office Republic of China. *Taiwan Yearbook 2006: The People & Languages*. Government Information Office Republic of China, Taipei, 2006.
- [12] He, W. W., and Jiao, D. Curriculum design and special features. In *Teaching and Learning Chinese: Issues and Perspectives*, J. Chen, C. Wang, and J. Cai, Eds. Information Age Publishing, Charlotte, 2010.
- [13] Heisig, J. W. *Remembering the Kanji I: A Complete Course on How Not to Forget the Meaning and Writing of Japanese Characters*, 4 ed. Japan Publications Trading Company, Tokyo, 2001.
- [14] Kara, L. B., and Stahovich, T. F. An image-based, trainable symbol recognizer for hand-drawn sketches. *Computers & Graphics*. 29, 4. (Aug. 2005), 501–517.
- [15] Liu, C.-L., Jaeger, S., and Nakagawa, M. Online recognition of chinese characters: The state-of-the-art. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 2 (Jan. 2004), 198–213.
- [16] Lunde, K. *CJKV Information Processing: Chinese, Japanese, Korean & Vietnamese Computing*, 2 ed. O'Reilly Media, Sebastopol, 2009.
- [17] Mandarin Daily News. Chinese education online - bopomofo. <http://www.mdnkids.com/BoPoMo/>, 2015. [Online; accessed 31-May-2015].
- [18] Matic, N. P., Platt, J. C., and Wang, T. Quickstroke: An incremental online chinese handwriting recognition system. In *Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02) Volume 3 - Volume 3, ICPR '02*, IEEE Computer Society (Washington, DC, USA, 2002), 30435–.
- [19] McNaughton, W., and Ying, L. *Reading & Writing Chinese: Traditional Character Edition*, 2 ed. Tuttle Publishing, North Clarendon, 1999.
- [20] National Taiwan Normal University. *Practical Audio-Visual Chinese 1*, 2 ed. Zheng Zhong/Tsai Fong Books, Taipei, 2008.
- [21] National Taiwan Normal University. Mandarin training center. <http://bit.ly/1FmNBaJ>, 2015. [Online; accessed 31-May-2015].
- [22] Pittman, J. A. Handwriting recognition: Tablet pc text input. *Computer* 40, 9 (Sept 2007), 49–54.
- [23] Sigma Sky, LLC. write bopomofo. <http://apple.co/1I2X11V>, 2014. [Online; accessed 31-May-2015].
- [24] Su, Q. G. Bopomofozhuyin fuhao: an alternative to pinyin. <http://abt.cm/1GTaebc>, 2008. [Online; accessed 31-May-2015].
- [25] Syson, M. B., Estuar, M. R. E., and See, K. T. Abkd: Multimodal mobile language game for collaborative learning of chinese hanzi and japanese kanji characters. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 03, WI-IAT '12*, IEEE Computer Society (Washington, DC, USA, 2012), 311–315.
- [26] Taele, P., Barreto, L., and Hammond, T. Maestro: An intelligent educational sketching tool for learning music theory. In *Proceedings of the Twenty-Seventh Innovative Applications of Artificial Intelligence Conference, IAAI '15 (2015)*, 3999–4005.
- [27] Taele, P., and Hammond, T. Hashigo: A next-generation sketch interactive system for japanese kanji. In *Proceedings of the Twenty-First Innovative Applications of Artificial Intelligence Conference, IAAI'09*, AAAI Press (2009), 153–158.
- [28] Taele, P., and Hammond, T. Lamps: A sketch recognition-based teaching tool for mandarin phonetic symbols i. *J. Vis. Lang. Comput.* 21, 2 (Apr. 2010), 109–120.
- [29] Takezaki, K., and Godin, B. *An Introduction to Japanese Kanji Calligraphy*. North Clarendon, Tokyo, 2008.
- [30] Tian, F., Lv, F., Wang, J., Wang, H., Luo, W., Kam, M., Setlur, V., Dai, G., and Canny, J. Let's play chinese characters: Mobile learning approaches via culturally inspired group games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10*, ACM (New York, NY, USA, 2010), 1603–1612.
- [31] Valentine, S., Vides, F., Lucchese, G., Turner, D., hoe Kim, H., Li, W., Linsey, J., and Hammond, T. Mechanix: A sketch-based tutoring system for statics courses. In *Proceedings of the Twenty-Fourth Innovative Applications of Artificial Intelligence Conference, IAAI '12 (2012)*, 2253–2260.
- [32] van Sommers, P. *Drawing and Cognition: Descriptive and Experimental Studies of Graphic Production Processes*. Cambridge University Press, Cambridge, 1984.
- [33] Wang, H.-H., and Honig, A. S. What difficulties do children experience while learning to read and write chinese. In *Teaching and Learning Chinese: Issues and Perspectives*, J. Chen, C. Wang, and J. Cai, Eds. Information Age Publishing, Charlotte, 2010.
- [34] Wobbrock, J. O., Wilson, A. D., and Li, Y. Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, UIST '07*, ACM (New York, NY, USA, 2007), 159–168.
- [35] Wolin, A., Eoff, B., and Hammond, T. Shortstraw: A simple and effective corner finder for polylines. In *Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling, SBM'08*, Eurographics Association (Aire-la-Ville, Switzerland, Switzerland, 2008), 33–40.