LAMPS: A Sketch Recognition-Based Teaching Tool for Mandarin Phonetic Symbols I

Paul Taele, Tracy Hammond
Sketch Recognition Laboratory
Department of Computer Science and Engineering
Texas A&M University
{ptaele, hammond}@cse.tamu.edu

Abstract
The non-Romanized Mandarin Phonetic Symbols I (MPS1) system is a highly advantageous phonetic system for native English users studying Chinese Mandarin to learn, yet its steep initial learning curve discourages language programs to instead adopt Romanized phonetic systems. Computer-assisted language instruction (CALI) can greatly reduce this learning curve, in order to enable students to sooner benefit from the long-term advantages presented in MPS1 usage during the course of Chinese Mandarin study. Unfortunately, the technologies surrounding existing online handwriting recognition algorithms and CALI applications are insufficient in providing a “dynamic” counterpart to traditional paper-based workbooks employed in the classroom setting. In this paper, we describe our sketch recognition-based LAMPS system for teaching MPS1 by emulating the naturalness and realism of paper-based workbooks, while extending their functionality with human instructor-level critique and assessment at an automated level.

1 Introduction
One of the core difficulties in achieving fluency of Chinese Mandarin for native English users is the language's substantially contrasting written component. The written script used in the various spoken languages of Chinese (e.g., Mandarin, Cantonese) differs greatly from the written script of western languages (e.g., English, Spanish) in that it is not phonetic. In other words, the pronunciation corresponding to written Chinese cannot be determined directly from solely reading it. This particular trait of written Chinese thus poses a difficult challenge to novice students, especially those with primarily native English fluency, as they encounter new Chinese Mandarin words in the language. Specifically, these novice students are unable to make use of newly acquired words from readings for oral communication, unless they resort to the aid of dictionaries or the help of native Chinese Mandarin speakers to provide the pronunciation.

In order to resolve the non-phonetic nature of written Chinese, various phonetic systems have been devised that can also be used to complement the learning of new vocabulary in the language. These various phonetic systems in general can be divided into two different subgroups: Romanized (i.e., phonetic systems that use letters from the Roman alphabet to represent the sounds of Chinese Mandarin) and non-Romanized phonetic systems.

Romanized phonetic systems, such as the Hanyu Pinyin system developed in China, are some of the most popular phonetic systems currently utilized in numerous textbooks and language programs. One advantage of Romanized phonetic systems to novice students is their low initial learning curve, since they incorporate a writing script (i.e., Roman alphabet) that native English users already know. On the other hand, this initial advantage ends up also being its greatest disadvantage. Specifically, letters in the Roman alphabet were designed to represent the sounds of western languages and not the sounds of Chinese Mandarin. To illustrate this point, the Romanized phonetic representation without tone markings of the Chinese characters 是 and 吃 in the popular Hanyu Pinyin system are shi and chi, respectively. At first glance, a native English user will intuitively assume that the pronunciation based on these Romanized representations are equivalent to the sounds of the English word “she” for shi and initial “chee-” sound from the English word “cheese” for chi, when in fact the sounds for these Chinese characters lack an equivalent sound in spoken English. Another example involves the Romanized phonetic representations without tone markings for the Chinese characters 比 and 花, which are bi and ri, respectively. At first glance to a native English user, the pronunciation of the -i ending in both Romanized representations would appear to mutually be the same, when in fact the sounds of their vowel endings differ from each other greatly (i.e., the former phonetic representation bi in Chinese Mandarin is exactly like the pronunciation of the letter “b” in English, while the latter phonetic representation ri in Chinese Mandarin has no corresponding sound in English).

Non-Romanized phonetic systems avoid the ambiguities of Romanized phonetic systems, since symbols in these phonetic systems do not rely on letters in the Roman alphabet to represent the contrasting sounds of Chinese Mandarin. One such non-Romanized phonetic system is Mandarin Phonetic Symbols I (MPS1), which also goes by the name Zhuyin Fuhao and Bopomofo, a
legitimate phonetic system developed and still widely used in Taiwan (Figure 1). Since the symbols in MPS1 were originally designed to map strictly to Chinese Mandarin phonetics, they eliminate the dependency of consonants and vowels in the Roman alphabet that can cause novice language students to confuse them with existing sounds in English. This is because of the inherent inability of symbols in Romanized phonetic systems to accurately replicate many sounds in Chinese Mandarin that are achievable in non-Romanized phonetic systems such as MPS1 [DeF09][RoC06].

Figure 1. The thirty-seven MPS1 symbols. The adjacent Latin characters are their romanized equivalents (source: Wikipedia).

Yet, the lack of visual semblance between the symbols in MPS1 and the familiar letters in Romanized phonetic systems adds an extra layer of complexity to students, which involves requiring them to skillfully master an additional set of symbols before they map the sounds of Chinese Mandarin to the written Chinese characters. This additional layer typically drives language programs to immediately accclimate students to learning the pronunciation of written Chinese. The consequence of this action is that language programs sacrifice the instruction of MPS1 for less dependable Romanized phonetic systems due to the sake of time.

Approaches that can be used to reduce the initial learning curve would thus be beneficial in motivating the practice of MPS1 as a phonetic guide for students with native English fluency. One approach that aids in effective memorization of the more general scripts in written East Asian languages includes the teaching of systematic written techniques called stroke order and stroke direction [BOSS99], where the stroke is defined to be a single mark made by the writing utensil from the initial contact to the final contact on the surface. Since the visual structures of symbols in MPS1 are based on existing Chinese characters [Su08], they share the same property as Chinese characters in that they also employ written techniques. These written techniques are often taught in language courses to help facilitate the memorization of written Chinese, and can apply equally as well in the instruction of MPS1 to language students [NTN04]. Additionally, correct application of these written techniques are heavily stressed early in the learning process, in order to discourage bad writing habits that may hinder effective memorization [MY99].

While proper written technique is vital for effective memorization of the symbols in MPS1, the dominant medium used in the classroom to facilitate the instruction of the writing aspect of Chinese Mandarin, including MPS1, consists of traditional paper-based workbooks and assignments. The advantage of paper-based tools is that they allow students to learn through writing, which plays a key role in learning the written component since it allows students to improve the aesthetic appearance of their writing and acquire a “natural feel” for them that cannot be achieved simply by remembering them [Hei07]. Although paper-based workbooks are effective in allowing students to physically perform actual writing during the learning process, workbooks by themselves are only effective to instructors for critiquing the visual structure of students’ writing and not the written technique. Assessing whether students are using correct written technique currently requires the aid of an instructor physically monitoring students’ writing (Figure 2). Such assessment is also not only time-consuming on the instructors’ part, but also unrealistic to perform as the number of students increase [CJL07][CJYL08].
With advances in online sketch and handwriting recognition algorithms for written East Asian, and with decreasing costs of Tablet PCs (TPCs) that permit pen input, computer-assisted language instruction (CALI) applications with pen-input recognition capabilities are an affordable and potential solution to resolve the limitations of traditional paper-based workbooks [vDBS07][WM04]. An ideal intelligent user interface for teaching tools specific to MPS1 would thus involve emulating the natural flow of free handwriting that is done with paper-based workbooks, recognizing stylus input that is handled by online handwriting recognition algorithms specific to written East Asian with reasonable accuracy, and assessing written technique and providing feedback that are performed by human instructors.

In this paper, we therefore discuss our LAMPS (Language Assistant for Mandarin Phonetic Symbols I) system, a sketch recognition-based CALI system for teaching MPS1 to novice students of Chinese Mandarin. With our system, students are provided a free-form writing space to practice the natural writing of symbols in MPS1 with a stylus, and are later provided valuable human instructor-level feedback on the visual and technique correctness of their written MPS1 input. In essence, we created an intelligent workbook for MPS1 that can be used to both complement the standard classroom curriculum and supplement distance education programs.

2 Related Work

While prior research work has been conducted in novel methods for inputting Chinese Mandarin using MPS1 [CAM05][LLA05], these systems were designed to give users who were already familiar with MPS1 more convenient input on mobile devices. In comparison, CALI applications specific to MPS1 have been scant, and such applications for handling the recognition of students’ handwritten input are even more lacking. Since constructing a system capable of recognizing students’ handwritten MPS1 input plays an important factor, we therefore examine research literature for the related domain of Chinese characters.

2.1 Traditional OCRR Written Technique Approaches

A subset of the literature focuses exclusively on using handwriting recognition as a method for the instruction of written Chinese [NAOK98], and even though CALI applications can assess students’ visual technique, correctness of students’ stroke order and direction still has its limitations.

Two particularly distinct machine learning techniques commonly implemented for OCRR with high accuracy rates are neural networks (NNs) and hidden Markov models (HMMs) [LJN04]. The former technique, NNs, is used in the handwriting recognizer for Microsoft’s Window Vista Operating System, and functions similarly to other NN implementations in that the recognition is based on various features from users’ digital handwritten input [Pit07]. While NNs generalize well to variations in users’ handwriting, their effectiveness in handwriting recognition-enabled pedagogical tools is limited, since the information on students’ written technique (e.g., stroke order and stroke direction) is not taken into consideration.

The latter technique, HMMs, differs from NNs in that stroke order and stroke direction can be exploited in the recognition process [NSS03]. The weakness of HMMs for assessing students’ written technique though is that stroke order and stroke direction information is used primarily to aid in the handwriting recognition process. As an implementation in CALI applications, HMMs alone also suffer in that they cannot provide feedback that can differentiate between students’ handwritten input that is visually correct but technique-wise incorrect to input that is visually incorrect. Furthermore, HMMs, by design require a different model for each possible set of stroke orders and directions that students may feasibly write. Otherwise, HMMs will misclassify some students’ handwritten input that has unaccounted stroke order and direction possibilities, since these possibilities are assigned extremely low probability in the recognition process by default.

2.1 OCRR Written Technique Approaches

In order to develop an MPS1-specific CALI application that can function as a “dynamic” extension of traditional paper-based notebooks, the system must be able to:

- recognize students’ handwritten input that language instructors would deem correct,
- provide an input space that mimics the natural free sketching found with the paper medium to accommodate the writing of multiple symbols (e.g., provide direction-independence of writing, since written Chinese and MPS1 can be written both vertically and horizontally),
- achieve recognition of provided input independent of the size or number of written symbols,
- and provide human instructor-level feedback and assessment that can seamlessly critique the visual and technical correctness of students’ writing.

Since commonly-used machine learning techniques for OCRR have their limitations in assessing both the visual and technique correctness of students’ handwriting for written Chinese, this issue has motivated research work to devise alternative approaches.
Earlier research work from [SHC98], and later improved upon in [SF04], utilized two separate techniques for assessing the stroke order and stroke direction of the prompted character. Assessment of stroke order correctness involved first defining a stroke as the endpoints of the lines that make up a stroke, and then critiquing the correctness of the sequence of spatial positions relative to the other strokes in the character. Assessment of stroke direction correctness, in comparison, first determined the direction of the lines in each stroke based on the temporal sequence of the endpoints, assigned them a numerical code that corresponds to one of the eight compass directions, and then analyzed the correctness of the sequence. The weakness of this system in creating a “dynamic” workbook is that the system was designed to have users trace over the outline of Chinese characters. Therefore, the correctness of the visual structure in the students’ writing is never assessed, since no actual handwriting recognition takes place.

Later research works from [CJL07][CJYL08] provided more freedom in how users write the characters. Their method in assessing stroke order and stroke direction involved grouping all possible lines into six different slopes, and then critiquing the order and direction based on the temporal sequence written by the user. The limitations of the system for use in the envisioned “dynamic” workbook though stems from the handling of the visual structure assessment. Their system assumed the entire drawing area of the character was dedicated to one character, and the correctness of the character was based on all the features within this coordinate space. Since this restriction required that a single character be drawn relative to the drawing area, both the size independence and multiple written input capabilities necessary for a “dynamic” workbook is lost.

More recent research work by [Qi08] proposed a solution to the size independence issue by first defining a bounding box for handwritten input of characters, splitting the input into a three-by-three grid, and then comparing the features from the pixels in those grid blocks to possible candidate characters. While the size independence issue was resolved, several concerns restricted this approach from achieving a “dynamic” workbook. One concern was that this system always assumes that the strokes originated from a single character, and provided no information on how it can handle the recognition of separate characters within the same writing space. Another concern was that the system always assumed the characters it was attempting to recognized lied within a quadrilateral block, which fails for single-line Chinese characters such as the Chinese character – and similar-looking MPS1 symbols. An additional concern with the approach was that the written technique assessment is handled separately in a different system, which can yield to the incompatible scenario of when the written technique assessment from one system responds as correct for a visually incorrect character.

A different approach from another recent research work by [TH09] adapts free-sketch recognition techniques capable of recognizing unconstrained writing with reasonable accuracy [HEP*08], and applies them to the instruction of Chinese characters. While the system described in the paper achieves most of the requirements deemed necessary for constructing a “dynamic” notebook, a solution for the difficult problem of seamless recognition of multiple-character input was largely absent in the paper. We therefore use the system from that paper as a model to expand on, in order to permit multiple-symbol recognition necessary for MPS1 instruction with a “dynamic” workbook medium.

### 3 Implementation

The core ideas for building a “dynamic” workbook to teach MPS1 stem from our previous work in [TH08]. Several unresolved issues from this previous work include an inelegant workaround for the recognition of single-line strokes by requiring an additional overtraced stroke, an inefficient solution to the handling of arcs that involved approximating them as multiple lines, and the lack of multiple interpretations for how MPS1 symbols were written. In this section, we first provide the relevant details on the resources employed in our implementation. We then explain how these resources were employed in order to make our envisioned “dynamic” workbook system possible. Lastly, we describe in further detail how we addressed the above mentioned issues that plagued the previous iteration of our system.

#### 3.1 Corner-Finding Algorithms

The lowest layer worth discussing regarding our system’s implementation is the capturing of data that represents the students’ writing. At this fundamental layer, students using our system on pen-enabled computers (e.g., Tablet PCs) will use a stylus to write their input, which will then be stored by the computer and represented back to the student as a given set of pixels. By treating this set of pixels as points in Cartesian space, we take advantage of different algorithms that can process these series of points and later approximate them as basic geometric primitives (e.g., lines, curves, arcs, ellipses) [SSD07][PH08].

We further make an assumption that symbols in MPS1 can be visually approximated entirely as line and arc primitives. This allows us to exploit stroke processing algorithms that are capable of fragmenting the points collected from students’ stylus input into their elementary parts, which is then used for later written recognition and assessment feedback.

To aid in the task of processing the strokes into their representative geometric primitives, we selected the Sezgin [SSD07] and the PaleoSketch [PH08] algorithms for their strengths in detecting the line and arc primitives of the given strokes, respectively. Since we explicitly define a stroke in this paper as being a temporal sequence of points collected by the computer, from the pen-down motion on the writing surface to the next pen-up motion, the key idea shared by these two algorithms is that the
corresponding corners detected in a stroke serve as the endpoints of recognized lines and arcs. The Sezgin corner-finding algorithm's ability to detect corners for lines from these strokes stems from the observation that people slow down during the formation of strokes in their writing. Therefore, the algorithm relies on curvature and velocity data of the pen's direction during the writing process in order to make its selection on the stroke's corners (Figure 3).

Alternatively, the PaleoSketch corner-finding algorithm's ability to detect corners specifically for arcs uses the same concepts of computing the direction, speed, curvature, and corner values from the Sezgin algorithm [PH08]. The PaleoSketch algorithm further expands upon the Sezgin algorithm by calculating the normalized distance between direction extremes (NDDE) and direction change ratio (DCR), two additional features that have proven very useful in the algorithm's ability to recognize a larger set of geometric shapes that include arcs.

![Figure 3. Fragmenting the strokes of sample MPS1 symbols using the Sezgin and PaleoSketch recognizers into their primitive line and arc components. (a) Segmentation using only the Sezgin recognizer. (b) Segmentation using the Sezgin and PaleoSketch recognizers.](image)

We stress the importance of processing the strokes from their explicit temporal sequence of points into their geometric primitives, because this enables our system to later achieve handwritten recognition of those geometric primitives from the students' handwritten input and independent of the writing space. With handwritten recognition of MPS1 symbols independent of the writing space, we accomplish free-sketch recognition capabilities that are a necessary step for achieving a "dynamic" workbook. The next important step for providing handwriting recognition of students' written MPS1 symbols involves how the interaction between those very geometric constraints can be exploited for categorizing them to their corresponding MPS1 symbol, which we further expand on in the next section.

### 3.2 LADDER Sketching Language

Once the strokes for the students’ writing are processed into their geometric primitives, we must categorize the groupings of those geometric primitives into their corresponding MPS1 symbols using pattern recognition. In order to provide this pattern recognition with reasonable accuracy for this domain, we resort to the LADDER sketching language [TD07] to aid in our sketch recognition needs. Since LADDER is a general purpose sketching language for describing how sketch diagrams for various domains are drawn, displayed, and edited, we adopted this sketching language for the domain of MPS1.

What differentiates the pattern recognition of students' handwriting with the LADDER language from traditional pattern recognition techniques (e.g., neural networks, hidden Markov models) for the domain of MPS1 is the emphasis on recognizing the writing. Specifically, our system focuses more on recognizing students' handwriting based on whether it fulfills a set of requirements. Not only does this free our system from using training data that restricts the recognition to existing training data from model users, but it is also similar to how language teachers determine whether students succeeded in correctly writing symbols in MPS1 by verifying if all the necessary visual structure requirements for those symbols have been met.

Contrast how our system recognizes students' handwriting of MPS1 symbols to alternative systems which employ traditional pattern recognition techniques for the instruction of MPS1. A major disadvantage of the latter pattern recognition approaches involves instances where a student may write a particular symbol slightly visually incorrect (e.g., missing or extra strokes, sloppiness), yet still obtain the correct recognition from the system. This is a consequence of traditional pattern recognition techniques which inherently recognize written input based on the closest match in the training set. This greater leeway in recognition may be appropriate for native writers of MPS1 symbols, since they would prefer writing symbols with the convenience of higher recognition over the perceived hassle of pedagogical-based feedback on a domain that they already master. For students learning MPS1 symbols though, this extra leeway in recognition is less suitable, since it would deteriorate students' learning of the symbols due to the system not correcting those slight mistakes.

The actual recognition of students' handwritten MPS1 symbols using LADDER involves the use of shape descriptions, which are structures primarily containing geometric information for categorizing the handwritten input using the sketching language’s syntax. The shape descriptions that are constructed in LADDER can be used to describe a wide variety of shapes such as MPS1 symbols.
symbols, and these shape descriptions consist of multiple specifications. These specifications and how they are used in recognizing the visual structure and written technique of students’ handwritten MPS1 symbols are elaborated in the next section.

3.3 Visual Structure Assessment

With the use of shape descriptions, our system can assess the correctness on the visual structure of students’ written MPS1 symbols, which is done by the various specifications that make up the LADDER sketching language’s syntax. The visual structures and a subset of the specification types used in partial shape descriptions of representative MPS1 symbols can be found in Figure 4.

The first specification we make use of is components, which consist of a combination of predefined and user-defined shapes that serve as the building blocks of the MPS1 symbols we wish to recognize. Predefined shapes in LADDER consist of primitive shapes such as lines, arcs, curves, and ellipses, while user-defined shapes consist of shapes in the domain that the user creates. For the domain we’re working with, lines and arcs are the only predefined shapes used, while partial and complete MPS1 symbols encompass user-defined shapes. It follows then that not only can user-defined shapes in LADDER be built as a collection of primitive shapes, but they can also be built using simpler user-defined shapes. In our system’s implementation, we take advantage of this particular trait of the LADDER sketching language in order to simplify the construction of the more visually complex symbols in MPS1.

The next specification we utilize is constraints, which explicitly defines the behaviors related to the components and the relationships between them. The LADDER language already has a large built-in set of constraints, and the complete list of constraints that we use for our shape definitions from this larger set can be found in Figure 6. For ease of explanation of the constraints used to describe the interactions used within and between the primitive shapes in our system, we categorize the employed constraints by functionality:

- **Orientation:** Checks whether a line is a slope or an anti-diagonal.
- **Point Relationships:** Compares the center, endpoint, or bounding position of one shape to another.
- **Position:** Compares the position of two shapes relative to each other.
- **Proximity:** Compares the proximity of two shapes relative to each other.
- **Length:** Compares the length of two shapes relative to each other.
- **Logical:** Involves negating a constraint or operating disjunction on two constraints. By default, all constraints in LADDER are mutually conjunctive.

The first specification we make use of is components, which consist of a combination of predefined and user-defined shapes that serve as the building blocks of the MPS1 symbols we wish to recognize. Predefined shapes in LADDER consist of primitive shapes such as lines, arcs, curves, and ellipses, while user-defined shapes consist of shapes in the domain that the user creates. For the domain we’re working with, lines and arcs are the only predefined shapes used, while partial and complete MPS1 symbols encompass user-defined shapes. It follows then that not only can user-defined shapes in LADDER be built as a collection of primitive shapes, but they can also be built using simpler user-defined shapes. In our system’s implementation, we take advantage of this particular trait of the LADDER sketching language in order to simplify the construction of the more visually complex symbols in MPS1.

The next specification we utilize is constraints, which explicitly defines the behaviors related to the components and the relationships between them. The LADDER language already has a large built-in set of constraints, and the complete list of constraints that we use for our shape definitions from this larger set can be found in Figure 6. For ease of explanation of the constraints used to describe the interactions used within and between the primitive shapes in our system, we categorize the employed constraints by functionality:

- **Orientation:** Checks whether a line is a slope or an anti-diagonal.
- **Point Relationships:** Compares the center, endpoint, or bounding position of one shape to another.
- **Position:** Compares the position of two shapes relative to each other.
- **Proximity:** Compares the proximity of two shapes relative to each other.
- **Length:** Compares the length of two shapes relative to each other.
- **Logical:** Involves negating a constraint or operating disjunction on two constraints. By default, all constraints in LADDER are mutually conjunctive.
Figure 6. The complete list of constraints used in the LAMPS system, a subset of the constraints in LADDER.

The format style we use to order the constraints involves grouping them into line orientations, endpoint ordering, and spatial relationships. Since we can approximate the symbols entirely as lines, arcs, or ellipses, we constrain the lines particularly as either sloped, anti-diagonal, or the negation of one of the two. In addition, each line defined in LADDER has endpoints and midpoints assigned $p_1$, $p_2$, and $center$, respectively. Since the assignment of endpoints $p_1$ and $p_2$ in each line changes depending on how the line is drawn when context is not provided, we explicitly assign those endpoints in a systematic fashion. That is, we systematically assign endpoint $p_1$ to the left relative to endpoint $p_2$ for all non-vertical lines, and $p_1$ on top relative to $p_2$ for vertical lines. Once we constrain the lines, we finally define the interrelationships between the line components.

The third specification we employ is aliases, which allows us to assign alternate labels to existing component names. One of the benefits of using aliases is their ability to provide more intuitive names to either the primitive shape components or points within those components. Therefore, we can label a particular primitive shape either by its physical feature or by some enumeration. For symbols in MPS1, labeling a line or arc primitive by its stroke enumeration is also vital to written technique recognition in our system, and is elaborated further in the next section. Figure 7 below shows two practical instances of using aliases.

![Figure 7](image)

The last specification which we use is the display methods, which alters how components look in the written input when it is recognized. This specification serves more as a convenience to convey visual information to users of the system, and also does not contribute to the recognition or assessment of either the visual structure or the written technique of students’ written symbols. Nonetheless, we consistently set the display methods in the shape descriptions to maintain its original look (i.e., no beautification) and to be displayed in dark grey upon successful recognition.

### 3.4 Written Technique Assessment

Assessing the written technique of students’ written MPS1 symbols involves assessing the stroke orders and stroke directions of those symbols. This assessment takes place subsequent to the visual structure assessment step, and requires enumerating the strokes and points used in the shape descriptions directly, with the labels indicating target stroke order and direction for each symbol. Figure 8 illustrates how the given alternative alias labels compare to Figure 4. First, this information is used during the
written technique assessment step by referring back to the timing data of the primitive shape components and corresponding end- and midpoints segmented by the corner-finding algorithms. Since we already have stroke enumerated labels assigned from the constraints specification in the previous visual structure assessment step, we reference the timing data on those primitive shape components and points by their labels. Lastly, we determine whether the enumerated shape components and corresponding points are listed in ascending temporal order. If there is a discrepancy in the ordering, this implies that the user had written the stroke or its direction differently from the target written technique.

![Image](image_url)

**Figure 8.** Visual structures of MPS1 symbols (a) \( \text{eng} \) and (b) \( \text{s} \), and their corresponding partial shape descriptions (c) and (d), respectively.

### 3.5 Single-Line Symbols

Of the thirty-seven symbols and four explicit tonal marks in MPS1, three consist of a single line (Figure 9). While symbols and characters consisting of a single line are the simplest shapes to recognize in OCCR, their recognition for the multi-symbol recognition we desire in our “dynamic” MPS1 workbook application is far from trivial. The reason is because in a free-sketch recognition environment, where symbols and characters in written Chinese can be written either vertically or horizontally, these single lines have multiple interpretations. In other words, single lines may either represent a single-line symbol, or as a stroke in a multi-line symbol within the MPS1 domain. The non-triviality of this problem was demonstrated in related works which either did not address the problem with a single-character input assumption, or ignored the problem completely.

In order to address this issue, we rely on LADDER's approach to achieving faster interpretations, which involves enabling the option of doing greedy hierarchical recognition on students' written input. There is an option to disable greedy hierarchical recognition to have multiple interpretations of the written input, but doing so for our particular domain would come at a lengthy time cost.

**Figure 9.** The three single-line tone markings in MPS1. (a) The “i” symbol. (b) The 2nd tone. (c) the 4th tone.

### 3.6 Symbol Template-Based Matching

The previous iteration of this work was strongly dependent on the conjecture that a single symbol can be recognized using a single shape description. This conjecture would hold if users had writing styles that were very similar in nature. The reality though is that users can write these symbols with some degree of variation and still be considered correct by a human instructor, which would thus break our assumption of being able to recognize a single written symbol with only one shape description. To overcome this limitation, we take a template-based recognition assumption by constructing shape descriptions to compensate for such possible variations. This necessity of template-based matching can arise in situations such as when a set of users are writing a complex symbol two different ways (e.g., writing a subpart of the symbol which matches an existing simpler symbol’s shape description vs. writing without it).

The case for template-based matching is further justified regarding our solution for the non-trivial issue of online recognition of single-line symbols when multiple symbols are taken into account. The consequence of our solution was that we lose the generality of single lines as pure primitive shapes, since we are emulating the three single-line user-defined shapes previously mentioned as pure primitive shapes. To illustrate, suppose users draw a particular single-line stroke in a symbol as either horizontally or positively-sloped, then we can state in the shape description that the orientation of the line is the negation of being negatively-sloped. With our solution, we no longer can exploit this attribute since horizontal and sloped lines are now user-defined shapes instead of pure primitive shape components. Symbol templates resolve this issue by creating separate shape descriptions for ambiguously-orientated line components.

### 4 Application

We have developed a fully operational system for providing a “dynamic” workbook experience, one which incorporates the recognition techniques necessary described in the previous section for providing human instruction-level assessment (Figure 10). The current form of our system tests the MPS1 knowledge of students based on the vocabulary from the first chapter of [NTN04], a textbook used by several language programs in Taiwan.

At the start of running our system, the user is provided two additional windows. The top window informs the
user of the next character to write, while the bottom window provides an assessment of both the visual structure and also the written technique correctness of what the user had written. This window will provide the previous word prompted, the target pronunciation(s), and the assessment of what was written. If the user wrote a symbol technically incorrect, our system also provides a visual guide on how to correctly write that symbol with the correct written technique.

The first user study we conducted therefore evaluated the recognition rates of our system for the visual structure of written MPS1 symbols. Ideally, we would like for our system to sufficiently recognize the handwritten symbols of expert MPS1 writers based on our constructed shape descriptions, since the objective of our “dynamic” workbook is for students to eventually emulate the visual structure writing made by these native writers. For this user study, we recruited nine Taiwanese graduate students at our university with proficient MPS1 writing knowledge, with the additional prerequisite that they write the symbols as if though they were teaching someone not familiar with them. This was desired since casual writing was not representative of the type of model writing we wish to base our system’s recognition on as a pedagogical tool. The users were each asked to write each symbol in MPS1 twice, which we later evaluated for visual structure correctness on all-or-nothing recognition. In other words, the correctness of a written symbol in our system is only counted if the entire symbol is correctly recognized.

5 Evaluation

To attain the “dynamic” workbook interface catered for MPS1 instruction, it must be able to handle following:

- achieve recognition of students’ handwritten symbols with accuracy levels within the range of other handwriting recognizers for the domain, and
- provide correct feedback and assessment on par with a human language instructor

We therefore conduct three different user studies to gauge the performance our system for its effectiveness as the envisioned “dynamic” workbook.

The result of this first user study was that our system attained 95% accuracy on the visual structure of the study
participants’ handwritten input, where the expert writers wrote the correct symbol and that the misrecognized symbols were considered too sloppily drawn for our system to recognize. Since the system performance of similar online handwritten recognizers in the domain achieved accuracy within the range of 85% to above 95% [LIN04], we concluded our system attained sufficient recognition comparable to other recognizers. Since our system does not rely on the use of training data to improve the recognition of our system found in traditional machine learning techniques (e.g., neural networks, hidden Markov models), we generalized our system on the expert writers’ model handwritten input to further tweak our constructed shape definitions.

We then evaluated our system by determining whether our system would be able to adequately recognize the correctness of novice users’ sketched MPS1 symbols based on the visual structure and stroke order. A second user study was conducted to collect handwritten symbols from a second group, this time consisting of five American university students with no knowledge of East Asian writing whatsoever. Like in the previous user study, each participant was asked to write each symbol twice.

Since these participants had no knowledge of MPS1, we prompted the symbols for these participants to write. The result of this user study yielded 100% accuracy on visual structure recognition. We attributed this higher accuracy rate to the students writing the symbols more carefully and with less variation than the native writers, which is the type of learning behavior we predicted from novice students learning MPS1. In addition, for each symbol from this set of collected handwritten data that was recognized correctly for visual correctness, our system was also able to correctly assess their written technique (i.e., the stroke order and stroke direction) with an accuracy rate of 100%. The accuracy of correct assessment on written technique for these symbols is comparable to other systems based on LADDER for similar domains [TH08][TH09]. We also note that our approach have improved students’ learning of closely-related domains [TH09], and we expect it to similarly hold true for the learning of the symbols in MPS1.

We believe that our system has achieved reasonable assessment capabilities on the visual structure of students’ handwriting, while also providing complementary assessment for the written technique that is on par with that of human instructors. The next step in this research work is to conduct additional user studies and expand on the application in order to provide a more robust system for teaching MPS1. That is, we strive to further tweak how handwritten input is recognized to allow students greater flexibility in what they write, while still educating them in maintaining proper form and technique of their writings.

In addition, we plan to use our system as a way to motivate language programs to more strongly consider the adoption of TPCs in augmenting current instruction of Chinese Mandarin, including the instruction of MPS1 and future work in written Chinese. With decreasing cost of TPCs and the addition of CALI applications such as ours, we hope that we can help language instructors offer improved methods in the learning process.

7 Contributions
Due to the complexity of learning MPS1 before students can fully exploit its phonetic advantages over Romanized phonetic systems for Chinese Mandarin, updated solutions must be devised in order to overcome the assessment limitations inherent in paper-based tools and the time-consuming process with direct human instructor oversight on students’ writing. With our LAMPS system, we provide an extension to paper-based workbooks that allows students to maintain natural writing of paper while also gaining valuable feedback on the visual structure and written technique in their writing during MPS1 learning.

Acknowledgments
This research is supported by the NSF IIS Creative IT Grant #0757557 Pilot: Let Your Notes Come Alive: The SkRUI Classroom Sketchbook.
This research is also supported by the NSF IIS HCC Grant #0744150 Developing Perception-based Geometric Primitive-shape and Constraint Recognizers to Empower Instructors to Build Sketch Systems in the Classroom.

The authors would like to thank the members of the Sketch Recognition Laboratory for their valuable assistance in this research.

References


