High Resolution Hand-Pose Estimation System for Automatic Lecture Notes Transcription

Ming-Chun Huang, Sunghoon I. Lee, Xiaoyi Zhang, Wenyao Xu, and Majid Sarrafzadeh, *Senior Member*, IEEE

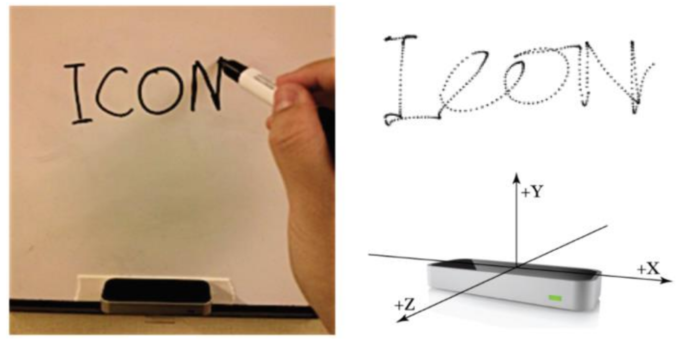
Abstract — *Lecture notes plays an important role in helping students understand and review the material taught in class. Fail to record notes in full directly impacts students' learning and reduces their willingness in participating the follow-up classes. Existing technology of lecture notes recording facilitates the note-taking burden for students, but the burden shifts to the lecturer end since special accessories are required for lecturers to record their handwriting. Therefore, lecturers may be reluctant to use the new technology because that usually means that they should adjust their used teaching habits. In this study, a lightweight lecture note transcribing framework is introduced using a portable high resolution hand-pose sensing device. Transcribed notes can be used as indexes to search taped class video contents. Three notes quality evaluation results demonstrate that the proposed design can effectively facilitate the effort for students in note-taking without increasing extra burdens for the lecture instructors.*

Index Terms — Hand-pose tracking, Handwriting Recognition, Lecture notes transcription, Infrared Sensor.

I. Introduction

Learning from class notes is an important process for students. Taking lecture notes is an active learning method to keep students concentrated during lectures and to help student's reviewing the material after lectures. However, students usually have trouble collecting the entire class notes due to insufficient note taking speed, being late to class, or occasional and unexpected interruptions. In fact, incomplete class notes is one of the major reasons that restrict students from efficiently reviewing the knowledge after class. As a result, more and more students record the lectures using cameras or microphones and replay them afterwards for reviewing. On the other hand, some schools require instructors to post their lecture notes online before the class. Nevertheless, if lecturers make any impromptu updates in their lecture notes or answers unexpected questions on the black/white board, these modifications will not be included in the prior-posted notes. Therefore, much research has been done to help automatic lecture notes recording.

Two straightforward solutions have been discussed: (i) to extract notes from videos and (ii) to use sensors such as touch surface or special designed pens for recording writing movement, commonly called pen and paper computing. Video-based techniques usually require relatively expensive equipment such as video or audio recorders in a designated classroom. Additionally, a sophisticated calibration process is often required. Furthermore, these techniques suffer from the lecturers' blocking the line-of-sight of the camera, which may result in incomplete class notes. As a consequence, sensor-based system with efficient hand movement tracking algorithm is considered as a promising solution to these problems.



(a) (b)

Fig. 1. The infrared sensing device setup is shown in Fig. 1(a) (Y-axis is the direction of infrared sensors). A comparison of written characters on the board and the handwritten trajectories from infrared sensors are plotted in Fig. 1(b).

Using the sensor-based systems, lecturers usually use a touch pen or their fingers to write materials on a touch surface. All written information can be automatically recorded and saved as downloadable files. However, installing touch-sensitive surface of a size of a black/white board is extremely expensive. Furthermore, traditional chalk or markers cannot be used on such touch surfaces. Sensing pen can be an alternative solution, but lecturers are required to adapt to these special designed pens. As a consequence, it is especially interesting to explore the possibility of combining the benefits of video-based system (i.e., less interference with lecturers in teaching) and the pen-based system (i.e., infrastructure-less system) for inconspicuously supporting lecture activities. For example, these techniques allow lecturers to write, erase, and overwrite existing notes. Moreover, student may search notes or even search taped lecture clips by keywords learned from a lecture for reviewing course materials.

This paper proposes a high mobile and inexpensive lecture note recording system that utilizes infrared sensors (Fig. 1) to capture handwritten trajectories. The proposed system can be easily deployed at different environments (e.g., different classrooms with various characteristics such as the lighting condition) without any calibration process. It also avoids technical problems of extracting non-duplicated class notes from a sequence of videos. The proposed system extracts geometric variables such as distance or location between the writing tool and a black/white board by tracking writing movements using mobile and inexpensive infrared sensors. A novel handwritten tracking algorithm based on conventional English writing behaviors.

The contributions of this work can be summarized as follows:

1. This work proposes a novel infrared-sensor based system to capture the handwritten trajectory;
2. This work develops a complete processing framework from data processing, notes generation, modification, assembling, to bidirectional search;
3. The performance (e.g., note quality) of the proposed system is evaluated based on the standard letter recognition techniques such as optical character recognition (OCR), Stroke Heuristics, and Stroke Tables.

II. RELATED WORKS

Before we proceed to system design, some prerequisite terminologies are explained in this section: hand-writing recognition and existing sensor assisted notes recorders. Handwriting recognition techniques are utilized to process sensor data stream and to evaluate the readability of the notes content.

A. *Overview of Sensor Assisted Note-Recorders*

Some universities recorded lecture videos, therefore, a variety of researches focus on photo/video analysis in order to generate class notes. One practical solution was blackboard segmentation [1]. This system first analyzed video to segment written regions on blackboard. Then it exported these regions of notes in photo format. This solution was effective especially when processing lectures in the past. However, it could only generate photo, which was not a search-friendly content. Another method [2], [3] was to use a camera scanning whole white board with several photos. Then it stitched those images together as a panorama and eliminates background and keep enhanced notes image. This solution was practical and generated readable notes image. There was also a system [4] which could recognize handwriting text from these photos by OCR. However, these two solutions could not automatically generate notes with time line in every line. In real settings, this system might miss content if it missed a single photo. The modification on notes also could not be preserved. Most significantly, the writer must not block the notes on the board, which was hard to achieve during lecture.

Due to drawbacks of photo/video analysis, companies released interactive whiteboard with huge touch screen. A commercial product, IR Touch, is a 65 inch LCD screen with infrared frame to detect touch events. Touch screens can provide fast response, and nowadays users get used to touch screen from Apple products. However, a screen can only achieve a relatively small range from 60 to 100 inches. Most importantly, the large touch screens are expensive, currently priced $3000-5000 based on screen sizes. Compete with high price of touch screen, current researchers have investigated methods to automatically record notes using cheaper sensors. Microsoft Kinect RGB-D camera could provide depth information based on its infrared camera. It stimulated researchers to innovate in remote control [5] and education [6]. The Low-Cost Efficient Interactive Whiteboard [7] used a Kinect to detect and track user hand movement on the whiteboard. Unfortunately, Kinect cannot provide accurate enough depth resolution when the hand is near the board, so this work combined visual data with depth information to track hand movements. Another revolutionary sensor, Nintendo Wiimote could provide relative position from a fixed IR emitter. Researchers explored its potential in 3D User Interface [8], Medical Data Visualization [9] and Human Computer Interaction [10]. Lee et al. designed an interactive projection solution [11]. In this system, a writing tool was equipped with an IR emitter and its position was tracked by using a fixed Wiimote on its field of view. A computer screen was projected onto a wall, the IR emitter writing tool was tracked accordingly and it allowed the user to write onto the projected computer screen image. Unfortunately, the Wiimote's field of view could be blocked by user body in daily usage, and therefore the system loses valuable tracking information.

I
cor
Ee-
Stu
que
cor
(pa
not
of
sen

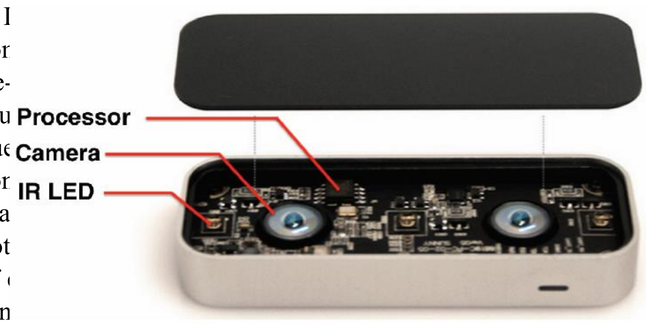


Fig. 2. An off-the-shelf hand-pose estimation sensing device

recognized human writing by reconstructing the relative position from acceleration with noise. This system used threshold to segment strokes, and then used grammar tree to recognize letters. However, its grammar tree forced writing order to reduce ambiguity in letters with the same strokes. Also it required users to hold the phone in a fixed pose, in order to fix patterns in accelerometer reading. Moreover, its average writing speed, 3s/letter, was not acceptable to write long content like lecture notes. ActivBoard, a commercial available product did not use infrared touch screen. Instead, it used special ActivPen to interact with whiteboard. The sensor inside the pen could track the relative position of writing on the board. Unfortunately, the writer needed to get used to ActivPen, which was thicker and heavier than marker or chalk. Moreover, the need of battery in ActivPen can be troublesome in actual working environment.

B. Overview Hand-writing Recognition Research

Hand writing recognition is the task of transforming a language represented in its spatial form of graphical marks into its symbolic representation. There are two kinds of handwriting input, on-line and off-line [14]. On-line handwriting input maintains the time series of writing points, order of strokes and additional information about pen tip (velocity, acceleration). For example, handwriting input methods on cell phones and tablets receive on-line handwriting input when users touch the screen. Preprocessing of on-line recognition includes noise removal, stroke and character segmentation. Off-line handwriting input only preserves images of the completed on-board writing area. For example, banks recognize handwriting amounts on checks. Preprocessing of off-line recognition includes setting thresholds to extract writing points, removal of noise, segmentation of writing lines, and finally segmentation of characters and words. Our system is a type of on-line handwriting input system and records time stamps of each points on the handwriting trajectory. However, the sensor tracks both on-board and off-board movement of writing tools. Thus, we need classification after segmentation to determine which strokes belong to on-board writing or off-board hand movement.

III. Sensory Device for Hand-Pose Estimation

The high resolution hand-pose estimation system shown in Fig. 2 is a new dual camera tracking system that uses infrared light to detect location, velocity and orientation information of targeted objects. It is equipped with infrared LEDs, infrared sensors, and an embedded processor (Fig. 2). By receiving reflected infrared signals from the objects with emitting range, it can compute the 3-D position of the objects relative to the device. It is designed to precisely track movements of thin objects such as fingers or pens. Moreover, multiple objects (fingers) can be tracked at the same time. Its sampling rate can range from 30 to 200 frames per sec based on CPU performance. Therefore, developers have mainly focused on using this sensor to track fingers movements and build gesture control interfaces of computers. Nevertheless, we realize its potential in assisting lecture notes creation in classroom environment. A flat eraser-holder commonly accompanies with black/white boards in a classroom. Chalks, markers, and erasers are placed on top of the holder. The sensor can be placed on the holder and vertically project LED lights, when a lecturer writes notes. This sensor has theoretical 0.01mm accuracy in 3-D position, which is 200 times more accurate compared to other competitors in the market. Ideally, all movements within the field of view of the sensor can be easily captured and recorded under the claimed precision. The precision of this device is also validated by the authors by steadily placing a marker tip on the board for 10 seconds. From the examination, the devices provided 0.35mm variance in XY-plane and 0.14mm in z depth. This accuracy has negligible effect on general gesture control of computer control, as gesture sequences are much more important than the accurate position. However, this accuracy highly affects handwriting movement collection, because identifying whether the collected movement is writing in the air or writing on the board becomes non-trivial. The variance of Z direction will dominate the ″writing or not” judgments and we do not want to restrict lecturers’ performance to move chalks or pens from the board for a large enough distance when they are not writing.

IV. System Design

This section discusses a sequence of information processing steps that automates the construction of lecture notes based the sensory device discussed in the previous section. Then, five main modules of the proposed prototype are elaborated in detail.

A. System Flowchart

There are five main components in the system architecture as shown in Fig. 3: Data Processing, Notes Generation, Notes Modification, Notes assembling, and Bidirectional (between lecture notes and lecture clips) search. Raw data stream returned from the infrared sensor contains position, velocity, and orientation information of every point within the

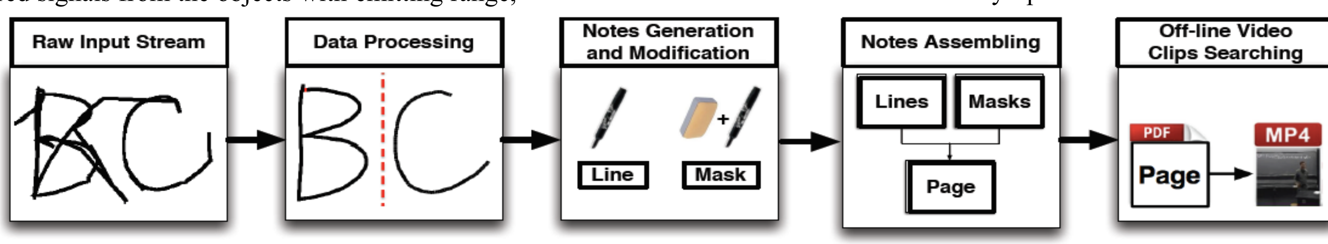


Fig. 3. System flowchart

handwriting trajectories. Segmentation, classification, and grouping algorithms are applied on the raw data based on this collected information, which contains human writing behaviors and conventions (i.e., English writing conventions). All alphabets written in a row are stored as an image. Two image types (Line and Mask) are assigned depending on whether the lecturer writes on a new row or simply modify identifiable notes.

B. Data Processing

Mask is an appended patch of the existing Line. The content in Mask will overlay on top of an existing Line when a series of Lines and Masks are assembled to form a page of lecture notes. Each page is created when the lecturer starts a new column and these pages can be used to index video clips with stored time stamps of each Line, Mask, and Page. Point series to characters conversion can be done with OCR and alphabet stroke table and it can be used to support keyword-based notes or clips search across all text

Raw data stream is composed of a sequence of points with writing tool's position, velocity, orientation and timestamp. A subset of points can form a stroke, then those strokes further form a character, then a word, and then a sentence. Data processing steps shown in Fig. 4 aim to segment data stream into strokes, remove the off-board movement, and group the on-board strokes into a character. The heuristics used for segment data points are oriented from our observations of English handwriting behaviors and conventions. However, the classification is not trivial, because full trajectories, including both on-board writing and off-board movement of writing tool are loyally returned by the sensor. Due to the accuracy limitation of the infrared sensor, it is improper to classify on-board and off-board trajectories simply based on the z-axis coordinates. Since single point information is not sufficient to make appropriate classification, we attempt to segment a set of points into a stroke, and make on-board or off-board classification for each stroke. After segmentation and classification, on-board handwriting strokes are ready for grouping. Grouping step is optional when producing image-based lecture notes, but it is useful for notes quality evaluation and character recognition in the keyword searching service.

1) Segmentation

The goal of segmentation is to identify individual strokes from the raw data stream based on human English handwriting behaviors and conventions. Three types of features can identify the start and end points of a stroke, including slow start and slow end in XY-plane, big jump in Z-direction speed diagram, and sharp angle transition in a handwriting sequence.

1. Slow start and slow end in XY-plane: When a writer attempts to write a stroke, he/she will start a stroke from slow speed, then speed up to a constant speed, and then slow down when the end of a stroke is reached.
2. Big jump in a Z-direction speed diagram: If the following stroke is not connected with the current written stroke, a writer should lift the writing tool so that he/she can move to the start of the next stroke. Thus, the speed impulse (big jump) in Z-direction can be used to segment between two consecutive strokes.
3. Sharp angle transition in a handwriting sequence: If a handwriting velocity sequences contains a sharp direction transition, the hand-writing sequence should be split into basic smooth strokes to facilitate character recognition.

Hand-writing movement of a character "A" can be split into five strokes from s0 to s4 marked in Fig. 5. Raw data returned by the sensor loyally record all handwriting movements;

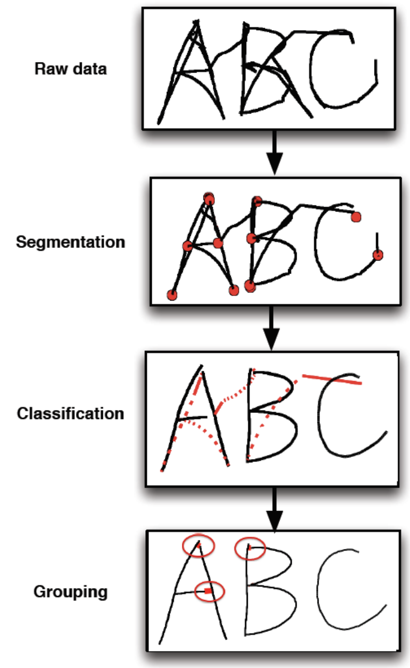
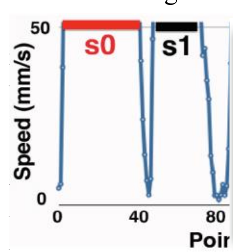
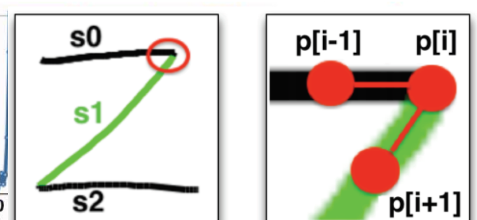


Fig. 4. Data processing algorithm flow

including s1 and s3 which are the strokes written in the air. Nevertheless, we can clearly identify each stroke by picking low XY-plane speed points as the start or end points. When writing a stroke, the XY-plane speed is much larger than 50mm/s, and the low writing speed distinguishes two strokes, as shown in Fig 5.

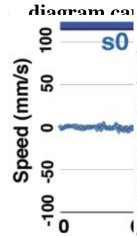




within an alphabet as well. In order to recognize sharp angle

Fig. 5. Cha... Segmented ... with the X... diagram ca...

Fig. 7. Character "Z" can be segmented into three straight strokes based on a sharp angle detection formula. Three consecutive points can form an angle if they are not in a line. The velocity change of character "Z" from s0 to s1 is more than 120 degree which can be



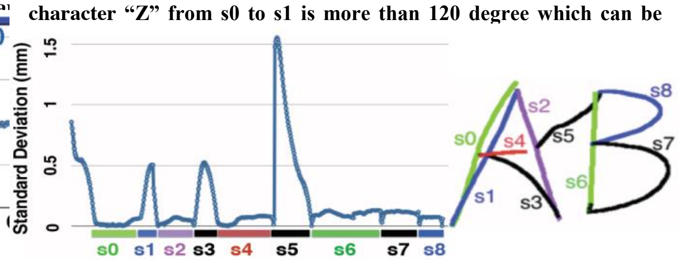


Fig. 6. Two observing ... stroke in K... bridge strok... trajectory i... Nevertheles... the writer t...

Fig. 8. s1, s3, and s5 are three strokes caused by off-board hand movements. In the depth standard deviation diagram, these strokes contain larger variance. On-board writing has much small depth deviation because a writing tool should be closely against to the board in board writing. From the Z direction standard deviation information, s1, s3, and s5 can be identified and removed from the segmented strokes list.

$$angle(i) = \cos^{-1} \frac{(p[i]-p[i-1])^2 + (p[i]-p[i+1])^2 - (p[i+1]-p[i-1])^2}{2 \times |p[i]-p[i-1]| \times |p[i]-p[i+1]|} \quad (1)$$

An example is given to split character "Z" into three straight strokes in Fig 7. This segmentation method is often used when a writer writes a new stroke without lifting his/her tools.

2) Classification of a Stroke

Classification step identifies whether a stroke is a trajectory in the air or on the board. Redundant strokes caused by off-board hand movement should be removed. After segmentation, a list of strokes is generated. However, this list contains not only on-board hand-writing, but also off-board hand movement. One intuitive solution is to check the distance of every sampled data point between the writing tool and the black/white board (commonly called depth information) when a lecturer is writing. Nevertheless, due to the limited accuracy of the infrared sensor, only a set of discontinued or missing strokes are returned. This intuitive solution may even break the segmented strokes. To overcome this problem, strokes are utilized as the fundamental unit for binary classification. Instead of measuring depth information of every single data point, moving window standard deviation of the depth information in a stroke is calculated. (Fig. 8) It has been observed that the depth changes of on-board writing are insignificant and standard deviation calculation can remove measurement error caused by limited sensor accuracy. On the contrary, the depth changes of off-board hand movement can be considerably prominent. Therefore, binary classification with training or simple threshold can effectively make reasonable judgments.

3) Grouping

Grouping step attempts to group a set of on-board strokes into letters if possible. This is an optional step for image-based notes but it is required for character recognition (and for the validation purpose of the proposed work). A distance between every two consecutive strokes is used to test if these two strokes belong to two characters. A recursive method (Grouping Algorithm) is used to split the two neighbor characters and connect all strokes within an alphabet after splitting.

C. Class Note Generation

Output from the data processing module will be images. If a grouping function is applied in advance followed by letter recognition, the output can be provided in a form of texts. This section describes a sequence of process to generate lecture notes from images or texts. Some terminologies are defined below.

1. Notes: Notes stand for all class notes Pages for a given class and date.
2. Pages: Pages stand for the contents written on a column of the black/white board. Pages should be assembled by a list of Lines and Masks with their geometric and chronically information.

```
function GROUPING(strokeList)
  Pick largest distance from s[i] to s[i+1]
  for j = 0 → i do
    Find rightmost point pR in s[j]
  end for
  for k = i + 1 → n do
    Find leftmost point pL in s[k]
  end for
  if pR.x ≤ pL.x then
    Split strokes into two groups Left(s[0]...s[i]) and
    Right(s[i+1]...s[n])
    GROUPING(Left)
    GROUPING(Right)
  else
    break
  end if
end function
```

Grouping Algorithm

3. Lines: Every time a writer finishes a row on the board, a snapshot image (called a Line) should be created with on-board writings.
4. Masks: Every time a writer does a modification on a created Line, a Mask will be generated to record new writing on the modified location.

Note generation is event-driven. The system keeps track of Lines creation and Pages creation events according to English writing convention, which is writing from left to right and from top to bottom. Mask creation events are triggered when a lecturer overrides or erases written notes occasionally.

New Lines and Pages are created based on the knowledge of English writing convention: the writing order is commonly from left to right and from top to bottom. A Page should contain all contents in a column of the black/white board (Fig. 9). Based on these assumptions, a new Line should be created when the writing apparently moves from the right side to the left side. Similar rule can be applied to track a new Page generation. If a writer apparently performs a movement from bottom of the board to the top level, a new Page should be generated accordingly.

1) Detect New Masks Action

Erasing or modifying written words and sentences are common in a lecture. Every modification should be reflected on the updated notes accordingly. The infrared sensor system is specially designed for tracing thin objects, such as fingers or chalks, so the eraser will be treated as two objects moving in parallel in the returned data stream. From experiments, we find that when an erasing motion happens, the sensor API

(a) (b)

Fig. 9. A common note writing convention: English sentence starts from left to right and from top to bottom. When a writer finishes a sentence or reaches the border, he/she should move back to the left and start a new line below the previous line. In addition, once a writer fills a column of the board, he/she should move from bottom to the top of the board and write a new page. Both right-to-left and bottom-to-top movements are treated as special events for notes

returns two points at a time. These points indicate the bottom of the eraser in every frame. This is a reasonable indication of note modification. In order to prevent false alarm of erasing event caused by off-board hand movement, erasing trajectories should also be passed through the proposed data processing algorithm to remove off-board movements. The on-board erasing movement can be useful to estimate the location of the Masks. If a lecturer fills new contents in the erased area, these contents should be filled into a new Mask rather than creating a new Line.

D. *Lecture Note Assembling*

A complete lecture note is composed by several Pages. Each Page is assembled by Lines and Masks. Fig. 10 provides an example to describe how to form a page based on Lines and Masks. Each Line and Mask contains its time stamp and geometric information on the black/white board. With time stamp, each page of the lecture note can be assembled under version control. Every update will trigger a Mask creation event. Masks can be overlaid on top of the existing Lines in order to display modifications. Moreover, by applying Mask in chronically order, a Page can contain multiple versions. Default option is the most up-to-date modifications applied.

E. *Off-line Video Segments Indexing*

This prototype can be integrated with video recordings. Thus, it helps students to search lecture contents from notes and video clips. Since every Page contains time stamp, once the taped video is synchronized with the proposed system, students can find the corresponding video clips when a lecture note is selected. This mechanism is also applicable in the reverse direction. Students can also find the corresponding lecture note Page based on video time. If character recognition is applied, students can further lookup their interested lecture notes or video clips by keywords.

V. NOTES QUALITY EVALUATION

In order to demonstrate that the proposed framework can generate human-readable notes, this section introduces three objective evaluation methods to evaluate the recording notes. Three methods are used to evaluate notes readability: OCR, OCR with stroke number heuristics, and primitive table methods. These heuristics and primitives are collected in the data processing stage as a side product. They are not used for note generation and assembling but are used for enhancing character recognition results. With sufficient accuracy in character recognition, keyword-based searching can be a useful feature. In addition to note-readability analysis, we also use these character recognition tools to analysis the robustness of the proposed framework under a variety of writing styles, such as different size of texts, tilted texts, and different spaces between two consecutive characters.

A. *Data Stream Processing Evaluation with Character Recognition Tools*

TABLE I
COMPARISON OF THREE NOTES TRANSCRIBING METHODS WHEN PARTIALLY PROCESSED DATA ARE USED AS INPUTS

	OCR[1]	OSN[2]	PT[3]
Raw	0%	0%	0%
Segmentation	0%	0%	26.9%
Classification	67.3%	80.8%	34.6%
Grouping	78.8%	90.4%	92.3%

[1]Optical Character Recognition
[2]OCR+Stroke Number
[3]Primitive Table

OCR software is commonly used for character recognition. Texts typed by keyboards can be identified reliably in most commercialized OCR software. Handwriting OCR is more challenging because human beings tend to have distinguishable writing styles and habits. Off-line recognizing characters captured from the black/white board is even harder. The prototype captures notes by tracking chalk or marker movements to resolve dirty board surface problem, but different artifacts are created because both on-board writing and off-board hand motion are loyally recorded. In fact, this redundant information causes much more interference for all character recognition tools. However, in a series of experiments conducted, we demonstrate the possibility of having reasonable recognition results with appropriate data processing and character recognition tools. Three character recognition tools are used: online available OCR software, stroke number heuristics, and primitive table.

1) *Traditional OCR Method*

The off-the-shelf handwriting OCR tools work well on typed characters and even handwriting texts on a clean paper. However, it has obvious difficulties in recognizing raw data collected by the infrared sensor system because of the redundant artifacts caused by off-board hand movements. Once the artifacts are removed after classification the accuracy is boosted to 70%. (Table. 1) The accuracy can be

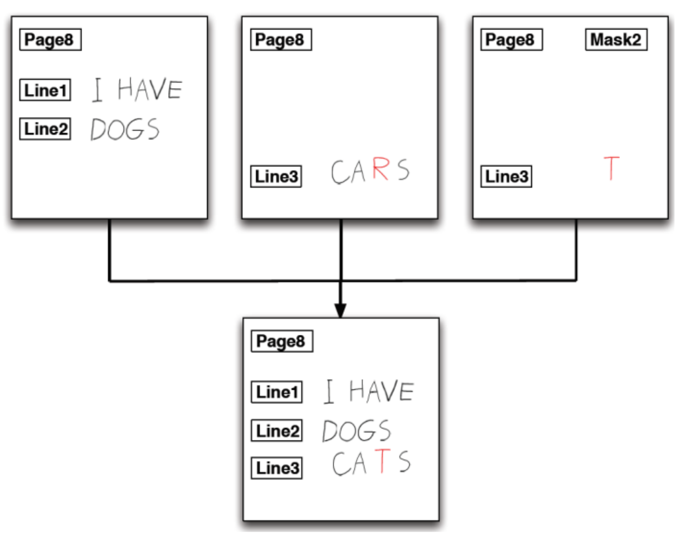


Fig. 10. There are two Lines (Line1 and Line2) already in Page8. Line3 is appended below according the geometric information contained in the Line. Mask2 is generated due to a detected modification event. It contains the geometric information regarding which Line it

even improved with grouping. However, the best accuracy that has been achieved is 80%. Table. 1 provides an example of data stream input “ABC”. Traditional OCR method will recognize these texts as “4BD”.

2) OCR and Stroke Number

TABLE II

PROBABILITY LIST OF TRADITIONAL OCR FOR CHARACTER “O” AND “Y”

O		Y	
Candidate	**Probability**	**Candidate**	**Probability**
D	0.328	T	0.203
O	0.319	V	0.201
0	0.277	Y	0.156
9	0.231	7	0.138

TABLE II

PROBABILITY LIST OF TRADITIONAL OCR FOR CHARACTER “C” AND “A”

C		A	
Candidate	**Probability**	**Candidate**	**Probability**
D	0.241	4	0.190
L	0.218	B	0.096
B	0.196	H	0.095
U	0.190	S	0.094
E	0.174	P	0.082
O	0.129	F	0.058
S	0.111	R	0.052
R	0.097	A	0.009
C	0.882	L	0.005

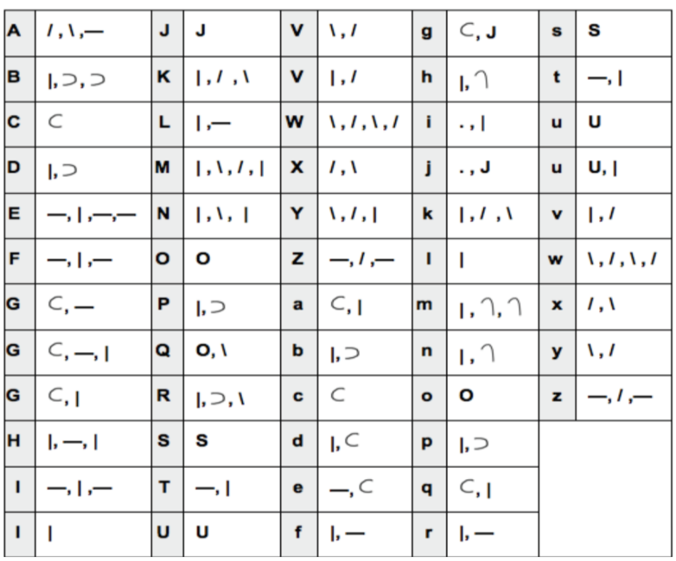


Fig. 11. Primitive Table: English Alphabets.

The handwriting OCR software we utilized generates probability lists for each input character. The probability list changes with different writing styles and input quality. Through filtering out inappropriate candidates existing in the list, it is possible to have better recognition results with reliable heuristics. Stroke number is chosen because it can be reliably determined in the data processing stage. This heuristics should simply improve the recognition rate rather than make it worse because if a character is on the OCR software, its stroke number should be agree with the strokes we sensed from the sensor. As a result, stroke number is a reasonable candidate to be exploited to walk through the OCR probability list to search for correct characters. Table. 2 provides two examples to pick up characters “O” and “Y” from the OCR probability list with the heuristics of stroke numbers.

Table. 3 shows two examples when OCR plus stroke number heuristics fails. It is because the ranks of “C” and “A” are very low in the list. Although we may be able to filter out the first candidate “4” in the third column because it is not an alphabet, but “F” in the list is written in three strokes will be selected next.

3) Primitive Table

A primitive table is shown in Fig. 11. It is designed to improve the recognition accuracy by enumerating all possible strokes of English alphabets by the knowledge of strokes. We decompose alphabets into several primitive type of strokes without ordering. For example, if a sequence of strokes are received as [/, n, -], the grammar table will return “A” as the recognition result. If strokes are [/, n, -], grammar table will also return “A”. In addition, a character can be consisted of multiple strokes combinations. Take G as an example, there are three ways to write G (Fig. 12). Whenever a new writing style is discover, the strokes combinations can be inserted into the primitive table. Nevertheless there are some of characters which are hard to categorized, such as O, S, J, and U. Hence, they are treated as basic primitives as well.

B. Evaluate Tilted Characters Identification

Writer can write characters on board with an arbitrary angle, but in general he/she may tilt within 5 to 10 degree and tile with counter-clock direction. This phenomenon may not significantly affect the data processing step because the human behaviors in writing strokes do not depend on a tilted writing angle. However, tilted characters tend to increase recognition error for common handwriting recognition system. To evaluate this situation, we intentionally rotate a character ”K” with 5 and 10 degree counter-clock-wise (Fig. 12). The rotated images are used as simulated inputs for traditional OCR, OCR with stroke number heuristics, and primitive table methods. The accuracy drops apparently for OCR based methods as shown in Table. 4.

Fig. 12. A variant of handwritten styles of character “G”. These variants should be recorded in the primitive table. Three tilted character “K” are simulated by rotating original images with 5 and 10 degree. We exploit image rotation rather than writing tilted characters on board because we want to quantify the rotated angles.

TABLE IV

COMPARISON OF THREE NOTES TRANSCRIBING METHODS WHEN TILTED CHARACTERS ARE USED AS INPUTS

	OCR[1]	**OSN[2]**	**PT[3]**
Original	78.8%	90.4%	92.3%
5° Tilted	76.9%	84.6%	92.3%
10° Tilted	65.4%	80.7%	92.3%

[1]Optical Character Recognition
[2]OCR+Stroke Number
[3]Primitive Table

C. Evaluate Identification Rate Changes Caused by the Distance between Characters

In most of our experiments, we leave 10mm (25% of average letter width) spacing between characters. When there are larger space between characters, results from the grouping function are better. As writer has to move further between two letters, handwriting behaviors in segmentation (slow start/ end and big jump) and classification (Z-direction variance) are clearer in the waveforms. On the other hand, if there are merely 5mm spacing between characters, it will dramatically increase difficulties in grouping and severely affects the effect of the primitive table method.

VI. Discussion

Four difficulties and design tradeoffs are brought into discussion: Sensor Range, OCR Accuracy, Modification Event Detection, and Synchronization problems.

A. Sensor Range

The sensor system used in the design is still in developer version; hence, the infrared sensor's range is very limited. One sensor cannot fully cover enough range for a full sentence. As a result, multiple sensors are require to be used in the experiments with careful arrangement to avoid overlapping to their view fields. This problem should be alleviated when commercialized products or customized hardware design are available in the future.

B. OCR Accuracy Limitation

There are some pairs of characters share with the same strokes primitive combinations; for example character "D" and "P". In this case, the grammar table will return both candidates D and P and we can only use traditional OCR probability list as the tie breaker. To compensate the consequence of the incorrect recognition, dictionary lookup may be necessary.

C. Modification Event Detection

Erasing action is a signal for modification. Erasing events can be detected when both sides of eraser are detected. In this case, the sensor will return two points per frame which indicate the bottom of the eraser. However, the two points per frame pattern can be simulated if two fingers or two chalks slides against the board. Although this is not common in most of lecture sensors, we believe some cheating prevention mechanism should be included in deployment.

D. Synchronizing Between the Recorded Notes and Video Clips

If the note transcription system and a video recording system do not launch at the same time, students cannot directly use the recorded time stamps to find the corresponding section in video. We manually keep both system synchronized in the experiments but we believe that using the recorded information of notes and video clips can achieve context based synchronization with keywords in the recorded notes and the video clips.

VII. Conclusion

This paper attempts to show a feasible prototype of an electronic lecture notes generator on the aspects of sensing and computing. Compare with other practical and theoretical solutions, our design reduces the complexity of system infrastructure while enhances user experience. Lecturers do not need to worry about adapting new sensor technology and changing their used teaching habits. All lecture notes are inconspicuously recorded because the proposed system can be seamlessly integrated with existing class room facilities: white/black board, markers and chalks. It can automatically generate and assemble class notes with our proposed data processing and software architecture design. With character recognition, created notes can integrate with lecture tapes and allow students bidirectional review course materials with bidirectional search.

References

[1] M. Onishi, M. Izumi, and K. Fukunaga, "Blackboard segmentation using video image of lecture and its applications," *in Proc. International Conference on Pattern Recognition*, Barcelona, Spain, pp. 615-618, Sep. 2000.

[2] Z. Zhang and L. He, "Note-taking with a camera: whiteboard scanning and image enhancement," *in Proc. International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Quebec, Canada, pp. 533-536, May. 2004.

[3] Z. Zhang, "Computer Vision Technologies for Remote Collaboration Using Physical Whiteboards, Projectors, and Cameras," *in Proc. Computer Vision for Intelligent Environment*, Washington DC, USA, pp. 109-122, Jan. 2005.

[4] T. Platz, C. Thurau, and G. A. Fink, "Camera-based whiteboard reading: new approaches to a challenging task," *in Proc. International Conference on Frontiers in Handwriting Recognition*, Montreal, Quebec, Canada, pp. 385-390, Aug. 2008.

[5] T. Chu and C. Su, "A Kinect-based handwriting digit recognition for TV remote controller," *in Proc. International Symposium on Intelligent Signal Processing and Communications*, Tamsui, New Taipei City, Taiwan, pp. 414-419, Nov. 2012.

[6] S. N. Cheong, W. J. Yap, R. Logeswaran, and I. Chai, "Design and development of Kinect-based technology-enhanced teaching classroom," *In Embedded and Multimedia Computing Technology and Service*, Gwangju, Korea, pp. 179-186, Sep. 2012

[7] M. Camplani, L. Salgado, and R. Camplani, "Low-cost efficient interactive whiteboard," *in Proc. IEEE International Conference on Consumer Electronics*, pp. 686-687, Singapore, Nov 2012.

[8] C.A. Wingrave, B. Williamson, D. P. Varcholik, J. Rose, A. Miller, E. Charbonneau, J. Bott, and J. J LaViola, "The Wiimote and beyond: spatially convenient devices for 3D user interfaces," *Computer Graphics and Applications, IEEE*., vol. 30, no. 2, pp. 71-85, Apr. 2010.

[9] L. Gallo, G. De Pietro, and I. Marra, "3D interaction with volumetric medical data: experiencing the Wiimote," *in Proc. IEEE International Conference on Ambient Media and Systems*, Quebec, Canada, pp. 14:1-14:6, Nov. 2008.

[10] A. Holzinger, S. Softic, C. Stickel, M. Ebner, and M. Debevc, "Intuitive e-teaching by using combined HCI devices: experience with Wiimote applications," *in Proc. Universal Access in Human-Computer Interaction, Applications, and Services*, San Diego, CA, USA, pp. 44-52, Jul. 2009.

[11] J. Lee, S. Hudson, P. Huert, and P. Dietz, "Hybrid infrared and visible light projection for location tracking*," in Proc. ACM symposium on User interface software and technology*, Newport, Rhode Island, USA, pp. 57-60, Oct. 2007.

[12] K. K. M. Sugimoto and N. Ezaki, "Ee-class – interactive lecture support system using wii remote*," in Proc. Computers and Advanced Technology in Education*, Lahaina, Maui, USA, pp. 709, Aug. 2010.

[13] S. Agrawal, I. Constandache, S. Gaonkar, R. R. Choudhury, K. Caves, and F. DeRuyter, "Using mobile phones to write in air*," In Proc. International Conference on Mobile Systems, Applications, and Services*, Maryland, USA, pp. 15-28, June. 2011.

[14] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Trans. Pattern Analysis and Machine Intelligence.*, vol. 22, no. 1, pp. 63-84, 2000.

[15] S. Choi and S. Lee, "3D stroke reconstruction and cursive script recognition with magnetometer-aided inertial measurement unit," *IEEE Trans. Consumer Electronics.*, vol. 58, no. 2, pp. 661-669, 2012.

[16] R. Aoki, M. Ihara, A. Maeda, M. Kobayashi, and S. Kagami, "Expanding kinds of gestures for hierarchical menu selection by universal gesture interface," *IEEE Trans. Consumer Electronics.*, vol. 57, no. 2, pp. 731-737, 2011.

[17] B. Lee, W. Bang, J.D.K. Kim, and C. Kim "Orientation estimation in mobile virtual environments with inertial sensors," *IEEE Trans. Consumer Electronics.*, vol. 57, no. 2, pp. 802-810, 2011.

[18] Y. Han, "A low-cost visual motion data glove as an input device to interpret human hand gestures," *IEEE Trans. Consumer Electronics.*, vol. 56, no. 2, pp. 501-509, 2010.

[19] D. Lee and Y. Lee, "Framework for vision-based sensory games using motion estimation and collision responses," *IEEE Trans. Consumer Electronics.*, vol. 56, no. 3, pp. 1356-1363, 2010.

[20] Y. Jang and K.H. Lee, "Depth video based human model reconstruction resolving self-occlusion," *IEEE Trans. Consumer Electronics.*, vol. 56, no. 3, pp. 1933-1941, 2010.

[21] D. Lee, K. Park, and Y. Park, "Collision detection and response method for markerless camera-based games using motion boundary estimation," *IEEE Trans. Consumer Electronics.*, vol. 56, no. 4, pp. 2178-2184, 2010.