

# Technical Disclosure Commons

---

Defensive Publications Series

---

April 19, 2017

## Direct Handwriting Editing

Megan Austin Renshaw

Wei Li

Follow this and additional works at: [http://www.tdcommons.org/dpubs\\_series](http://www.tdcommons.org/dpubs_series)

---

### Recommended Citation

Renshaw, Megan Austin and Li, Wei, "Direct Handwriting Editing", Technical Disclosure Commons, (April 19, 2017)  
[http://www.tdcommons.org/dpubs\\_series/469](http://www.tdcommons.org/dpubs_series/469)



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

## Direct Handwriting Editing

### Abstract

This publication describes techniques for enabling handwriting and text editing on touch-sensitive surfaces (e.g., touchscreens, touchpads, interactive whiteboards) without the use of a physical or virtual keyboard. These techniques enable users to perform free-form handwriting gestures anywhere on a touch-sensitive surface, and also to perform editing gestures that modify previously created text without a mode change. A computing system (e.g. mobile phone, tablet computer, personal digital assistance, laptop computer, interactive whiteboard) may analyze the shapes of user gestures input at a touch-sensitive surface, as well as the relative positions of these gestures compared to existing displayed text, and determine whether the user intended to edit existing text or instead to write new text. If the system determines that the user intended to edit existing text, the system applies an appropriate editing operation to the text (e.g., text selection, text deletion/replacement, text insertion/movement, text merging, text separation). Otherwise, the computing system interprets the user's strokes as regular handwriting and interprets the strokes for new text entry using a handwriting recognizer.

The techniques described in this publication may provide one or more advantages. For example, the described system does not require the use of a physical or even virtual keyboard. Because the system does not provide a popup virtual interface for text entry or editing operations, the system increases the amount of available screen space for text entry or editing operations. A user can write and/or edit anywhere on the touch-sensitive surface (e.g., directly on target text or at a target location), instead of having to write or edit in a dedicated region of the screen, and the user does not need to move or otherwise manipulate a cursor before performing editing operations. The system is able to create a new text label with a desired size, position, and/or orientation as soon as user finishes writing. In addition, multiple different users may be able to simultaneously enter new

text or edit existing text on a large touch-sensitive surface, such as an interactive whiteboard.

### **Technical details**

To achieve direct handwriting editing functionality, the system determines whether inputted user strokes are to be interpreted as handwriting gestures for text entry or as editing gestures. As shown in Figure 1 below, the system initially assumes that one or more inputted strokes corresponds to an editing gesture. The system performs gesture recognition to determine if the properties of the stroke(s), including the shape and relative position of the stroke(s) to existing text, satisfy the criteria of a recognized editing gesture (e.g., criteria for text selection, text deletion/replacement, text insertion/movement, text merging, or text separation gesture). If the properties of the stroke(s) satisfy the criteria of a recognized editing gesture, the system performs the corresponding editing operation. Otherwise, the stroke(s) are interpreted as a handwriting gesture and are fed to a handwriting recognizer for entry of new text. In this way, the system may use an existing handwriting recognizer for interpreting handwriting gesture for text entry, and may use an additional module that performs gesture detection to determine if inputted strokes correspond instead to an editing gesture.

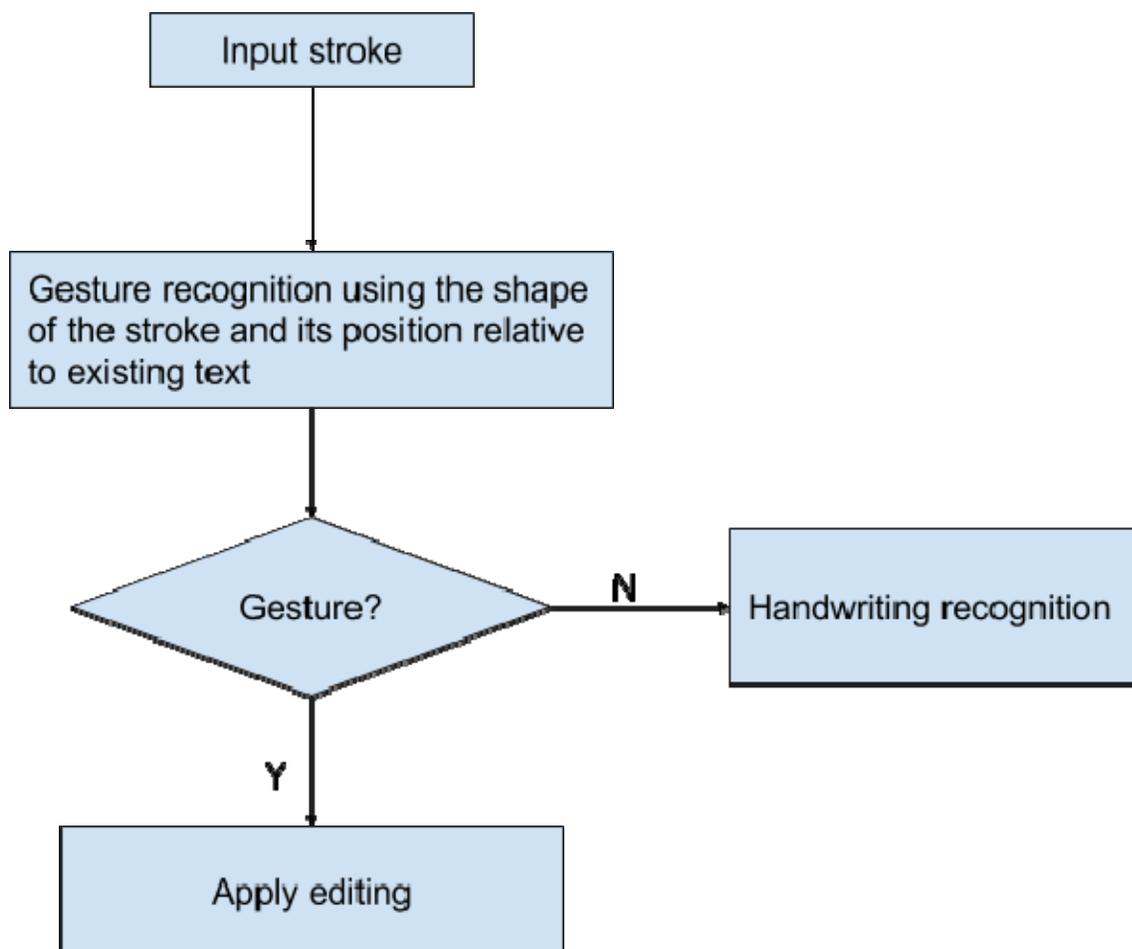


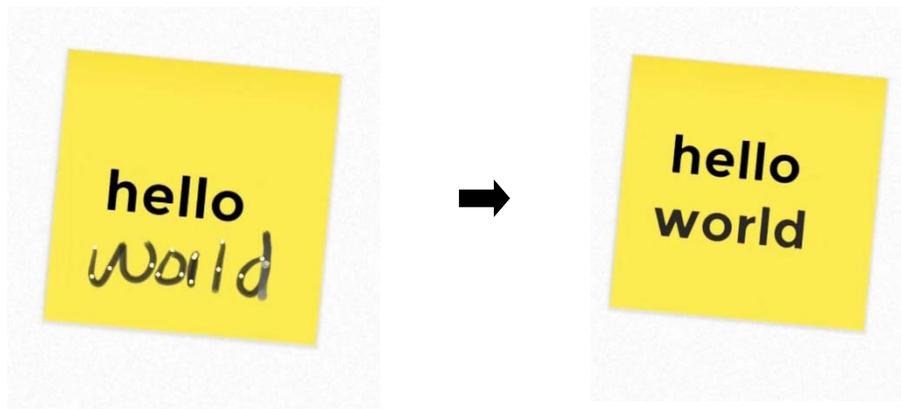
Figure 1

### Text Creation via Handwriting Recognition

A user can create new text by directly writing one or more strokes on any empty space of a screen or other area (e.g., area of an interactive whiteboard) having a touch-sensitive surface. If the properties of the strokes, such as the shape of the strokes or their relative position to existing text, does not satisfy the criteria of a recognized editing gesture, the system processes the strokes as handwriting gestures using a handwriting recognizer for text entry. The system may insert the new text at a position and an orientation that corresponds to the position and orientation of the inputted strokes. The system may also use a default size and/or font for inserted text. The default

size and/or font settings are fully customizable by a user.

A user may also input one or more strokes to enter text on a note that contains existing text, such as shown in Figure 2 below, where a user has inputted strokes, corresponding to the word “World,” below the existing text “hello.” Once a note is created, a user can simply input one or more strokes that are recognized as a handwriting gesture, causing the handwriting recognizer to enter text directly onto the note. If the recognized handwriting is fully contained in the boundary of the note, the corresponding text is added either to the beginning (e.g., top) or end (e.g., bottom) of the note, depending on which is closer to the center point of the new text.



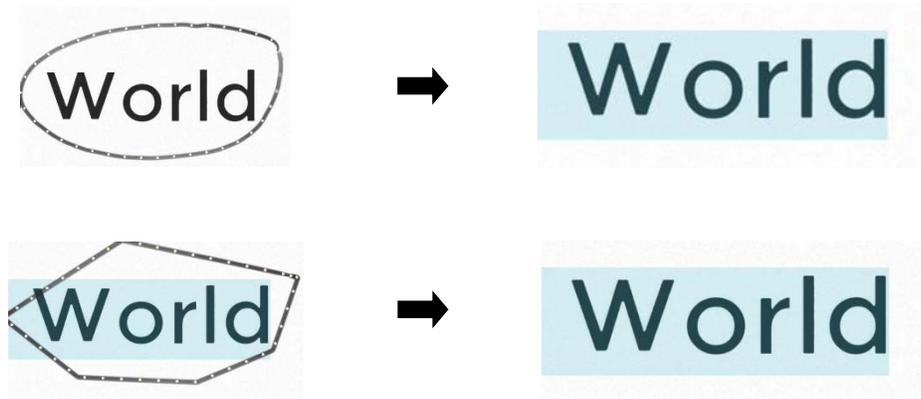
**Figure 2**

## **Editing Operations**

### **(a) Text Selection**

A user may select existing text by performing a selection gesture to create a closed or partially closed shape around the existing text. The created shape may be elliptical in shape, or it may alternatively include a finite number of polylines. Figure 3 below shows two different examples of such selection gestures to select the word “World.” The system may impose a limit on the number of polylines to decrease a probability that the user will perform selection with something other than a polygon. The selection gesture has its midpoint inside a text target that includes one or more characters, and the bounds of the selection gesture should not be too far away

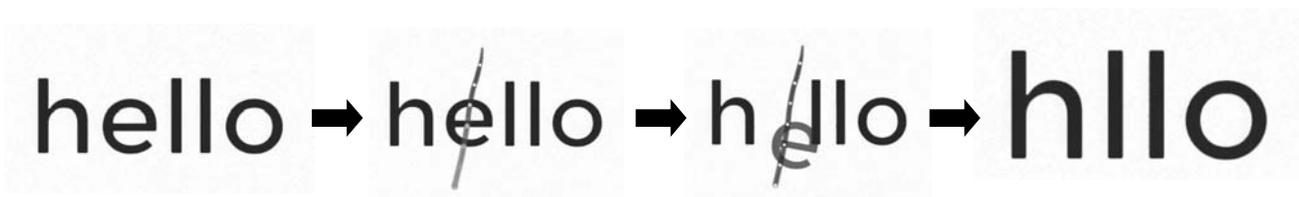
from the text target. Selection is successful if the contour defined by the vertices of the polyline shape, or select points on the ellipse, fully or partially enclose one or more characters. The selection gesture may only resort to partial enclosure if no characters are fully enclosed.



**Figure 3**

### **(b) Deletion and Text Replacement**

A user may delete characters, words, or even a full line of text by performing a deletion gesture to draw a slash through the text to be deleted. To determine if a slash stroke is valid, the system determines if a midpoint of the stroke intersects a text element (e.g., character(s), word(s), line) and crosses at least one character. In the example shown in Figure 4 below, a user inputs a slash stroke to delete the “e” character from the word “hello.”



**Figure 4**

To delete or remove a space between characters, a user may input the slash stroke in the white space between the characters. Additionally, in order to delete an entire word, rather than drawing a slash stroke through the entire word, a user may alternatively perform a long-press gesture on the word and then quickly (e.g., within a defined duration of time) perform a drag or throw gesture to an edge of the touch-sensitive surface to delete the word.

To perform text deletion and replacement as a combined operation, the user may write new text outside of a text target and then quickly (e.g., within a defined duration of time) draw a slash stroke through the text to be replaced. As shown in the example of Figure 5 below, the user may handwrite the word “World” below the existing text “hello Megan.” The user may then quickly perform a deletion gesture by drawing a slash stroke through the word “Megan.” In response, the system deletes the word “Megan” and replaces it with the word “World.”



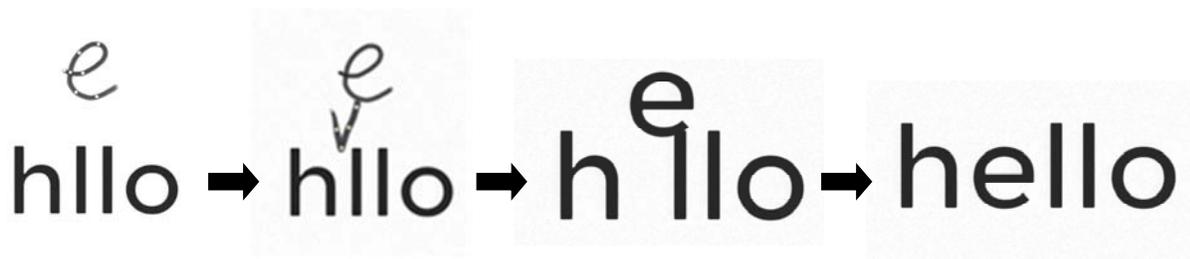
**Figure 5**

### **(c) Insertion and Text Movement**

A user may insert text or white space by performing a wedge gesture, and the user may also move text by performing a drag-and-drop gesture. A wedge gesture is valid if the tip of the wedge is sufficiently close to a border of a text element, and if the ends of the wedge are not too far away from the text element. The wedge may either be a downwardly pointing wedge or an upwardly pointing wedge.

To insert a space character, the user may simply perform a wedge gesture at a desired location. To insert text, the user may write text outside of a text area and then quickly use the wedge gesture to insert the text, as shown in Figure 6 below. In the example of Figure 6, the user may write the character “e” above the existing text “hlllo” and may then quickly perform a wedge

gesture to input a downwardly pointing wedge between the “h” character and the first “l” character. In response, the system inserts the character “e” at the location of the wedge to display the word “hello.”



**Figure 6**

A user is also able to move or rearrange one or more characters of text using the selection gesture in combination with the wedge gesture. For example, in Figure 7 below, the user may first perform a selection gesture, as described above, to select the word “World.” The user may then perform a wedge gesture to input a downwardly pointing wedge before the “h” character in the word “hello.” In response, the system moves the word “World” and inserts it before the word “hello.”

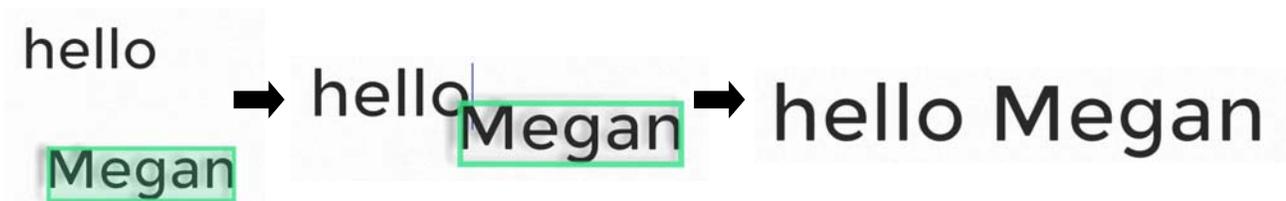


**Figure 7**

When moving multiple characters in the way demonstrated in Figure 7, the system may automatically insert a space either before or after the group of characters that have just been moved and inserted. However, if the user has selected only a single character for a move operation, the system may not automatically insert a space before or after the single character upon moving it to its new location.

#### (d) Merging Text

In order to merge text, a user may first perform a selection gesture to select a text element, and may then perform a drag gesture to move the text element such that it hovers over at least a portion of a second text element. The system may display a cursor when either the top or bottom corner (e.g., top or bottom left corner) of the selected text element intersects the second text element. In the example of Figure 8 shown below, after selecting the text element “Megan,” the user performs a drag gesture to move this text element such that it hovers over at least a portion of the “hello” text element. The system may output a cursor at a position that is closest to the midpoint, on the left edge of the selected text element “Megan,” as shown in Figure 8. The system then merges the two text elements to output the phrase “hello Megan.”



**Figure 8**

Similar to the text movement operation described above, when merging multiple text elements that each include multiple characters, the system may automatically insert a space between the merged text elements. However, if one of the text elements includes only a single character, the system may not automatically insert a space between the merged text elements.

#### (e) Separating Text

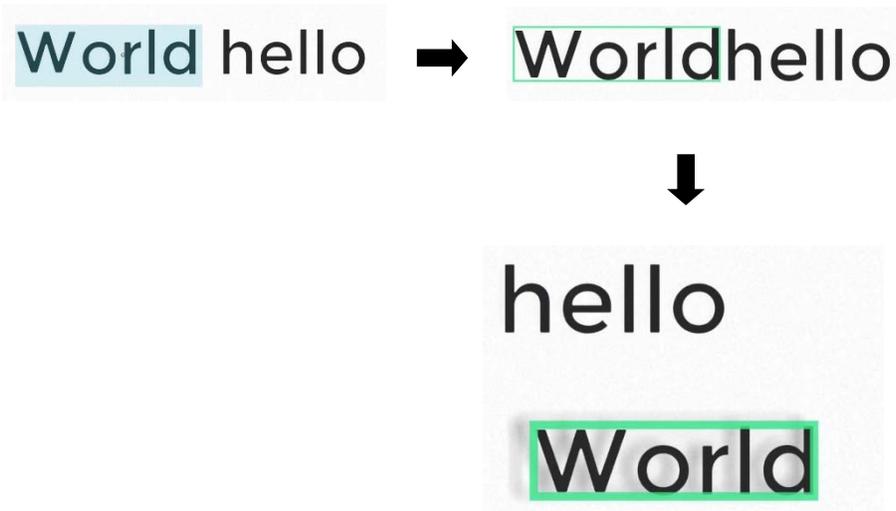
A user may separate text in a text element by performing a selection gesture to select a portion of the text, and then performing a long-press gesture. In response, the system will remove the selected text from the current text element, and will create a new text element for the selected text (e.g., in a label or note), as shown below in Figure 9. In this figure, the user first performs a

selection gesture to select the text “Megan” in the text element “hello Megan.” The user then performs a long-press gesture on the text “Megan.” In response, the system separates the text “Megan” from the current label or note and creates a new label or note with a text element “Megan,” as shown. After the separation operation is complete, a first label or note contains the text element “hello,” and a second and distinct label or note contains the text element “Megan.” When the individual, separated text elements each contain more than one character, the system may automatically insert a space between the separated text elements.



**Figure 9**

In certain cases, the user may perform selection and long-press gestures to separate out text that is not located at the end of the label or note that contains the original text element. For example, as shown in the example of Figure 10 below, the user may first perform a selection gesture to select the text “World” in the text element “World hello,” and may then perform a long-press gesture, causing the system to separate the text “World” from the text “hello.” In this case, the system may re-position the separated text elements such that the newly separated text element “World” is positioned below the new text element “hello,” as shown in Figure 10. In this example, the system may also shift the text element “hello” to the left, above the word “World.”



**Figure 10**

## **Conclusion**

The techniques described herein enable both free-form handwriting and text editing on touch-sensitive surfaces without the use of a physical or virtual keyboard. Through analysis of shapes and relative positions of input gestures compared to existing displayed text, a computing system is able to determine whether the user intended to edit existing text or instead to write new text. If the system determines that the user intended to edit existing text, the system applies an appropriate editing operation. Otherwise, the computing system interprets the user's strokes as regular handwriting and interprets the strokes for new text entry. A user can write and/or edit anywhere on the touch-sensitive surface, instead of having to write or edit in a dedicated region of a screen, and the user does not need to manipulate a cursor before performing editing operations.