September 2020

Automatic Grouping and Categorization of Objects in Drawing or Inking Apps

Claire Tauziet

Robert Mickle

Thomas Buckley

Automatic Grouping and Categorization of Objects in Drawing or Inking Apps

ABSTRACT

Drawing, note-taking, or inking apps that operate on electronic canvases do not differentiate between the various types of objects, e.g., paragraphs, lists, sketches, tables, doodles, etc., that an author can create on a screen. Consequently, it is difficult in electronic canvases for an author to perform operations commonly available in word processors, e.g., insert a sketch, move a block of text up or down, merge or separate sketches, etc. This disclosure describes techniques that automatically organize and group strokes drawn by a user on an electronic canvas into blocks of content, e.g., objects such as text-blocks, lists, sketches, tables, doodles, etc. The techniques thereby handle implicit structuring in an interactive inking or drawing app and provide the user the ease equivalent to writing on physical paper and the flexibility of electronically editing and restructuring diverse content.

KEYWORDS

- Inking app
- Note-taking app
- Drawing app
- Sketching app
- Touchscreen
- Stylus
- Bounding box
- Object categorization

BACKGROUND

An author of a physical paper document is not constrained by the structure of the document. Indeed, the author defines the structure implicitly through her writing. Thus, a page can include objects such as paragraphs, lists, diagrams, tables, doodles, etc. in any configuration decided by the author. However, current drawing, note-taking, or inking apps that operate on electronic canvases do not differentiate between the various types of objects that an author can create on a screen. Consequently, it is difficult, in mouse, stylus, or finger-drawn electronic canvases, for an author to perform operations commonly available in word processors, e.g., insert a sketch, move a block of text up or down, add a row to a table, merge or separate sketches, etc.

Stylus-based note-taking and drawing apps today at best offer handwriting recognition, auto-growing canvases, a set of brush tools, etc. However, the experience of writing, editing, and restructuring content using such apps is inflexible and unintuitive.

DESCRIPTION

This disclosure describes techniques that automatically organize and group strokes drawn by a user on an electronic canvas into blocks of content, e.g., objects such as text-blocks, lists, sketches, tables, doodles, etc. The techniques thereby handle implicit structuring in an interactive inking or drawing app, and offer to the user the ease of writing on physical paper and the flexibility of electronically editing and restructuring diverse content. The structure of the document is automatically interpreted and adapted, e.g., changed as necessary, as the user develops the document.

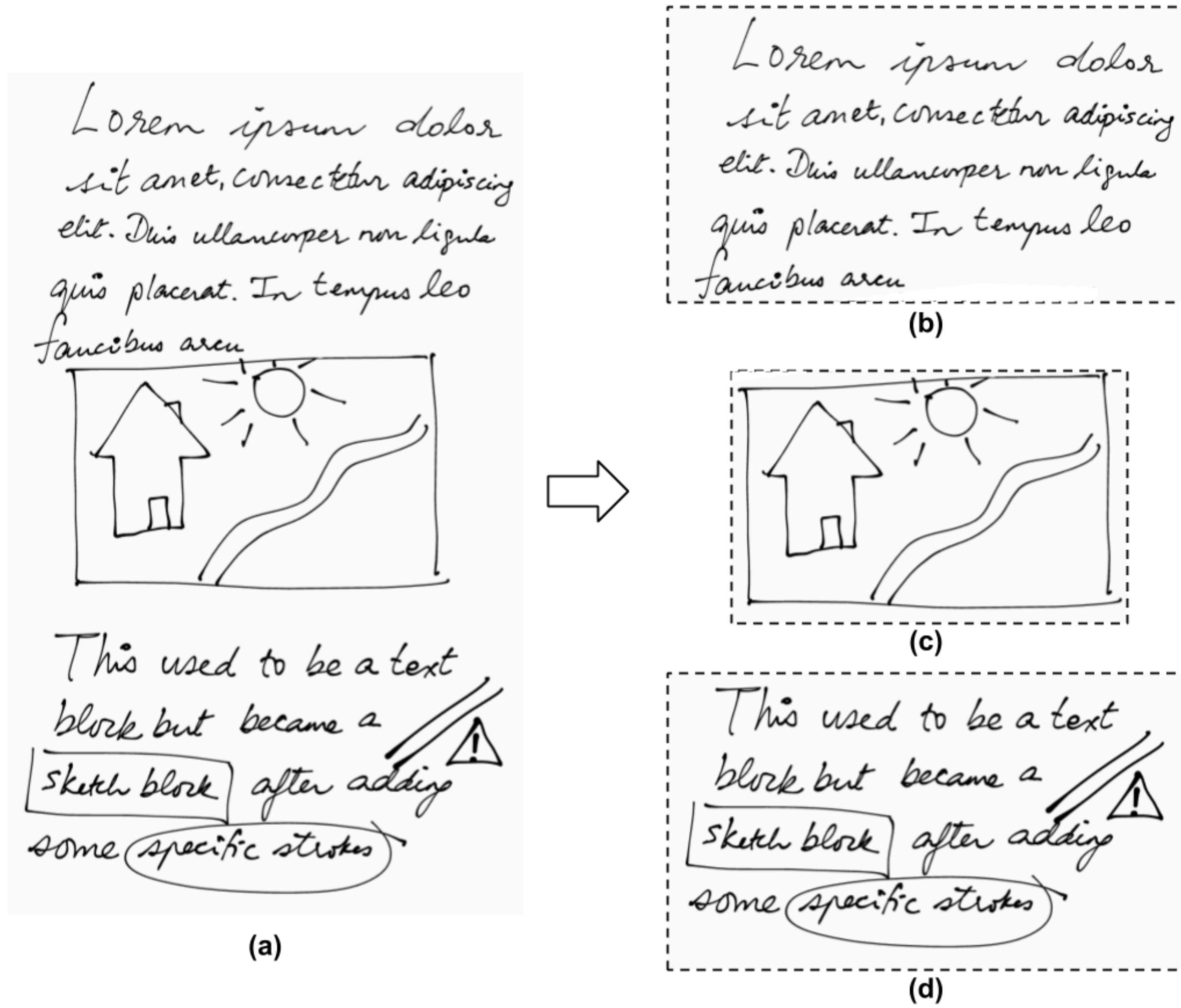

Fig. 1: Typecasting a raw set of strokes into blocks of content: (a) A set of raw strokes on an electronic canvas; (b) Automatic categorization of a subset of strokes as a text block; (c) Automatic categorization of another subset of strokes as a sketch block; (d) An initial categorization as a text block automatically re-categorized as a sketch block upon the addition of more strokes.

Fig. 1 illustrates the typecasting of a raw set of strokes into blocks of content. Fig. 1(a) illustrates a set of raw strokes on an electronic canvas. By default, the user can create any kind of stroke on the electronic canvas. Fig. 1(b) illustrates the automatic categorization of a subset of strokes as a text block. Fig. 1(c) illustrates the automatic categorization of another subset of strokes as a sketch block. In Fig. 1(d), an initial categorization of a subset of strokes as a text block is automatically re-categorized as a sketch block upon the addition of more strokes.

Although blocks of content are automatically and in real-time typecast (or re-typecast), e.g., as text-blocks, lists, sketches, tables, doodles, etc., the user has the option to declare or change the type of a given block. Grouping and categorization of strokes into types can be done automatically, e.g., using machine learning models. Every piece of content within the document, including stroke, image, text, widget, etc. is classified as part of a block. This organization of elements into blocks enables the user to keep the document organized and get appropriate tools for different types of content.

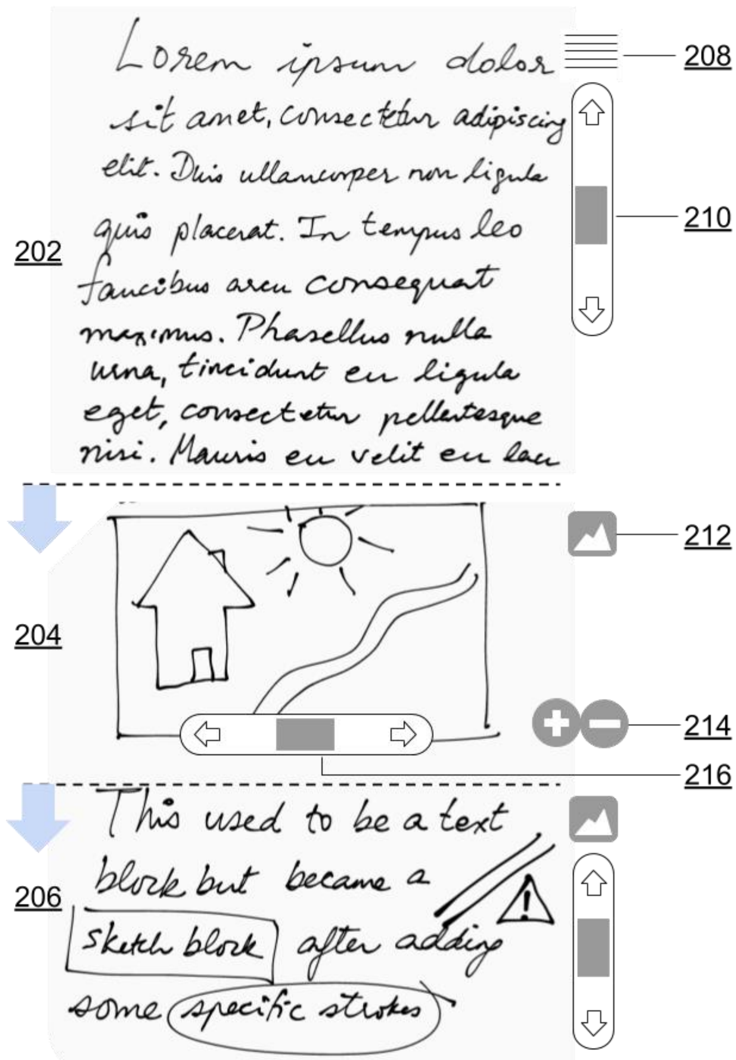

Fig. 2: Editing and re-structuring content

Fig. 2 illustrates editing and restructuring content, per the techniques of this disclosure. As illustrated, typecasting a block, e.g., as a text-block, enables users to easily edit its content independent of other typecast blocks. Manipulating the content of a block automatically causes a reflow of its content, an increase or decrease of the block size, or movement of other blocks around it as necessary. For example, adding text to block 202 automatically causes the downward movement of other blocks 204 and 206.

Fig. 2 also illustrates user-interface elements (208-216), e.g., identifying icons, insert ranges, drag handles, etc., that can be automatically provided, per the disclosed techniques, to indicate to the user the type of identified block, and to enable the user to control or edit it. Typically, the start of inking by a user automatically detects or creates a new block. With the addition of more strokes by the user, the block is identified and typecast, and the type of block communicated to the user along with the bounds of the block. Starting to write or draw outside of an existing block creates a new one.

Users can interact with different types of blocks in different ways, e.g., text blocks can be edited with text-appropriate gestures for deletion, insertion, etc.; sketches can be appended with additional strokes, zoomed in or out, etc. The identifying icon that displays the block type enables the user to know what behavior to expect from the block.

As mentioned previously, a block can grow vertically or horizontally as the user fills in additional material within the block. The height of a text block is thus defined as the user writes. A new block includes an empty line at the bottom to let users append text to a text block or continue drawing on an existing sketch block. When the user starts writing on the empty line at the bottom, a new empty line is added to the text block. The height of a sketch block is similarly defined as the user is drawing. To expand its height, users can either use the expand-control of the user interface or continue to draw on blank space.

If the user writes something that isn't text, e.g., a doodle, inside a text block, the techniques attempt to preserve the user's strokes, e.g., avoid stroke-rejection. A small sketch (or unrecognized handwriting) within a text block is treated as an inline sketch. Thus, small drawings, e.g., stick figures, emojis, etc., reflow naturally along with the text. For cases where the sketch cannot be treated as inline, e.g., if the sketch is large or complex enough to warrant its own block, the block is automatically converted to a sketch block. As mentioned before, an incorrectly categorized block, e.g., a text-block incorrectly typecast as a sketch-block or vice-versa, can be manually re-typecast by the user.

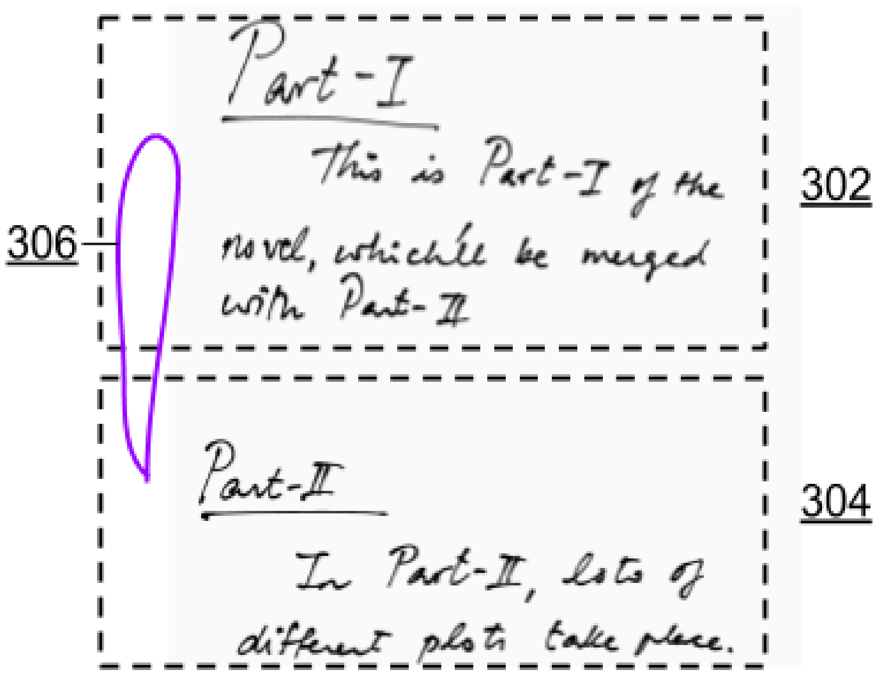

Fig. 3: Merging blocks

Fig. 3 illustrates merging of two blocks, per the described techniques. While block boundaries are displayed, a user can simply draw some strokes (306) across multiple blocks (302 and 304 in the example of Fig. 3) to merge them. Alternatively, the user can merge blocks by selecting them and providing a merge command.

Upon the movement of a block vertically or horizontally, space is automatically inserted between the block that is being moved and the other blocks. The movement of blocks can happen automatically, e.g., when additional material is appended to a block by the user, or manually, e.g., when the user selects a block and moves or otherwise reorganizes it. A block can be moved, e.g., to the top of the page; until it hits the immediate previous block; over the previous block(s) (which would cause the previous blocks to automatically move down); etc.

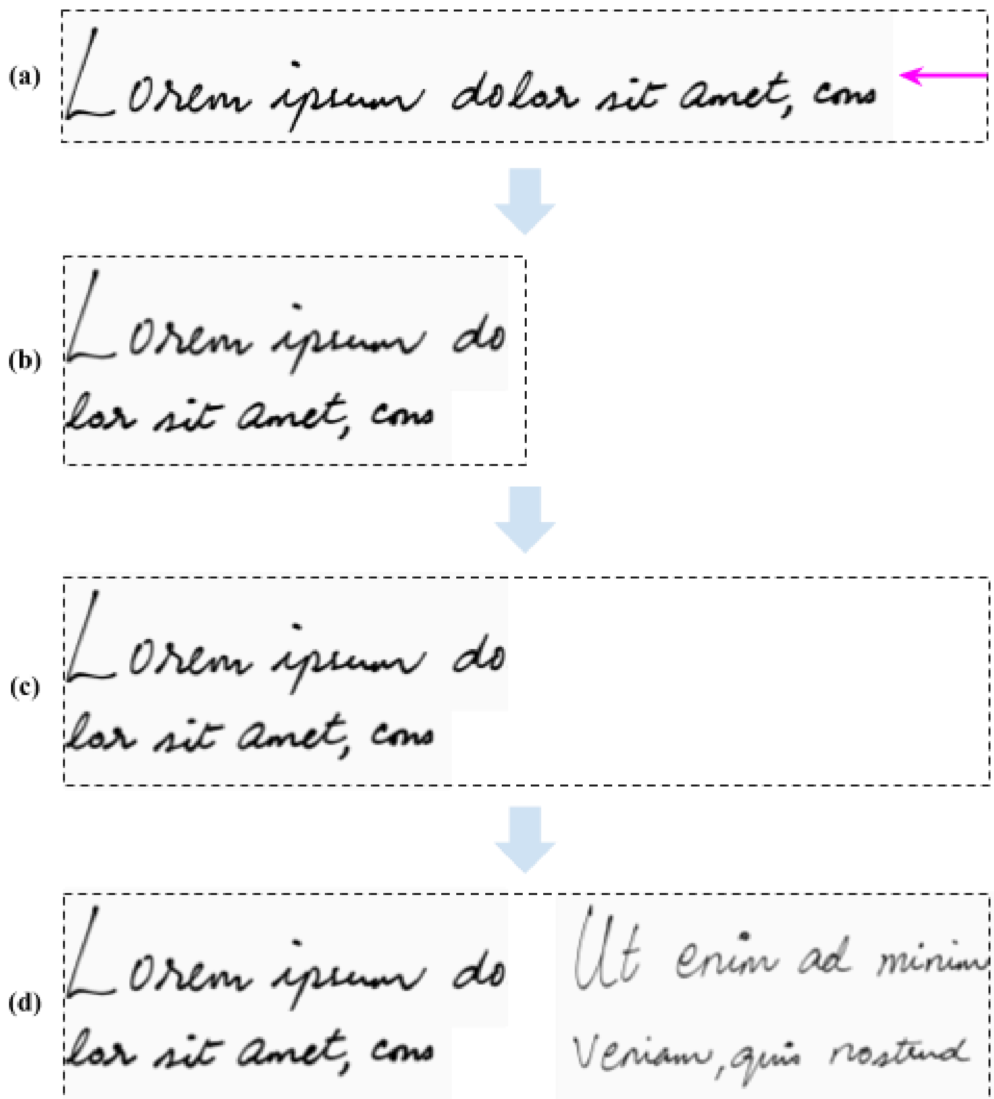

Fig. 4: Various operation on blocks (a) A text block (b) Automatic reflow of text (c) Creation of additional space within a text block (d) Two-column text format

Fig. 4 illustrates various operations that can be performed on blocks, e.g., text blocks. The size of a text block shown in Fig. 4(a) can be expanded or contracted by the user. If the user contracts a text block, it causes an automatic reflow of its text, as shown in Fig. 4(b). Space can be created at the side of text by expanding the text block, as shown in Fig. 4(c). In the newly-created space, additional text can be written, automatically enabling a multi-column format, as shown in Fig. 4(d). Sketch blocks can similarly be expanded or contracted, have new space created by their side, be auto-formatted in multi-column format, etc.

Fig. 5 illustrates the treatment of strokes outside an already-defined block, per the techniques of this disclosure. Fig. 5(a) illustrates two text blocks (502a) and (504a), and a sketch block (506a). The user extends her sketch (508) beyond the sketch block. The parts of the sketch beyond the sketch block may be marked as such or be temporarily masked, e.g., invisible. As shown in Fig. 5(b), the user extends the sketch block to encompass the entire sketch (506b); in doing so, the existing blocks (502b, 504b) dynamically re-shape themselves, their content re-flowing as necessary.

Users can select or multi-select a block (or a part of a block, e.g., a section of text, a stroke, a set of strokes, etc.) to reposition it somewhere else in their note. Repositioning can be done, e.g., by long-pressing to drag-and-drop, by copying-and-pasting, by cutting-and-pasting, etc. If the pasted content does not fit in the target area, content below it will be pushed down. If it is pasted into a column, the content is resized, re-flowed, or transformed to fit.

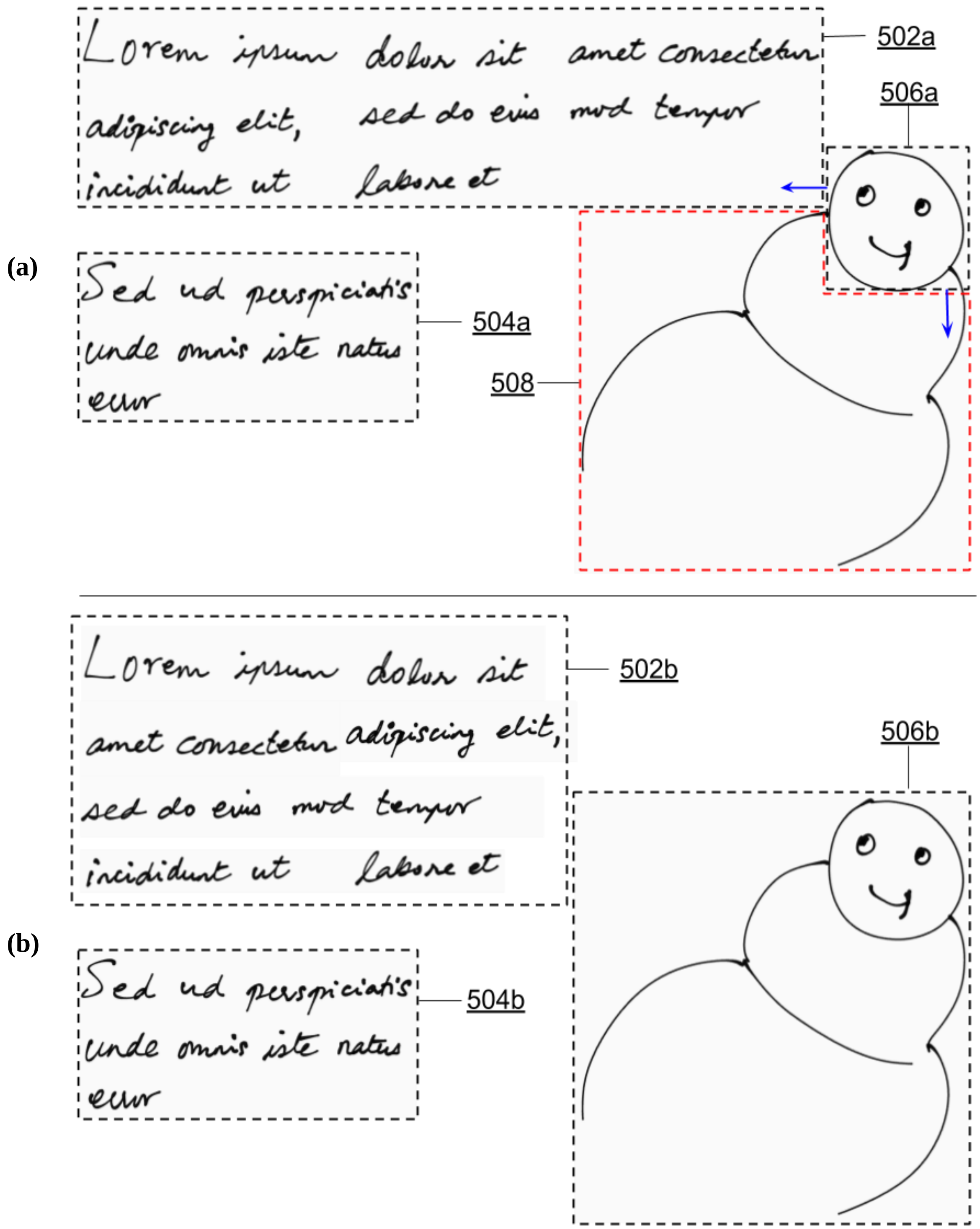

Fig. 5: Treatment of strokes outside an already-defined block

In this manner, the techniques of this disclosure enable a user to capture notes with the ease of physical paper and the flexibility of an electronic canvas, and have the captured content automatically organized into blocks that make written or sketched material editable, interactive, and interoperable with the other digital workflows of the user. The techniques effectively handle the ambiguity of mixed handwriting and sketches; enable the user to correct mistakes; keep the document ordered as the user adds to it; and enable the user to define and refine the document structure over time.

CONCLUSION

Drawing, note-taking, or inking apps that operate on electronic canvases do not differentiate between the various types of objects, e.g., paragraphs, lists, sketches, tables, doodles, etc., that an author can create on a screen. Consequently, it is difficult in electronic canvases for an author to perform operations commonly available in word processors, e.g., insert a sketch, move a block of text up or down, merge or separate sketches, etc. This disclosure describes techniques that automatically organize and group strokes drawn by a user on an electronic canvas into blocks of content, e.g., objects such as text-blocks, lists, sketches, tables, doodles, etc. The techniques thereby handle implicit structuring in an interactive inking or drawing app and provide the user the ease equivalent to writing on physical paper and the flexibility of electronically editing and restructuring diverse content.