March 20, 2018

Automatic contextual recognition of hand-drawn content

Kyle Williams

Milton Ribeiro Filho

Megan Renshaw

Automatic contextual recognition of hand-drawn content

ABSTRACT

Digital whiteboards and touchscreens enable users to express concepts using free-hand drawing and writing. Although applications exist that recognize handwritten matter or geometric shapes, such recognition is not based on the user context. This disclosure describes techniques to transform user input on a digital canvas, e.g., hand-drawn strokes on a touchscreen or digital whiteboard, into software editable objects, based on determined context. The context is determined based on user permission for analysis of the strokes. The software editable objects are provided in a format suitable for a software application matched to the context, e.g., as editable graphical structures such as graphs, charts, block diagrams, circuit schematics, etc. The techniques enable an expressive form of digital content creation and create clean vector content from intelligible rasterized strokes on a canvas, enhancing user experience and productivity.

KEYWORDS

- Scene recognition
- Digital whiteboard
- Contextual recognition
- Sketch recognition

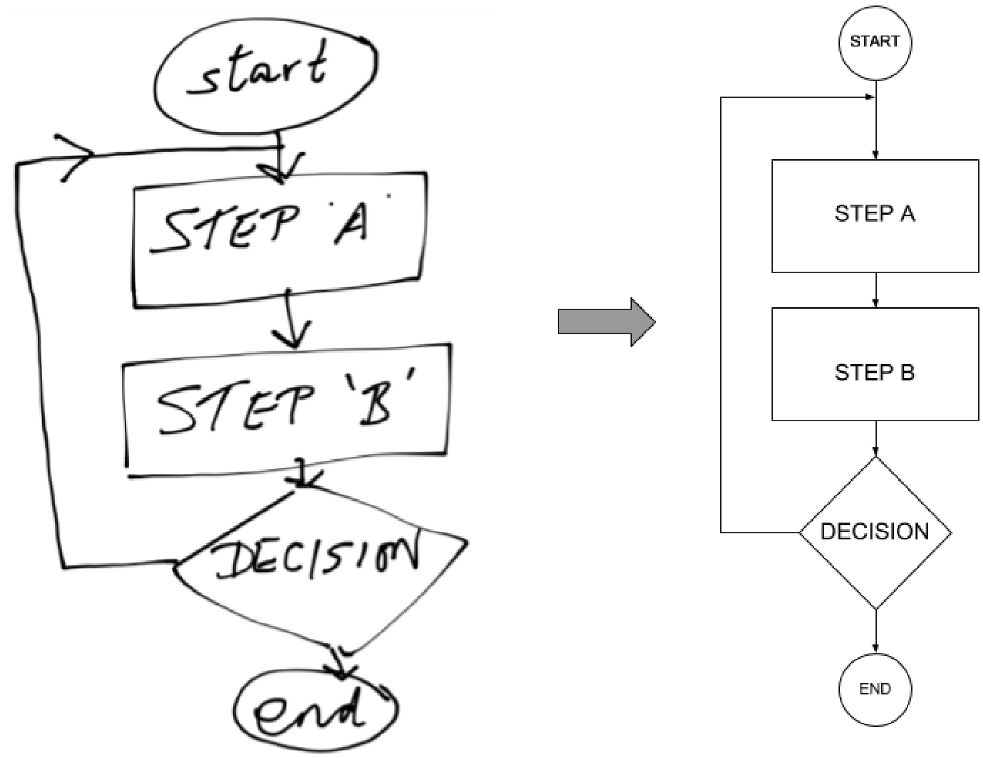
BACKGROUND

Collaborative digital whiteboards are a recent product category that feature an internet-connected touchscreen (or another internet-connected input device enabled to recognize touch or other gestures on or near the screen, including writing) on which several team-members that may be in different physical locations, collaboratively express concepts. Digital whiteboards facilitate new ways of collaboration, and encourage new content forms suited for natural interaction.

Although a digital whiteboard (or other touch-enabled device) recognizes handwriting or hand-drawn geometrical shapes, such whiteboards currently do not recognize an entire scene within its context. Hand-sketched or handwritten matter on a touchscreen is interpretable in a variety of ways based on context. For example, depending on the intent of the user, a hand-drawn figure could represent a business plan, an industrial process flow, an electronic circuit, an architectural blueprint, an urban map, a block diagram, a plumbing layout, etc. There are special-purpose software applications for use by professionals in each industrial or academic domain. For example, architectural blueprints are created and viewed in computer-aided design (CAD) software, electronic circuits in electronic design software, maps in cartographic software, etc. The lack of context-based, whole-scene recognition in a whiteboard causes a hand-sketched scene to not be adequately translated to a software object that is editable by the appropriate software application.

DESCRIPTION

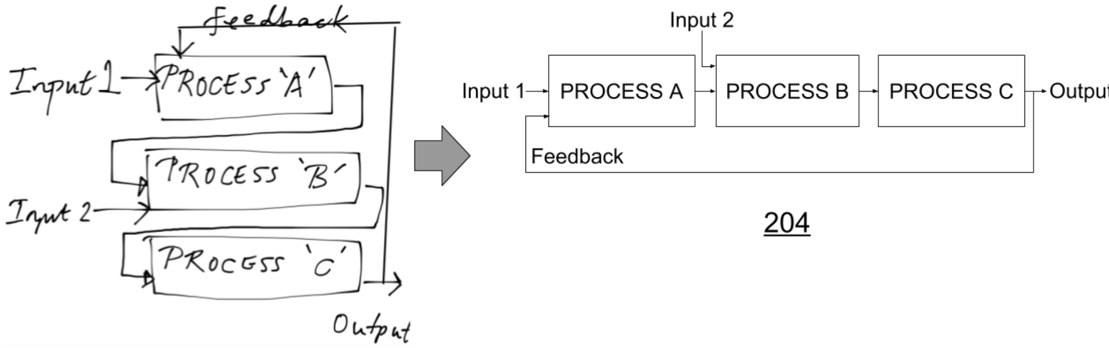
This disclosure describes techniques to recognize a hand-drawn figure in context. With user permission, the hand-drawn figure is analyzed to recognize user context, and upon analysis, the figure is converted and rendered in a language or format suitable for an appropriate software application. This is illustrated below with several examples. The techniques are implemented with specific user permission for analysis of hand-drawn content. User are provided options to indicate that one or more hand-drawn portions are to be excluded from analysis and to turn off the recognition features.



102 104

Fig. 1: A hand-drawn flowchart is transformed into a software-editable flowchart

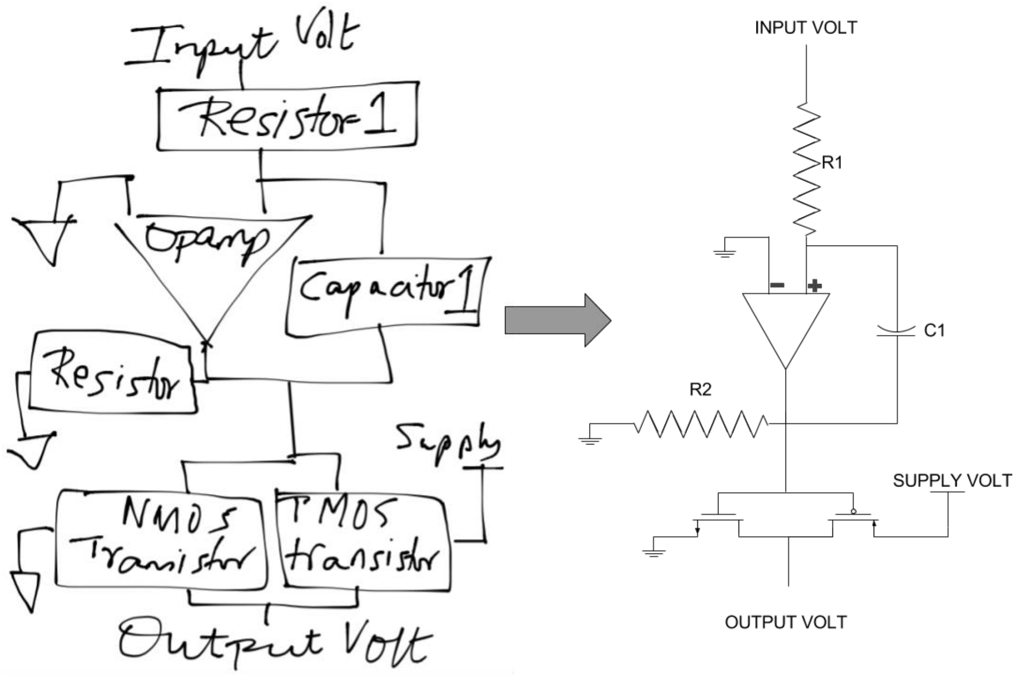
Fig. 1 illustrates transformation of a hand-drawn flowchart (102) to a software-editable flowchart (104), per techniques of this disclosure. The software-editable flowchart is a formal representation of the original hand-drawn flowchart, with objects that are understandable by editing software. While the original hand-drawn flowchart may be raster-mapped and hence subject to pixelation when magnified, the software-editable flowchart can be stored in a vector-mapped format supported by popular diagramming software applications.



202

Fig. 2: A hand-drawn block diagram is transformed into an editable software object

Fig. 2 illustrates transformation of a hand-drawn process-flow block diagram (202) to a software object (204) editable by standard design tools, per techniques of this disclosure.

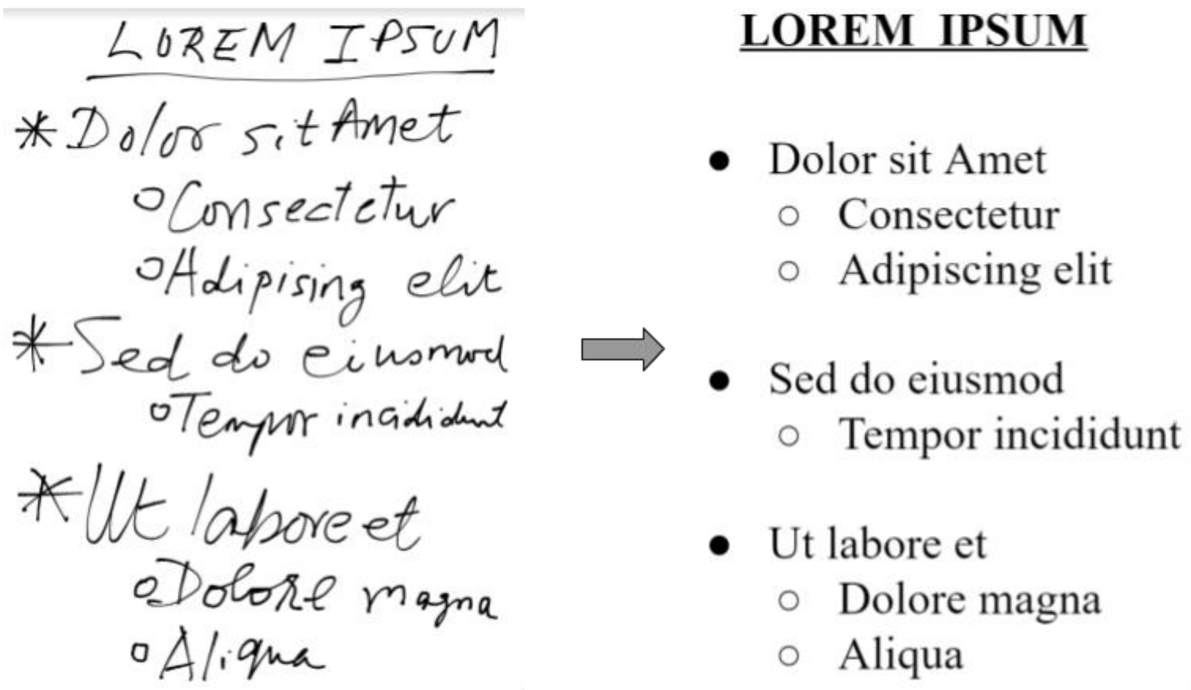


302

304

Fig. 3: A hand-drawn electronic schematic is transformed into a design-tool editable electronic schematic

Fig. 3 illustrates transformation of a hand-drawn electronic schematic (302) to an electronic design tool editable software object (304). While the original hand-drawn schematic uses blocks to represent electronic componentry, the techniques described herein automatically determine corresponding objects and use standard electronic circuit symbols, e.g., for parts such as resistors, capacitors, transistors, etc., as seen in Fig. 3.



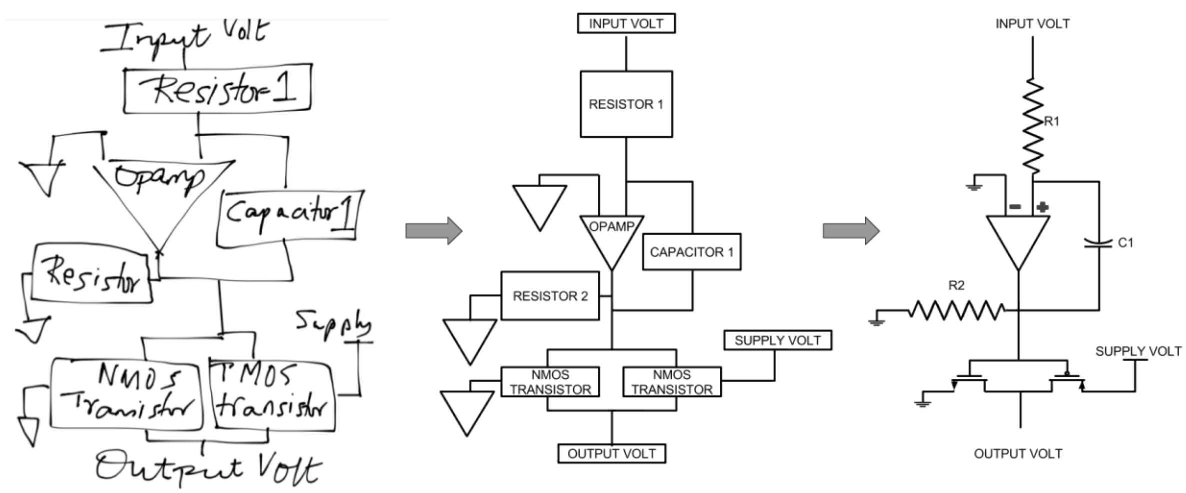
402 404

Fig. 4: Transformation of handwritten matter to an editable file

Fig. 4 illustrates an example of hand-written matter (402) being transformed into a file (404) editable by a software application such as a word processor or presentation application. Per techniques of this disclosure, the transformation preserves the typesetting and formatting indicated in the original handwritten matter, e.g., the underlined text “Lorem Ipsum” and bullet-point text is transformed per the formatting indicated.

Features of the techniques of this disclosure are elaborated upon below.

- *Automatic detection of context*: With user permission for use of such techniques, trained machine learning models or other recognition techniques are applied to the content detected on the whiteboard to automatically detect context. For example, detecting a preponderance of electrical symbols or terms on hand-drawn matter indicates that the context is "electronic schematic."
- *Provision for user to specify context*: Users of the whiteboard can also indicate a particular context, which may override auto-detected context.
- *Provision for user to correct context*: Under certain situations, it may be determined that the context is ambiguous. In such instances, the techniques enable users to correct the automatically determined context.



502 504 506

Fig. 5: Resolving ambiguous context

This is illustrated in Fig. 5, where the whiteboard has a hand-drawn sketch (502). In this example, the user intends the drawing to be an electronic schematic; however, automatic context-detection interprets the context as "block diagram" (504). In this case, the user simply corrects the context to "electronic schematic" and in response, the software editable drawing is automatically transformed to generate a design-tool compatible electronic schematic (506).

- *Training to recognize handwriting or sketching idiosyncrasies*: The aforementioned provision for a user to correct context includes training of the machine learning model that recognizes context and objects therein, when user permit such training.

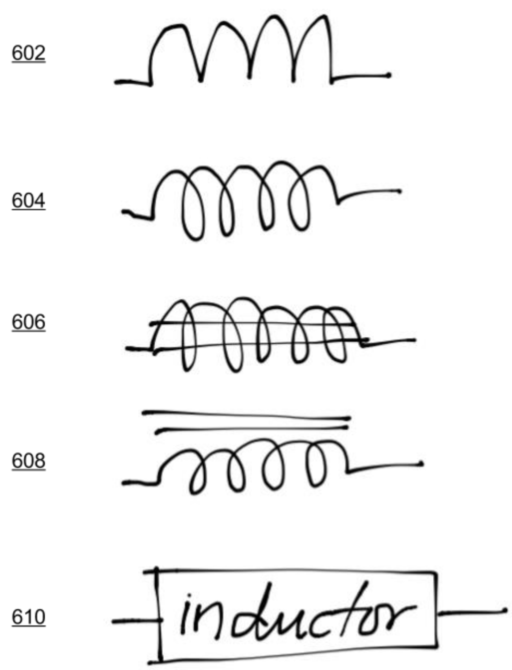


Fig. 6: Different ways for a user to sketch an inductor

In human expression, there are many ways to sketch a given concept or object. For example, in the field of electronic design, Fig. 6 shows several valid examples (602, 604, 606, 608, and 610) of sketches that correspond to the electric circuit component inductor. Moreover, a particular user can use their own ways to sketch and indicate user-specific meaning for symbols. The recognition techniques described herein enable the user to provide feedback, and thereby train, the machine learning models to recognize a user's particular idiosyncrasies, handwriting, use of symbols, etc. When users provide permission, anonymized learning from different users can be aggregated to train the machine learning models for global trends in contextual recognition of hand-sketched or handwritten matter.

To recognize hand-sketched or hand-written matter within context, the trained machine learning model of this disclosure use as input features the sequence of strokes made on the touchscreen. A stroke is mathematically defined by a set of position and velocity coordinates, color, and accompanying time-stamps. The machine learning model attempts to determine the context, e.g., interpretation intended by the user, when the user writes or draws on a touchscreen or digital whiteboard. If the context of the hand-sketch or handwritten matter is provided by the user, the machine learning model uses such context as an input or as a training label. The scene, as recognized by the machine learning model, is returned as a single object or as multiple artifacts that together make up a larger scene, both editable by a software application appropriate for the detected context.

In situations in which certain implementations discussed herein may collect or use personal information about users (e.g., user data, information about a user’s social network, user's location and time at the location, user's biometric information, user's activities and demographic information), users are provided with one or more opportunities to control whether information is collected, whether the personal information is stored, whether the personal information is used, and how the information is collected about the user, stored and used. That is, the systems and methods discussed herein collect, store and/or use user personal information specifically upon receiving explicit authorization from the relevant users to do so.

For example, a user is provided with control over whether programs or features collect user information about that particular user or other users relevant to the program or feature.

Each user for which personal information is to be collected is presented with one or more options to allow control over the information collection relevant to that user, to provide permission or authorization as to whether the information is collected and as to which portions of the information are to be collected. For example, users can be provided with one or more such control options over a communication network. In addition, certain data may be treated in one or more ways before it is stored or used so that personally identifiable information is removed. As one example, a user’s identity may be treated so that no personally identifiable information can be determined. As another example, a user’s geographic location may be generalized to a larger region so that the user's particular location cannot be determined.

CONCLUSION

This disclosure describes techniques to transform user input on a digital canvas, e.g., hand-drawn strokes on a touchscreen or digital whiteboard, into software editable objects, based on determined context. The context is determined based on user permission for analysis of the strokes. The software editable objects are provided in a format suitable for a software application matched to the context, e.g., as editable graphical structures such as graphs, charts, block diagrams, circuit schematics, etc. The techniques enable an expressive form of digital content creation and create clean vector content from intelligible rasterized strokes on a canvas, enhancing user experience and productivity.