



formatting text with face gestures

Alice Strunkmann-Meister (CarolineAlice.Strunkmann-Meister@HS-Augsburg.de)

Rodrigo Blásquez (Rodrigo.Blazquez@HS-Augsburg.de)

Winter Semester 2015/16

Interaction Engineering

Prof. Dr. Michael Kipp



Augsburg University of Applied Sciences

Abstract

Formatting text in text editors can be uncomfortable: One has to stop the writing-flow and grab the mouse to highlight the text and then press the formatting-buttons. With *faceTYPE* that problem can be solved: the user formats the text by using his facial gestures. If a gesture is performed, the format is activated and the user can keep on writing the text without having to interrupt the writing-flow. There are three formats, which are caused by a face gesture: first, the format "bold" is caused by rising up the eyebrows, second, the format "italic" is caused by raising the head to the right, third, the format "underlined" is caused by compressing the lips. If none of those three gestures is performed, the "regular" text-format is activated.

Introduction/Motivation

Usually, when people write a text with their preferred text-Editor (like word or OpenOffice), they don't use shortcodes, to format their text. If they want to make a text passage bold, italic or underlined, they use the buttons, that the editor usually provides to them in the menu-band. But the use of the buttons - either before or after the text is written - by highlighting the text passage first, then formatting it by clicking the buttons, is always a little bit uncomfortable for the user: One has to move one hand away from the keyboard and grab the mouse, to press the formatting-buttons or to highlight the text. But if the user is in the writing-flow and doesn't want to interrupt it, which possibilities could he have instead of using the Shortcuts? With `faceTYPE`, the answer to this question is to use face gestures for text-formatting. By detecting those face gestures in realtime while the user is writing, some predefined gestures can be detected, which then cause the format to be activated.

Related Work

For disabled people, who are greatly restricted in their freedom of movement (for example people who suffer from Paraplegia), the face-muscles can be the only part of their body which they are able to move. For those people, it can be highly important and helpful, to control a computer with their face. One project, which aims to help in those situations, is the "project ACAT" by Intel.¹ They provide an Interface, in which the user is able to write text, to create folders and data, and to navigate through the computer only by moving his right jaw.

A technology, which is often used to detect movement in general and face gestures in particular, is OpenCV. Based on that technology, Jason Saragih developed a model based algorithm, which is called "FaceTracker".² Based on that work, the artist Kyle McDonald developed an application, which can communicate with other Applications which is called "FaceOSC".³ That application makes it possible, to easily control other computer programs, via the Open Sound Control (OSC)-Protocol. Due to that easy acces to face tracking data in realtime, there have been made many different applications, which allow to control different computer software, like for example Ableton⁴ or Rhino⁵ by face gestures.

¹ <https://01.org/acat>

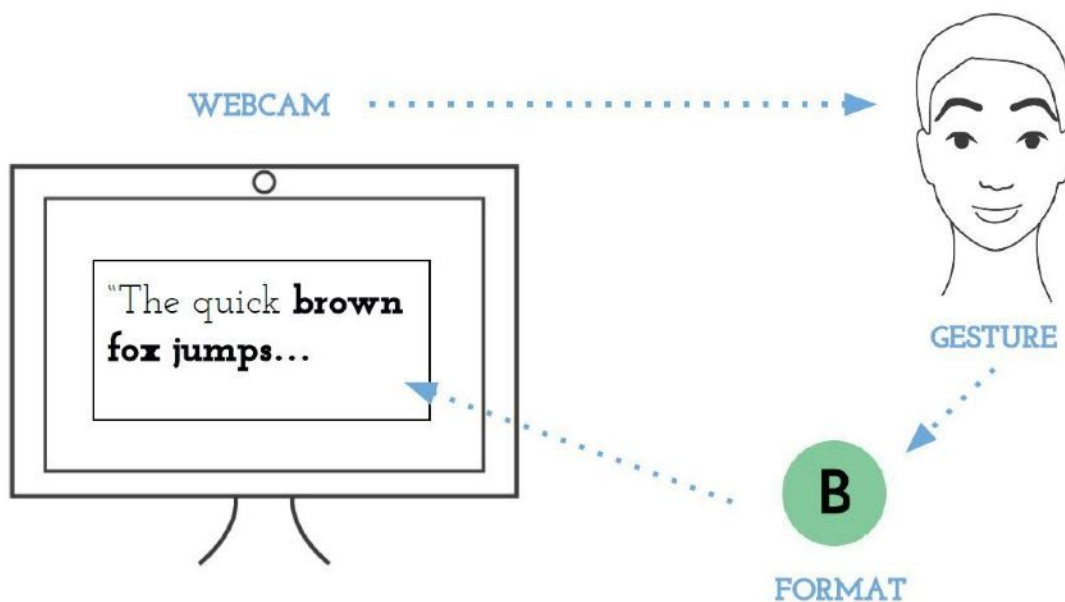
² <https://github.com/kylemcdonald/FaceTracker>

³ <https://github.com/kylemcdonald/ofxFaceTracker/releases/tag/v1.1>

⁴ <https://vimeo.com/26098366>

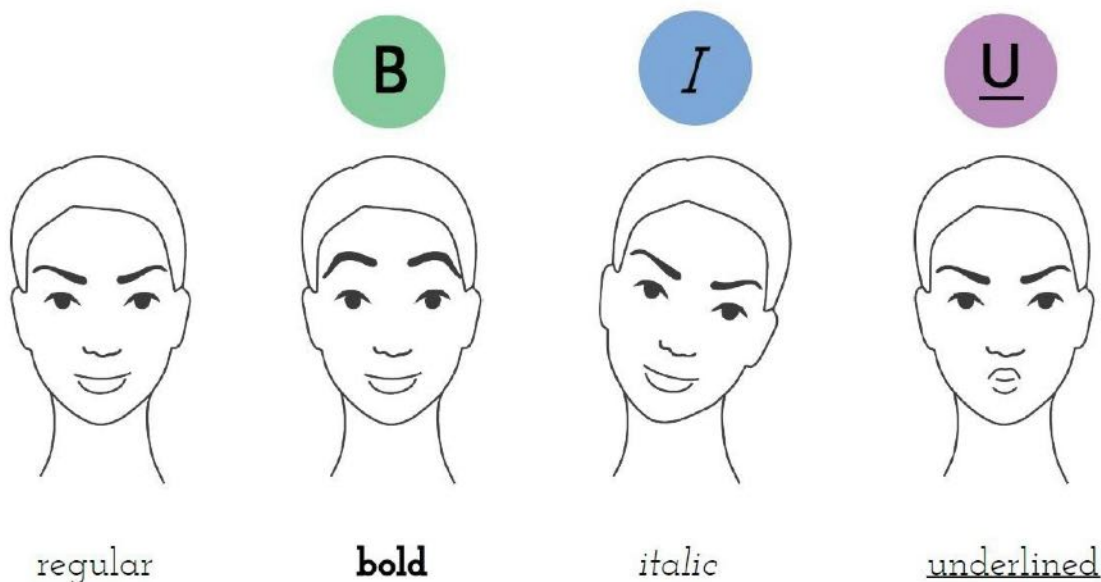
⁵ <https://vimeo.com/58430608>

Concept And Implementation



the face gesture is captured via webcam and transformed into the formatted text

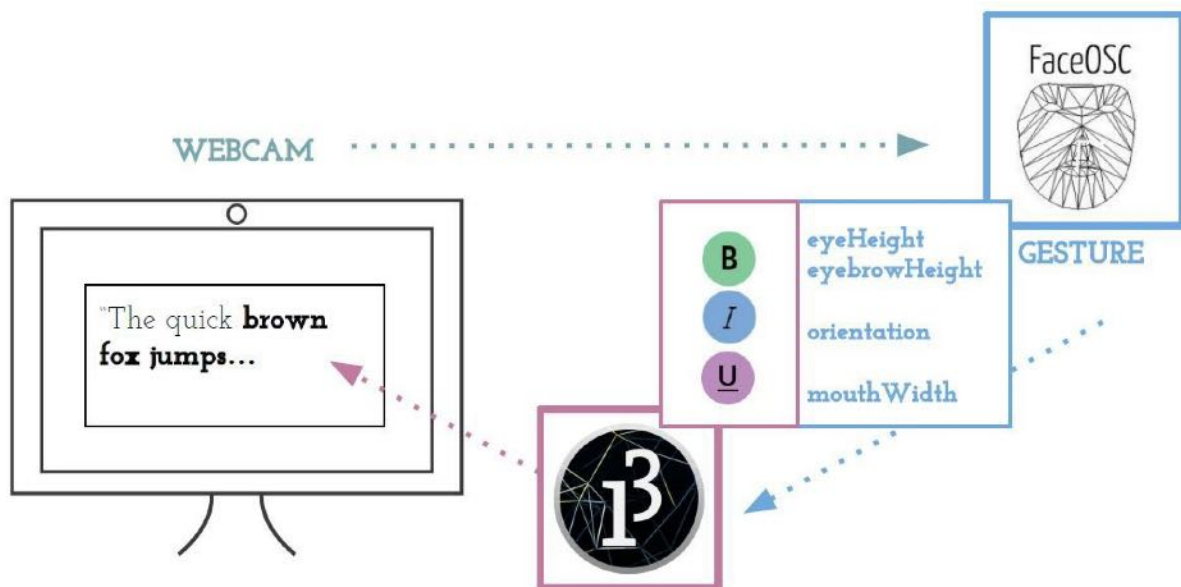
For the face-detection, a webcam, either built-in or attached to the computer, is used. If the user performs a specific Gesture, the format for the text the user is typing, is activated. The possible formats are “bold”, “italic” and “underlined”. If the user is not performing a special gesture at all, the font is displayed in a “regular” style. The gestures, which are connected to the format are “eyebrows up” for “bold”, “tilt head to the right” for “italic” and “compress lips” for the format “underlined”.



there are three gesture/format combinations. If no gesture is performed, the applied text format is “regular”

Prototype

For the face-recognition, the application “faceOSC”⁶ is used. The received data is then processed in Processing 3.0.1..⁷ The application gets the facial data via a network connection, which is build via the oscp5 library for Processing⁸. If the face is recognized, the “faceOSC”-application sends information about the facial data, such as the float values for the height and the width of the mouth, the height of the right and the left eye and eyebrow, as well as information about the orientation of the head as vectors. In Processing, the calculation of the border values for each gesture is made.



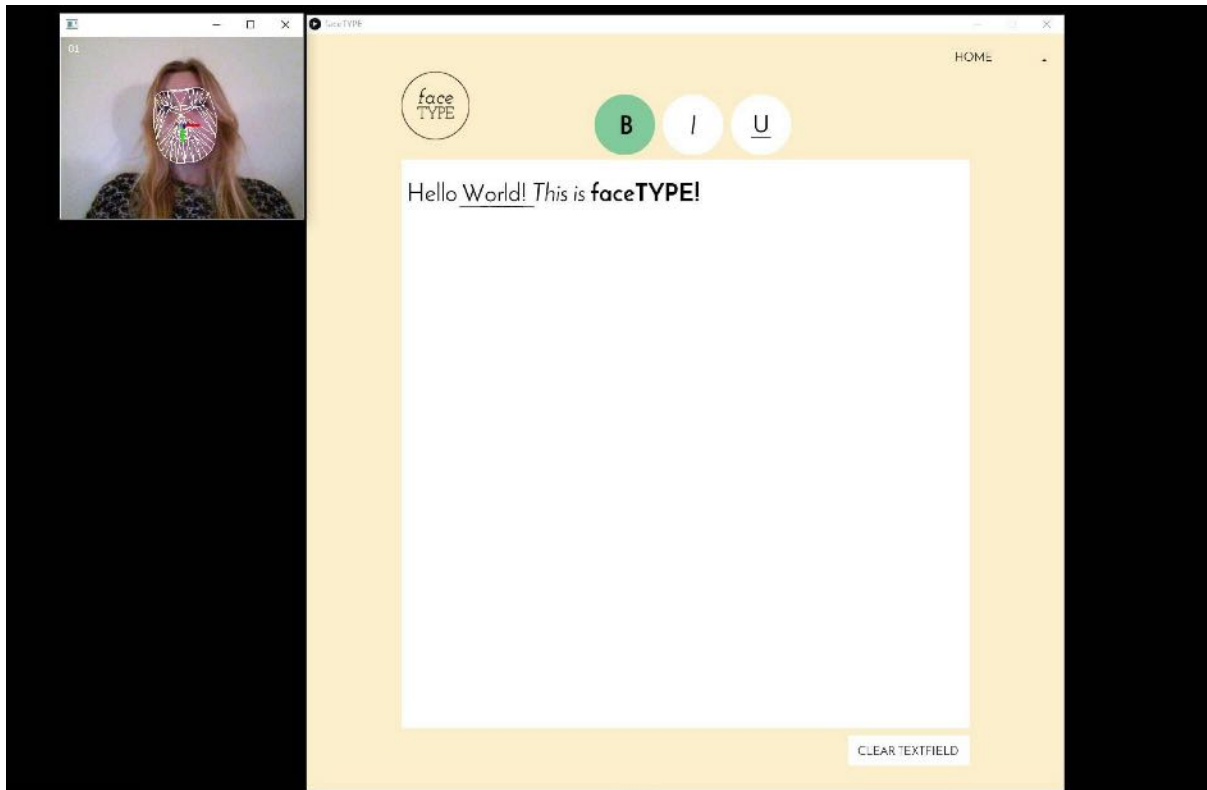
the face recognition is performed by faceOSC. The data is sent to the Processing application, where the text-format is applied.

The interface is written in Processing and consists of a “format-bar”, which shows the user, which format is currently activated, a “calibration-menu”, which allows the user to calibrate each gesture, and a simple text editor, which provides basic text-editing functions like entering text, deleting it with the backspace key and displaying the formatted text. There is also a “clear text field”-button, which allows the user to clear the whole text field. The received face-mesh from “faceOSC” is shown in an extra window. The application was implemented and tested on a Windows 10 Laptop with an integrated webcam.

⁶ <https://github.com/kylemcdonald/ofxFaceTracker/releases/tag/v1.1>

⁷ <https://processing.org/download/?processing>

⁸ <http://www.sojamo.de/libraries/oscp5/>



the interface provides basic text editing functions, a format-bar and a calibration menu to the user. The received face-mesh is shown in an extra window.

User Study

For the user study, two different prototypes were built. For both applications, the possibility to erase letters has been removed, to prevent users from making “unwanted” mistakes, as they should only concentrate on formatting text, not on correct spelling.

The first prototype is similar to *faceTYPE*, with the functionality of formatting text with face-gestures, and the second prototype was “shortcut”, where the user could format text with different shortcuts. The format “bold” could be activated by pressing CTRL + SHIFT + ‘b’, “italic” by pressing CTRL + SHIFT + ‘i’ and “underlined” by pressing CTRL + SHIFT + ‘u’. In both prototypes the format could be activated, until the first letter of the next word has been typed in. After that, the format is applied to the whole word, the user is typing in, until he presses the spacebar.

The users had to use both prototypes, “shortcut” and *faceTYPE*. Before the test-phase of both prototypes, there was a training phase, where the interface was introduced to the users and where they could get familiar with the formatting-technique (and to calibrate the interactions for the face gestures).

After the training phase, a simple text was shown to the user sentence by sentence. Each sentence had one format involved two times and the last sentence had every format involved one time (see appendix [1] for example text). For each sentence, the time was measured independently. The text, that each user typed in, was also saved as a screenshot.

The order, in which the prototypes were tested by the users, was mixed. During the tests, the users feedback was logged and after the tests, the users had to fill in a short survey about the test (see appendix [2]).

Results

As only four participants took part in the user study, the results should not be considered as representative, but as an helpful advice, how the prototype(s) can be improved.

The participants were between 28 and 41 years old, two male and two female. The native tongue of all of them was German.

Two of them started with *faceTYPE* as the first prototype and two of them started with “shortcuts”. The average time, the participants needed to write a sentence with *faceTYPE* was in all cases higher, than the time they needed with “shortcuts”. The

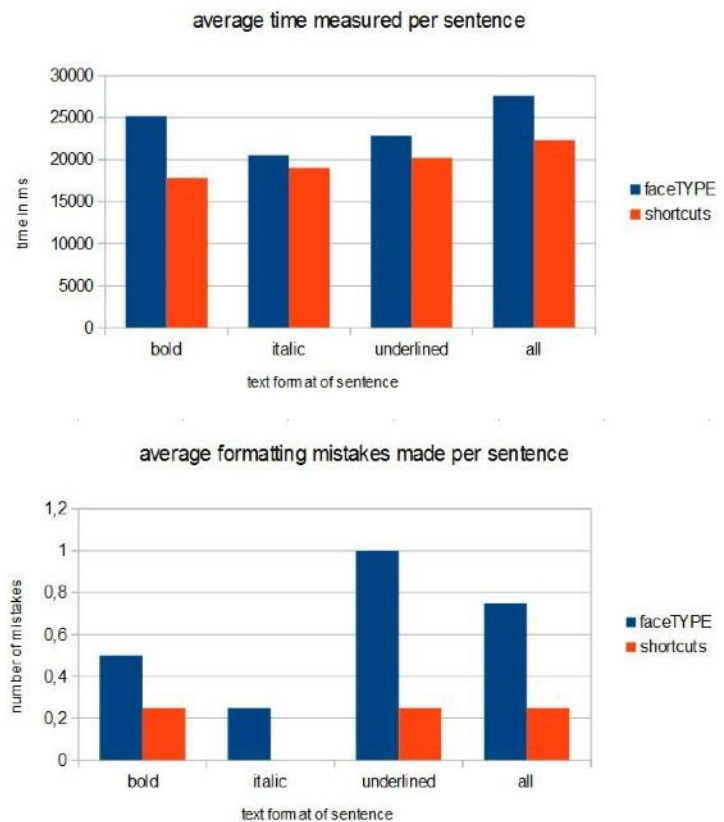
average number of mistakes the participants made per sentence while using *faceTYPE* was also higher than the number of mistakes, they made with “shortcuts” (every time, a participant didn’t apply the correct format to a word or applied the wrong format to a

word, it was counted as one mistake. Spelling-mistakes hadn’t been counted at all).

100% of the participants stated, that generally their preferred method of formatting text was “highlighting text and use of buttons”. After the test was finished, 100% stated, that “shortcuts” is easier to use, 50% that *faceTYPE* is more intuitive (the remaining 50% that “shortcut” was), 100% that *faceTYPE* is more fun and 50%, that they would prefer *faceTYPE* in general (the remaining 50% prefer “shortcuts”).

Asked about what the participants liked about *faceTYPE*, they mentioned again the fact, that it is fun to use the application and they found it helpful, that one format could be applied more easily to multiple words, than with “shortcuts”. They also mentioned, that they liked the colored format-bar very much, because they got feedback about which format is currently activated.

Asked about what they didn’t like about *faceTYPE*, all of the participants stated, that they had to concentrate a lot to apply a format and to remember, which gesture



belonged to which text format. Some of them found it difficult having to keep the head rigid and not to be able to express themselves by other face gestures (f.e. laughing) or by language while using the application.

Asked about how `faceTYPE` could be improved, all the participants stated, that they would have liked the training phase to be longer. Also all of them had problems with the text editor: there was no test cursor implemented, so the users didn't know where their actual position in the text field was and they were also missing the functionality of the return-key to start a new line. Some of the participants also had problems with the space bar having to be pressed, before a new format could be activated. Using `faceTYPE`, the format, the most easy to use was "italic". The participants all stated, that it was the most intuitive gesture and it was also the least error prone in terms of face detection.

Conclusion

One thing, all the participants agreed on is, that using `faceTYPE` is fun. To really find out, if the usage could also be helpful in terms of intuitiveness, writing speed and usability, the prototype should be improved more: The text-editor should provide the functionality of being able to use the return key and the text cursor should be implemented. Also the benefits of word-wise formatting - by only being able to activate a new format after having pressed the spacebar - should be verified, because some participants had problems with that feature. The training phase should be extended, to ensure, that the participants are able to use the face gestures in the proper way and maybe small illustrations of the depending face gestures could help the participants, to remember the gestures more easily. It could also be interesting to find out, if there are other face gestures for the format "bold" and "underlined", which are more intuitive to the user.

Appendix

[1] Example of the text, that was shown to the users in the user study (same font size and type, as used in the tests; the order of the applied format varied for each participant):

Hallo Anna. Wie geht es Dir?

Danke, **heute** geht es **mir** gut.

Das *Wetter* ist heute *sehr* schön.

Die Sonne scheint **und** der *Himmel* ist blau.

[2] Survey, the users had to fill in:

"faceTYPE" Fragebogen / Survey

1. How old are you ? Wie alt sind Sie?

2. What is your preferred method to format text? Welche Methode bevorzugen Sie, um Text zu formatieren

Markieren Sie nur ein Oval.

- Highlighting text and use of buttons/ Markieren des Textes und Verwendung der Buttons
- Use Shortcuts / Shortcuts verwenden

after the test / nach dem Test

3. Which software is easier to use? Welche Anwendung ist einfacher zu Bedienen?

Markieren Sie nur ein Oval.

- faceTYPE
- Shortcuts

4. Which software is more intuitive? Welche Anwendung ist intuitiver?

Markieren Sie nur ein Oval.

- faceTYPE
- Shortcuts

5. Which software is more fun? Welche Anwendung macht mehr Spaß?

Markieren Sie nur ein Oval.

- faceTYPE
- Shortcuts

6. Which software would you prefer in general? Welche Anwendung würden Sie im Allgemeinen bevorzugen?

Markieren Sie nur ein Oval.

- faceTYPE
- Shortcuts

7. What did you like about the use of the software "faceTYPE"? was hat ihnen an der Bedienung der Anwendung "faceTYPE" gefallen?

8. What did you not like about "faceTYPE"? Was hat Ihnen an "faceTYPE" nicht gefallen?

9. How could "faceTYPE" be improved? Wie könne "faceTYPE" verbessert werden?
