Daniel van Eijk-Bos, V2A, 17-May-2020

Goal:

This test rapport is written to document the way I will analyse the effectiveness of a new algorithm. The to be tested algorithm is a freshly programmed Canny Edge Detection algorithm that will be compared to an already existing Sobel Edge Detection algorithm.

With this test I want to research whether the Canny Edge Detection provides better edges than the Sobel Edge Detection. “Better edges” in this case means thin edges that clearly mark important points of someone’s face: the eyes, the nose, and the mouth. Whether the edges mark other things like the shape of the head or the hair is irrelevant. The only thing the chosen algorithm must do is to make it easier for the next algorithm in line to quickly pinpoint the areas of importance.

I cannot just give the results that flow out of these algorithms to the next algorithm, because it does not exist yet. Because of this, I will instead perform an A/B-poll in which participants have to make a choice between the outputs of these two algorithms. The question which will be asked to participants, and will be resolved in this test rapport, is: “Which algorithm pinpoints the important areas on the face, eyes; nose; mouth, in a clearer way?”.

Other details of the algorithms, like speed or memory efficiency, doesn’t matter in this test rapport and thus shall not be cleared up in this test rapport.

Hypothesis:

My hypothesis is that the Canny Edge Detection will produce a better result. An image that clearly pinpoints the areas where the important parts of the face are located. Considering the existing Sobel algorithm produces a blurry image with thick lines, where every little edge is highlighted, I doubt the Canny algorithm will lose.

Method:

The test shall be run as an A/B-poll in which participants will have to choose between the output of the Canny and Sobel algorithms. These outputs will both be created using exactly the same input image, which will not be shown to the participants. The participants will not be told which image is which, as in which image came from which algorithm. The ideal number of participants would be twenty, but the test will work just fine with a few more or a few less participants.

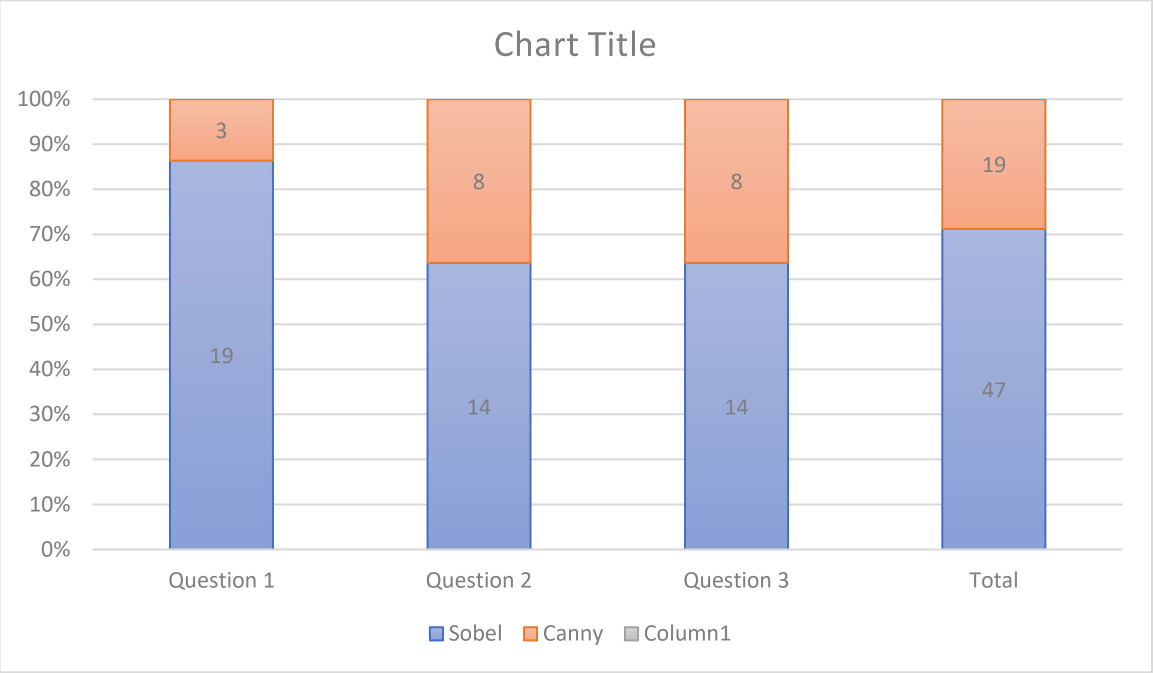
The data from this poll will be analysed and a conclusion will be formed based upon it.

Workplan:

1. Create a poll in which participants must choose between the two output images. The images must remain unidentified, the participant must not know which image came from which algorithm.
2. Distribute the poll to a wide audience of TI-students and ask them nicely whether I could have a minute of their time for them to fill in the poll.
3. Be ignored and get only six entries.
4. Spread the poll around among family and friends instead.
5. Tally the now 20+ entries and use this newfound data to determine which algorithm performs better in the eyes of the participants.

Results & Processing:

22 participants, each answered three questions.



	Canny %	Sobel %
Question 1	14%	86%
Question 2	36%	54%
Question 3	36 %	54%
Total	29%	71%

Most of the participants voted in favour of the Sobel algorithm. I think this is because of the clearer and nicer looking lines it produces in combination with these lines being more plentiful than the ones appearing in the Canny output. Because of this, the facial structure is defined clearer instead of the sparse few edges appearing in the Canny output that only pinpoint the most important of edges.

I honestly do not know how to take this result. Not just because of my personal bias towards the Canny algorithm I created. But also, because this test involved humans analysing data meant for computers, these two analyse data in completely different ways. This might mean that regardless of what the participants think, the Canny algorithm is considered better than the Sobel one by the computer. It might also mean that the Canny algorithm is considered worthless to the computer, I do not know. The problem with this test is that it tests something computers need to use by giving it to non-computers and this might give incorrect results.

The reason I chose for this faulty testing mechanism is because it was the only option. There is as of now no algorithm yet which should/could process my output images in the specific way I require. And thus we have to satisfy ourselves with these less than reliable results.

I was originally planning to calculate a binomial proportion confidence interval for how much more effective Canny was. But considering the, at least to me, surprising results I will not do so. That is because this does not add anything substantial to the test rapport considering this algorithm is already implemented and the results do not show this changing any time soon.

Conclusion:

The answer to the question “Which algorithm pinpoints the important areas on the face, eyes; nose; mouth, in a clearer way?”, at least according to my results, is the Sobel Edge Detection algorithm.

Evaluation:

The implementation plan went well enough as far as I can tell. The only mistake I made there was that I made up a hard to test scenario for my goal. This was because I tried to manipulate the results, in the big table, in such a way that in the selected scenario the Canny algorithm would come out on top. The algorithm seemed like such an interesting one that I just had to program it for this assignment.
And that is where I messed up, because of the weirdly specific scenario I did not have any program to test the output from my algorithms and I had to resort to a less than reliable testing method. This testing method backfired when I learned that the participants generally preferred the lines that the Sobel algorithm provided over the ones from the Canny algorithm. Thanks to this my hypothesis got thrown off and I kind of feel like I did it for nothing. Though I still believe that if a program got both as an input, and if it satisfied al the requirement from my scenario, the Canny output would come out on top. Except for this chain of mistakes, I think everything went pretty well.

Next time when making such an assignment I should either pick a simpler to test scenario or pick a specific scenario which already has something to test it with. That ways I will avoid being forced to poorly test the algorithm I made.