Reproducing a result from the paper "Retinal Prosthetic Vision Simulation: Temporal Aspects" by Avraham et.al

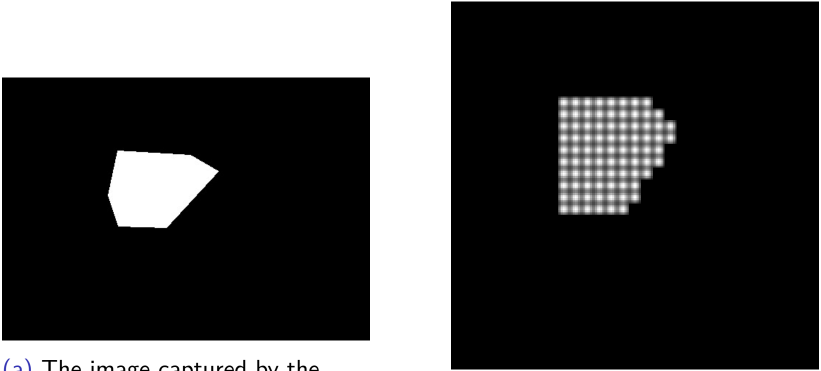
Omer Rosler

August 4, 2024

The research field - Prosthetic Vision

- This paper [4] is in the field of prosthetic vision. Certain types of degenerative blindness can be treated with prosthetic devices.
- The prosthetic architecture studied in this paper consists of two parts - a mounted camera, and an electrode array mounted inside the retina. Each electrode emits an electric pulse at a steady pace, (with frequencies of 6, 10, or 30 Hz for the main prosthetic simulated in the paper, the Argus II).
- This has the effect of seeing a spot of light in the patient's field of view. This white light is called a "phosphene" in the literature. Using an array of these electrodes enables "pattern vision".

Example Vision



(a) The image captured by the camera

(b) The image simulated

Figure: Comparison of real object and simulation (static)

The novelty in this paper

This paper contains simulations of this prosthetic vision. The main novelty of the paper is a simulation architecture that includes temporal effects. Using this new simulation technique, it analyzed the temporal effects and reached several conclusions regarding user training and the planning and development of future prosthetic devices.

Temporal effects

These are measurable quantities, and they vary between users:

1. Persistence (PER): It was found that some subjects perceived phosphenes that lasted as long as 8 s after the end of the stimulation.
2. Perceptual fading (PF): With continuous stimulation, phosphenes begin to fade out after a variable duration up to 10 s, over a fading out duration for individual users ranges from 0.5 s to more than 60 s

Videos 1,2

Results of the paper

The paper runs simulations to understand the effects of these "parameters" on the perception of the user.
The other perceptual effects discussed in the paper include "flickering" (3.1), "the stroboscopic effect" (3.2), "smearing" (3.3), and "contour vision" (3.6), we will focus only on the last one.

The Testable Claim - Contour Vision

Exploiting persistence, by using "scanning head movements" we can reproduce the contour of a static object **after** the perceptual fading started.
In other words, we can counteract the negative effect of perceptual fading using persistence.
"Two wrongs ~~don't~~ make a right."
Video 3

Mimicking head movements

To mimic head movements that scan a static object, the paper "inverted the perspective" i.e., created videos of the objects moving in direction opposite to the movement.
One may argue this is a wrong approach, as the object is in 3D, and one needs to apply projective transformations to change perspective. However, due to the low resolution of the prosthetic, the claimed contour vision works only for close and large objects (relative to the field of view). In this case, the difference between the approaches is negligible. The author addressed this fact indirectly in the "future research" section which suggested using virtual reality.

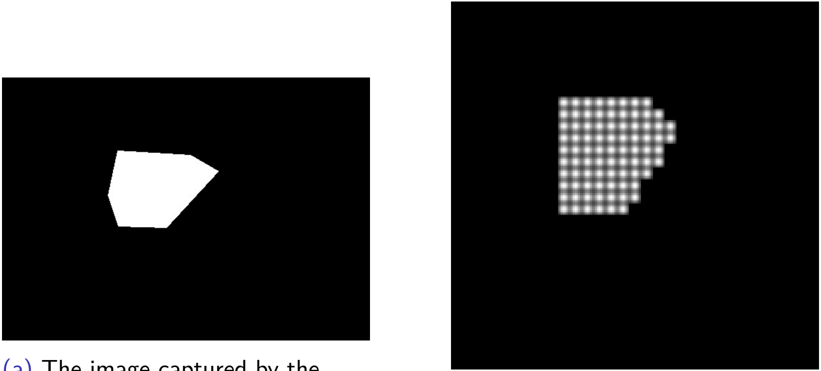
The Data available - Code and 3 Videos

The original paper ran simulations on very specific videos using very specific values for the temporal parameters. The claim is based on "seeing" the result with our eyes (figure 9 in the paper). Originally, I thought this claim to be robust, but after examining the videos used more closely, it seems to be less so.
Therefore, in order to reproduce this claim, I first needed to quantify it.
The "input data" is a video of an object scanned using head movements, and the result is a number representing the similarity between the contour of the object and the apparent contour at the end of the simulated video.

Quantifying the claim

- The code in the paper generates a simulation of what the patient sees with the prosthetic. By extracting the frames at the end, we compare these to the original video (before the movement).
- To obtain a quantifiable result, we aimed to extract the contour and use a standard shape-matching algorithm. Specifically, I use the ‘matchShapes‘ function from the python OpenCV library [2] , with the chosen method being ‘CV_CONTOURS_MATCH_I2‘ which is more suitable for noisy conditions (as is the case here due to "persistence").

Example frame



(a) The image captured by the camera

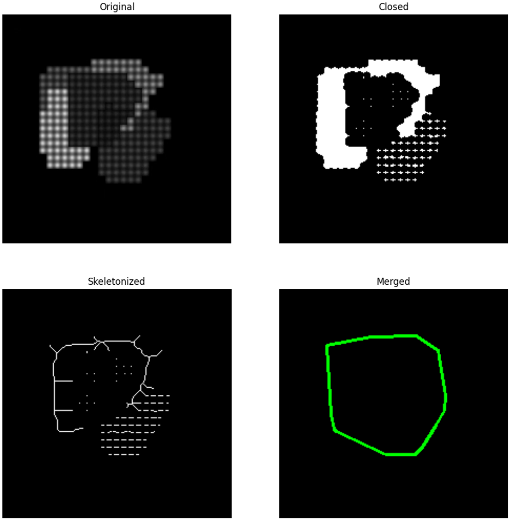
(b) The image simulated

Figure: Comparison of real object and simulation (static)

Image Processing required

First we apply "closing" which is dilation followed by erosion. This removes the tiny holes which are due to the distance between phosphenes in the image. Next, we apply "skeletonization" to recover the shape (we used the open source python library [6]). This results in a thin contour-like shape which is disconnected and branched. We then merge the disconnected pieces as described in [1]. This final result is a convex hull that approximates the shape.

Example Processing



Limitations - Choice of Threshold

We need to choose a threshold, and this affects the results. Therefore the exact value of the metric is meaningless, but only certain ranges are meaningful.

Choosing the input videos

Using the apparatus described above, we input videos of objects being scanned and obtain a metric describing the accuracy of contour detection.
This is a very open-ended dataset, therefore we need to decide what objects to test. My approach is to try and refute the claim of the author. Therefore I developed a list of conjectures regarding potential biases in the videos used in the paper.

Conjectures

1. The shapes used were too symmetrical.
2. The directions of the head movement are perpendicular to the contour of the object.
3. There is no delay between the perceptual fading onset and the start of the head movements. In reality, there would be a small delay, or maybe even premature movement.
4. The contour vision is most prominent when the PFO and PER are close to each other.

Dataset generated

The conjectures lead to "useful" datasets.
To test the first three conjectures, I created a dataset of videos of objects which are being scanned with the following parameters:

1. Randomized shape of the object
2. Randomized *delayed or premature* start of the movement (relative to the Perceptual Fading Duration)
3. Randomized directions and speed of movement

To test the fourth claim, I ran the simulation code with different values for the PFO and PER.

More Limitations

I had trouble determining the actual speed used by the author.
The paper cites a certain range in section 2.3, but the videos used in the simulation follow a much lower speed.
In our simulations, we used the value cited in the paper.
This does not affect the validity of our claim, as these movements simulate head scanning of an object. Increasing the speed induces a similar effect to moving the object further away. Therefore, even if our code used an incorrect speed, the results would still be applicable to more distant objects, and remain meaningful.

Results

First we note the result of the matching algorithm is a number that is interpreted as "smaller" the better match. We don't have absolute meaning, but we can still compare it.

Results - Quantifying the original data

We want to quantify the original 3 videos used by the author, i.e. figure 9. This is done by setting PER to 0 or 3 and PFO to 0.5. The result was as follows, for PER=0 the match score was the special value '1.7976931348623157e+308' which indicates the shapes were drastically different. For the value PER=3 we got the following results:

Ellipse	Rectangle	Triangle
12.884957614701374	0.08348188961649206	20.25174440803744

Table: Results of the original data

The first two conjecture seems to explain the fact the rectangle is a very good match.

Results - Quantifying the generated data

The simulation parameters are PER and PFO. We put the result along with the suite of parameters into a python "pandas" database and performed some queries, described in the "data_analysis.py" file. A sanity check is that every time the match completely failed (i.e. no contour detection at all) was when PER was zero, i.e. no persistence.
There were a number of queries with graphs not shown, but are in the repository.

Results - Quantifying the generated data

We analyzed the data, first per video (to compare the simulation parameters):
For most of the videos, higher PER was directly correlated with a better match. For others this ratio was more complex.
There was no apparent correlation between higher PFO and the match score.
The scale of the results for each video is the same (except for the case PER=0.0), which means the randomization process made no significant difference. This refutes completely conjecture 3 and only partially conjecture 2.
The fourth conjecture is also blatantly false.

The relation between PFO and the match

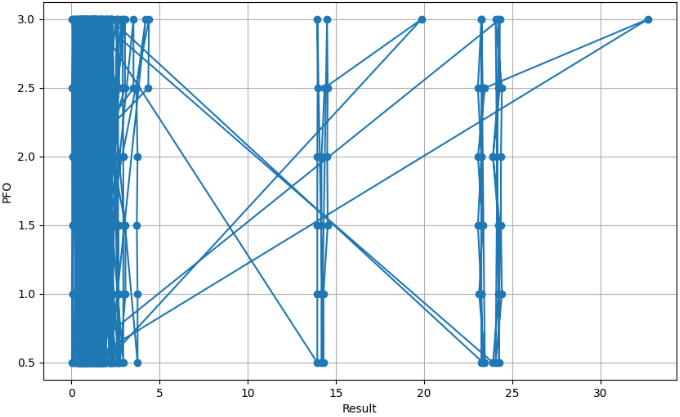


Figure: The relation between PFO and the match

It seems the best matches are for low PFO.

Higher PER correlates with better match

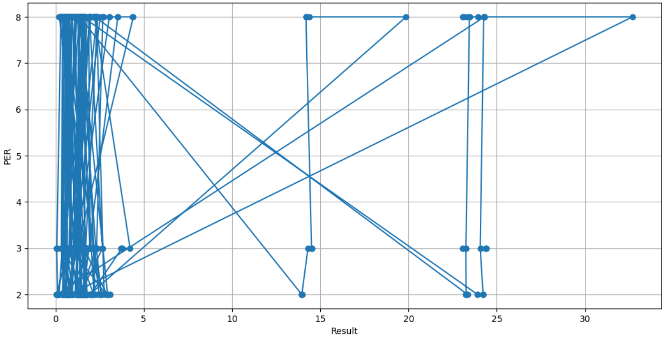


Figure: Higher PER correlates with better match

Proving randomization didn't matter

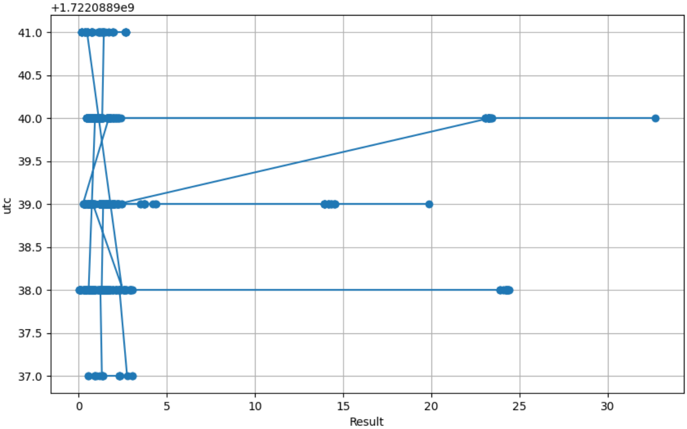


Figure: Proving randomization didn't matter

The UTC represents the different videos.

Conjecture 4 is false

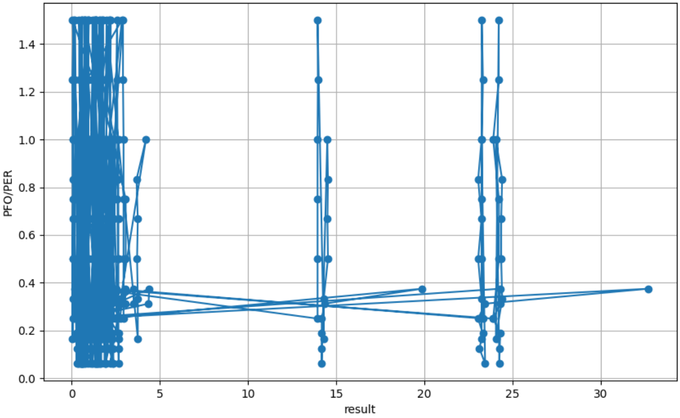


Figure: Conjecture 4 is false

Interpretation

There is still an unexplained smaller number of videos with very high match scores. We do not understand this wholly and this needs further analysis.

Conclusions

Most importantly, the shape we want to look at matters most.
We didn't disprove the second conjecture wholly. There coulb be a bias towards preferred directions of movement. We can definitely accept this, as the directions of movements are chosen by the user, and the training would be to choose these directions based on the perception of the object before the PFO.
In fact, we proved the conjecture true for most temporal parameters and refined the conjecture (without proving it completely, this needs further analysis):

Final Conclusion

Conjecture

To achieve contour vision of an object, the user should use head scanning motions in directions perpendicular to the most prominent direction in the convex hull.

- How to merge contours in opencv?
  https://stackoverflow.com/a/65434912/5005049.

- The opencv reference manual.
  http://docs.opencv.org/.

- David Avraham, Jae-Hyun Jung, Yitzhak Yitzhaky, and Eli Peli.
  MATLAB code of prosthetic vision simulation.
  https://matlab.mathworks.com/open/fileexchange/v1?id=116005, 2021.
  Version 2.0.

- David Avraham, Jae-Hyun Jung, Yitzhak Yitzhaky, and Eli Peli.
  Retinal prosthetic vision simulation: Temporal aspects.
  Journal of Neural Engineering, 18:0460d9, 08 2021.

- Omer Rosler.
  Code and data used in this paper.

https://github.com/OmerRosler/ProthesisSimulationMeasure.

Stefan van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors.
scikit-image: image processing in python.
PeerJ, 2:e453, 2014.