Aalto-yliopisto
Aalto-universitetet
Aalto University

ELEC-E7852
Computational Interaction and Design

Assignment A5a

Computational Rationality

Aitor Urruticoechea Puig
aitor.urruticoecheapuig@aalto.fi
Student N°101444219
November 2024

Assignment A5a
Aitor Urruticoechea Puig
November 2024

Acknowledgement

Parts of the code implemented in this work were generated with the assistance of GitHub Copilot. None of its proposals were, however, employed without refinement and necessary tweaks to adapt it to the nuances of the tasks at hand; making it impossible to identify and subsequently mark which lines of code were human or machine-made, for many were the fruit of the combination of both.

Contents

List of Figures

1 GazeModel overview

Let's first look at what kind of predictions we can generate with GazeModel. First, generate data for different target widths using GazeModel (you can change the widths in the code template). Also generate some movement times. We assume that the movement duration is the sum of (1) the saccades and (2) the fixations. A helper function add_movement_time is given for a rough estimate. Describe what each of the rows in the generated dataset represent. Aggregate the data so that you get total movement times for each trial (episode). (1p)

Each row represents a single step in the simulation. Specifically, in each column, for each time step / row, the dataset captures the state, action, and outcome information for each step of the model's operation as it progresses through the task environment. From that, we can showcase the actual aggregates of movement time for each tested width (Table 1).

Target Width	Total Movement Time
0.1	28014.687957
0.2	12574.288906
0.3	4773.949123
0.4	515.035159

Table 1: Total movement time per each tested target width, using GazeModel.

The key in generating this data is in properly looping around with different widths:

```python
for target_width in target_widths:
    model = GazeModel(target_width=target_width)
    data = gazetools.run_model(model, controller, 100, "behaviour_trace.csv")
    data["target_width"] = target_width  # Add a column for target width
    datas.append(data)
```

2 Dependence on target width

How do total movement times and number of fixations predicted by GazeModel change depending on target width? Plot target width vs. movement time; and target width vs. number of fixations. Note that you might need to do some averaging over trials or select an appropriate function for plotting. (1p)

In Figure 1 many natural assumptions regarding gaze can be corroborated. First, that wider targets generally reduce the need for precise gaze adjustments, resulting in less average movement times as the width increases (a notable outlier in the trend is $\texttt{width} = 0.3$). Similarly, the number of fixations decreases as well, as wider targets demand fewer saccades and adjustments.

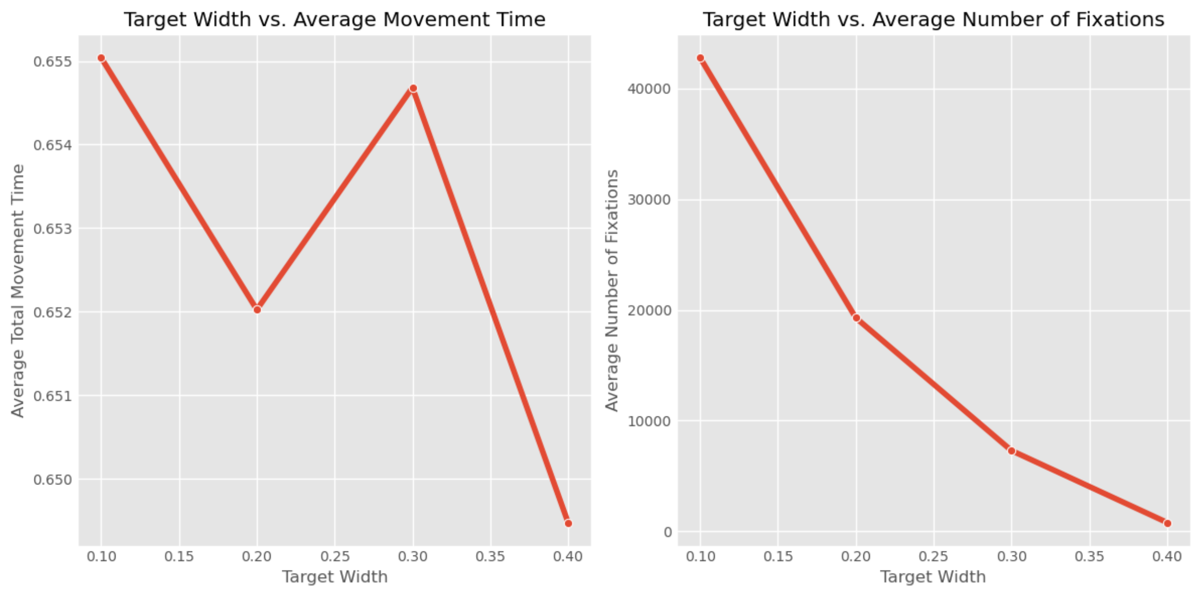

Figure 1: Target Width vs. Average Movement time vs. Average Number of Fixations.

3 Improving the Human model

Let's then improve the human model. The add_movement_time function currently randomly samples fixation durations and saccades to generate total movement time. Why is this not a great model? Using empirical literature, modify the model that generates fixation durations and saccades. For instance, you will want to assume that the time required to make a single saccade is linearly related to the distance moved, such that* $\text{saccade\_time} = \text{distance}/\text{saccade\_speed}$. *Also check (and cite) some realistic bounds for the model from the literature. (1p)*

In short, the current model is not great as it assumes randomness when in actuality saccades depend strongly on a myriad of human factors. As explored in the notebook, eccentricity plays a major role, but there are also other actors like temporal coherence (longer fixations are often followed by shorter saccades, and vice versa), cognitive influences, etc.

Following the advice on taking saccade speed as an input, we can work on a better yet simplified model taking a normal saccade speed of 500mm/s [1]. This can be implemented like:

```
import numpy as np
saccade_speed = 500   # mm/s

def calculate_saccade_time(fixation, target):
    distance = np.linalg.norm(fixation - target)

    # Convert distance to mm (assuming 1 degree = 10 mm at the fixation point)
    distance_mm = distance * 10

    saccade_time = distance_mm / saccade_speed # in seconds
    return saccade_time
```

```python
def add_movement_time(fixation, target, eccentricity_weight=0.1):
    saccade_time = calculate_saccade_time(fixation, target)

    # Simulate fixation duration based on eccentricity
    eccentricity = np.linalg.norm(fixation - target)
    # Simulate fixation duration with some random variation
    fixation_duration = np.random.normal(loc=0.3 + eccentricity_weight * eccentricity,
    ↪  scale=0.1)  # 0.3s base fixation time
    # Ensure that the fixation duration is not negative
    fixation_duration = max(fixation_duration, 0.05)  # minimum fixation duration of 50ms

    total_movement_time = saccade_time + fixation_duration
    return total_movement_time, saccade_time, fixation_duration
```

4 Visualizing the new proposed model

Plot target width vs. movement time for GazeModel with your new movement time model. How do these predictions compare to the original results that used add_movement_time? (1p)

This new model shows an inverse of the previously seen trend (see Figure 2), where this time target width and movement time have a direct and clear positive correlation. This is understood to happen because with a larger target width, the distance between the fixation and the target increases. This results in longer saccades and understandably longer fixation durations.

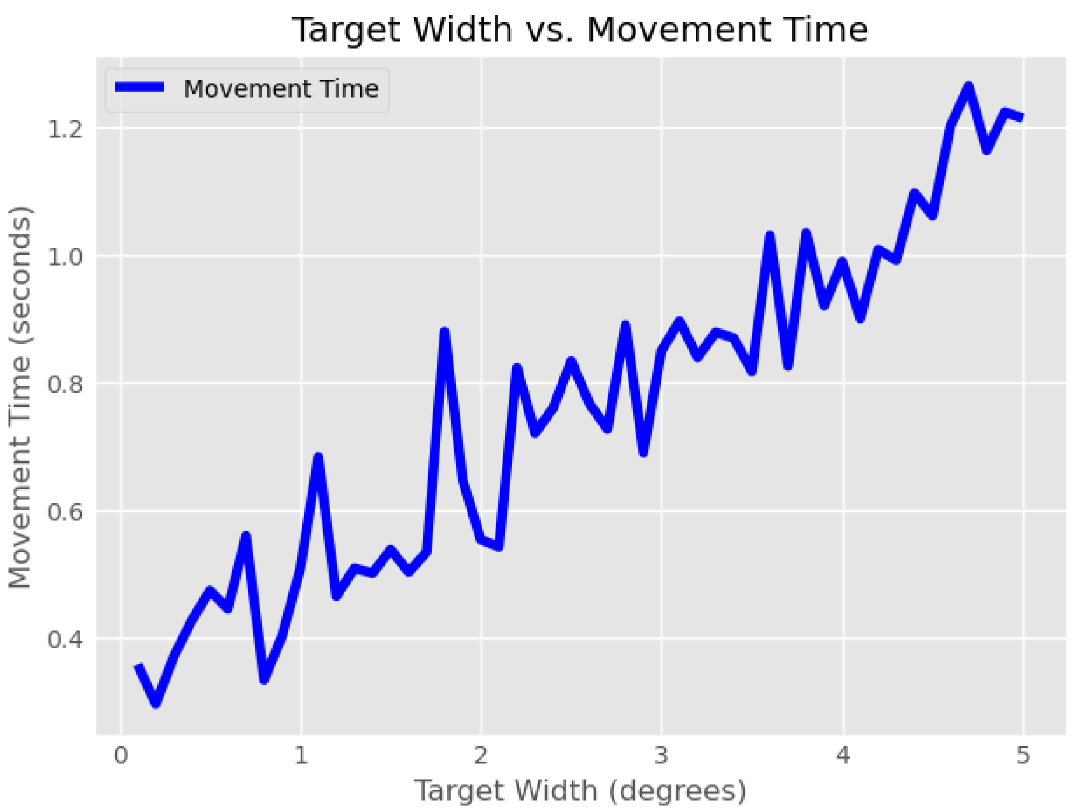

Figure 2: Target width vs. Movement time with the new model

This inversion of the trend with the new model stems from the underlying assumptions and how the model now factors in saccade time in relation to eccentricity and fixation duration. This model is, at its core, more deterministic and accounts for both the saccade duration based on distance and the fixation duration based on eccentricity, making the relationship between target width and movement time more consistent and predictable.

5 Additional bounds

Suggest an additional bound that you hypothesize to impact gaze-based selection in addition to those accounted for in the model. Is the bound internal or external? Suggest how you would implement this bound. (1p)

An interesting bound to consider is how visual acuity affects the user capacity to discern the smallest details at a given distance. This could impact the precision with which a user can select small targets: when a target is indeed too small (or placed in an area with low acuity at higher eccentricities), it becomes harder for the user to accurately fixate on it. Does the visual system compensate with increased fixations, or with more saccades in that area? In any case, this is an external limit to precision, as it is based on the physical realities of the human eye.

To implement this, I would go about implementing one acuity function. This should be able to adjust the probability of successful target fixation or selection based on the target's size and its eccentricity. How this escalates with both eccentricity and target size should, however, be investigated in-depth in order to properly model that behaviour computationally.

References

1. ROBINSON, David A. The mechanics of human saccadic eye movement. *The Journal of Physiology*. 1964, vol. 174, no. 2, pp. 245–264. Available from DOI: 10.1113/JPHYSIOL.1964.SP007485.