Chapter 10:
Eye-tracking
Today

- Learn about eye-tracking
- Hands (eyes) on eye-tracker
Eye-tracker: then
The first eye-tracking study
The first eye-tracking study

1. Free examination.
2. Estimate material circumstances of the family.
3. Give the ages of the people.
4. Surmise what the family had been doing before the arrival of the unexpected visitor.
5. Remember the clothes worn by the people.
6. Remember positions of people and objects in the room.
7. Estimate how long the visitor had been away from the family.

3 min. recordings of the same subject.
Eye-trackers: now
How does an eye-tracker work?
## Raw Data

<table>
<thead>
<tr>
<th>Timestamp [ms]</th>
<th>Category</th>
<th>Pupil size R [mm]</th>
<th>Pupil size L [mm]</th>
<th>Point of regard X</th>
<th>Point of regard Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>87542.5</td>
<td>Blink</td>
<td>3</td>
<td>2.9</td>
<td>936.3</td>
<td>691.7</td>
</tr>
<tr>
<td>87575.7</td>
<td>Blink</td>
<td>3</td>
<td>2.8</td>
<td>908.6</td>
<td>639.5</td>
</tr>
<tr>
<td>87609.2</td>
<td>Visual Intake</td>
<td>3</td>
<td>2.9</td>
<td>873.7</td>
<td>613.7</td>
</tr>
<tr>
<td>87642.5</td>
<td>Visual Intake</td>
<td>3</td>
<td>2.9</td>
<td>851.3</td>
<td>608.9</td>
</tr>
<tr>
<td>87675.8</td>
<td>Visual Intake</td>
<td>3</td>
<td>3</td>
<td>828.5</td>
<td>603.1</td>
</tr>
<tr>
<td>87709.2</td>
<td>Visual Intake</td>
<td>3</td>
<td>3</td>
<td>809.1</td>
<td>613.9</td>
</tr>
<tr>
<td>87742.3</td>
<td>Visual Intake</td>
<td>3.1</td>
<td>3</td>
<td>794.1</td>
<td>618.1</td>
</tr>
<tr>
<td>87775.6</td>
<td>Visual Intake</td>
<td>3.1</td>
<td>3.1</td>
<td>783.7</td>
<td>627.1</td>
</tr>
<tr>
<td>87808.8</td>
<td>Visual Intake</td>
<td>3.2</td>
<td>3.1</td>
<td>771.4</td>
<td>633.7</td>
</tr>
<tr>
<td>87842.1</td>
<td>Saccade</td>
<td>3.1</td>
<td>3.2</td>
<td>769.3</td>
<td>651.5</td>
</tr>
<tr>
<td>87875.3</td>
<td>Saccade</td>
<td>3.2</td>
<td>3.2</td>
<td>767.7</td>
<td>671.3</td>
</tr>
<tr>
<td>87908.6</td>
<td>Saccade</td>
<td>3.2</td>
<td>3.2</td>
<td>764</td>
<td>679.8</td>
</tr>
<tr>
<td>87941.8</td>
<td>Visual Intake</td>
<td>3.2</td>
<td>3.2</td>
<td>759</td>
<td>686.1</td>
</tr>
<tr>
<td>87975.3</td>
<td>Visual Intake</td>
<td>3.2</td>
<td>3.2</td>
<td>758.9</td>
<td>690.9</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Fixations and Saccades

Fixations: when the eyes rest momentarily on a small part of screen, 120 - 1000 ms

Saccades: when eyes jump rapidly from one part of the screen to another, 40 - 120 ms
Accuracy and Precision

**Accuracy**: average difference between real position of the stimuli and the measured position of the gaze.

**Precision**: ability of the eye-tracker to reproduce the same gaze measurement (variance)

![Images showing examples of accuracy and precision](image.png)
“Eye-mind” Hypothesis

What I see is what I process

1. Free examination.
2. Estimate material circumstances of the family.
3. Give the ages of the people.
4. Surmise what the family had been doing before the arrival of the unexpected visitor.
5. Remember the clothes worn by the people.
6. Remember positions of people and objects in the room.
7. Estimate how long the visitor had been away from the family.

3 min. recordings of the same subject.
Applications

1. Marketing: what attracts our attention?
2. Security: where are we paying attention?
3. Research: education, cognition, perception…
1. Marketing
1. Marketing
2. Security
3. Research

1. Expert versus novice
2. Task based performance
3. Programming
4. Massive Open Online Courses (MOOCs)
3.1 Expert versus Novice
3.1 Expert versus Novice
3.2 Task based performance
3.2 Task based performance

Solvers

Non-solvers
3.3 Programming

public static int goolbaka ( int vv )
{
    int c = 0 ;
    while ( vv >= 0 )
    {
        vv /= 10;
        c ++ ;
    }
    return c ;
}
3.3 Programming

```java
public static int goolbaka ( int vv )
{
    int c = 0 ;
    while ( vv >= 0 )
    {
        vv /= 10;
        c ++ ;
    }
    return c ;
}
```

**Poor student**

**Good student**
3.4 MOOCs

The substitution model

This scheme of expression evaluation is called the substitution model.
The idea underlying this model is that all evaluation does is reduce an expression to a value.
It can be applied to all expressions, as long as they have no side effects.
The substitution model is formalized in the \(\lambda\)-calculus, which gives a foundation for functional programming.
4. Groupware

1. Gaze-dialogue coupling
2. Dual eye-tracking
4.1 Gaze-dialogue Coupling

- Eye-voice span: time difference between what I see and what I say
- Voice-eye span: time difference between what you say and what I see
- Eye-eye span: time difference between what you see and what I see
4.1 Gaze-dialogue Coupling

Eye-voice span

4.1 Gaze-dialogue Coupling

Voice-eye span

4.1 Gaze-dialogue Coupling

Eye-eye span

4.1 Gaze-dialogue Coupling

Gaze cross-recurrence
4.1 Gaze-dialogue Coupling

Gaze cross-recurrence
4.1 Gaze-dialogue Coupling

Gaze cross-recurrence
4.1 Gaze-dialogue Coupling

Gaze cross-recurrence

[Diagram of gaze cross-recurrence]
4.1 Gaze-dialogue Coupling

Gaze cross-recurrence
4.2 Dual eye-tracking

One task to complete.

Two synchronised eye-trackers.

Synchronised screen: actions, viewports…

Peers can not see each other but can talk.
4.2 Dual eye-tracking

Example: pair-programming

Good collaboration quality  Poor collaboration quality
4.2 Dual eye-tracking

Example: collaborative visual search
4.2 Dual eye-tracking

Example: collaborative visual search

Good collaboration quality
Low cross-recurrence

Poor collaboration quality
High cross-recurrence
Eye-tracking and Learning Analytics

Learning Analytics: capture, compare, evaluate, intervene
Eye-tracking and Learning Analytics
Eye-tracking and Learning Analytics

1. Gaze-awareness: where am I looking?

2. Gaze-contingency: where is the speaker looking?
1. Gaze Awareness

With-me-ness

1. Perceptual with-me-ness: how much do you follow the teacher’s explicit gestures?

2. Conceptual with-me-ness: how much do you follow the teacher’s dialogues?
1. Gaze Awareness
Perceptual With-me-ness

Parameter and Return Types

Function parameters come with their type, which is given after a colon.

```java
def power(x: Double, y: Int): Double = ...
```

If a return type is given, it follows the parameter list.

Primitive types are as in Java, but are written capitalized, e.g:

- `Int` 32-bit integers
- `Double` 64-bit floating point numbers
- `Boolean` boolean values true and false
1. Gaze Awareness

Perceptual With-me-ness

Parameter and Return Types

Function parameters come with their type, which is given after a colon.

```python
def power(x: Double, y: Int): Double = ...
```

If a return type is given, it follows the parameter list.

Primitive types are as in Java, but are written capitalized, e.g.:

- Integer: 32-bit integers
- Double: 64-bit floating point numbers
- Boolean: boolean values true and false

Entry time
1. Gaze Awareness

Perceptual With-me-ness

Parameter and Return Types

Function parameters come with their type, which is given after a colon.

```java
def power(x: Double, y: Int): Double = ...
```

If a return type is given, it follows the parameter list.

Primitive types are as in Java, but are written capitalized, e.g:

- `Int` 32-bit integers
- `Double` 64-bit floating point numbers
- `Boolean` boolean values true and false

First Fixation Duration
1. Gaze Awareness
Perceptual With-me-ness

Parameter and Return Types

Function parameters come with their type, which is given after a colon:

```java
def power(x: Double, y: Int): Double = ...```

If a return type is given, it follows the parameter list.

Primitive types are as in Java, but are written capitalized, e.g:

- Int: 32-bit integers
- Double: 64-bit floating point numbers
- Boolean: boolean values true and false

Revisits
1. Gaze Awareness

Conceptual With-me-ness

Time $t = 0$

```java
for ( int i = 0; i < n; i++)
{
    sum += array [i];
}
```

Teacher: the iterator loops

Time $t = 1$

```java
for ( int i = 0 ; i < n ; i++)
{
    sum += array [i];
}
```

Teacher: over the array

Time $t = 2$

```java
for ( int i = 0 ; i < n ; i++)
{
    sum += array [i];
}
```

Teacher: updating the sum
1. Gaze Awareness

With-me-ness

- Perceptual with-me-ness is correlated to learning gain.
- Conceptual with-me-ness is correlated to learning gain.
- What happens when we give students the feedback about their with-me-ness?
1. Gaze Awareness

With-me-ness
1. Gaze Awareness

With-me-ness

Experimental conditions

Learning gain (normalised between 0 and 1)

Baseline group: n=50
Experimental group: n=27

Time (minutes)
2. Gaze Contingency
2. Gaze Contingency
2. Gaze Contingency

Number of pauses per minute
2. Gaze Contingency

Proportion of paused time

Video ID
2. Gaze Contingency

Number of seek-backs per minute vs. video ID.
2. Gaze Contingency

- When the teacher’s gaze is displayed the video-watching behaviour reflects low perceived difficulty.

- Is gaze better than a pen pointer?
2. Gaze Contingency

Cumulus Clouds
Low & Puffy
2. Gaze Contingency

Cumulus Clouds
Low & Puffy
2. Gaze Contingency
Next?

• Take any of the contexts mentioned in the slides and design an eye-tracking experiment.

• Eyes on eye-tracker.

• Work on your project (recommended).