# DIALOGUE SYSTEMS GROUP

# TAKE CORPUS

Casey Kennington
E-mail:dsg@uni-bielefeld.de

V1.0, March 26, 2014

# Contents

# 1 Summary

The goal of the TAKEcorpus is to have a collection of multimodal data for interactive dialogue using reference resolution as the primary task.

# 2 Materials

### Sensors and Recording Equipment

- Microsoft Kinect
- Seeingmachines eye tracker
- microphone
- video camera

### Furniture

- large-screen television
- table between participant and television screen
- participant chair

### Computers and Software

- Microsoft Windows Machine hosting the service for the MS Kinect
- Microsoft Windows Machine hosting the service for the eye tracker
- Ubuntu Linux 12.04 laptop running InstantReality and logging tool
- Ubuntu Linux 12.04 laptop running the software for the Wizard
- Gigabit network switch connecting all of the above mentioned machines

# 3 Physical Lab Layout

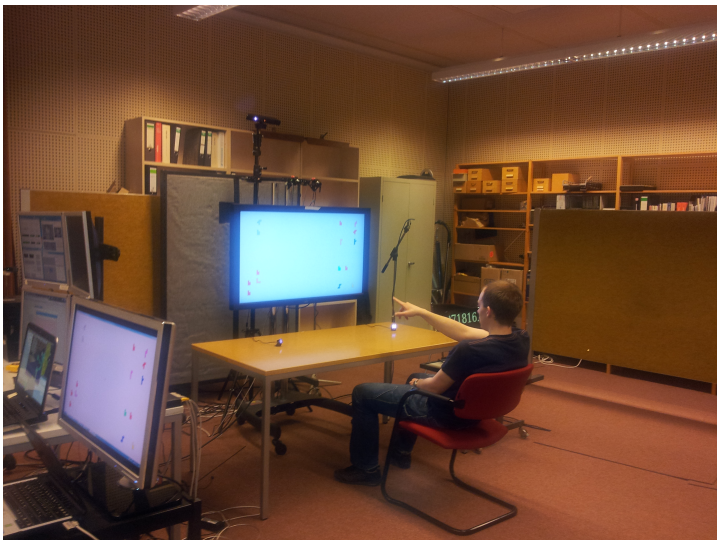

**Figure 3.1:** Lab Setup with Participant

Figure 3.1 shows how the lab was setup. The main focal point was a large TV screen. In front of that was a table (needed to place components of the eye tracker and space the participant from the screen) and a chair. Above the screen was the eye tracker, and above the eye tracker was the Kinect sensor. To the right of the screen was a free-standing boom microphone. Behind and to the right of the participant was a camera, which could record the participant and most of the screen, as well as a monitor with timestamp information. The participant did not have to wear any special equipment.

To the left were the computers running the sensors and logger. The Wizard was seated to the left and behind the participant and could see the large screen and participant at an angle.

# 4 Task Description

The overall goal was to collect as many episodes of reference resolution as possible in the time the participant interacted with our system. An episode would begin when the large screen showed a Pentomino game board with various pieces of various shapes and colors in seemingly random places on the screen (See Figure 3.1). The participant was to choose a piece, any piece, and describe and refer to the piece such that the system (wizard) could isolate it from the others and select it with a visual outline. When the outline appeared around a piece, the participant was to utter some kind of confirmation, if it was the intended piece. If not, then the process repeated until the correct piece was selected. Then a new episode would begin.

# 5 Software Architecture

**Overview** All of the sensor data were logged with timestamps (from the source and received time) which were logged to a single file. This log file, with video and accompanying audio constitute all of the data that were recorded. This was done with the Fraunhofer InstantReality framework. This framework is able to collect data from various sources across a network of computers via an *InstanIO network node*, project that data into a defined virtual reality scene which is a recreation of the scene, and can log the data to a single file. The `main` laptop, Ubuntu 12.04, ran the software that receives the data from the sensors, played the virtual reality recreation in real time, and logged the data to a single, time-aligned file. Figure 5.1 gives an overview of how the components were connected in terms of communication between software.

I will now describe the sensors and task software.

**Sensors** One MS Windows machine controlled the MS Kinect sensor and sent data, via the InstanIO network, to the `main` laptop. Each 3D joint point as recorded by the MS Kinect were logged. Another MS Windows machine controlled the eye tracker. This sent information to the logger about what pixel on the screen the sensor thought the participant was looking.

**Task Software** The `wizard` laptop had a Java program that was controlled by the wizard. This extended a game board onto the large screen which the participant was to look at. The state of the game board and events made by the wizard were sent to the `main` laptop and logged.

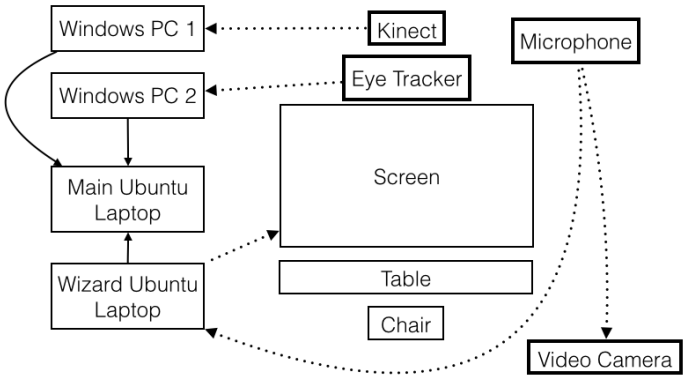


**Figure 5.1:** Overview of communication between software components.

# 6 Procedure

## 6.1 Before Task

Participants were brought into the lab, and given a form to read, which was a release form that the data we collect with their participation could be used for scientific analysis. After singing the form and assuring agreement, they were brought to the participant chair and were seated, facing the large screen.

By this point, the services running the Kinect and eye tracker were started and running and collecting data, but those were not as of yet being logged.

## 6.2 Task Procedure

The participant was instructed by a lab assistant in calibrating the eye tracker and the pointing region. The eye tracker calibration was a built-in function of the eye tracker; the participant followed, with his eyes, a moving dot on the screen. Also, the lab assistant set tracking points (via the eye tracking software) on various points on the face. This process took around 5 minutes. To calibrate pointing, the participants were instructed to hold their arms in the air (in the so-called *post* position), and then they were instructed to point to each corner of the large screen. Following this, the task was explained. The participants were told that they were talking with a computer dialogue system that is in development. At this point the Wizard started the software which displayed the first game board on the large screen. They were instructed to select a piece and describe it such that the computer system could select it by describing it vocally and they were encouraged to point. They were explained their task (as previously described) and were allowed to ask questions. At this point the Wizard began a new episode and the task began. Each episode took roughly 10 seconds. The wizard was instructed to select a piece as it was understood by the speech and pointing gestures of the participant. The wizard had an identical, smaller screen where he could use his mouse to select a piece. After a piece was selected, it showed as outlined in yellow on both the wizard screen and the large main screen. The participant was instructed to, after noticing a selected piece, to utter a confirmation. If it was indeed the intended piece and a positive confirmation was uttered, then the wizard pushed a button and a new episode

began. If not, the participant was to attempt again to describe the piece until the system (wizard) selected the correct piece.

For each episode, an episode ID was logged, as well as the state of the game board, to the logging laptop. Also, an xml representation of each game board was generated on the wizard computer. The wizard computer also recorded the audio from the microphone in a wav format. Further, when a piece was selected, that piece ID was written to a file. If something went wrong with the episode, the wizard could push a button to produce a file that flagged the episode as unusual (i.e., the wizard did not understand the uttearnce, there was a technical problem, etc.).

## 6.3 After Task

After the allotted time was up, the participant was informed that the task was complete and was thanked for his participation. He was then asked to fill out a questionnaire and was paid for his time.

# 7 Pointers

Pointers to outside documentation (instructions, forms, README files, offset files, etc.).

TAKE/DerivedData/LAB_MEASUREMENTS.txt - cantons all of the exact measurements of the instrumentation used in this study
TAKE/DerivedData/README.txt - contains information on derived data formats
TAKE/DerivedData/video_time_starts.txt - the start times as seen in the videos, to use for aligning data with the video in ELAN

TAKE/Documentation/Pre - contains all the documents for before a participant started the task
TAKE/Documentation/Post - contains all documents for after a participant completed the task
TAKE/Documentation/Questionnaires - Template and scanned questionnaires as filled out by each participant

# 8 Log

Notes taken on the individual participants during the tasks:

- **r0** described the pieces with his hands, nothing done by the wizard could get him to point, in other words he did not perform the task at at, this led us to change the instructions
- **r1** non-native speaker (Eastern-European), but good pointing and easy utterances
- **r2** refused to point, took a lot of time to refer to a single object
- **r3** did not want to point, sometimes changed the referred piece if the wizard selected the wrong one, went for easy-to-describe pieces
- **r4** lots of pointing and speaking
- **r5** he figured out in the end that it was being controlled (though he did not guess the right person), did not like to point
- **r6** good pointing and references, interesting data, often attempted to select more difficult objects
- **r7** excellent, often used sentence fragments like “das rote oben links” and always pointed

# 9 Conclusions

Overall, we were successful and happy with the data that were collected in this corpus. However, there were some issues that could be handled better in the future. First, in order to log data from the wizard computer, we sent UDP packets to a client listener that then put the data into the logging tool. In many cases, the UDP packets were lost completely, so we attempted to overcome this by sending 10 consecutive events. This has caused some problems with our analysis (duplicates in some cases, and in others even 10 repetitions were not enough and some things weren't logged at all). A better approach would be to send the data once and guarantee data transmission.

# List of Figures