Augmented Meeting Table
Project Report
Multi Modal Interfaces Course

Nathalie Mottet, Tom Forrer, Benoît Pointet

May 28, 2009

Contents

1 Introduction

Are you tired of attending "yet-another-of-those-meeeetings" where everybody talks at the same time, nobody listens and in which no concrete decision is taken ? Then read on . . .

Meeting is a delicate art, which sometimes makes the world go round, but too often fails to achieve its purpose. Methodologies have come to the rescue, based on preparing a coherent meeting agenda with topics, discussions, decisions, . . . but everyone gathers around the table, the meeting agenda and people's goodwill (or even civilized manners) soon disappear.

The "Augmented Meeting Table" project aims at ensuring paces and consistency throughout a meeting by guiding the participants through the meeting agenda, while purveying them with insights and actions regarding the meeting itself.

2 Meeting Description

2.1 Meeting Structure

A meeting has the following (informal) structure:

- a **meeting** contains
  - **participants**, seated ,
  - **topic**(s), each topic contains
    - **discussion**(s)
    - **decision**(s), each decision contains
      - **choice**(s).

This agenda is provided to the system as a `XML` file format (Figure 1). The system aids/augments the meeting at various levels, in various ways. We detail them hereafter.

2.2 Aided Meeting

The predefined timing of each discussion and decision step is garanteed by the system. Each step (including a topic) has aperture and closure phases, lasting a few seconds, that present, respectively summarize them on the table's surface (Figure 2).

```
<meeting title="test meeting : simple discussion">
    <participants>

        <participant seat="2" title="Nath"/>
        <participant seat="3" title="Ben"/>
        <participant seat="4" title="Tom"/>
    </participants>
    <topics>
        <topic title="Project One">
            <discussion duration="2" title="Meeting Engine : a cool tool!" />
            <decision duration="2" title="Is this really cool ?">
                <choice title="yes"/>
                <choice title="no"/>
                <choice title="maybe"/>
            </decision>
        </topic>
    </topics>
</meeting>
```

Figure 1: Agenda of the meeting in a XML format

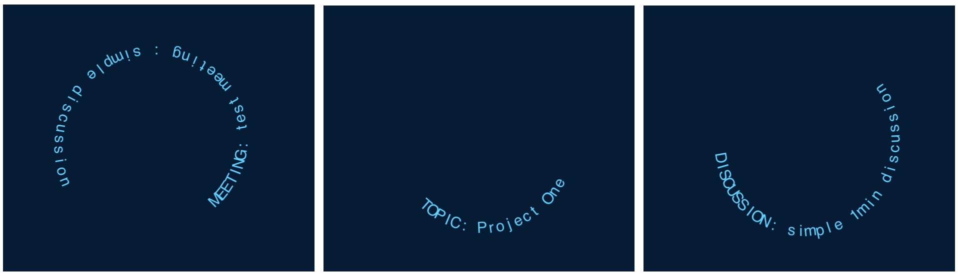

Figure 2: The opening phases of a meeting, its first topic, the first discussion.

2.3 Augmented Discussion

The right to speak is represented on the table's surface by a white bubble that travels from participant to participant following hand gestures and sound inputs (Figure 3). The size of this voice bubble is proportional to remaining discussion time. Soft blow gestures enable participants to orient the voice bubble toward another participant, stealing or giving it to further the discussion. The speech of each participant also attracts the voice bubble proportionally to its volume. The voice bubble traveling on the table's surface leaves traces on each position which accumulates. The map of those traces represent the geography of voice (ab)use.

2.4 Aided Decision

In-between discussions of a same topic, or at the end of them, decisions may come. They are modeled as the democratical vote of the participants for predetermined choices (yes/no, variant 1/2/3, etc.). The proposed choices are represented by circles gathered in the table's center, while every participant's vote sits at in front of him, represented by a smaller discs (Figure 4).

The set of choices slowly rotates to present each choice in front of each participant. Each participant votes for a choice by dragging his vote inside his desired choice with the gesture of a pointed finger (Figure 4).

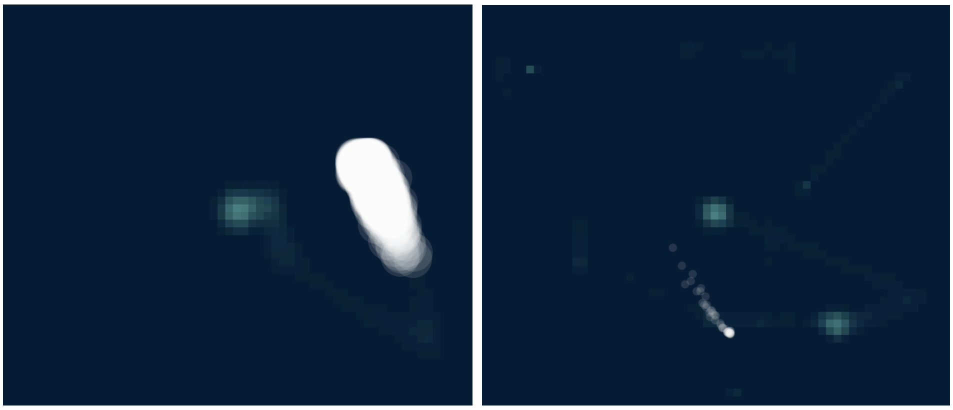

Figure 3: The (foggy white) voice bubble at the beginning and end of a discussion, leaving (green pixelated) traces.

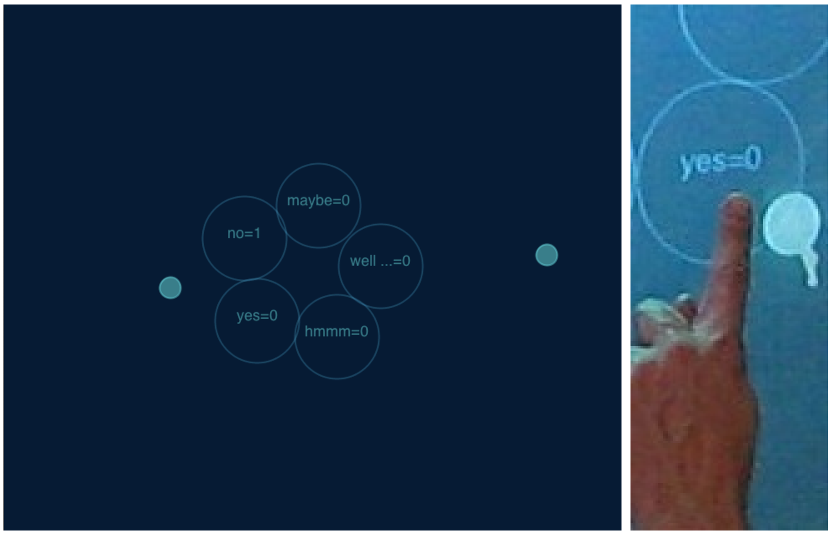

Figure 4: A decision with 5 possible choices (one of which already got one vote) with two remaining votes, and a pointing finger dragging a vote.

3 Architecture & Implementation

3.1 Hardware

The meeting table is basically a horizontal touch screen, around which people sit, equipped with a microphone. Our prototype uses a simple DV camera above the table to track hands on the table surface, and a beamer to project (from under with a mirror) the visuals on the table's surface, which is a matte translucid surface (Figure 5).

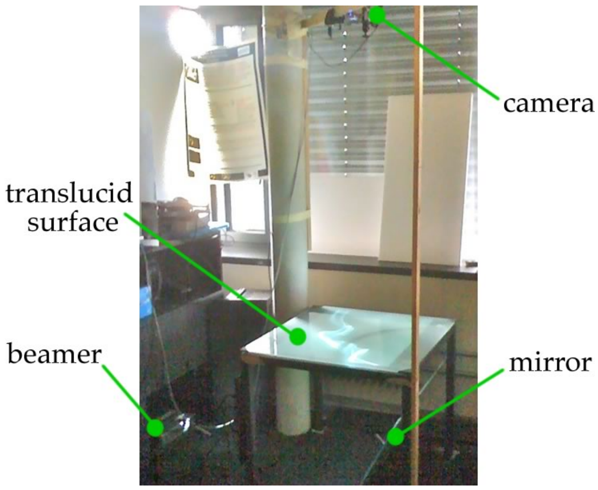

Figure 5: Augmented meeting table

3.2 Software

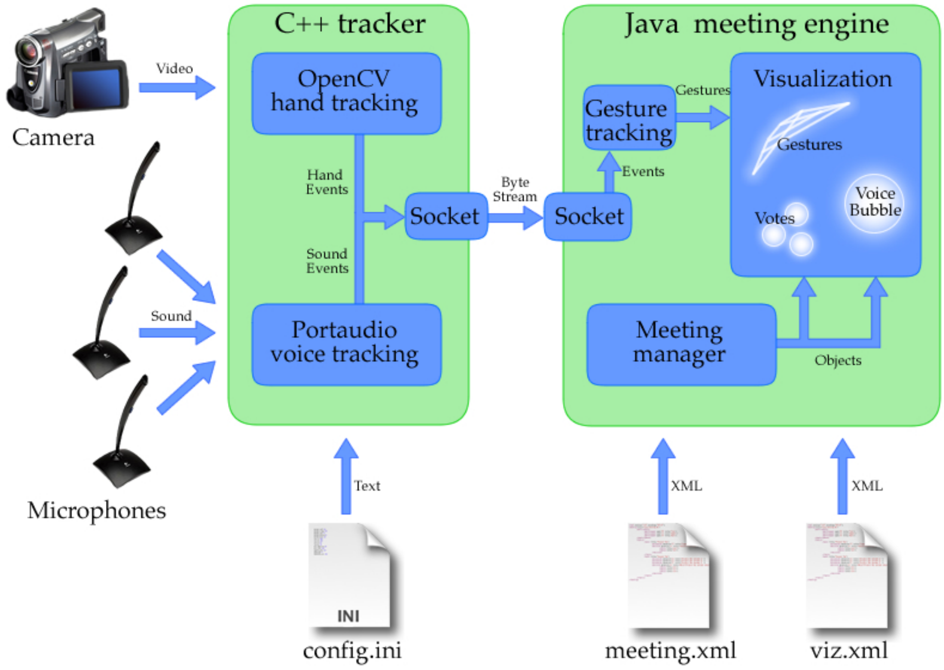

Figure 6: Software architecture

The software is architectured around its mains functions:

1. **Hand and voice tracking**
   C++ based, the hand and voice tracking uses the OpenCV[1] computer vision library and the Portaudio library[2]. OpenCV is known for its speed, its rich catalogue of image manipulation functions and its ease of use in different operating systems and with different cameras. The hand tracking program (Figure 7) uses a Luv color space filtering, finds the contours of each hand and detects the fingers on the contours (Figure 8). The Luv color space has the advantage of having invariant colors with different lighting conditions. Other methods of tracking were tested (Haar classifier, Hu Moments, Canny thresholding, . . .), but the results of the final implementation were unsurpassed in accuracy, indifference to hand orientations, in tracking different gestures and in speed (up to 30 frames per second).
   Originally the Java Sound API was considered for speech detection,

[1]http://sourceforge.net/projects/opencv/
[2]http://www.portaudio.com

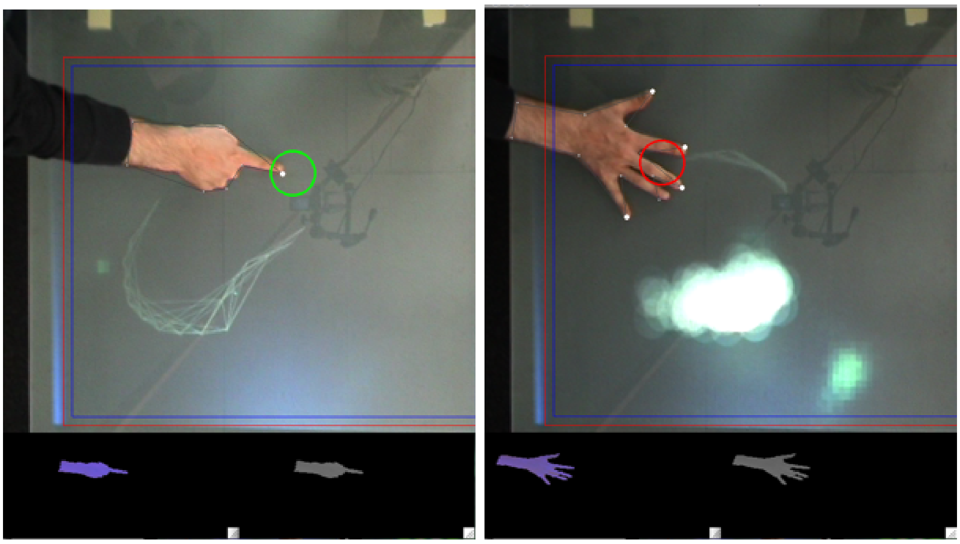

Figure 7: Hand tracking program with two gestures: one pointing finger and a spread hand

but under the used operating system (MacOS X 10.5) there was no possibility for multiple audio inputs. The Portaudio library can do low-level, high-performance audio processing with multiple inputs. It is portable for the most common platforms but uses the native Core-Audio framework on MacOS X.

Figure 8: Hand detection algorithm

2. **Meeting management**
   Java based, uses SimpleXML[3] serialization framework. The meeting management loads a description file, which is fully customizable.

3. **Visual output**
   Java based, uses Processing [4] visualization library. The Processing library is specialized in complex animations and physics simulations. Every aspect of the visualization is also customizable via an `XML` description (i.e. frame rate, voice bubble, dirt stain factor, . . .).

[3]http://simple.sourceforge.net/
[4]http://processing.org/

4. **Gesture tracking**
   Java based, the gesture tracking interprets the hand events received from the OpenCV tracker and provides them to the visualization part of the meeting engine.

Although initially conceived to be implemented on the Java platform, performance issues drove the project toward an hybrid solution, with all tracking happening in C++, while the rest still relying on Java. This induced the addition of a communication channel:

5. **Communication channel**
   Socket based network communication. The tracker application can send the tracked hand and speech events directly to the meeting management.

4 Tests

Testing was done throughout the entire project. OpenCV was first tested in a JNI wrapper, but was considered too slow. Once opting for a hybrid solution, several communication channels were tested. For a working prototype a SQL database (based on the SQLite database engine) was chosen, but once again, the performance was too slow and the socket based network communication was the next choice. In the final software tests, the application was improved considerably in fluidity aspects.
A few informal tests were done on the meeting table prototype. Although the pertinence of the system for real business meeting was not tested, the users highlighted the following aspects:

- gestures are natural, no special glove is required,
- each participant still needs to act with fairplay,
- the hand tracking works surprisingly well,
- there is a small delay when grabbing the camera frame,
- further work in animation fluidity should be done for a final product

5 Conclusions

This project reached the main goal, which was a prototype for a multimodal interface. The different modalities include hand recognition with several gestures, sound input from different microphones, and visualization of speech, gestures, decisions and choices. The prototype is at a functional stage, where gestures and sound inputs are tracked and displayed in a visualization, which reacts to specific gestures. The meeting system aids the participants as expected and augments the meeting in a novel way.

The augmented meeting table could be further developed by adding negative and veto votes, by summarizing the meeting and sending a meeting report by email to the participants.

Although this projects halts at an early stage, the authors were able to meet the challenges they had put in this project, and most important, to have a practical experience of building a multimodal system, rich of learnings and new ideas. They are now fully aware of the necessity of standards along the multimodal chain, and of the possibilities now open by computer vision technologies.