Vrije Universiteit Brussel

FACULTY OF SCIENCE AND BIO-ENGINEERING SCIENCES
Department of Computer Science
Web & Information Systems Engineering Laboratory

Multimodal Interaction for the MindXpres Presentation Tool

Graduation thesis submitted in partial fulfillment of the requirements for the degree of Master in Applied Computer Science, by:

David Bos

Promoter: Prof. Dr. Beat Signer
Advisor: Reinout Roels

AUGUST 2015

Abstract

Presentation tools have become an important milestone in communication media. It is also the first milestone that has been accompanied with so much criticism, being blamed for being a bad stand alone medium to communicate and for elevating format over content. However, we should not characterise presentations by their slides alone, since the bodily and spoken performance of the presenter is equally important for effective communication. Therefore, we analysed how we use and interact with current presentation tools so we could identify current limitations in term of navigation and free creation of content. In current presentation tools we can for example only traverse slides linearly and there are no room for free creation of content, making more complex narrative difficult. Once these limitations were identified, we proposed a solution that solves them by integrating multimodal interaction.

In our solution, we introduced new features into MindXpres, a content-oriented presentation tool, which enables multimodal interaction. By using gestures which are all familiar to us, such as pointing and touch, we were able to offer a solution that feels natural to use and that copes with the identified limitations.

Acknowledgements

First I would like to thank my promoter Prof. Dr. Beat Signer for giving me the opportunity to conduct my research. I also would like to thank him and my advisor Reinout Roels for their patience and continuous support during the process of making the thesis.

Next I would like to thank my girlfriend and my family for their continuous support. I could always count on them to help me or to get suggestions for improvement.

Contents

1 Introduction

Presentation tools such as Microsoft PowerPoint[1], Apple Keynote[2] or OpenOffice Impress[3] have become an important milestone in communication media. Slideshare[4] alone, which distributes and shares presentations, has 216 million views a month. However, contrary to the overhead projecter from which digital slideware have evolved, slideware has been accompanied with extensive criticism. The base for this criticism lays in the cognitive style of these tools. They all lack a high resolution, interaction and make use of similar templates which point the user to create presentations which are full of so-called PowerPointPhluff:

> *"serious analysis with chartjunk, over-produced layouts, cheerleader logotypes and branding, and corny clip art"*
>
> *"weakening verbal and spatial reasoning, and almost always corrupting statistical analysis"* [45].

The criticism further blames current slideware for not only the Columbia disaster, but also for far more widespread communicative disasters in business meetings and lecture halls [45].

The presentation tool MindXpres was developed with this in mind, taking a new approach on how to create, share and deliver presentations. It differentiates itself by providing new features such as non-linear traversal,

[1] http://office.microsoft.com/en-us/powerpoint/
[2] http://www.apple.com/iwork/keynote/
[3] http://www.openoffice.org/product/impress.html
[4] http://www.slideshare.net/

a zoomable interface, new ways of visualising specific types of information, semantic highlighting and so on. Presenting us with a second problem in current slideware: the lack of interaction support. Most of the interaction in digital presentations is still done with a mouse and keyboard. Supporting such a limited amount of inputs restrains the functionalities. Although there exist multiple solutions for enhanced interactivity for the most common presentation tools, they often require expensive specialised hardware and do not try to enhance the tools themselves.

So the question we ask ourselves is: *how do we interact with slideware during a presentation?"*. A first obvious interaction is navigating through the slides. In current slideware, this is a simple task, since the slides themselves have a linear traversal. Nevertheless, this requires the presenter to go to his computer and use the keyboard or use a remote if one is available. When using MindXpres, we can also have a non-linear traversal. Needless to say that a remote won't do anymore thus obliging the presenter to move towards his computer, unless he only makes use of linear traversal. Furthermore the presenter will often point towards his presentation. To point things out is to give meaning to something. We could even say that if something isn't pointed out, it has no meaning, it isn't important [4]. This is often done with a pointer, a stick or by hand. In Chapter 2, through a literature study, we'll start our investigation on how current slideware tools are used and what we use them for. We also reveal the hidden effects of slideware using McLuhan's power points. Only by fully understanding how we use a specific technology, and how it affects our habits, can we know where and how to improve it. We also take a look at the importance of interaction during a presentation, as we should consider slides as enactments in which speech and images are interrelated with technology and media [25].
Chapter 3 will first discuss multimodal interaction. We'll take a look at its history, its objectives and possibilities. We then look at some related work in the context of presentations, with their advantages and disadvantages.
Chapter 4 will introduce our ideal presentation tool, and how MindXpres is part of this solution.
Chapter 2 and 3 will thus be used as the foundation for the proposed solution described in Chapter 4. We'll use what we've learned from literature and related work to integrate multimodal interaction, together with a multimodal interface, for the MindXpres presentation tool. The actual solution and implementation is discussed in Chapter 6, based on the technologies introduced in Chapter 5. All of the previous work is then followed by a use case in Chapter 7. And finally, in Chapter 8 we discuss the contributions together with possible future work.

2

Limitations of Current Slideware

In this chapter, we'll take a closer look at the literature about common presentation tools like PowerPoint, Keynote and OpenOffice Impress. Because these tools were designed with keyboard and mouse interaction in mind, we would like to analyse what changes are required if we want to integrate multimodal interaction. Therefore, we'll investigate how we use and integrate current slideware in our daily lives.

2.1 Common Slideware

Most of the literature talks about Microsoft PowerPoint. However, because of the similarities between each of these tools, sharing a similar intuitive graphical editor for slides with support for multimedia, charts, animation and effects, we will consider PowerPoint, Apple Keynote and OpenOffice Impress to be equivalent.

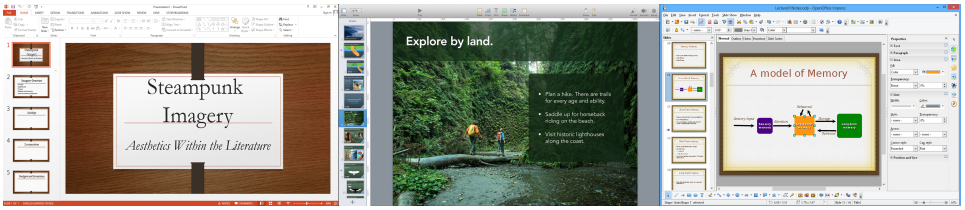


Figure 2-1: Microsoft PowerPoint, Apple Keynote and OpenOffice Impress side to side.

2.1.1 Criticism on Common Slideware

PowerPoint has become a preferred method of communicating, presenting and sharing knowledge. However, a lot would agree by saying that presentations often leave something to be desired. This can partly be solved by knowing how to make good presentations, and a lot of literature has emerged on doing just that. However, not everybody agrees that it is just a problem on how we use PowerPoint or other common slideware, but rather a problem with the tools themselves. Tufte [45], who generated a tsunami of criticism, goes as far as calling PowerPoint 'evil'. He says that PowerPoint has the worst signal/noise ratio of any communication method known on paper or on the computer screen and that it 'elevates format over content', turning everything into a sales pitch.

2.1.2 Habits of mind

Following his footsteps, some further interesting analyses were made on explaining *why* PowerPoint may not be the best way to communicate, present and share knowledge.

> *"There is a deep link between humankind and our machines. Our tools or techne extend our reach, abilities, sensory perception, locomotion and understanding. In adopting a tool, we invite it to enhance, or more dramatically transform what we do and how we perceive the world."* [4]

Since PowerPoint is heavily used in not only the office, but also in classrooms and online, we are changing the way we share and gather information. An

example to support this statement, as given by Catherine Adams, is the usage of mobile phones. Mobile phones have drastically changed the way we keep in touch with each other, challenging and re-framing notions such as availability and autonomy.
We can safely assume that every new technology, such as slideware used in classrooms, embodies a new *form of thinking* [4]. The question we ask ourselves now is how PowerPoint and other common slideware affect our habits of mind? Once we can answer this question, we can also *change* it, and propose a solution.

2.1.3 McLuhan's Power Points

McLuhan composed 'four laws of media' to reveal the hidden effects of technology [29]. These are:

- What does the medium enhance or intensify?
- What does it render obsolete or displace?
- What does it retrieve that was previously obscolesced?
- What does it produce or become when pressed to an extreme?

The responses to these questions compose a tetrad. This tetrad focusses its attention on simultaneous effects of the technology. Catherine Adams composed following tetrad for PowerPoint:

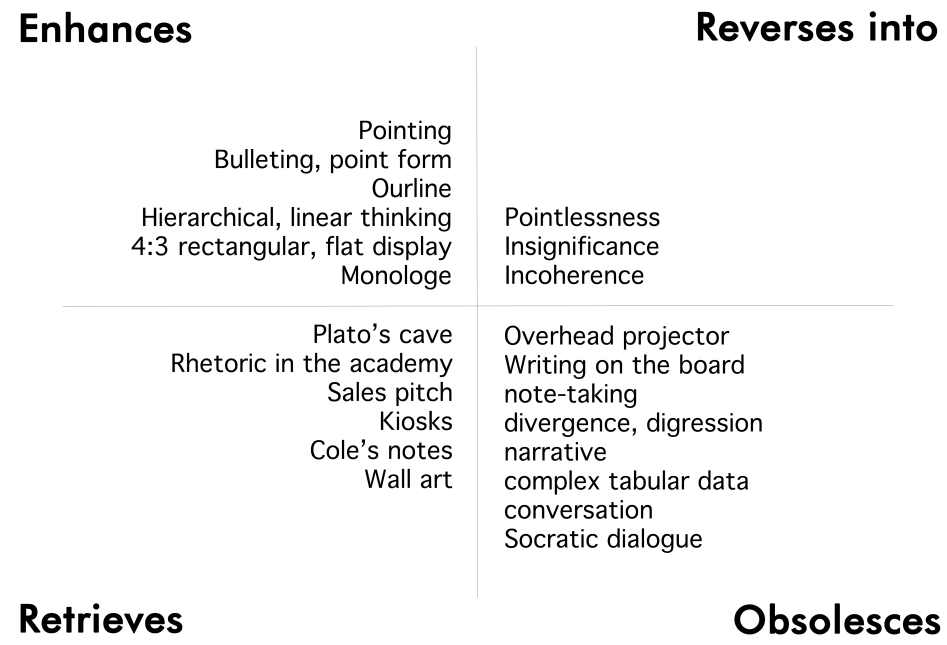


Figure 2-2: PowerPoint tetrad, Catherine Adams

Through common slideware, the presenter can point more accurately, vividly and rapidly at text and images. *Pointing*, or the act of signifying, is a central pedagogical practice. It goes as far as saying that, when something isn't pointed out, that is has no significance [4]. This has its downside. If something that is being said is not on the slides, people often perceive it as being *insignificant*. And when pointing at everything, you are *pointing at nothing* at all. The *Columbia* Shuttle disaster demonstrates this wonderfully. Because every bullet in the slides presented critical information, the most critical point, 'buried' several levels deep, couldn't be discerned from the rest and the meaning of this information passed unseen.
PowerPoint favours information that can be displayed on a single slide, that is, a *4:3 or 16:9 rectangular*. All information that does not fit often has the disadvantage that it must be abbreviated, thus suffering loss of information. (ref) There is the possibility to distribute information over multiple slides, but in opposition to a book, the audience does not have the possibility to look back, making complex *narrative* difficult [4]. This is further enforced by the *linearity* of the slides which forces the presenter to think in a sequence of 4:3 frames. However, the linearity is also a strength. It helps mapping out a clear, singular course to follow.
Because of the reasons mentioned above, the audience will more often experience a projected product, the *'sales pitch'* and less often the *process* of the knowledge-in-action. In other words, it reclaims *rhetoric or persuasive*

speech, aiming straight at the mark. PowerPoint is designed to eliminate 'unnecessary' sub-steps, favouring the most efficient path to an end, resolving in its turn in Cole's notes. Cole's notes are a simplified version of some complex book, instruction, or narrative, often used by students while studying.
PowerPoint is completely presenter oriented, rendering *conversation* obsolete. It enforced a *monologue* from the presenter towards his audience. It renders *Socratic dialogue*, a form of teaching and learning that involves the flowing juxtaposition of like and unlike ideas over time in complex discourse, impossible since this kind of conversation is not easily transferable to a predetermined slide format.
Another strength and weakness is the ease in which one can show digital content. It renders obsolete direct experience or apprehension of the world compared to the whiteboard alone, since one can show an image or video of an artefact on the *flat display* instead of bringing it to class. The actual is more and more replaced by the virtual. In a certain way, it revives *Plato's Cave* [4]. Rather than the shadows projected on the wall, the audience now witness a projection of bright light upon the wall to share knowledge.

2.2 The Importance of Interaction

The arguments criticising PowerPoint have become quite popular and many seem to agree, at least partially, with Tufte's remarks and opinions. However, not much has been said about the factual use of common slideware. It is equally important to consider how the usage of common slideware affects the *performance* of speakers and audiences. In other words, we can't just reduce technically supported presentations to slides, texts and visualisations only, but we should consider them as enactments in which speech and images are interrelated with technology and media [25]. A presentation can thus be considered as a communicative event, characterised not only by its slides, but also by the bodily and spoken performance, as by the activities of audiences.

2.2.1 Forms and Functions of Pointing

A study has been made on the performance of such presentation, with the focus on pointing, since pointing is most specific to this genre of communicative event [25]. When we speak about pointing, we refer to body movements and gestures which cannot be understood without additional contextual information, sometimes accompanied by speech, typically occurring within interaction. It is an interaction that we are able to perform without explicitly knowing when and how.
The meaning of gestures can be reconstructed by identifying the particular gestural forms that 'carry' meaning [23]. In the case of pointing, we can distinguish iconographic gestures that appeal to visual images or mimetic gestures that mimic other processes. Most popular is pointing with the index finger, but a lot of technical instruments are also available to support pointing: sticks, pens and pencils as well as a computer mouse can be used to point in PowerPoint presentations. The most popular for common slideware is the use of the laser pointer. As noted by Hubert Knoblauch, the fact that pointing also includes pointing by technical aids means that the signification of pointing is not dependent on the sign itself as gesture studies would suggest. Pointing in this case is directly related to other aspects of communication: speech and visuals. It is this relationship which make certain passages of a presentation decipherable: the presenter can point at an illustration to make his talk decipherable. Pointing movements can also be understood as interpretations of what the speaker is saying and what he is hinting at [25]. It is safe to say that pointing does much more than just refer to something given. Its the interplay between pointing, speech and visuals that give meaning. Taking one of these out of the equation leads to a reduce in communicative performance of the presentation as a whole.

2.2.2 Second Degree of Pointing

Not only by pointing, but also by changing the body or face orientation, the speaker can draw the attention from or to the screen. While we can hardly call this a gesture, it somehow performs the function of pointing. The same can be done by a slide on which not much information is shown, but which supports the talk of the speaker visually to create a duality in the structure of the talk. The slide is then not decisive for understanding the presentation, but it is the spoken text in combination with certain elements of the slide which create a dual structure. This paralleling between the spoken and visual can be understood as pointing since what is said becomes evident by being seen and vice versa.

2.2.3 Reflecting on Tufte's Criticism

When criticising PowerPoint, Tufte ignored the fact that presentations as a whole cannot be reduced to the slides alone. It is important to notice that the communication process is much more than what is shown on slides alone. PowerPoint integrates the visual media into an overall structure of speech. The typical slide does not much more than illustrating what is being said. Pointing thus produces a 'surplus' of meaning that allows us to understand what goes on [25].

2.3 Doumont's Response

The same remarks have been made by Jean-Luc Doumont in his paper responding to Tufe [14]. A first highlight in his argument is that *oral presentations are not written documents.* A presentation should thus be a complementary communication media which try to convince the audience of key messages with selected evidence and non-verbal communication. Another key point however is that the *tool is not the product.* It goes without saying that many people see the product as being different if it were transparencies or Powerpoint slides. The end product should be the same, however, because of the emphasis slideware put on deeply hierarchical bullet lists and the preoccupation with format over content, there exists a confusion between the end-products and their tools. While, according to Doumont, Tufte uses a faulty reasoning, he's not denying that Powerpoint's overall structure does not help us by avoiding common mistakes in the creation of slides. There are still too many presentations out there which primary function is for the speaker to remember his text, or worse, to simply read his text from to slides to his audience. Slides should help the audience to understand the material better. But how can we guarantee that our slides do just that?

2.3.1 Doumont's Guidelines

Doumont believes that it is not an impossible mission to create slides which are helpful to the audience. To do this, he created a set of guidelines based on his the *laws of communication.*

- Adapt to your audience
- Maximize the signal-to-noise ratio
- Use effective redundancy

An audience should furthermore be able to understand the message of the presentation without the slides. And the slides on their own should be clear enough so that a deaf person should be able to understand the message by looking at the slides. Effective slides are thus redundant, stand-alone and visual. They integrate both the verbal and the visual and contain a message by using a minimum of text, generally framed in complete sentences [14]. G. Gross et al. believe that, in order to make a good presentation, a second condition needs to be fulfilled: we need to be able to grasp the relationship of a single image to *the narrative or argument* in which they are embedded [20]. This can be done by carefully integrating every slide as being part of the narrative or argumentation that is being used when speaking towards the audience.

2.3.2 Beyond Bullet Points

The book "Beyond Bullet Points" by Cliff Atkinson [6] describes in detail how to make a good presentation, as well as how to present one. While it doesn't explicitly mention the Doumont's guidelines, it follows them well. We are not going to cover how to make a good presentation in detail, as this falls out of the scope of this thesis. Some aspects are worth mentioning though, as they are complementary to the analyzes made in previous sections. For example, the book immediately starts off with an example where the interplay between the presenter and his slides is clearly described:

> *As he began speaking, Mark's thumb pressed the button on a remote control device which he cupped in his hand at his side where the audience would not notice it. This remote would be his constant companion for the next couple of hours, as he used it to advance the PowerPoint slides while he spoke, giving him the flexibility to slow down or speed up to match his narration and ensure that the experience appeared seamless to the jurors.* [6]

This description shows immediately the importance of interplay between the presenter and the presentation towards the audience. The audience is carefully considered in the design for a presentation, such as the consideration of the *working memory* of the audience when preparing a presentation. People tend to learn better from a presentation when information is split up in digestible pieces. Every slide should thus contain only one main idea, so that your audience can absorb the information one piece at a time [42]. However, one should also be aware of the previous knowledge of their audience when working on the working memory, because while the working memory is limited in its capacity for new information, it is unlimited to process existing information from long-term memory [6] as illustrated in Figure 2-3.

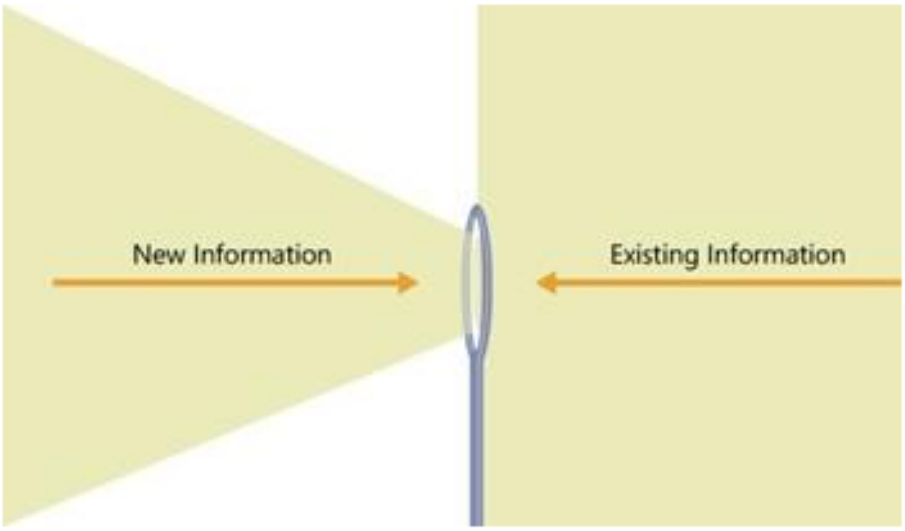


Figure 2-3: Capacity of new information versus existing information.

Finally, two following topics from the book "Beyond Bullet Points" [6] are also worth mentioning:

- *The difficulty of linearising thought*: Ordering all the gathered information and put them inside a presentation in a linear way is a difficult task. While the linearity is a strength as well as a weakness, in common slideware it renders complex narrative much more difficult because the presenter does not have the possibility to quickly jump back to information available on a previous slide.

- *How to handle Q&A effectively*: The book proposed "as an advanced technique" to use the PowerPoint storyboard on screen as a navigational aid during Q&A. This enables the presenter to quickly jump to a certain slide if more information on the related topic would be requested, as can be seen on Figure 2-4

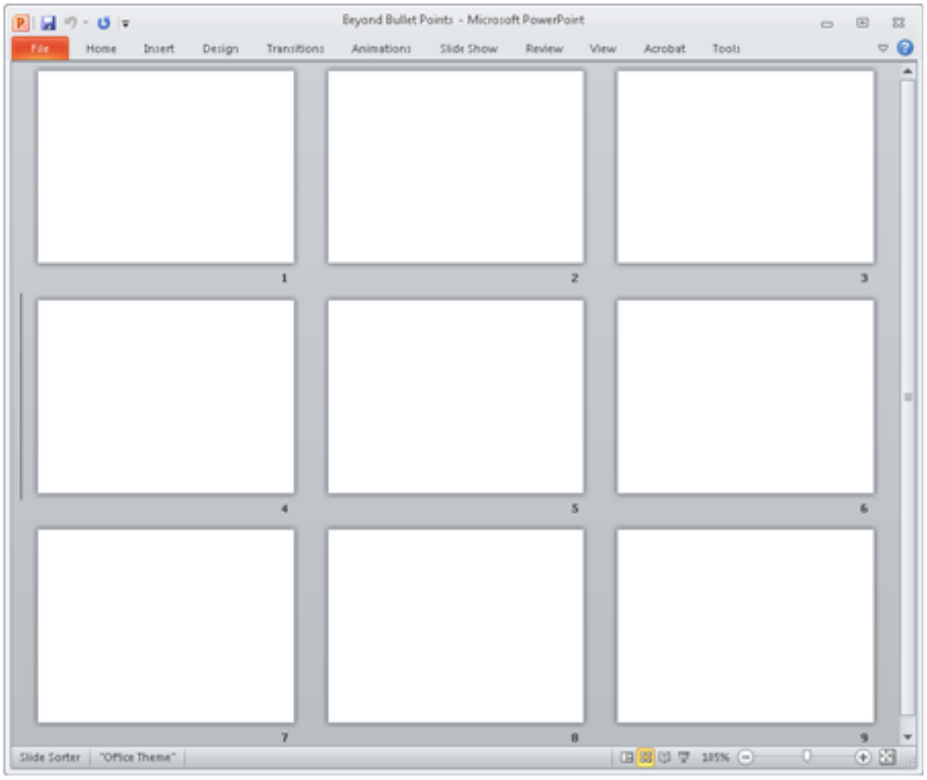


Figure 2-4: Handle Q&A effectively.

2.4 Conclusion

In this chapter, we have taken a closer look the literature about common slideware. Current slideware is often criticized for elevating format over content and having a bad signal/noise ratio. Using McLuhan's power points, we analysed common slideware in more detail and discussed major implications on our habits of minds. We also found that the interaction with our presentations play an important role in their performance. It is thus irrelevant to reduce presentations to the slides alone, but we should consider them as enactments in which speech and images are interrelated with technology and media. However, the fact remains that the tools themselves would benefit a lot of from improvements since the guidelines that we saw on how to make presentations better, too often try to get around the default structure that common slideware impose.

3
State of the Art

There exist a lot of tools that enhance interactivity through multimodal interaction and multimodal interfaces. Multimodal interaction is a cross-disciplinary field with its foundations in cognitive psychology. This chapter will take a look at the history of multimodal interaction and multimodal interfaces, some new concepts that multimodal interaction and multimodal interfaces introduce, as well as some better known multimodal interfaces available for presentations.

3.1 The History of Multimodal Interfaces

The first multimodal interface can be traced back to 1980, where Richard A. Bolt presented his "Put-That-There" [11]. In his paper, he introduced an interface where he made use of various modals to control a spatial user interface. He combined multiple input modalities as input for the commands of the interface, with the intention of providing the user with a more natural interaction. Making use of two new technical offerings at the time, speech recognition and position sensing in space, the user could point to a shape and command the interface to move it, remove it or to create a new shape at the area he was pointing to as can be seen in Figure 3-1.

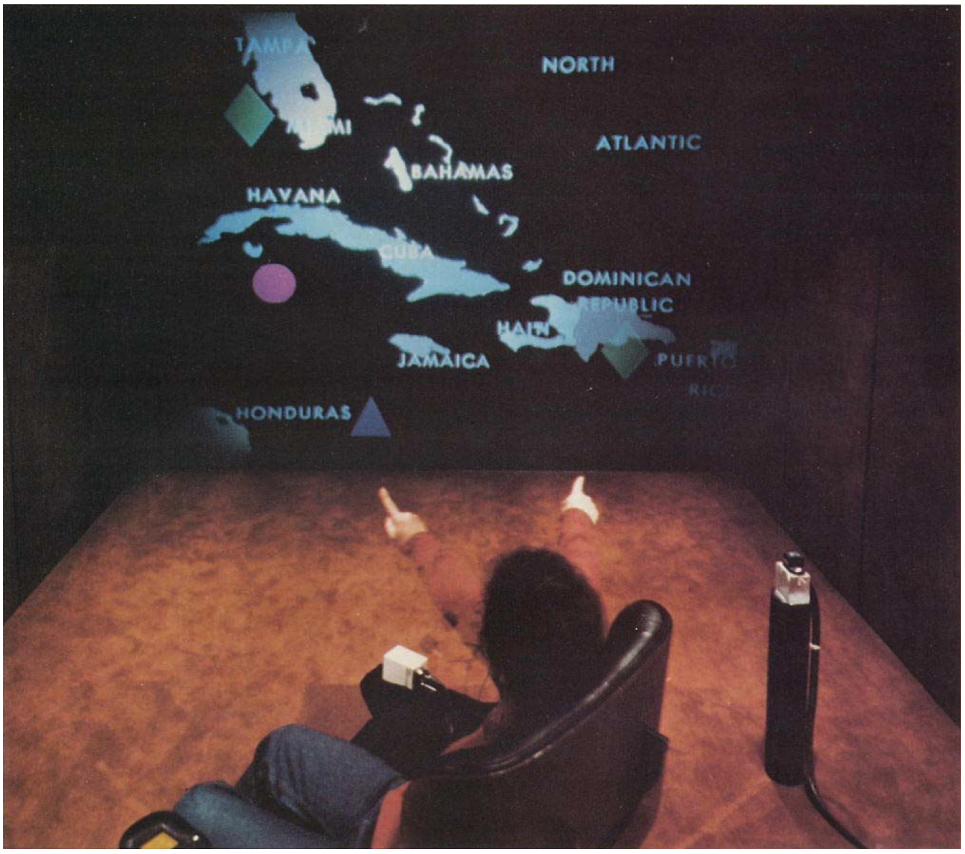


Figure 3-1: Media Room

3.2 Multimodal Interfaces

Multimodal interfaces target a more natural way of interacting with computers by offering a set of "modalities" to users. According to Oviatt [34]:

> "*Multimodal interfaces process two or more combined user input modes in a coordinated manner with multimedia system output. They are a new class of interfaces that aim to recognize naturally occurring forms of human language and behavior, and which incorporate one or more recognition-based technologies.* "

To do this, multimodal interfaces need to fusion different types of data and process them in real-time given some temporal constraints imposed on information processing [33]. Therefore, multimodal interfaces are considered to be a new class of user-machine interfaces, different from WIMP (windows, icons, menus, pointer) interfaces. They support users' perceptual and communicative capabilities and integrate computational skills of computers in the real world, by offering more natural ways of interaction to humans [15].

3.2.1 Cognitive Foundations

Multimodal interfaces target a more natural way of human-machine interaction. Therefore, the foundations of multimodal interfaces come from cognitive psychology. The cognitive psychology tells us that humans are able to process modalities partially independently and thus using multiple modalities increases human working memory. Human performance is thus improved when interacting multimodally.
Mousavi et al. [32] experiments suggested these statements by showing students content partly visual and auditory. By combining both instead of only auditory or visual content, the effective working memory could be increased. These findings were later confirmed by Tindall-Ford et al. [44] who did similar experiments using multimedia learning material. Oviat et al. [35] used these finding in to experiment different interfaces, where she found out that multimodal user-interfaces minimised cognitive load which improved student performance.

3.2.2 Guidelines

The user interface design is based on the user requirements and system capabilities within a given domain. However, since there is a growing interest in multimodal interface design, some general considerations are given [38]:

- *Design for the broadest range of users and contexts of use*: If we want our application to be accepted and valued by a lot of users, it needs to be usable for a lot of users and in more than one manner. Since multiple modalities

support flexibility, a good multimodal interface should support the best modality or the combination of modalities in changing environments. For example, keyboard input in noisy environments or speech when nobody's around.

- *Address privacy and security issues*: According to the user's preferences, different modes of the interface should be enabled. For example, if the user does not wish to use speech in public, a non-speech input modality should be available.
- *Maximise human cognitive and physical abilities*: Combining different output modalities such as combining audio and visuals lowers cognitive load.
- *Integrate modalities in a manner compatible with user preferences, context, and system functionality*
- *Error Prevention and handling*: Complementary modalities can be used to improve robustness. A multimodal system can give users control over modality selection so they can avoid errors for given lexical content.

3.3 Multimodal Interaction

Following concepts are popularly accepted for multimodal interaction: fusion, fission, dialog management, contextmanagement and time-sensitive architectures [15]. Using these concepts, Figure 3-2 can be drawn using the major concepts that should be considered when building a multimodal system.

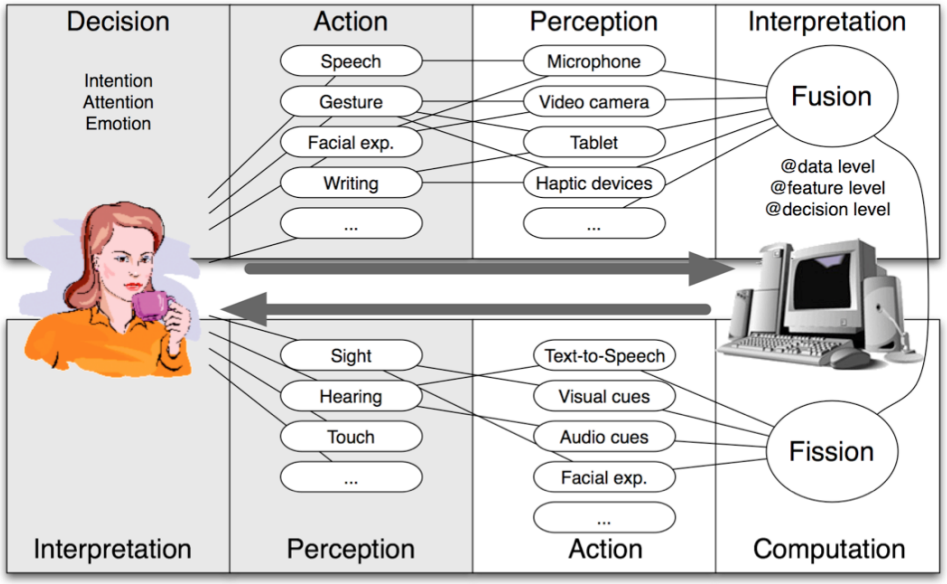


Figure 3-2: A representation of multimodal man machine interaction loop [15].

The communication between a human and a machine can be divided in four different states. First the user is in a *decision state*, in which the communication message content is prepared consciously for an intention, or unconsciously for attentional content. Then comes the *action state* where the communication means to transmit the message are selected. The machine in turn will capture the message through capturing the different communication means with different modules. At first, in the *perception state*, the machine will gather information from one or multiple sensors. In the *interpretation state*, the multimodal system will try to give meaning to the different information collected in the perception state, typically using fusion. In the *computational state* action is taken following the business logic and dialogue manager rules defined by the developer. And finally, in the *action state* an answer is transmitted back to the user, in which a fission engine will determine the most relevant modalities to return the message.

3.3.1 Fusion of Input Modalities

One of the features that distinguish multimodal interfaces from unimodal interfaces is the fusion of information coming from different input sensors. Fusion extracts meaning from the input modalities and passes it to a human-machine dialog manager, where business logic and dialogue manager rules determine what to do. Fusion can be done at three different levels: at *data level*, at *feature level* and at *decision level* [15].

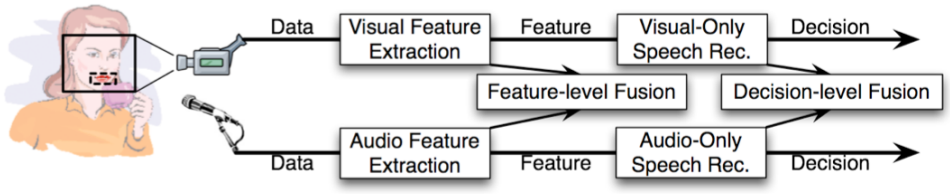


Figure 3-3: The different levels of multimodal fusion [15]

Data level fusion is used when very similar modality sources are being used. No loss of information occurs as the signal gets processed immediately. This advantage is also its disadvantage, since it is highly susceptible to noise and failure.
Feature-level fusion is used when tightly-coupled or time synchronized modalities are used. This type of fusion is susceptible to low-level information loss but it handles noise better.
Decision-level fusion is the most common type of fusion. It manages loosely-coupled modalities. It has a low failure and noise sensitivity since the data has been preprocessed. However, it means it has to rely on the quality of previous preprocessing of data.

	Data-level fusion	Features-level fusion	Decision-level fusion
Input type	Raw data of same type	Closely coupled modalities	Loosely coupled modalities
Level of information	Highest level of information detail	Moderate level of information detail	Mutual disambiguation by combining data from modes
Noise/failures sensitivity	Highly susceptible to noise or failures	Less sensitive to noise or failures	Highly resistant to noise or failures
Usage	Not really used for combining modalities	Used for fusion of particular modes	Most widely used type of fusion
Application examples	Fusion of two video streams	speech recognition from voice and lips	Pen/speech interaction

Figure 3-4: Characteristics of fusion levels [15]

Decision based fusion typically makes use of following types of fusion:
Frame-based fusion uses frames for meaning representation of data coming from various sources or modalities, represented as attribute-value pairs.
Unification-based fusion recursively merges attribute-value structures to obtain a logical whole meaning representation.
Symbolic/statistical fusion adds statistical processing techniques to the fusion techniques described above.

3.3.2 Fission of Output Modalities

When multiple output modalities are available, such as audio cues, visual cues, ..., we need to select output modalities to adapt to a context of use, the type of task or the type of user.
Fission consists of three tasks:
Message construction, where the information which gets transmitted back to the user is created.
Output channel selection, where interfaces are selected according to context and user profile in order to convey all data effectively in a given situation.
Construction of a coherent and synchronised result: when multiple output modalities are used. Coordination of layout and temporal constraints need to be taken into account.

3.4 Interactive Whiteboards

Interactive whiteboards (IWBs) are the most well-known multimodal interfaces used in classrooms and office meetings [41]. It are mostly large, touch-sensitive boards which control a computer connected to a digital projector where the presenter can interact with the whiteboard by touch, using his fingers or by using a non-ink pen tool. However, similar systems exist without the requirement of a large touch sensitive board. These solutions provide the same functionality, be it using cheaper hardware such as infrared pens.

3.4.1 SMART Boards

SMART board is a large touch-sensitive board. It was first introduced in 1991, combining the simplicity of a whiteboard together with the power of a computer. [10] There are other manufacturers who have similar touch-sensitive boards, but we will not list them all in this thesis.

Figure 3-5: SMART board

In addition to the possibility to interact with the computer using touch, which basically emulates a mouse, additional software is added to maximise interaction opportunities. In the case for smartboards, a software called SMART Notebook is available.

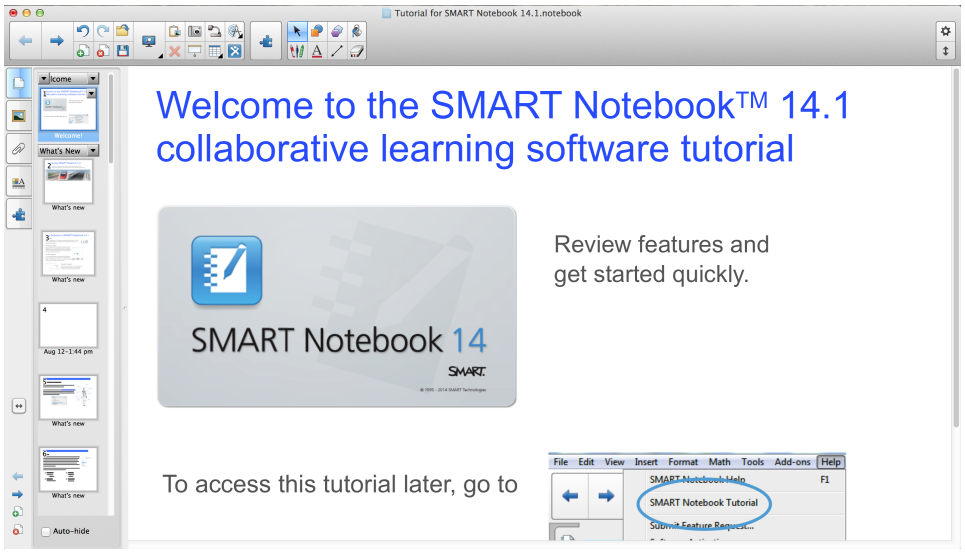


Figure 3-6: SMART Notebook

On first sight, SMART Notebook is a lot like Microsoft PowerPoint as can be seen on Figure 3-6. Both interfaces share the same intuitive graphical slide editor, where one can easily import images, add text or other media files with a simple drag and drop. It is even possible to import existing PowerPoint presentations into SMART Notebook.
But once you start using the software, more and more differences start to emerge. In PowerPoint, a teacher is in some ways charged with re-fashioning a space especially designed for office use [4]. For example, when creating a new presentation in PowerPoint, the template will "invite" you to create a title slide followed by slides with a center title above a box of bulleted, textual information or points. A user is thus invited to add first the title, and then bullets. If he does not want to add bullets, he is forced to first erase the bullet and adjust the text placement if no prior custom template has been made. In SMART notebook, there is no such template, hence there is no "invitation" to make a title and add bullets. The slide is just blank. According to the reasoning of Tufte, this is actually not a bad thing, since this "invitation" is part of the cognitive style of PowerPoint that is criticised [45].

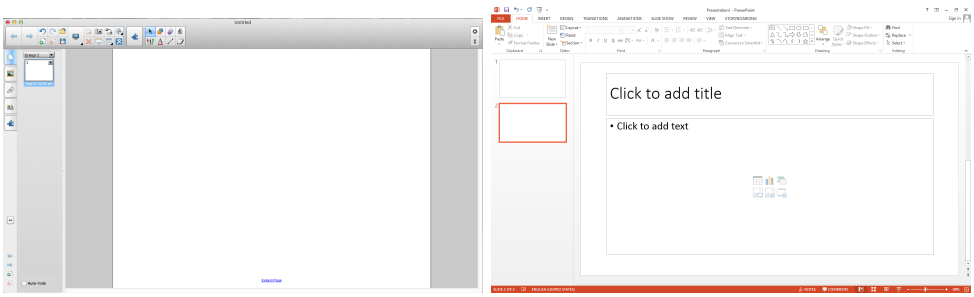


Figure 3-7: SMART Notebook new slide versus Microsoft PowerPoint new slide

Then there is the possibility to annotate the page. Since SMART Notebook was developed for SMART boards, annotating is easy with the non-ink pen tools while using touch will give you the ability to select and drag objects on the slide. PowerPoint has no such tools, unless in presenter mode where PowerPoint also has the possibility to annotate the slides as can be seen in Figure 3-8.

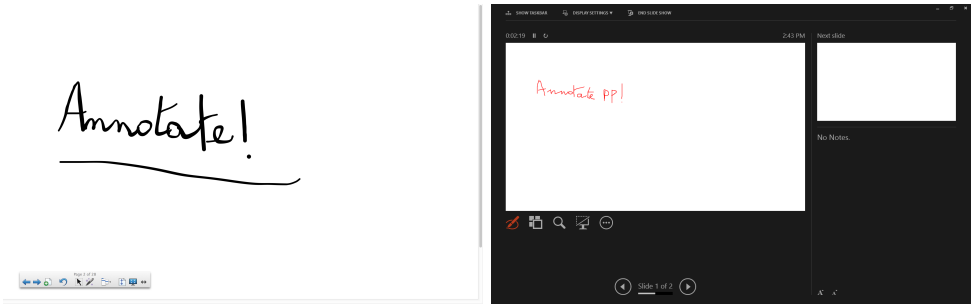


Figure 3-8: SMART Notebook presenter view versus Microsoft PowerPoint presenter view

Note that PowerPoint supposes the presenter view to be on a separate screen, dedicated to the presenter. With SMART Notebook, designed to work with a SMART board, no such separation exists. The full functionality that one has as when creating the slides is always present, to easily offer the possibility to easily add slides, insert images or animations as he sees fit directly from within the same presentation. These animations are a big part of the SMART Notebook software. For example, SMART notebook provides a math module where you can write equations and it gets plotted immediately on a graph, with the possibility to change it on the go.

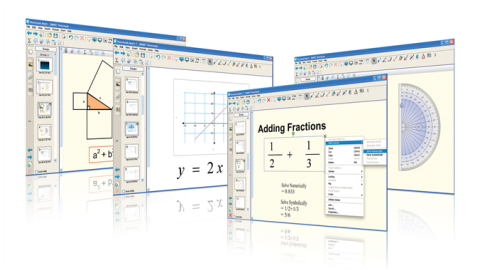


Figure 3-9: SMART Notebook math tools

Finally, another addition is the so-called "Activity" pages. There are some pre-built examples to use, but the presenter can make some themselves. These activities are designed to augment interactivity in class, and can be seen as small flash-like applets designed for educational purposes. Figure 3-10 is such an example, where a magnifying glass reveals cities when it moves over the map.

Figure 3-10: SMART Notebook Activity

But SMART didn't stop there. Additional software has become available to enhance its product. It also provides an Audience Response System (ARS), calles SMART Response interactive system among other collaboraive learning software to enhance the interactivity with the audience. This software is out of the scope of this thesis, since ARS focus on the interactivity with the audience, this we will not discuss it any further.

3.4.2 eBeam

eBeam is a more recent technology developed to mimic a large touch-sensitive board. It does this by using a infrared and ultrasound receivers which will scan for the positioning of a transmitter-equipped pen, as can be seen on Figure 3-11. Because of that, it does not support touch, but all the same functionality as a regular interactive whiteboard is available through similar eBeam software by making use of the pen as input device.

Figure 3-11: eBeam technology

The main advantage of this technology is its portability and flexibility. As long as you have a projector and a whiteboard, eBeam technology will enable interaction on the screen. It is also, as one can imagine, a much cheaper solution.

3.4.3 Wiimote and Smoothboard

In 2007, Johnny Chung Lee developed his own interactive whiteboard by using a Wiimote and an infrared pen. Although he did not provide any additional software for interactive use, he shared the source from his project so everybody could make his own low-cost interactive whiteboard. The principle is quite similar to that of the eBeam. The infrared camera of the Wiimote will capture the infrared light of the pen when it is activated to calculate its position. Once an infrared light gets captured, it will emulate a mousepress in order to interact with the PC. The setup of his solution can be seen on Figure 3-12

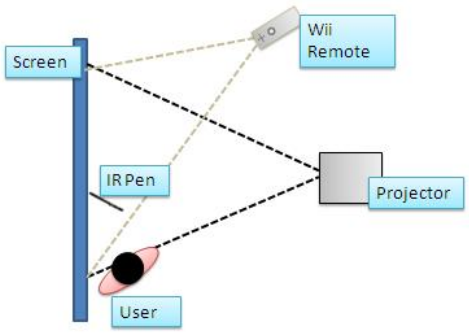


Figure 3-12: Wiimote Setup

This quickly led to the development of Smoothboard. Smoothboard uses the Wiimote as its primary input device, instead of a touch-sensitive board,

and which gives you the same possibilities as the SMART Notebook software described above in Section 3.4.1. However, it is not a standalone solution to replace common slideware, such as SMART Notebook.

3.4.4 Open-Sankoré

Open-Sankoré can be seen as the open source brother of Smart notebook. It provides a similar interface as can be seen on Figure 3-13, offering the same empty slide templates and an easy way to incorporate various media into each slide. It also has the same so-called activity pages, which are basically dedicated applets to illustrate a certain educational material interactively. Thanks to Open-Sankoré being open source, it has a big community creating new content every year which tutors can use in their teaching.

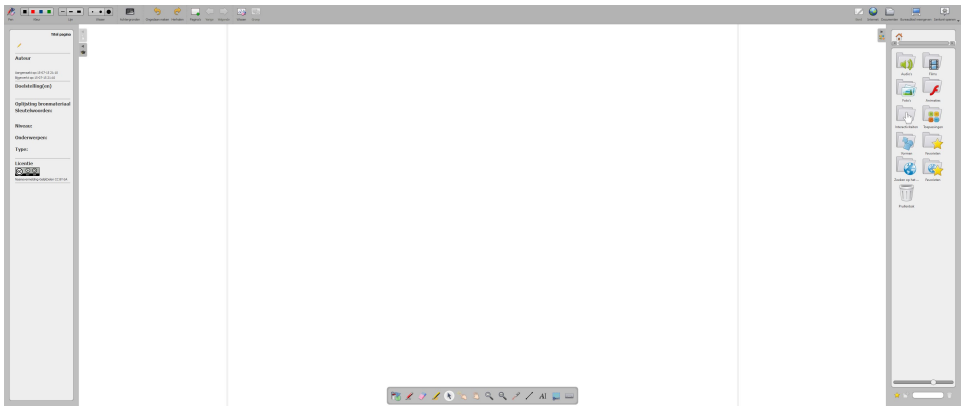


Figure 3-13: Open-Sankoré

Open Sankoré also immediately incorporates a web browser in the software so that one can easily open a webpage for content, and annotate on this web page if desired.

3.4.5 Benefits

Interviews, surveys and questionnaires relating to the perceptions of interactive whiteboards revealed that the main benefits of using interactive whiteboards are:

- flexibility and versatility
- multimedia/multimodal presentation
- efficiency
- supporting planning and the development of resources
- modelling ICT skills

- interactivity and participation in lessons

Flexiblity and versatility: Teachers reported that IWBs are a flexible and versatile teaching tool across age groups and settings [7]. Younger pupils for example prefer using a IWB to a computer because they found the keyboard and mouse difficult to operate [19]. Teachers also appreciate the facility to flip pages on a IWB screen for supporting a range of needs within a class spontaneously. [27]
Multimedia/multimodal presentation: In comparison to traditional whiteboards, the presenter can now color the presentation with sound, video and images, depending on the topic. [13] They furthermore have the possibility to further annotate, draw diagrams and label items which can be saved, shared and reused later. [27]
Efficiency: The most obvious advantage of using IWB is the facility to control the computer using touch, resulting in a more efficient presentation and more professional delivery of multimedia resources. [43]
Supporting planning and the development of resources: Although it takes some time to prepare lessons using an IWB and to become familiar with the system, teachers reported that they believe planning time will be reduced thanks to the ability to save, share and reuse the materials. [27] [26]
Modelling ICT skills: By using the computer for the lessons, pupils get familiar with some ICT skills who then don't need to be explicitly explained later on. [26]
Interactivity and participation: Traditional IWBs can augment interactivity and participation by using dialogic teaching because IWBs allow a flexibility in the marshalling of resources that enables teachers to create interesting multimodal stimuli for whole-class dialogue [30]. teacher-pupil interaction can for example be enhanced by encouraging students to answer to questions which can in turn be noted on a flipchart. The strong visual appeal of the information and learning resources that are displayed will overall encourage the pupil to participate. [27] This interaction can however be also done without an interactive whiteboard, and is a pattern that often is used in classrooms. [41]. The augmented interactivity is therefore largely dependent on how the teacher makes use of the technology and if the available software provides the potential and structure to do so [24]. Audience response systems (ARS) are a better example of such software where the primary focus lies in the participation of the audience. However, this is out of the scope of this thesis and will not be discussed further.

3.5 Conclusion

In this chapter we discussed multimodal interaction and multimodal interfaces. They propose a more natural form of interaction with computers by recognizing naturally occurring forms of human language and behaviour. To do this, a multimodal interface makes use of various input modalities to perceive and interpret human language and behaviour. Once all the input data is processed, a message will be constructed that is transmitted to a selection of various output channels.
We furthermore have taken a look at interactive whiteboards. Interactive whiteboards simulate a regular whiteboard with the advantage of using other digital communication media instead of written text or drawings alone. It furthermore offers the possibility to use animations to explain a certain topic which helps retention by using the combined input channels of speech and visuals towards an audience. Although they do not provide a direct solution to the issues found in common slideware, a lot of interviews, surveys and questionnaires have shown that this type of interaction had its fair share of benefits.

4

The Ideal Presentation Tool

In Chapter 2, we have taken a critical stance against common slideware. While the overall nature of slideware has been criticised as being a bad stand alone medium to communicate, we also studied that interaction with slideware is a crucial part in the effective communication when using slides, partially making the argumentation of Tufte [45] invalid.
In Chapter 3, we have taken a look at current available solutions that enable multimodal interaction and some new concepts that this form of interaction introduces. In this chapter, based on the ideas presented in previous chapters, we'll present our ideal tool and, more importantly, how we want the presenter to interact with it.

Multimodal interfaces target a more natural way of interacting with our computers. If we want to integrate multimodal interaction effectively in our ideal presentation tool, we need to understand the impact of certain interactions and know which type of interaction is being perceived as being a natural way to interact with our presentation. Once those interactions are identified, it is imperative to follow the guidelines for multimodal interface design, if we want to guarantee a high degree usability and public adaptation.
Unfortunately, not all issues can be solved by introducing new interaction possibilities. Therefore it is also important to choose the right existing presentation tool which is able to solve many of the issues found in popular common slideware.

4.1 MindXpres

MindXpres [39] is a tool that has been designed to provide more flexible and semantically enhanced presentations. It introduces a radically new presentation format to solve the lack of features found in common slideware. It offers for example the possibility to have a non-linear traversal over slides so that complex narrative becomes possible and solves the limited resolution issue by having a zoomable user interface, which allows to show more than one slide on the screen. It furthermore offers excellent support for multiple multimedia sources and allows us to introduce new types of media, such as the "Activity" pages seen in SMART Notebook, thanks to its plug-in mechanism. All of this without losing the ease of use found in other common slideware.
MindXpres offers a lot of the desired improvements that we would like to see in current common slideware. That is why we choose MindXpres as the presentation tool in which we want to integrate our solution.

4.2 A Natural Way of Interaction

In this section we will try to find what kind of interaction can be perceived as being a natural way of interacting within presentations. To do this, we will use the PowerPoint tetrad seen in Chapter 2, composed by Catherine Adams [4], as it reflects how common slideware are used in our daily lives. It will thus help us in identifying the features we want to improve, and how interaction can help.

4.2.1 Pointlessness and Insignificance

One of the first gestures which comes to mind when talking about presentations must be the pointing gesture. Pointing is a act of signifying, which is a central activity of pedagogical practice. It comes as no surprise that pointing is enforced when using common slideware, as the sole purpose of a presentation is to aid the talk of the speaker visually. Thanks to the presentation, the speaker can now point more accurately, vividly and rapidly at text and image. However, pointing at everything on the slides can also lead to pointlessness. This is due to our perception that if something isn't pointed out, it isn't important. However, if everything is being pointed at, and thus marked as important, nothing is. [4]
Doumont's guidelines [14] state that slides should have the least amount of text possible and that they should be stand alone, meaning that "deaf" audience members should be able to understand the messages by looking at the slides alone, and "blind" audience members should be able to understand the messages by listening to the presenter only. This will of course result in everything on the slides being important since it will already be a summarised booklet containing the most important data required to understand the message that the presenter wants to transmit. However, *limited text is no excuse for cryptic slides or for arbitrary word counts*. A behaviour that gets enforced when using bulleted lists. We can thus conclude that the automatic template with bulleted lists misleads us in doing the wrong thing.
Figure 4-1 shows a slide used as a part of the report used to judge whether the Columbia space shuttle could re-enter the atmosphere after it had taken some damage at lift-off. From the report, NASA officials decided that reentry was possible, resulting in the shuttle being burned up in the atmosphere. This slide is a good example to illustrate the problem at hand, where wrong judgement was made because of bad slides [45].

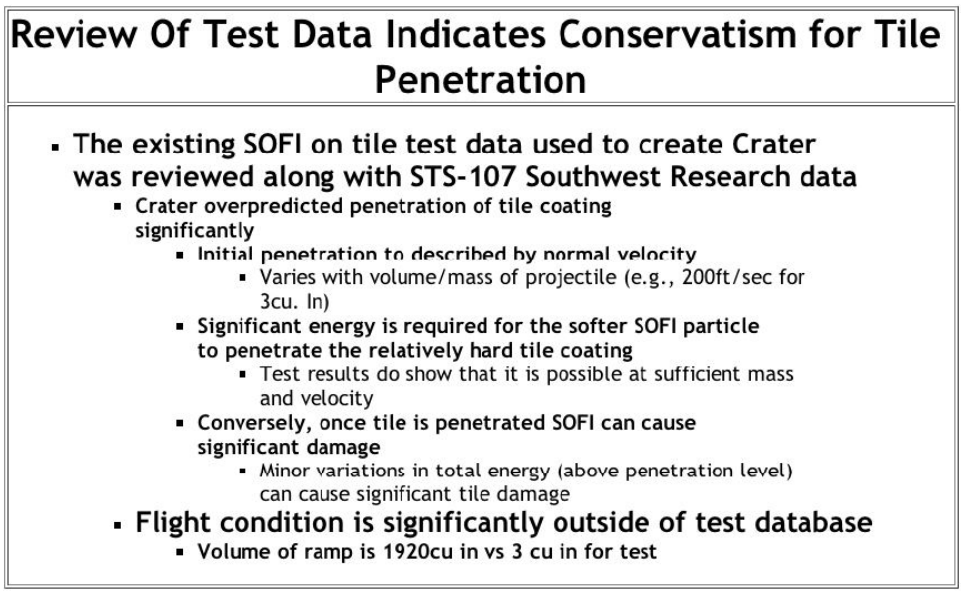


Figure 4-1: Key slide of the official Boeing report from the Columbia disaster.

This is an excellent example of a cryptic slide since no single clear message can be extracted from this slide. It is also an excellent example to illustrate how pointing at every bullet point would lead to pointlessness where the most crucial information, saying that the test data is way out of tested range, gets completely lost.
Now compare it with Doumont's version of what a good slide could have been, even while it contains less information Figure 4-2 [14].

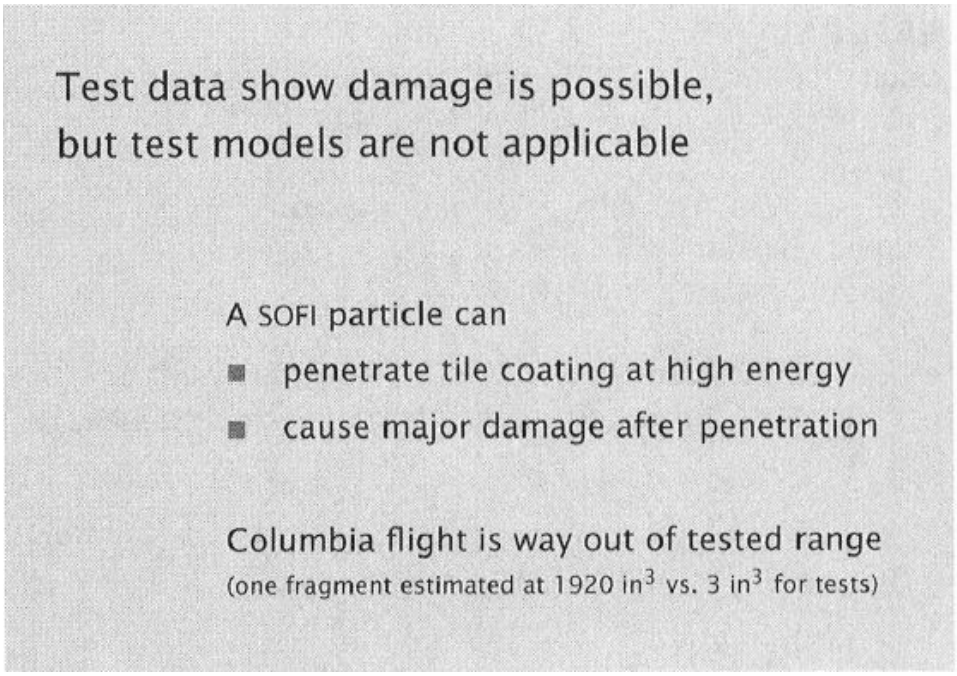


Figure 4-2: Doumont's version of the key slide of the official Boeing report.

By just looking at the slides, more information is absorbed when the slide is less cluttered and clear. If we would point out the bulleted phrases in Doumont's example, it would not lead to insignificance. Where the initial and cluttered version from the official report, where every point, no matter where on the slide, is equally important, pointing at the bulleted phrases would indeed lead to insignificance. A well designed slide is thus a big part of the solution. Because MindXpres is content oriented and doesn't implicitly propose us this kind of template, it already solves this issue partially. Unfortunately, the other part of the problem still remains in the hands of the presenter as no tool will ever guarantee a good presentation. What we can do however is try to improve the gesture of pointing, or the act of signifying.

4.2.1.1 The Gesture of Pointing

Most speakers interact with their slides by pointing towards it using their hands, a pointer, a stick or a pencil. According to Hubert Knoblauch, pointing does more than just refer to something on the slide, it relates the spoken and the visible by creating a distinction that parallels with what is being said. The gesture of pointing which gives contextual meaning, and by doing so, it helps us to understand what is being displayed [25]. In a certain way, it is as if we were highlighting the part on the slide that's being pointed at. Highlighting text has the same contextual meaning: it signifies the highlighted text and lays of a path for the reader to determine more easily what

is important. This highlighting has proven to have a positive influence on retention if the highlighted part is significant. However, highlighting insignificant and unrelated parts of a text can have the opposite effect [21]. In our ideal tool, we thus believe that the effectiveness of pointing can be improved by highlighting the areas being pointed at, automatically, and in a synchronized way. The speaker would have to determine those significant areas beforehand, since we do not want to highlight any insignificant parts. Once his pointing gesture point to the significant part of the slide, it will be highlighted once that the spoken relates to the visual being pointed at. The pointing could of course be done by hand, with a laser pointer, a stick or a pencil, as those are most used form of pointing.

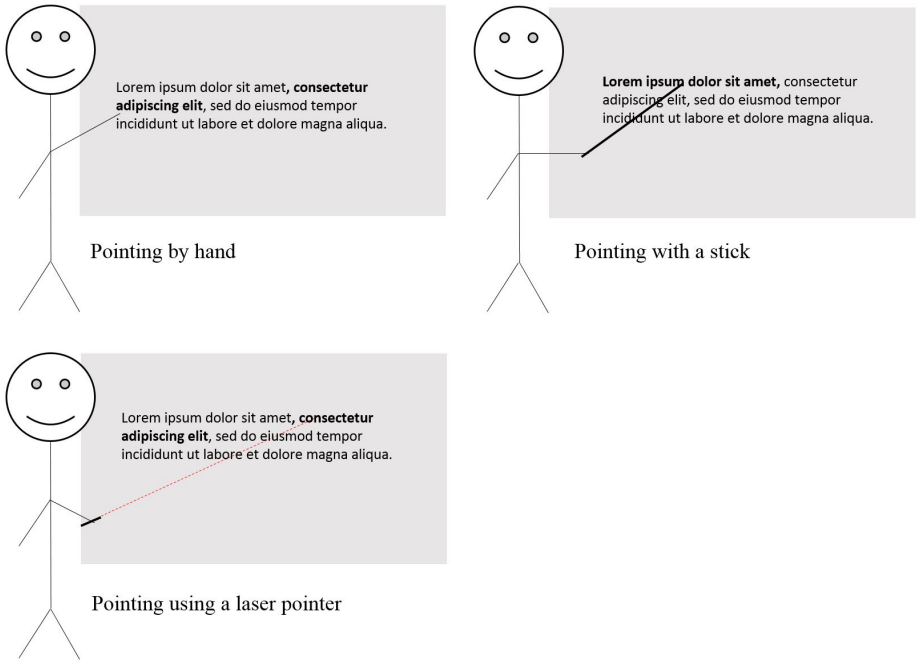


Figure 4-3: Different ways of pointing at the slide.

Finally, we also believe that the automatic highlighting will have the side effect of helping those who are less attentive during the presentation, as the automatic highlighting would help them to follow the flow of the presentation.

4.2.2 Limited Resolution,Linear Thinking and Complex Narrative

The limited resolution of a single slide has always been criticised in the literature. While presentations can be good for general information when the

information is limited, detailed information needs to be abbreviated rendering complex narrative difficult [31]. Knowledge that requires more space just translates badly on slides because we need to break the natural cohesion and coherence of different fragments, distributed over different slides [4].
MindXpres provides a solution to this problem: Using a zoomable user interface (ZUI), the audience can see an overview of all the different fragments before looking at each slide separately, preserving natural coherence. But what if more information is required, for example if the presenter badly estimated previous knowledge of the audience? In a linear presentation, the slides would become obsolete and unusable since the presenter has no flexibility to move out of his pre-determined path. But by allowing non-linear traversal, MindXpres offers again a way out. However, unless the presenter is behind his computer or if he has access to a SMARTboard, this feature can't be used. We would thus like to introduce new interaction possibilities to enable this key feature and improvement.

4.2.2.1 Using the Laser Pointer

The laser pointer is one of the most used pointer device for presentations. Most laser pointers now have dedicated buttons which can be used to navigate through the slides and turn on or off the projected screen. While this is already a big improvement over the default mouse and keyboard interaction, it only works for the linear traversal of slides.

Figure 4-4: Typical laser pointer used in the context of presentations.

The idea of using the laser pointer as an input device is nothing new [37,46]. While these solutions have been around for quite some time, it has seen no significant adaptation into the office or classroom, as we believe it didn't offer any added value in current common slideware. However, using the pointer as an input device in MindXpres becomes a lot more relevant because of its extended functionality. We can for example use simple sliding gestures with

the laser pointer over the slide to change slides in the absence of dedicated hardware buttons or use the same type of gestures to zoom in and to zoom out. To keep the interaction using the pointer intuitive, following gestures are proposed to enable interaction with MindXpres:

- Next slide

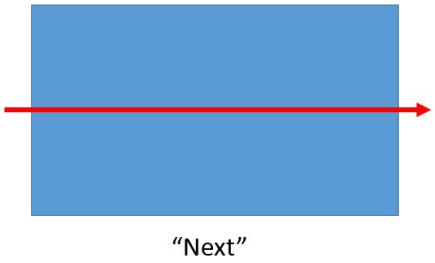


Figure 4-5: Swiping to the right with the laser pointer takes us to the next slide.

- Previous slide

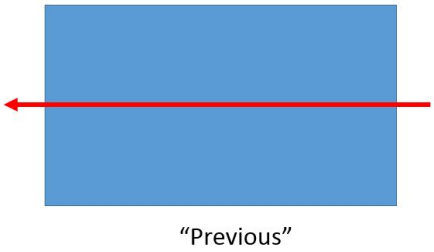


Figure 4-6: Swiping to the left with the laser pointer takes us back to the previous slide.

- Zoom out

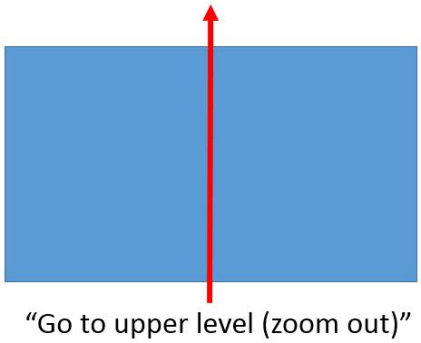


Figure 4-7: Swiping up will take us to the upper level of the presentation, thus zooming out.

- Click

Figure 4-8: Hovering over the same area for a short time will trigger a click event, enabling to zoom in to a selected slide or interact with other objects and animations.

4.2.2.2 Emulating Touch

Another possibility would of course be to interact by using touch. With interactive whiteboards becoming more common, this might be the best and most intuitive way to interact with our presentations. However, not everyone has the luxury of having a interactive whiteboard at their disposal. Therefore, in our ideal tool we would like to emulate touch interaction without the necessity of having an interactive whiteboard. This can be accomplished by using a much cheaper depth sensor.

4.2.3 The Process of Creating Things

By using common slideware, we eliminate the process of creating things progressively, as one would do when writing on the board. Instead, the audience sees the projected product instead of the process of the speaker's knowledge-in-action. In some sort, this is not such a bad thing as this method optimizes delivery. However, when presenting a complicated topic, the audience more often prefers the process of creating things progressively as it will lower cognitive load when dealing with complex narrative as it gives the opportunity to grasp the presented material at a slower pace [22]. It is thus rather context dependent whether one method is better than the other, and it may thus be convenient to allow both - such as with an interactive whiteboard.
Standard interfaces found in common slideware offer significant disadvantages when the presenter wants to create content freely. In our ideal tool, we want to make it easy to both annotate, create content and to add slides without having to look after the menu items to do so.
Unfortunately, MindXpres does not provide any of these features yet. But thanks to the plug-in mechanism of MindXpres, we can integrate these features ourselves.

4.2.3.1 Annotating Intuitively

We have always learned to write with a pen. Also on a traditional whiteboard or blackboard, a pen or chalk is used when creating content. It comes at no surprise that IWBs, such as SMARTboard, use a digital pen-tool for writing digitally. We also believe this is the most natural and intuitive way to write on digital slides.
In our ideal tool we want to enable annotating slides when holding a pen or something else with a similar shape.

4.2.3.2 Add and Erase Content

When we want to create things progressively, we will need a blank canvas to start with. And since we want the presenter to create content freely, it should also be made easy for the presenter to add a new slide and erase made annotations.
In our ideal tool, we would introduce following gestures to facilitate the free creation of content:

- Wave horizontally to insert a new slide.

Figure 4-9: Wave horizontally

- Wave vertically to erase all annotations.

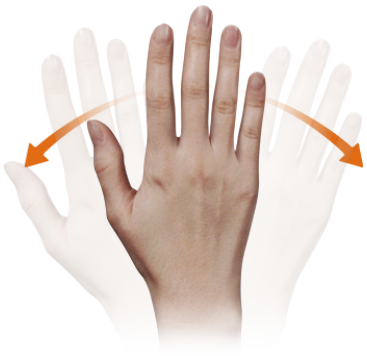


Figure 4-10: Wave vertically

4.3 Conclusion

In this chapter, we have seen what we believe to be the most crucial missing features where interaction can help. To make our analysis, we used the composed PowerPoint tetrad seen in Chapter 2, which describes how we interact with and use PowerPoint. Following improvements were then deducted and proposed:

- Automatic highlighting in a synchronized way by using pointing and speech

- Use the laser pointer as input device
- Simulate touch input
- Annotate using pen or pen shaped object
- Use gestures to facilitate the free creation of content

5 Towards Improved Interaction Modalities

In this chapter we are going to discuss some technologies and architectures needed to successfully implement multimodal interaction for MindXpres. We will start by explaining the Microsoft Kinect, as it is the chosen input device for various input sources. We will then discuss OpenCV, which is a computer vision library that we are going to use next to the Kinect API to extract more meaningful data out of our input sources. And finally we will take a look at how we can successfully transmit the output data to the MindXpres web application.

5.1 Microsoft Kinect

The first generation of the Kinect, which is a motion sensing input device developed by Microsoft, was available to the public in 2010 as an addition for their game console Xbox 360. It almost immediately received a lot of interest from the IT world which lead to the creation of various "hacks" to use the Kinect in a side projects. To respond to the high interest in the motion sensor, Microsoft release a non-commercial Kinect Software Development Kit (SDK) for Windows on June 16, 2011 [47].
In November 2013, Microsoft updated the Kinect along with the new Xbox One console. However, it was only in July 2014 that Microsoft released the Kinect SDK2.0 for Windows [48].

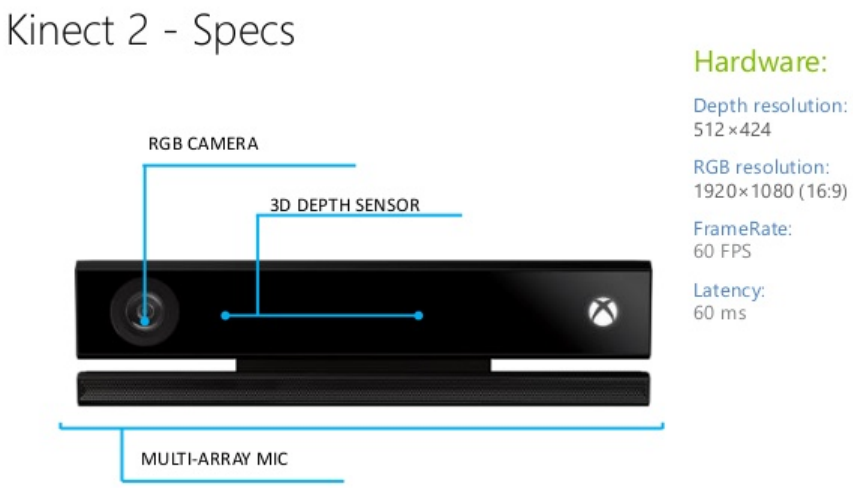


Figure 5-1: Microsoft Kinect 2

With a new Kinect sensor on the market, we had to determine which of both sensors would be best for our solution. As the features that we want to use are available on both iterations of the device, the determinant factor would be the accuracy.
In the setting of a presentation, the Kinect sensor would typically be a few meters away from the projected screen. From made measurements, we know that the accuracy decreases with an increasing range. For the first generation, the expected accuracy at 6 meters is as low as 60-70 mm, which is a bit low if we want to accurately simulate touch [18].
Although we do not have similar measurements for the Kinect V2, we know that the new depth sensor has a resolution of 512 x 424 pixels compared to 320 x 240 pixels. Together with the field of view, which is 70,6 x 60 degrees compared to 58,5 x 46,6 degrees for the first generation, we now have 7 x 7 pixels per degree compared to 5 x 5 pixels per degree [1, 2]. The depth is now also calculated using the time-of-flight of the reflected infrared light coming from the infrared emitters, compared to depth being calculated based on the structured light technique in the first generation. While it improves accuracy, it also makes the depth sensor insensible to external light. The second generation thus seems a better solution, especially since it also has an improved Red, Green and Blue (RGB camera) with a resolution of 1920 x 1080 pixels running at 30 fps.

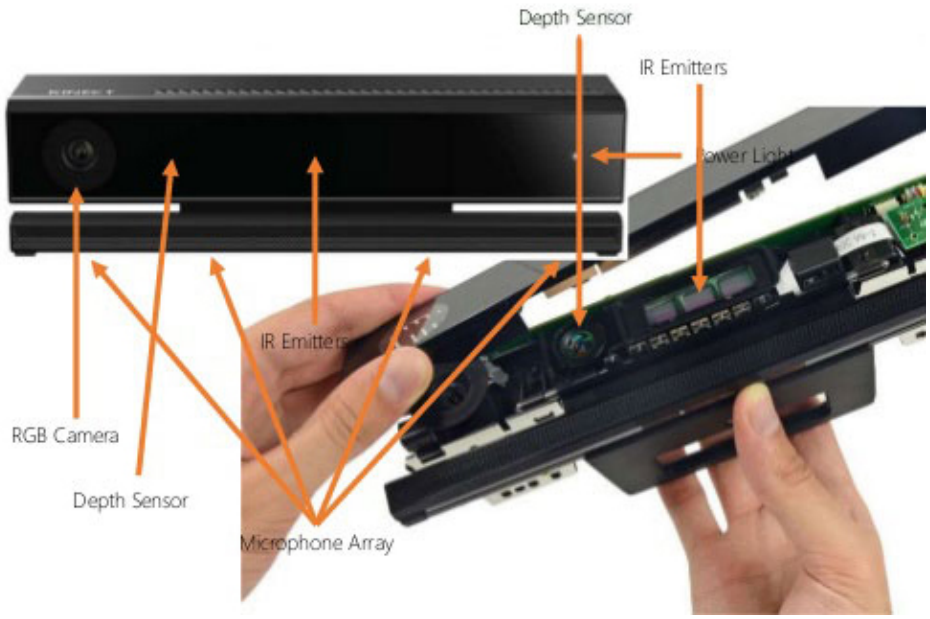


Figure 5-2: Microsoft Kinect 2 sensors, image from http://www.ifixit.com

5.1.1 Kinect API

With a new Kinect sensor came a new Kinect API, or rather three different API sets that can be used. A set of Windows Runtime APIs is provided for the development of Windows Store application, a set of .NET APIs is provided to support the development of WPF applications and then there is a set of native APIs to support applications that require the performance advantages of native code [2].

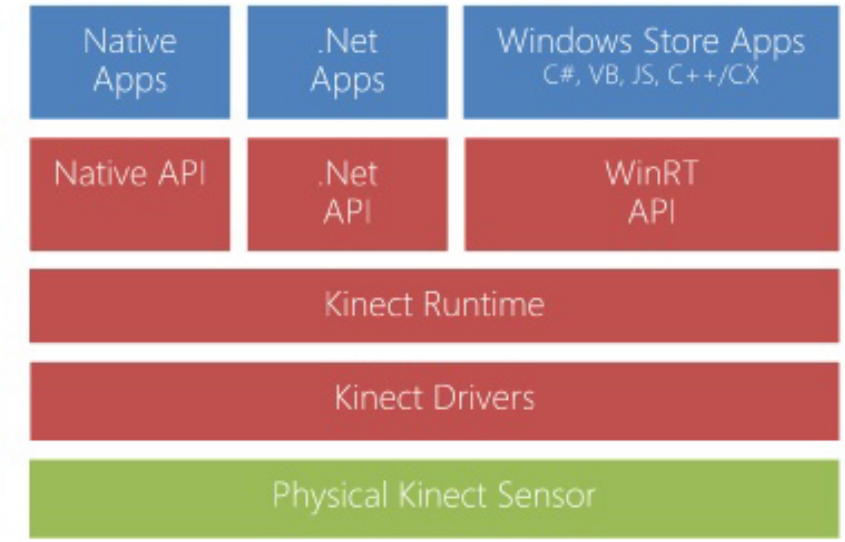


Figure 5-3: High-level architecture

The API translates video and audio data into various sources, listed below:

- Audio
- Body
- BodyIndex
- Color
- Depth
- Infrared
- LongExposureInfrared

The data of these sources can then be read with their corresponding streams, where every stream data is accessible through frames. The API also provides a multisource frame for video data, where different frames from different data sources can be accessed simultaneously, so that it is guaranteed that they contain a snapshot of the same recorded visual data. This can for example be used when we want to correlate data from the depth frame, coming from the infrared camera, with the data from the color frame, coming from the RGB camera.

5.1.1.1 Depth Frame

Every pixel within a depth frame represents the distance of the closest object seen by that pixel. This information is comes from the depth camera, where the distance is calculated using the time of flight of infrared light emitted from the Kinect V2.
The maximum depth is 8 meters, although reliability degrades starting at 4,5 meters.

5.1.1.2 Body Frame

The body frame contains information about the real-time tracked people that are in the view of the sensor. Every body includes information about the 25 tracked skeletal joints up to 6 people. The joints are illustrated in Figure 5-4.

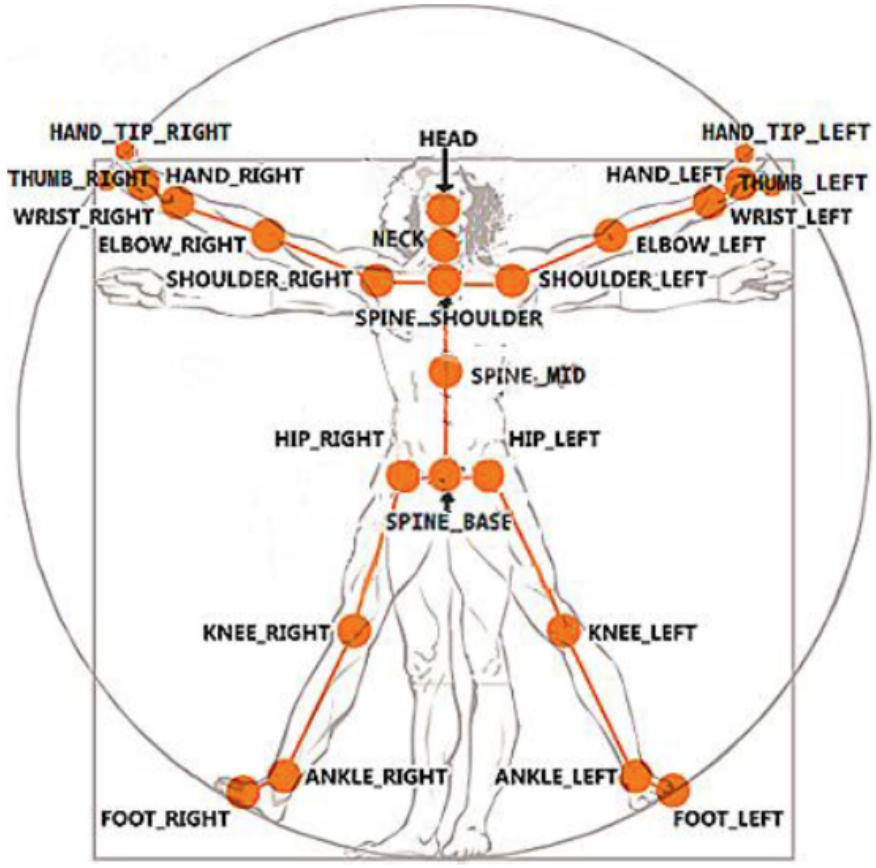


Figure 5-4: Kinect V2 skeletal joints

5.1.1.3 Body Index Frame

Based on the depth image, the body index frame tells you which depth pixels belong to the background or to tracked people. The value of these pixels is a value ranging between 0-5 when the pixel is belongs to a tracked person. The value is the index of the tracked body of that person (see previous section). If the value is out of this range, it means that the pixel contains to the background.

5.1.1.4 Color Frame

This is the most basic feature that we would expect from a device containing a camera: the color frame is contains image data which can be converted into a desired color image format. We can for example use this image if we would like to do some other visual processing outside of the Kinect API.

5.1.1.5 Infrared Frame

The infrared camera used for the depth frame can also be used to provide us with black and white images. It is a good source for computer vision algorithms where texture is important.

5.2 Computer Vision and OpenCV

The Kinect API is an excellent starting point to retrieve relevant data for our implementation. However, if we want to obtain more information from a given frame than body movements or depth, we'll need to make use of more specialised software to extract meaningful data through computer vision.
Computer vision is the transformation of data from a still or video camera into a new decision or a new representation [12]. This can be for example knowing if there is a person in an image, and if he is holding something in his hands. Because we are visual creatures, this may seem an easy task. However, while we, humans, can divide an image into multiple channels that stream different kinds of information into our brain, a computer only receives a grid of numbers from the camera. There is no built-in pattern recognition and no cross-associations with years of experience.

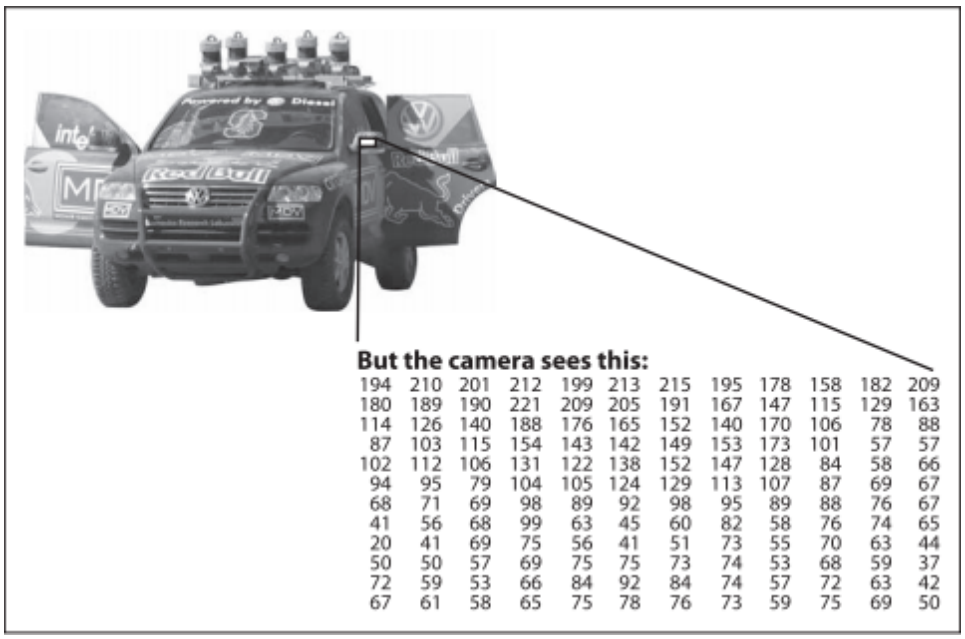


Figure 5-5: To a computer, the car's side mirror is just a grid of numbers [12].

The task of computer vision is to turn this grid of numbers into the actual object, which in this case is the side mirror of the car. This is a very complex problem, especially since we're representing the 3D world into a 2D projection. There is thus no definitive solution to this problem. The only thing we can do it use additional contextual knowledge to work around the limitations imposed by visual sensors and propose a solution.
OpenCV is aimed at providing the basic tools to solve this kind of computer vision problems with the focus on real-time applications. Sometimes its high-level algorithms might offer you a solution to your problem. And when they don't, OpenCV offers a complete set low-level functions so that you can create a complete solution on your own.
OpenCV began as a research project at Intel in 1998 [3] and became available to the public since 2000 under the BSD open source license. It was conceived as a way to make computer vision infrastructure universally available. The goals of OpenCV were the following [12]:

- Provide an open and optimized code for basic vision infrastructure. No need to reinvent the wheel.
- Disseminate vision knowledge by providing a common infrastructure.
- Advance vision-based commercial applications.

Today, OpenCV has reached the critical mass at which the project becomes self-sustaining. It is an active area of development at several institutions and we can expect it to evolve further over time.

5.3 Client Server Communication

When using MindXpres, we have to define our presentation in an XML container format which would then get compiled to HTML5 output, visible in any modern browser [39]. Because the output result is in HTML5 format, the presentation can be viewed on most devices, even on portable and less powerful devices such as smartphones.

HTML5 also gives the possibility to access hardware such as the microphone and webcam. However, interpretation of this data in the browser is too limited for multimodal interaction. Therefore, we would need another application which is capable of doing the calculations needed for multimodal interaction and which can send the relevant data to the browser for output. This communication between the browser and another application is the client-server model, where the browser would act as the client, and the application who does the multimodal interaction calculations would act as the server.

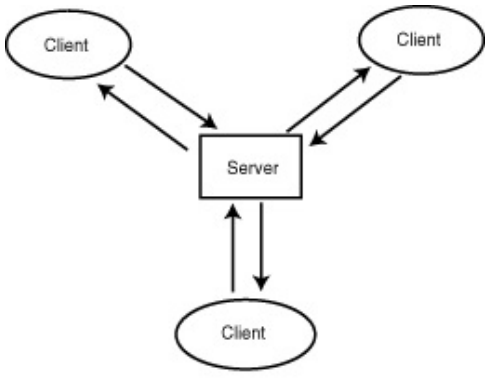


Figure 5-6: Client-Server model

This type of architecture is also often referred to as a two-tier architecture, as the clients acts as one tier and the server acts as the other tier.

5.3.1 Communicating Technologies

In client-server model, the client typically communicates with the server through HTTP. A big drawback however is that HTTP only works unidirectional. However, since we require that the server can push output data to the client, mechanisms on top of this protocol are required for our solution.

5.3.1.1 HTTP

In this section we are quickly going to take a look at HTTP (HyperText Transfer Protocol). HTTP is the foundation of data communication in the World Wide Web and was developed when the World Wide Web was first introduced [9] [8].
The HTTP protocol consists of a message type, message headers, and a message body. The message types is either request or response, where the client would send a request to the server and where the server would respond to that request accordingly, as shown on Figure 5-7.

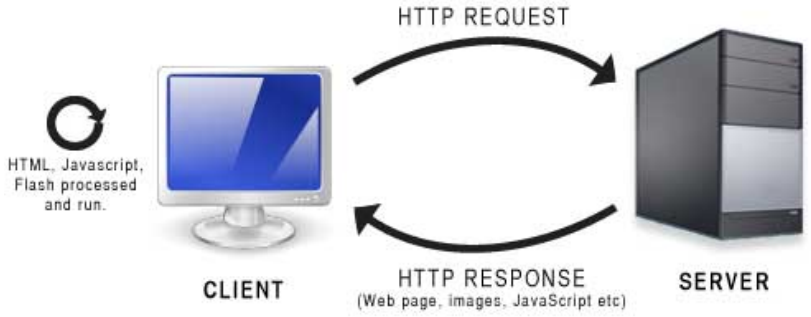


Figure 5-7: Client-Server model

There are different kinds of requests methods, each having a different semantic meaning [17]:

- GET: Retrieve whatever information is identified by the Request-URI.
- HEAD: Essentially the same as a GET request, except that the server must not return a message-body in the response.
- POST: Is used to request that the origin server accept the enity enclosed in the request as a new subordinate of the resource identified by the Request-URI.
- PUT: request that the enclosed entity be stored under the supplied Request-URI.
- DELETE: request that the origin server delete the resource identified by the Request-URI.
- TRACE: Used to invoke a remote, application-layer loop-back of the request message.
- OPTIONS: Request information about the communication options available on the request/response chain identified by the Request-URI

- CONNECT: Is reserved for use with a proxy that can dynamically switch to being a tunnel.

In practice, DELETE, TRACE, OPTIONS and CONNECT are rarely used. But we added them for completeness.

Below are the message headers for a GET request.

```
1 GET / HTTP/1.1
2 Host: www.vub.ac.be
3 User-Agent: curl/7.43.0
4 Accept: */*
```

In the header of the request we can read the type of user-agent that made the request, the requested URI and the type of data that is accepted. More information can be added to the header, such as the accepted language, so that the server better knows how to respond to the client.
Below are the message headers for the response of the request shown above.

```
1  HTTP/1.1 200 OK
2  Date: Mon, 10 Aug 2015 08:58:21 GMT
3  Server: Apache/2.2.22 (Debian)
4  X-Powered-By: PHP/5.4.4-14+deb7u14
5  X-Drupal-Cache: MISS
6  Expires: Sun, 19 Nov 1978 05:00:00 GMT
7  Cache-Control: public, max-age=1800
8  Content-Language: nl
9  X-Generator: Drupal 7 (http://drupal.org)
10 Link: </home>; rel="canonical",</node/6>; rel="shortlink"
11 Etag: "1439197101-0"
12 Last-Modified: Mon, 10 Aug 2015 08:58:21 GMT
13 Vary: Cookie,Accept-Encoding
14 VUBServerName: VM-01
15 Content-Type: text/html; charset=utf-8
16 X-Cache: MISS from www.vub.ac.be
17 Transfer-Encoding: chunked
```

On the first line we can see the version of the protocol together with a code. This code informs us in this case that the request was successful. However, a lot of other codes exist like the code 404 for example. This code informs us that the requested URI could not be found. Other information can be found in the response headers, such as the content-type, when it was last modified etc. The message body, not shown here, contains the actual HTML page which gets rendered by your browser.

This type of communication always goes from the client to the server but never from the server to the client. But as the web evolved and web communications became gradually more complex, such as in our solution, there emerged a need for bi-directional communication. This led to the development of asynchronous Javascript and XML (AJAX), also referred to as short

polling. The client would periodically send messages to the server to see if there new data available. If not, the server would send an empty response. This is a very inefficient way to enable server-client communication, as the client would still have to initiate server communication by pulling, increasing overhead on the network [36].

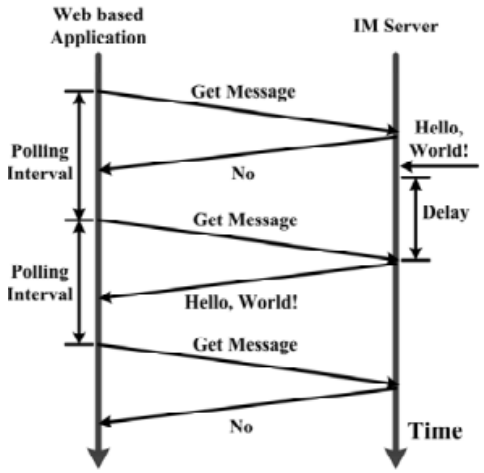


Figure 5-8: Short polling workflow [40].

5.3.2 Comet

In order to improve the situation of short-pulling, where continual polling consumes significant bandwidth by forcing the default request and response round trip when no data is available, new mechanisms were implemented and grouped under the label "Comet". These mechanisms would allow the server to deliver updates to clients without waiting for a poll request. These mechanisms also avoid the latency experienced by client applications using short polling due to the frequent opening and closing of connections for unnecessary data.
We will now focus on the two most-common server-push mechanisms: HTTP Long Polling and HTTP Streaming, based on [28].

5.3.2.1 HTTP Long Polling

Long Polling attempts to minimize the latency in server-client message delivery and to minimize the network resources. It achieves those attempts by responding only to a certain request when a particular event, status or time-out has occurred. Only once that a response is returned will there be a new request send to the server. This is a way for the server to asynchronously "initiate" communication.

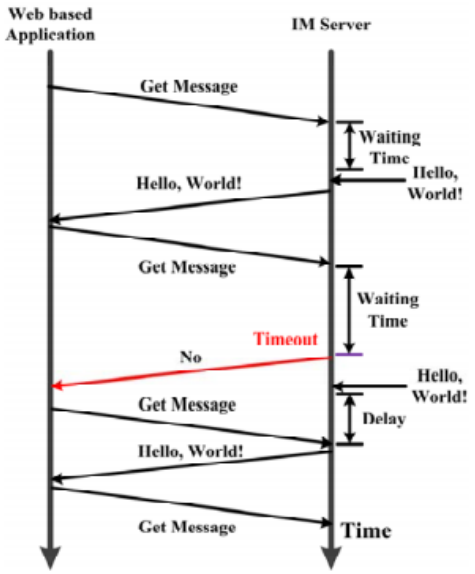


Figure 5-9: Long polling workflow [40].

Issues

- Header overhead: While this is a significant improvement over short polling, there still is the header overhead as this technique still used the standard HTTP protocol. These headers can represent a large percentage of the data transmitted and can thus have a significant impact on the network load.
- Latency: The latency should be much better than with short pulling due to the fact that there are a lot less requests and responses on the network, and because an update is send to the client only when a specific event, status or time-out has occurred, resulting in a latency of one network transit. However, when a server wants to transmit an update to the client when no request has been received, the maximal latency will be as long as three network transits (long poll response, next long poll request, long poll response). If packet loss occurs during these requests, it means that the latency gets even further increased.
- Allocated resources: Because the request doesn't get a response immediately, it requires the client to allocate resources for the HTTP request while it is held open. It is thus important to take this into consideration when designing an long polling application.

5.3.2.2 HTTP Streaming

HTTP streaming is another mechanism which attempts to reduce latency by keeping a request open indefinitely. This way, a connection never needs to

be opened or closed again once it is established, while the server can push data to the client.
The HTTP streaming mechanism is based on the capability of the server to send several pieces of information in the same response without closing a connection. The response defines the content length using three options:

- Content-Length header: contains the size of the message body, in bytes.
- Transfer-Encoding header: Setting this value to 'chunked' indicates that the message will break into chunks.
- End of File (EOF): Clients don't need to know the size of the message body. Instead, they expect to receive data until the server closes he connection.

If one wants to use HTTP Streaming, he can do so by using either defining the Transfer-Encoding as being chunked or by using the EOF principle.

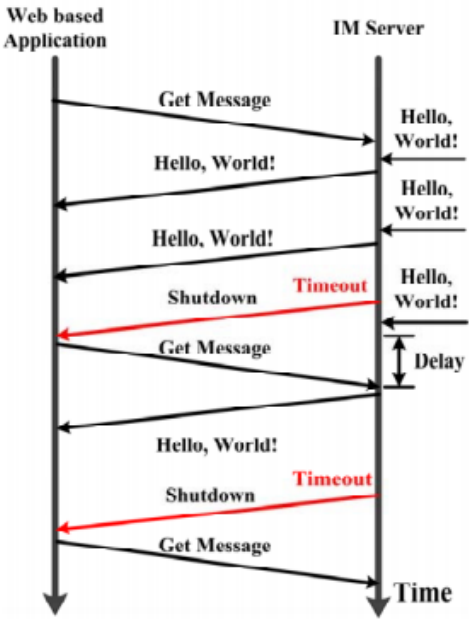


Figure 5-10: HTTP streaming workflow [40].

Issues

- Network intermediaries: It is not illegal for an intermediary, such a a proxies, gateways, etc., to bugger the entire response before sending it to the client. Obviously, HTTP streaming won't work if the complete response gets buffered before sending it to the client.
- Client buffering: Like with the network intermediaries, the client may buffer the total response before sending it to the client application. For Javascript, there is also no requirement to execute partial chunks.

- Latency: As with long polling, the theoretical latency should be only one network transit. But since HTTP streaming is often associated with Javascript and/or DOM (Document Object Model) elements that grow in size when data is received, the HTTP steaming implementation occasionally needs to terminate streaming to avoid unlimited growth of memory usage in the client. Thus the maximal latency will again be as long as three network transits (HTTP steaming response (close), next HTTP steaming request, HTTP streaming response) when the connection needs to be re-established. If packet loss occurs during these requests, it means that the latency gets even further increased.

- Framing technique: With long polling, we have a canonical framing technique: each application message can be send with a different HTTP response. However, with HTTP streaming, as everything is send in a single response, we need to separate the response stream into applications messages at the application level.

5.3.3 The WebSocket Protocol

HTTP was not initially meant to be used for bi-directional communication. The developed mechanisms that eventually did enable bi-directional communication - discussed in previous sections - have their resulting trade-offs between efficiency and reliability. Therefore, as a response to the need for a bi-directional communication in web applications, the WebSocket protocol got introduced in 2011 . It enables two-way, full-duplex communication between a client and a remote host without relying on multiple HTTP connections [16].

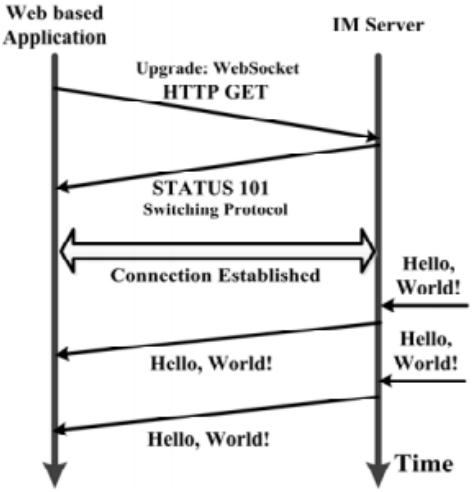
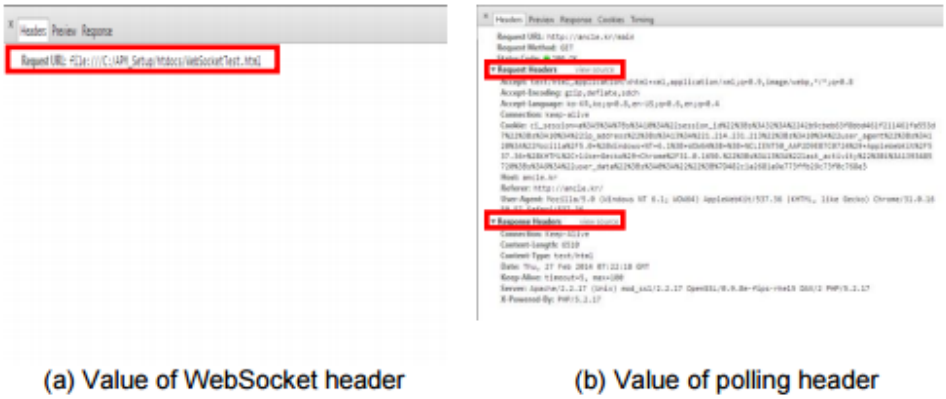


Figure 5-11: WebSockets workflow [40].

A WebSocket connection starts with an HTTP-based handshake to establish a connection, as can be seen on Figure 5-11. The handshake HTTP message is just a normal GET request with a field with name Upgrade whose value is WebSocket. If the server supports the WebSocket protocol, the connection will be upgraded to a full-duplex communication channel. The latency is thus always only a single network transit and the overhead caused by the request header is eliminated due to the much simpler header of WebSocket frames.



(a) Value of WebSocket header (b) Value of polling header

Figure 5-12: Comparison of the actual overhead of WebSocket and Polling methods.

5.3.3.1 A Comparison

Number of requests: Kai Shuang and Kai Feng created an experiment to test the different push methods [40]. In their experiment, an instant message generator would randomly distribute messages to different servers who made use of a single mechanism to allow bi-directional communication. Each server was connected to a single client and a total of 1000 messages were send to each client.
The chosen interval for short polling was 1000ms, and the chosen sleep time for long polling was 500ms during the experiment. The interval is the time between two following AJAX requests. Sleep time is the time between two queries on which the server processes the non-responded requests.

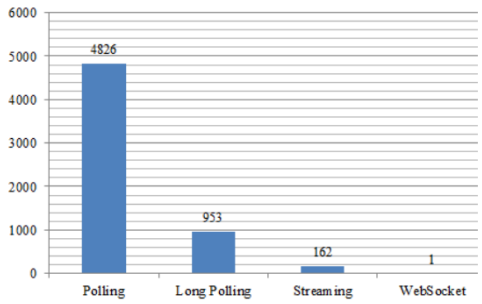


Figure 5-13: Number of requests [40].

Not surprisingly, short-pulling needed the most requests because an empty response can be returned if no new message is available at the server. Compared to WebSockets where only one request was needed to establish the connection, short polling does indeed increase server load significantly.
The long pulling had as expected 1000 requests for 1000 messages. The amount of requests in the setting was slightly lower because of the sleep time, which gave the possibility to aggregate several messages in a single response.
While theoretically one request would be enough for streaming, because of the increasing memory usage at the client, which requires to terminate the connection occasionally and occasional time-outs of the connection, more requests were recorded.

Network overhead: As discussed before, using HTTP requests and responses has the big disadvantage of sending the HTTP header message every time, resulting in an increase in network load. But how does this compare to WebSockets? A message send with the WebSockets protocol is called a frame. This frame contains one byte for the type of the message, one byte for the length of the frame, followed by 2 or 8 optional bytes if the length doesn't fit in the first byte dedicated for the length of the frame, and then the actual message. For the client, 4 bytes are added for the mask, which contains decoding keys [16]. We thus have a header side of minimum 2 bytes and maximum 14 bytes of overhead for WebSockets. If we compare this to an average header size of HTTP requests, which is typically between 200 bytes and 2 kilobytes, then we can agree that WebSockets do an outstanding job at minimizing network overhead.
To illustrate the impact of this overhead, we refer to the experiment done by [36]. They measured the increasing overhead on the network load with an increasing number of users, when sending one gigabyte of data through WebSockets and polling. For the polling, a HTTP header size of 1000 bytes was considered.

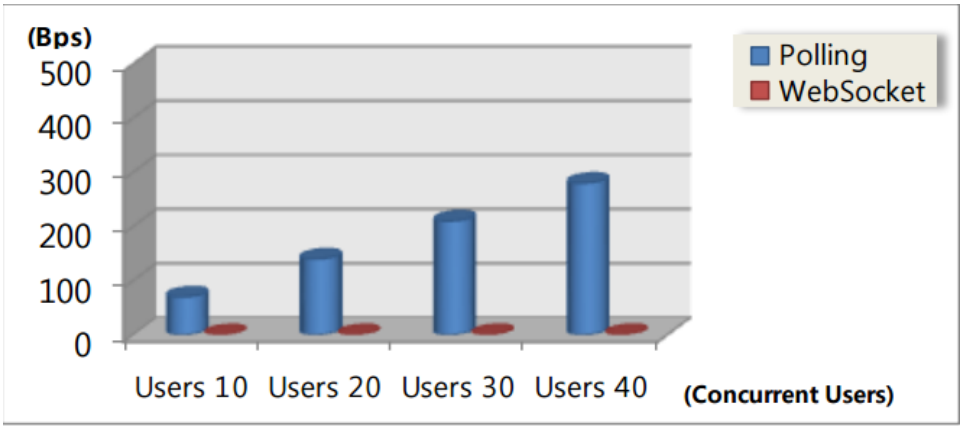


Figure 5-14: Comparison of the overhead generated by the polling and WebSocket method based on the concurrent number of users.

5.3.4 Conclusion

To communicate our output, coming from multimodal input, to the MindXpres web application, we require a bi-directional communication channel. The best solution would of course be the one having the best performance. The overall winner is without question the WebSocket protocol, who next to a lower overhead and better latency, also provides full-duplex communication.

5.4 HTML5 canvas

In order to allow hand-made annotations in MindXpres, we use HTML5 canvas. The canvas element allows to dynamically draw nd render 2D shapes and bitmap images. It is the perfect type of object if we want to draw in a web application.

5.5 Choosing our Framework

For our implementation, we want to have a client-server communication with MindXpres through the WebSocket protocol and we want to be able to use the Microsoft Kinect V2 as out input device. We would thus need a framework that provides us a webserver that can run the WebSocket protocol and that is capable of running the Kinect API. All requirements thus points towards a .Net application if we want to have everything running in a single solution.

5.5.1 C#,WPF and .NET

C# is a programming language created by Microsoft for its .NET framework. It builds on the lessons learned from C, C++, Java^{TM} and Visual Basic to create a general-purpose, type-safe, object-oriented programming language used for mainly the development of distributed Windows client and web applications [5].
WPF is a presentation system used in .NET for the graphical user interface of Windows desktop applications. Using XAML, which is similar to XML, one can declare a user interface while keeping application logic seperated in the background C# code.

5.5.2 Asp.net SignalR

SignalR is a library for ASP.NET that simplifies the implementation of bi-directional communication mechanisms into your web application. It uses WebSocket transport where available, and falls back to older transports where necessary. The library handles connection management automatically and makes it easy to send messages to a specific client, a group of clients or to broadcast messages to all connected clients.

5.5.3 EmguCV

EmguCV is a .Net wrapper to the OpenCV image processing library. As OpenCV does directly provide support in C#, we'll make use of this wrapper so that we can call any OpenCV functions within our application.

5.6 Conclusion

This chapter discussed all technologies that we are going to use in our implementation that will enable multimodal interaction for MindXpres. As MindXpres is a web application running in the browser and since we make use of the Microsoft Kinect as input, we will use client-server communication to communicate output from server to the client. However, default HTTP does not support bi-directional communication out of the box. Therefore we took a look at existing solutions of which the WebSocket protocol is best fit for our application.
We opted for a .Net WPF application in C# to have an easy access to the Kinect API. This Kinect application will do all the processing required for the multimodal interface and contains the webserver to communicate the output events to the browser.

6 Implementation

In this chapter, we are going to describe how we implemented our ideal presentation tool proposed in Chapter 4. We will first focus on how we extended the functionality in MindXpres through plug-ins to enable annotations, automatic highlighting and bi-directional client-server communication between MindXpres and our Kinect webserver application. Then we will look at how we calibrate the system so that we can map the pixels of the camera to the pixels of the projected screen. Finally, we will discuss how we implemented every new proposed feature using the the multimodal man-machine interaction loop that we saw in Chapter 3.

6.1 MindXpres Plug-ins

In MindXpres, all features are plug-ins. This allows us to replace, modify or add new features. Following major distinctions are made for different types of plug-ins:

- *Components*: A component is a container that provides visualisations and functionality for a specific content type.
- *Containers*: Containers are elements that contain components, and help the user to organise those components visually.
- *Structures*: Structures contain mechanism for laying out content. The main difference with containers is that structures have ties to the XML language used to define presentations, like for example chapters, sections and subsections.

6.1.1 Client-Server Communication

In order to communicate certain events to the MindXpres web application, triggered by the interpretation of different input modalities, we need to implement a basic web server capable of using WebSockets if we want to allow full-duplex communication. To achieve this requirement, we implemented a WPF Windows Kinect application containing a basic webserver running the ASP.NET library SignalR.

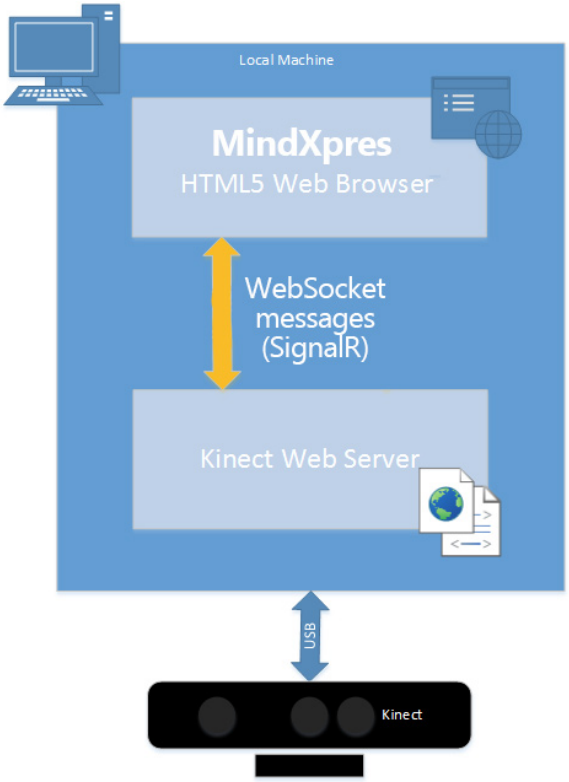


Figure 6-1: Kinect Webserver communicating with MindXpres running in the browser through WebSocket.

The next step would be to run the SignalR library in MindXpres. If we look at the major types of plug-ons for MindXpres described above, we find that WebSocket communication cannot be categorised in one of these three major distinctions that are being made, as WebSocket communication is not content based. Fortunately we can still add the desired functionality through other types of plug-ins.
First we will have to tell MindXpres which files need to be included at initialisation. This is done by registering our files to the dependency injector.

```
1 DI.register("signalr_dependency", "lib/jQuery/jquery.signalR
      -2.2.0.min.js");
```

```
2 DI.register("signalr_hub", "http://localhost:8080/signalr/hubs?.js
    ");
3 DI.register("signalr_client",libDir + "signalr/signalr.js")
```

Once registered, we have to also include them. This is done with the include function. In our signalr_client file, we included all dependencies needed for the client to make a connection.

```
1 DI.include("signalr_dependency");
2 DI.include("signalr_hub");
```

There is an optional callback function that we can pass as second the parameter of the include function. For the dependencies shown in the listing above, no callback is needed, thus we can leave it blank.
Finally, we need to include our signalr_client, which is the plug-in we want to implement to enable bi-directional communication for MindXpres with our webserver. Therefore, we include the signalr_client dependency in the bootstrapper of MindXpres. In this case, we make use of a callback function to initialize our signalr client and to make the connection with the webserver.

```
1 DI.include("signalr_client",function(){  signalr_client.init();});
```

6.1.2 Communication Between Plug-ins Through Events

In the signalr_client, every incoming message triggers a certain event. Other plug-ins, in charge of extending the functionality of MindXpres through multimodal interaction, can listen to these events and respond accordingly.

```
1  this.hub.client.triggerEvent = function (eventName, data) {
2                      document.dispatchEvent(new CustomEvent(
3                          eventName,
4                          {
5                              detail: data ,
6                              bubbles: true,
7                              cancelable: true
8                          }
9                      ));
10                 };
```

Listing 6.1: Trigger events in signalr_client

```
1  document.addEventListener("customEvent", function(message){/*Do
       something with the triggered event*/);
```

Listing 6.2: How other plugins can listen to those events

6.1.3 Automatic Highlighting

The automatic highlighting is the only feature that requires a component plug-in, as we need to define the significant part on a slide that we can be

highlighted automatically. The idea is to augment the interplay between the presenter and his presentation by highlighting the significant part(s) on a slide once it is being pointed at in a synchronous way.
Using the XML structure of MindXpres, a user can define these parts with a highlight tag, together with the keywords that trigger the highlighting.

```
1 <text>In this phrase, I am the <highlight keyword="significant">
    significant</highlight> part.</text>
```

These keywords will then be send to the server once the slide, on which the highlighted parts are defined, is in focus. If the keywords are being said by the presenter together with a pointing gesture towards the significant part, it will be highlighted. More about this in Section 6.3.

6.1.4 Annotate Slides

In order to allow annotations being made on slides, we modified the slide container to automatically include a HTML5 canvas as an overlay. By simply listening to mouse events, a user can now start drawing over the slide onto the canvas. However, this also means that underlying elements become inaccessible for user interaction. This can be disabled by using CSS (Cascading Style Sheets), setting the pointer-events property to none.
We will now thus enable and disable the ability to annotate over slides automatically by detecting if the user is trying to write on the slide, as will be discussed in Section 6.6. When a write event is send to MindXpres, the canvas will then be enabled or disabled.

6.2 Mapping the Projected Screen With Kinect Frames

A persistent problem for the new features that we want to implement is the mapping of the actual projection screen to one of the source frames that we retrieve from the Kinect API. We will thus have to calibrate our application in order to know in where we can "see" the projection within the captured camera image. Doing this for one source is enough, as the Kinect API has a coordinate mapper to map every source to each other.

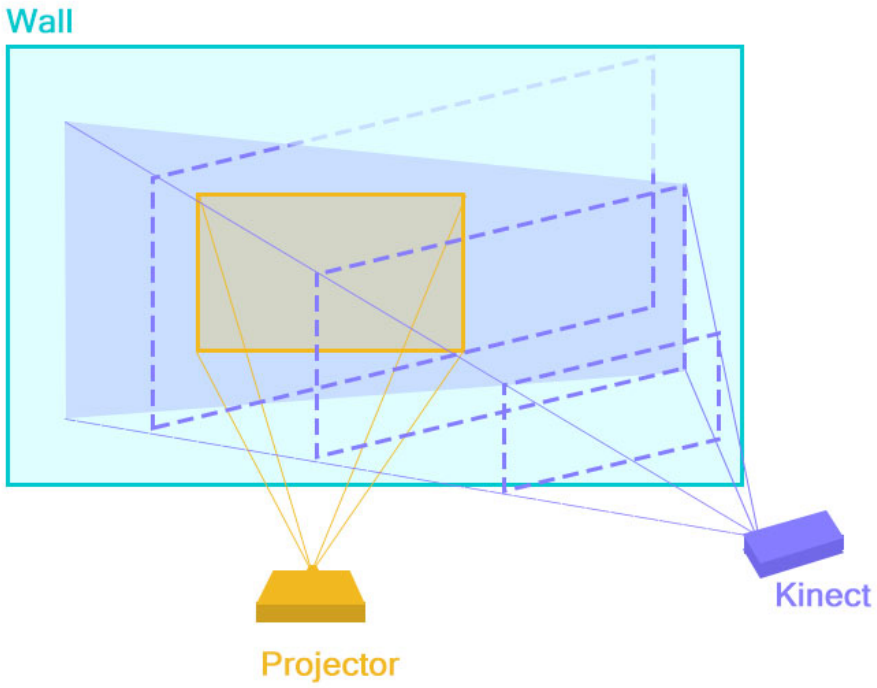


Figure 6-2: Projection within sensor field of view against wall.

For the calibration, we have to determine the 4 spacial coordinates delimiting the plane in which the projection screen will be found. With information we gather from the depth frame, as it contains spacial information about every pixel in the frame, we can exactly determine where the projected screen is positioned from the camera.

Figure 6-3: Coordinates from Kinect depth sensor.

During calibration we display the color frame to the user doing the calibration. He can manually then pick the pixels where the corners of the projected screen are displayed. By mapping these pixels to their corresponding depth

pixels, we have all necessary information needed for the new implemented features.
We are going to use this determined surface as the region of interest for the interpretations of data in following sections, with the exception of audio and body gesture recognition.

6.3 Automatic Highlighting by Using Point and Speech

Decision and Action State	When the speaker has to intention to signify a part of his speech by a visual aid (decision), he will point towards the presentation (action).
Perception	Following input sensors are used to gather all information: • RGB Camera • Microphone • Depth camera All of these sensors are integrated in the Kinect V2 and the data is extracted using the Kinect API.
Interpretation	Combining the extracted features from the microphone, the depth camera and the RGB camera, we try to detect whether the user uses gestures and speech to highlight a significant part on the slide.
Computation	If we found that a significant part is being pointed at during decision-level fusion, then we need to process fission of output modalities. *Message construction*: The message constructed will trigger en event in MindXpres, containing the detected keyword together with the position being pointed at. *Output channel selection*: The output channel is MindXpres, where we push the message to the client using the WebSocket protocol.
Action and Perception	MindXpres will process the incoming event. If the position of the significant part matches with the position being pointed at, then the application will highlight the significant part on the slide (action). The audience will then see the significant part being highlighted of the area being pointed at (perception).

We will now take a better look at the interpretation inside the man-machine interaction loop.

6.3.1 Interpretation

6.3.1.1 Audio

Audio-Only Speech Recognition: In order to highlight significant words or sentences, we first need to know which words are significant in the presentation. This is done by a new plug-in in MindXpres that sends the words that we want to recognize in a particular MindXpres structure to the server

when that same structure, containing the component, is in focus.
The speech recognition is built-in in the Kinect API. All we have to do now is listen to the audio input for those significant words.

6.3.1.2 Depth-camera

Body frame: We save body frame data that we gather from the Kinect API. We are interested in the hands of the detected body, as we will want to know if the user is pointing at something on the slides.
Body index frame: Using the Kinect API, we store the body index frame which contains information if a certain pixel belongs to a person.
Depth frame: Again using only the Kinect API, we store the depth of every pixel.

6.3.1.3 RGB-camera

Blob detection: A blob is a region of an image in which some properties are constant or approximately constant, for example a region with the same color can be considered a blob. Using OpenCV we are going to do a basic color blob detection on the retrieved color frame. These extracted blobs on the image will be used during feature-level fusion to eliminate false positives because of noise.
Visual-Only Laser Pointer Detection Using the RGB-camera, we wish to find whether or not there is a red dot coming from the laser pointer on the screen. Because of the lack of visualisation computations within the Kinect API, we make use of OpenCV to do the computation.
For the computation, we based our algorithm on the one proposed by E. Popovich [37].The algorithm works as follows:

1. Take the red channel of an image.
2. Look for the brightest point. This can be done using the minMaxLoc function in OpenCV
3. To reduce false positives, check if the detected point is less than 20 pixels away from the previously detected point.

6.3.1.4 Feature-level Fusion

All the data extracted from the different sources, such as depth, pixels belonging to a body and color blobs are various extracted features from the different input sources. The next step is to use feature-level fusion to determine if the user is pointing at the presentation using his hand, a pen or a stick.
Pointing using hand: The initial idea was to determine a cuboid in front of the projection plane. Using depth frame data, once an object would be

detected within this cuboid, we could assume the screen was being pointed at or touched.

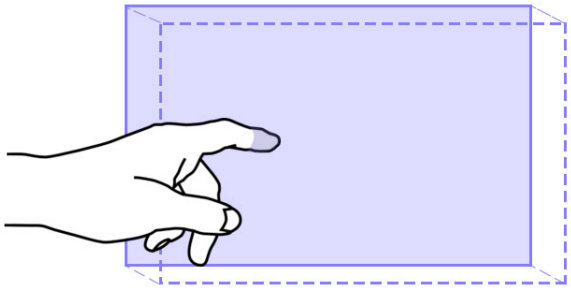


Figure 6-4: Cuboid to determine pointing and touch.

Unfortunately, this didn't work well as expected because of the noise in the depth frame when there is an object standing in front of the plane. While the measured coordinates of the object in front of it will we stable, the coordinates on the borderline between what the sensor sees as the object in front, and the plane in the background, will vary. This will then give false positives of measured points within the determined cuboid as can be seen in Figure 6-5.

Figure 6-5: Noise in red behind the hand in the cuboid on plane.

Adding distance between the plane and the cuboid we use to detect will reduce the amount of noise. Unfortunately, it does not reduce noise enough

so that we have usable results.
A solution for these false positives is to fusion and combine information from previously detected features to extract the requested information. In this use case the solution is provided with the data obtained from the body and body index frames. Using this data, we can detect whether or not the user is pointing within the projection screen. If we can detect a a hand joint in the projected screen then we'll check the depth of every pixel within a threshold radius of that joint and see if it is close enough to the projected screen to be considered as pointing. Noise in the background is ignored since we first use the body index frame as a mask, meaning that we only use pixels that are part of the body. If we can find pixels matching these conditions, then we take the closest point to the surface of the projected screen as the point being pointed at.
Pointing using a pen or stick: When pointing with a pen or stick, the pixels containing the pen or stick won't be part of the body. We thus can't use the body and body index frame to narrow our search. However, we can still detect if an object or something is close to the screen, using information from the depth frame. To reduce noise, we consider a stick or pen to have a almost uniform color. Using this property, we can use blobs to find objects being close to the projected screen. Unfortunately that won't be enough, as the red region in Figure 6-5 will often have the same properties as well. However, the detected blob with noise will most likely be completely within the cuboid, while the pen or stick approach the screen. Therefore, the blob for a pen or stick is at the same time inside and outside the cuboid. Using three smaller cuboid, noise can be eliminated if we expect the approaching object to be present in the three different cuboids. This technique will also work when a hand is approached when the body frame did not detect anyone inside the frame. The result can be seen in Figure 6-6.

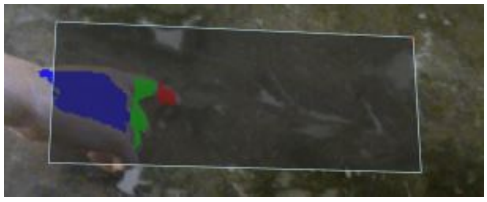


Figure 6-6: Usage of three cuboids for more accurate blob detection.

The closest point to the surface of the projected screen will again be used as the point being pointed at.

6.3.1.5 Decision-level Fusion

Based on the visual and audio data gathered, we now perform decision-level fusion. As we can now detect pointing towards the presentation and we have

speech recognition, we need to decide whether or not we want to trigger automatic highlighting, hence the need for decision-level fusion.
If we detected a pointing point using hand, pen, laser-pointer or stick, then we still need to know if a certain keyword was said by the presenter while he was pointing towards the presentation. If these two conditions are fulfilled, then the application will compute an outgoing message.

6.4 Laser Pointer as Input Device

Decision and Action State	When the speaker has the intention to make use of hypermedia, such as video, images and hypertext or when he has the intention of changing slide (decision), he can use the pointer gestures to interact with the presentation (action).
Perception	Following input sensors are used to gather all information: • RGB Camera
Interpretation	*Visual-Only Laser Pointer Detection*: We use the same implementation used for as in previous section for laser pointer detection. *Laser Pointer gestures*: The detected laser points are stored in an array for a short period. During every loop, we try to verify if there is a gesture that can be detected. Those gestures are: • Go to next slide • Go to previous slide • Go to parent view • Click on an element
Computation	*Message construction and channel selection*: The type of message constructed will vary with the registered gesture. All gestures with the exception of the gesture to click on an element will trigger en event in MindXpres containing the type of gesture. The click on an element gesture would simply send the (x,y) coordinates to the operating system to trigger a mouse click.
Action and Perception	In case of a non-click laser pointer gesture, MindXpres will process the incoming event and act accordingly.(action) In case of a click laser point gesture, a mouse click event is triggered in the operating system at the registered position of the laser pointer (action). Depending on the incoming event, the audience and can see the speaker navigating through slides or activate components which require user action (perception).

6.5 Simulate touch

Decision and Action State	When the speaker has the intention to make use of hypermedia, such as video, images and hypertext or when he has the intention of changing slide (decision), he can touch interaction to interact with the presentation (action).
Perception	Following input sensors are used to gather all information: • RGB Camera • Depth camera
Interpretation	The interpretation uses the same code that is being used for the automatic highlighting pointing detection during the feature level fusion. However, a more strict threshold is applied to make sure that the surface of the projected screen is being touched.
Computation	*Message construction and channel selection*: If touch is being detected during feature-level fusion, then we send the (x,y) coordinates to the operating system to trigger a mouse click.
Action and Perception	A mouse click event is triggered in the operating system at the registered position of the hand (action). Simulating touch will enable the presenter to interact with the operating system as well as with MindXpres. According to his action, the audience will hear or see different actions within a presentation (perception).

6.6 Annotate Using Pen or Pen Shaped Object

Decision and Action State	When the speaker has the intention to further extend or clarify the content that is displayed on the slide (decision), he can use a pen or pen shaped object to write digitally on the slide (action).
Perception	Following input sensors are used to gather all information: • RGB Camera • Depth camera
Interpretation	Basically, annotating is the same thing as simulating touch. However, in this case we expect to user to use movement while writing. In the next section, we will discuss these differences in more detail .
Computation	*Message construction*: The message constructed will trigger en event in MindXpres. This event will enable or disable the canvases used for annotations. *Output channel selection*: When we detect that the user is trying to make annotations, we send the (x,y) coordinates of the point where he is trying to make to the operating system as mouse events. These are mousedown, mousemove and mouseup according the sequence of detections made. At mousedown, the event to enable canvas annotations is send to MindXpres. At mouseup, an event to disable canvas annotation is send back so that the user can again interact with the slides.
Action and Perception	Annotations are being made on the slides in MindXpres. This was achieved by modifying the default slide container to include a HTML5 canvas as an overlay (action). The presenter now has the possibility to create, extend or clarify content on the slides with hand-made annotations (perception).

6.6.1 Interpretation

6.6.1.1 Depth Camera

Depth frame: We use only the Kinect API to store the depth of every pixel.

6.6.1.2 RGB Camera

Background removal: When the user annotates, we consider that the hand that is being used to write on the slide is in motion. OpenCv has built in algorithms to do this. In our case, we make use of the VideoSurveillance.BackgroundSubtractorMOG algorithm. *Blob detection*: We use the same blob detection as before, based on color.

6.6.1.3 Feature-level Fusion

We use feature-level fusion to determine if the user is trying to annotate on the slide using a pen or a stick. The fundamental difference with pointing using a pen, is that we consider that the hand of the user is in motion. To reduce noise, we can now use this mask to know which are needs to be looked at.
We will now detect using the depth frame whether the hand in motion is touching the surface of the projected screen. Using the three cuboids as we did before, we now want to make sure that the biggest blob in the cuboid closest to the surface is different from the biggest blob in the other two cuboids. This is a way for us to detect if a hand is holding an object with which he tries to annotate on the slide. The point closest to the surface is the input point.

6.6.1.4 Decision-level Fusion

In most cases, when we try to annotate our implementation will also detect touch. Therefore, we need to add another rule which states that no touch is allowed when annotations are detected.

6.7 Gestures for Free Creation of Content

Decision and Action State	When the speaker has the intention to further extend or clarify the content on the slide, or when he has the intention of removing annotated content (decision), he can use gestures to insert a blank slide or to remove made annotations (action).
Perception	Following input sensors are used to gather all information: • Depth Camera
Interpretation	We use the Kinect to detect body gestures, which will trigger various actions in MindXpres.
Computation	*Message construction*: The message constructed will trigger en event in MindXpres. This message send the command to either insert a new slide or to remove all annotations. *Output channel selection*: The output channel is MindXpres, where we push the message to the client using the WebSocket protocol.
Action and Perception	Insert a new slide or clear made annotations in MindXpres (action). The audience can see the creation of a new blank slide, or can see the annotation being removed from the slide (perception).

6.7.1 Interpretation

Body frame - Detecting gestures Using body frame data, we are now going to detect gestures. The Kinect API shines at detecting gestures. In our case, we have only two basic gestures that we want to recognize. Both being wave gestures.

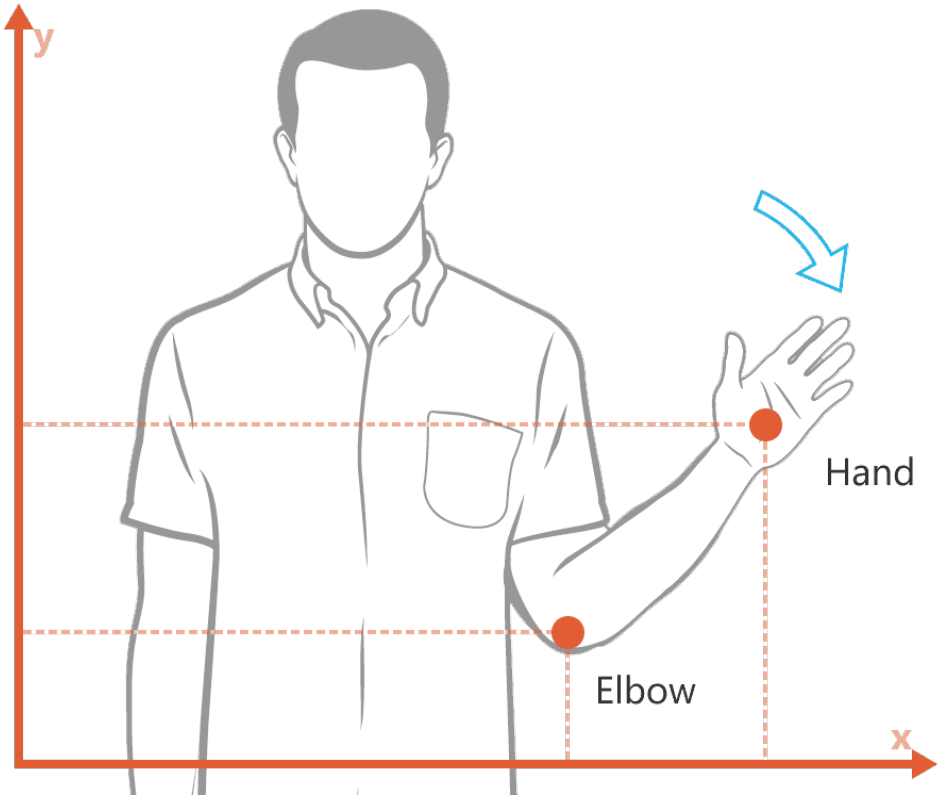


Figure 6-7: Detection of the wave gesture with Kinect.

To detect gestures, we track the position of specific joints over time. In the case of a wave gesture, we track the elbows and hands and see if we can detect segments of the gesture.
A segment is a part of a gesture, characterized by certain properties. For example, for a vertical wave gesture, a first segment would be that the hand position Y is higher than the elbow position Y AND the hand position X is higher than the elbow position X. In the second segment, the condition for Y would remain the same, while X position condition is inverted. If we are able to detect consecutive repeats for these segments, then we have successfully detected the gesture.

6.8 Conclusion

In this chapter we discussed how we enabled multimodal interaction for the MindXpres presentation tool. By using MindXpres plug-ins and client-server communication through WebSockets, all desired features could be inserted in the presentation tool. Furthermore did we discuss how every new feature was made possible with the Microsoft Kinect using the multimodal man-machine

interaction loop to give a complete overview on the complete process.

7 Use Case

In previous chapter we described the implementation of our ideal presentation tool with the insights that were gathered in Chapter 2 and Chapter 3 and the technologies discussed in Chapter 5. This chapter will walk through a scenario where the new implemented features are used. It will give the reader insight on the benefits of having multimodal interaction for MindXpres and how these features are used from a practical standpoint.

7.1 The Scenario

To illustrate our ideal tool in the best possible way, we define a scenario that will be used during this rest of this chapter which relates to a typical presentation that a student would receive during a lecture. For our scenario we use a hypothetical presentation that talks about the implementation of client-server communication in MindXpres to enable multimodal interaction. Moreover do we envision that the presenter does not have any information about the audience and their previous knowledge about the used technologies.

7.2 Creating the Presentation

We will not fully discuss how to make a presentation in MindXpres as this is not the primary focus of this thesis. However, it is good to have a general idea of how a presentation is created using the XML language of MindXpres that compiles into HTML5. For more information we kindly refer to [39].

7.2.1 The Structure

Every presentation starts with the root `<presentation>`. We have thus following code:

```
<presentation></presentation>
```

Which compiles into an empty presentation as is shown in **??**.

Figure 7-1: Empty presentation

We now start adding content to it. The first thing we'll do is define the global structure, with the title of our presentation.

```
<presentation>
  <structured title="Multimodal Interaction for the MindXpres
      Presentation Tool">
  </structured>
</presentation>
```

Next can be add various sections to the presentation. In our use case, we'll just add the sections that we need, with each section containing an empty slide.

```
<presentation>
  <structured title="Multimodal Interaction for the MindXpres
      Presentation Tool">
    <section title="Client-Server Communication">
      <slide></slide>
    </section>
    <section title="Integration in MindXpres">
    <slide></slide>
    </section>
  </structured>
</presentation>
```

The result of the code above can be seen on **??**

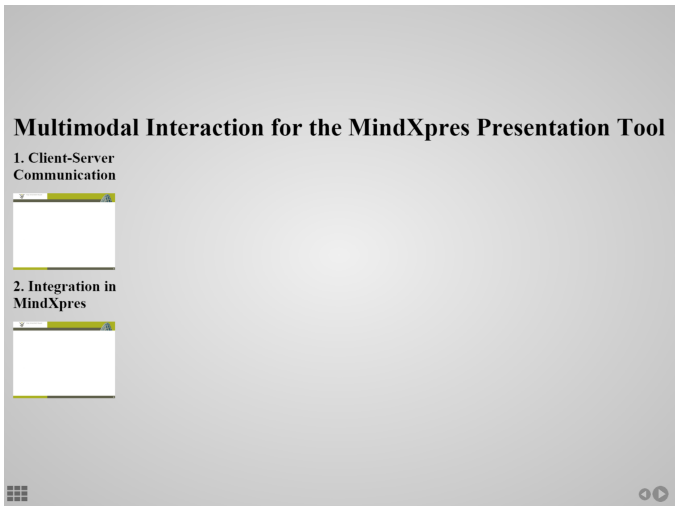


Figure 7-2: Presentation with title, sections and empty slides.

7.2.2 Add Automatic Highlighting

To enable automatic highlighting, we need to define which text that should be highlighted, together with the keyword that will trigger the automatic highlighting. To do this, we implemented the highlight module. Defining the part that needs to be highlighted can then be defined with the `<highlight>` tag.
If we would like to highlight automatically "bi-directional communication" in the slide shown on **??**, we would have the XML code defined in **??**.

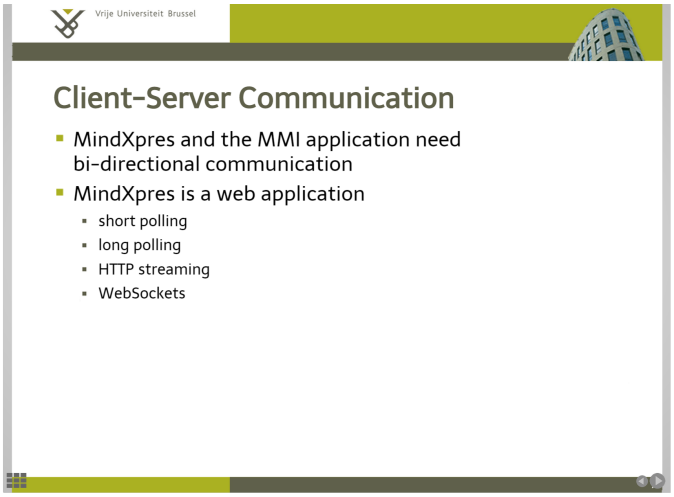


Figure 7-3: Slide where we want to automatically highlight "bi-directional communication".

```
1  <slide title="Client-Server Communication">
2    <bulletlist>
3      <item>MindXpres and the MMI aplication need
4        <highlight keyword="bi-directional">bi-directional
             communication</highlight>
5      </item>
6      <item>MindXpres is a web application
7        <item>short polling></item>
8        <item>long polling</item>
9        <item>HTTP streaming</item>
10       <item>WebSockters</item>
11     </item>
12   </bulletlist>
13 </slide>
```

7.3 Launching the Presentation

7.3.1 Setting up the Kinect Webserver

Before launching MindXpres, we need to launch the Kinect server if we want to enable multimodal interaction. If we do not, MindXpres will not be able to make a connection with the server and thus it won't receive any of the triggered events which enable most part of the newly implemented features.

7.3.1.1 Choosing the Projector

Upon opening the Kinect server application, the user is first given the option to pick the screen where the projection is being held. By doing so, the appli-

cation knows where to send the callibration points as well as the resolution of the screen that will be used to map any pointing or touch related input.

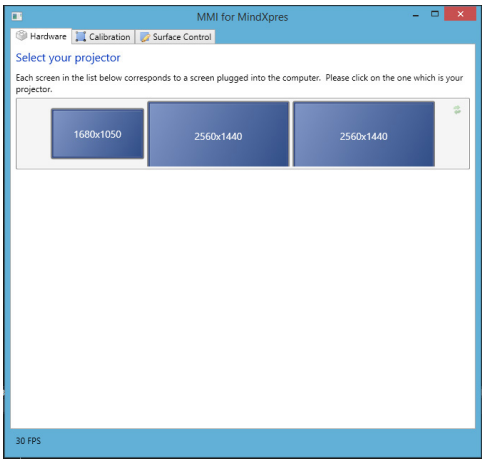


Figure 7-4: Select which detected computer screen is the projector

7.3.1.2 Calibrate the Projection Plane

Once the right projector has been selected, the user will be guided through the calibration process. The user will have to select the projected points on the camera visualisation to complete the process as can be seen on **??**

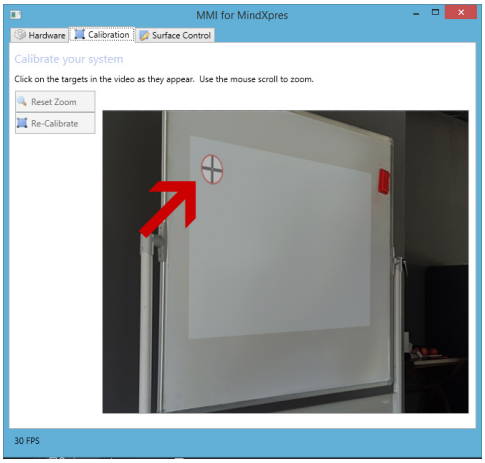


Figure 7-5: Calibration of the projection plane

7.3.1.3 Adjust the Surface of the Projection

Once the calibration is done, the user can still modify the calculated surface by moving the corners of the surface by clicking on the red dots as can be seen on **??**.

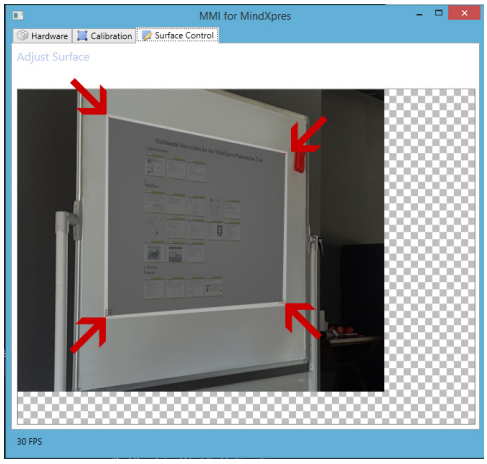


Figure 7-6: Adjust surface of the projection plane

7.3.2 Launch MindXpres

All that is left to do now is opening MindXpres with a modern web browser. The connection will automatically be made in the background for multimodal interaction.

7.4 Interaction During the Presentation

Let us consider a reduced presentation for the chosen topic. The presenter does not know the previous knowledge of the audience, so for the sake of completeness he included slides about the various communication mechanism for client-server communication.

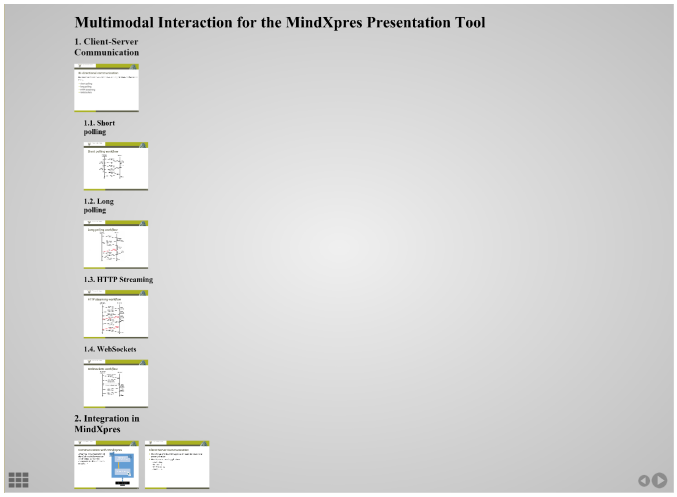


Figure 7-7: Reduced presentation overview

7.4.1 Interact Using Touch

The presenter decides that he will try to immediately skip the client-server communication section and moves on to the integration in MindXpres. He uses touch to go to the jump to the slide of his choice.

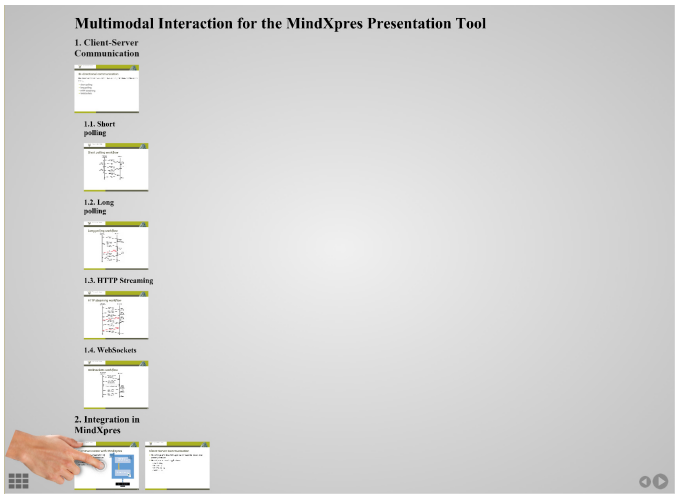


Figure 7-8: The presenter uses touch to select the slide.

7.4.2 Automatic Highlighting

On the next slide, the bi-directional communication is significant because it is this type of requirement which will have us use certain client-server communication mechanisms to fulfil this requirement. The presenter thus points towards this part on the slide using a laser pointer. Note that he could also have pointed using a stick, pen or simply his hand. When he says the word "bi-directional" while pointing, the significant part will be highlighted as can be seen on **??** and **??**.

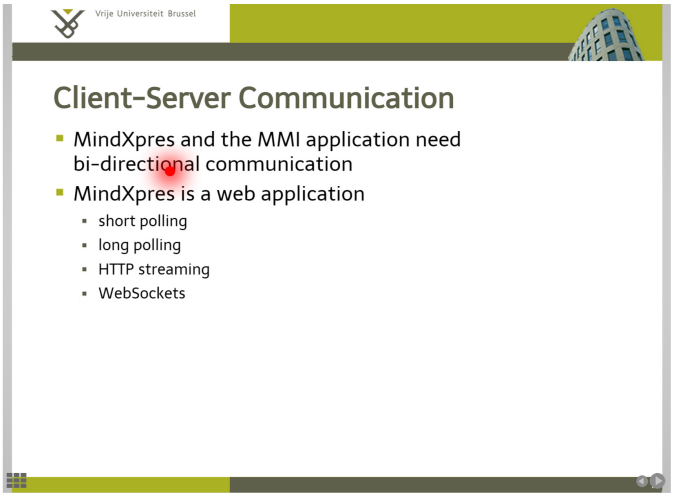


Figure 7-9: The presenter points to the significant part with a laser pointer.

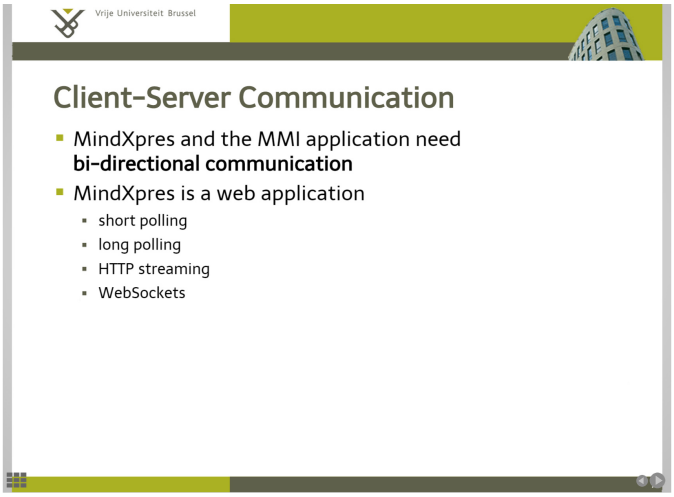


Figure 7-10: The significant part gets highlighted.

The pointing and highlighting thus give extra contextual information. By signifying the bi-directional communication, we can deduce that, since MindXpres is a web application, only the listed mechanisms provide us with a solution.

7.4.3 Interact Using the Laser Pointer

We now consider that the audience lacks prior knowledge of the WebSocket protocol. The presenter thus decides to go to the slides he has prepared in Section 1 which explain WebSockets. He has two option, he can either click on the bullet "WebSockets" (see **??**) if this is an hyperlink towards

the corresponding slide explaining WebSockets, or he can can use the sliding gesture to move to the upper level as illustrated in ??.

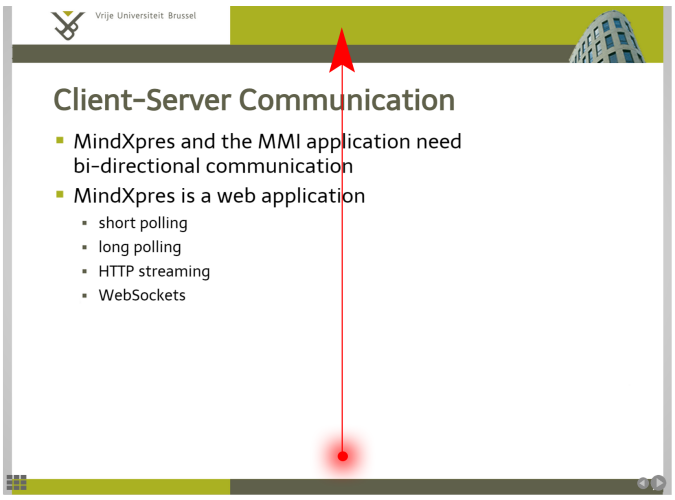


Figure 7-11: Use laser pointer gesture to go to the upper level.

Once at the upper level, the presenter can now "click" on the slide he wants to navigate to by pointing at the slide as can be seen on ??.

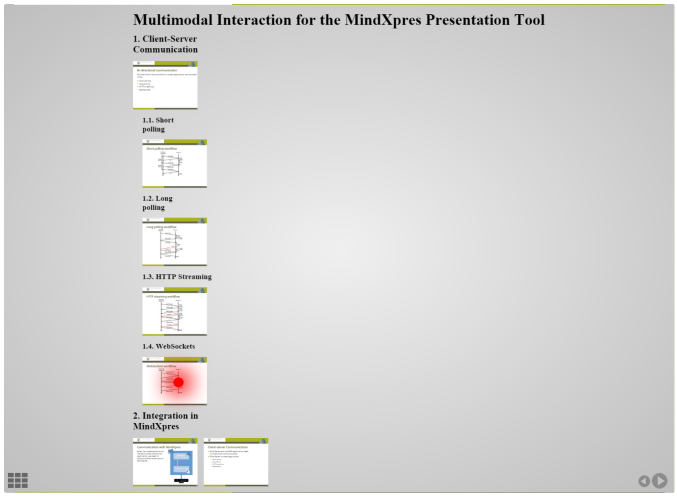


Figure 7-12: Select the slide using a laser pointer.

7.4.4 Annotate

Once the presenter is on the WebSocket slide, he wants to extent the content by annotating the given figure. Using a pen or pen-shaped object in his hand, he can now annotate the slide to better illustrate how WebSockets enable full-duplex communication.

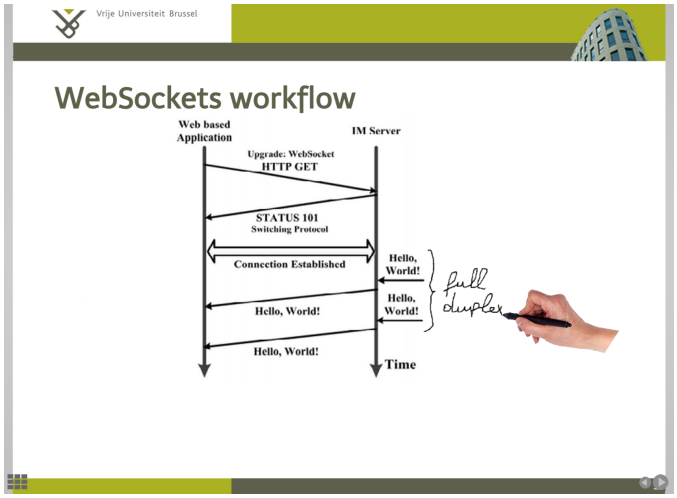


Figure 7-13: Annotate the slide using pen or pen-shaped object.

7.4.5 Using gestures

Finally, the presenter can also insert an empty slide or remove his made annotations using gestures. This gives him again an extra level of freedom for the creation of content on the slides.

7.5 Conclusion

In this chapter, we have seen how we can create a MindXpres presentation and how to calibrate the Kinect server to enable multimodal interaction. By using multimodal interaction, the presenter now has the possibility to use the zoomable user interface of MindXpres to its full extend. By doing so, he can for example use multiple navigation paths to adapt to his audience, as seen in our scenario. The presenter can also create content freely if required, and interact with any component that is implemented in MindXpres.
If we reflect on these new features, one may say that nothing extraordinary has been implemented. It is just a natural way to interact with what we already know... However, this is exactly the goal of our multimodal interface.

8 Conclusions and Future Work

In order to integrate multimodal interaction for the MindXpres presentation tool, we first analysed how current common slideware is used as well as how we could use multimodal interaction to cope with the current limitations that common slideware impose. We quickly concluded that not all criticism on common slideware is valid, since a presentation should not be characterized by slides alone, but also by the bodily and spoken performance.
Once we were familiar with these limitations of common slideware, as well as with some alternatives which make use of other input modalities, such as interactive whiteboards, we proposed a solution of our own. In this solution, we carefully picked some features which enhance the interactivity of common slideware in a way which we could perceive as being a natural way to interact. These features would focus on pointing, the main gesture used when giving a presentation, the lack of alternative traversal paths due to the linearity of common slideware and the unavailability to create content freely. By doing so, we believe we bring together the advantages of using digital media and the old-school way of using the blackboard.
All of the features were then implemented in MindXpres, which proved to be an excellent tool due to its modularity and content-oriented approach. Still, integrating multimodal interaction was a challenge nevertheless. As MindXpres is a web application, we had to make sure that all information would be passed quickly and with a minimum delay from our Kinect application. Moreover did we need to guarantee that the Kinect application would process all information quickly enough for it to become usable in real-time. Therefore, we made use of the visual processing library OpenCV next to the Kinect API, which focusses on real-time processing. For the communication we implemented a light-weight server capable of using WebSocket

communication.

8.1 Contributions

A first contribution would be an overall literature study of common slideware, looking both at the content as well as their interaction on a large scale. Our study tried to identify both the positive aspects and negative aspects of using common slideware, as well as how interaction could help in resolving the biggest concerns discussed in the literature.
Secondly, we proposed a solution to cope with the limitations previously identified, and created a cheap Kinect-based application which enables multimodal interaction in MindXpres. The features of MindXpres were also extended through plug-ins, to make it capable of creating content freely as well as augmenting the interplay between the presenter and his presentation.
On a final note, while the solution was originally developed to work with MindXpres, it can also be used to interact with the operating system. It could also be modified to work with other applications and thus leaves a door open for future development.

8.2 Future Work

In our solution, we analysed how we currently use and interact with common presentations tools to propose a solution. However, in the future a user evaluation is needed to test our expectations against reality.
Due to the fact that we have only taken a look at how we use current presentation tools, the input modalities used to enable multimodal interaction only use simple gestures and movements such as touch and pointing. It could however be beneficial to look at completely different interaction possibilities, such as for example the Myo armband[1].
Right now presentations are often printed as Cole's notes, as they give a good overview of the knowledge that is being communicated during a presentation. However, looking back at these often results in a loss of information, as the primary goal of a presentation is not to provide notes. Using multimodal input, we could now easily save the whole presentation with speech, video and interactions. Providing the right user interface, a user could now for example replay parts of the presentation while reviewing slides.

[1] https://www.myo.com/

Bibliography

[1] Kinect for windows sdk. https://msdn.microsoft.com/en-us/library/hh855347.aspx. Accessed: 2015-07-30.

[2] Kinect for windows v2 api. https://msdn.microsoft.com/en-us/library/dn782033.aspx. Accessed: 2015-07-30.

[3] Opencv library. http://code.opencv.org/projects/opencv/wiki. Accessed: 2015-08-03.

[4] C. Adams. Powerpoint, habits of mind, and classroom culture. *Journal of Curriculum studies*, 38(4):389–411, 2006.

[5] J. Albahari and B. Albahari. *C# 5.0 in a Nutshell: The Definitive Reference*. " O'Reilly Media, Inc.", 2012.

[6] C. Atkinson. *Beyond bullet points: Using Microsoft PowerPoint to create presentations that inform, motivate, and inspire*. Pearson Education, 2011.

[7] N. Austin. Mighty white. *The Guardian*, January 2003.

[8] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext transfer protocol–http/1.0. Technical report, 1996.

[9] T. J. Berners-Lee. The world-wide web. *Computer Networks and ISDN Systems*, 25(4):454–459, 1992.

[10] S. Board. The history of smart. http://smarttech.com/ca/About+SMART/About+SMART/Innovation/Beginnings+of+an+industry. Accessed: 2014-07-15.

[11] R. A. Bolt. *"Put-that-there": Voice and gesture at the graphics interface*, volume 14. ACM, 1980.

[12] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[13] J. C. Cuthell. The impact of interactive whiteboards on teaching, learning and attainment. In C. Crawford, R. Carlsen, I. Gibson, K. McFerrin, J. Price, R. Weber, and D. A. Willis, editors, *Proceedings of Society for*

Information Technology & Teacher Education International Conference 2005, pages 1353–1355, Phoenix, AZ, USA, 2005. AACE.

[14] J.-L. Doumont. The cognitive style of powerpoint: Slides are not all evil. *Technical communication*, 52(1):64–70, 2005.

[15] B. Dumas, D. Lalanne, and S. Oviatt. Multimodal interfaces: A survey of principles, models and frameworks. In *Human Machine Interaction*, pages 3–26. Springer, 2009.

[16] I. Fette and A. Melnikov. Rfc 6455: The websocket protocol. *IETF, December*, 2011.

[17] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol–http/1.1. Technical report, 1999.

[18] H. Gonzalez-Jorge, B. Riveiro, E. Vazquez-Fernandez, J. Martínez-Sánchez, and P. Arias. Metrological evaluation of microsoft kinect and asus xtion sensors. *Measurement*, 46(6):1800–1806, 2013.

[19] T. Goodison. Ict and attainment at primary level. *British Journal of Educational Technology*, 33(2):201–211, 2002.

[20] A. G. Gross and J. E. Harmon. The structure of powerpoint presentations: the art of grasping things whole. *Professional Communication, IEEE Transactions on*, 52(2):121–137, 2009.

[21] J. Harris. Text annotation and underlining as metacognitive strategies to improve comprehension and retention of expository text. 1990.

[22] W. Hürst and J. Meyer. A new user interface design for giving lectures and presentations. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, volume 2004, pages 2693–2700, 2004.

[23] A. Kendon. *Gesture: Visible action as utterance*. Cambridge University Press, 2004.

[24] S. Kennewell, H. Tanner, S. Jones, and G. Beauchamp. Analysing the use of interactive technology to implement interactive teaching. *Journal of Computer Assisted Learning*, 24(1):61–73, 2008.

[25] H. Knoblauch. The performance of knowledge: Pointing and knowledge in powerpoint presentations. *Cultural sociology*, 2(1):75–97, 2008.

[26] M. Lee and M. Boyle. The educational effects and implications of the interactive whiteboard strategy of richardson primary school.

http://www.richardsonps.act.edu.au/__data/assets/pdf_file/0020/83117/RichardsonReview_Grey.pdf, October 2003. Accessed: 2014-07-15.

[27] P. Levy and S. E. in Cities Partnership. *Interactive Whiteboards in Learning and Teaching in Two Sheffield Schools: A Developmental Study*. Sheffield Excellence in Cities Partnership, 2002.

[28] S. Loreto, P. Saint-Andre, S. Salsano, and G. Wilkins. Known issues and best practices for the use of long polling and streaming in bidirectional http. Technical report, 2011.

[29] M. McLuhan. Laws of the media. *ETC: A Review of General Semantics*, pages 173–179, 1977.

[30] N. Mercer, S. Hennessy, and P. Warwick. Using interactive whiteboards to orchestrate classroom dialogue. *Technology, Pedagogy and Education*, 19(2):195–209, 2010.

[31] R. O. Mines Jr. Do powerpoint presentations really work? *age*, 6:1, 2001.

[32] S. Y. Mousavi, R. Low, and J. Sweller. Reducing cognitive load by mixing auditory and visual presentation modes. *Journal of educational psychology*, 87(2):319, 1995.

[33] L. Nigay and J. Coutaz. A design space for multimodal systems: concurrent processing and data fusion. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 172–178. ACM, 1993.

[34] S. Oviatt. Advances in robust multimodal interface design. *IEEE Computer Graphics and Applications*, (5):62–68, 2003.

[35] S. Oviatt. Human-centered design meets cognitive load theory: designing interfaces that help people think. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 871–880. ACM, 2006.

[36] J.-t. Park, H.-s. Hwang, J.-s. Yun, and I.-y. Moon. Study of html5 websocket for a multimedia communication. *International Journal of Multimedia & Ubiquitous Engineering*, 9(7), 2014.

[37] E. Popovich and Z. Karni. Presentermouse laser-pointer tracking system. *www. mpi-inf. mpg. de/ karni/presentermouse/report. pdf*, 2006.

[38] L. M. Reeves, J. Lai, J. A. Larson, S. Oviatt, T. Balaji, S. Buisine, P. Collings, P. Cohen, B. Kraal, J.-C. Martin, et al. Guidelines for multimodal user interface design. *Communications of the ACM*, 47(1):57–59, 2004.

[39] R. Roels and B. Signer. An extensible presentation tool for flexible human-information interaction. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*, page 59. British Computer Society, 2013.

[40] K. Shuang and F. Kai. Research on server push methods in web browser based instant messaging applications. *Journal of Software*, 8(10):2644–2651, 2013.

[41] H. J. Smith, S. Higgins, K. Wall, and J. Miller. Interactive whiteboards: boon or bandwagon? a critical review of the literature. *Journal of Computer Assisted Learning*, 21(2):91–101, 2005.

[42] J. Sweller. Implications of cognitive load for multimedia learning. In R. E. Mayer, editor, *The Cambridge handbook of multimedia learning*, pages 19–30. Cambridge University Press, 2005.

[43] A. Thomas. Little touches that spell success. *Times Educational Supplement*, May 2003.

[44] S. Tindall-Ford, P. Chandler, and J. Sweller. When two sensory modes are better than one. *Journal of experimental psychology: Applied*, 3(4):257, 1997.

[45] E. R. Tufte. *The cognitive style of PowerPoint*, volume 2006. Graphics Press Cheshire, CT, 2003.

[46] F. Vogt, J. Wong, S. Fels, and D. Cavens. Tracking multiple laser pointers for large screen interaction. In *Extended Abstracts of ACM UIST*, pages 95–96, 2003.

[47] Wikipedia. Kinect — wikipedia, the free encyclopedia, 2015. [Online; accessed 21-July-2015].

[48] Wikipedia. Kinect for xbox one — wikipedia, the free encyclopedia, 2015. [Online; accessed 21-July-2015].