Laura Sofia Martins Oliveira

Licenciada em Engenharia Informática

Interfaces for Television Content Sharing and Annotation

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador : Professor Doutor Nuno Manuel Robalo Correia, Prof. Catedrático, Universidade Nova de Lisboa

Júri:

Presidente: Prof. Doutor José Augusto Legatheaux Martins

Arguente: Prof. Doutora Maria Teresa Caeiro Chambel

Vogal: Prof. Doutor Nuno Manuel Robalo Correia

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2013

Interfaces for Television Content Sharing and Annotation

Aos meus pais

Acknowledgements

The contributions of several people and entities have made this thesis possible and I would like to extend my appreciation especially to the following.

Foremost, I would like to express my gratitude to my supervisor Prof. Nuno Correia for the opportunity to work and explore this theme and for his guidance and support. Also, for the encouragement to take this project beyond the scope of this dissertation.

Furthermore, I would like to thank the Departamento de Informática da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (DI - FCT/UNL) and the project ImTV (On-Demand Immersive-TV for Communities of Media Producers and Consumers), that includes this dissertation, with the reference UTA-Est/MAI/0010/2009, funded by Fundação para a Ciência e a Tecnologia.

Also, I like to thank the participants in my survey, who have willingly shared their precious time for evaluating the application.

A special thanks goes to my dearest friend Inês, who volunteered for the tedious task of being my proofreader and whose critiques and comments improved the composition of this thesis.

Finally, I would like to thank my loved ones who have supported me through this process and through life, in particular my parents, for their immense patience, encouragement and endless love.

Abstract

The ways of television consumption and production are changing significantly, with the viewers moving away from the traditional linear model. The various devices for accessing content have a significant role in these changes and suggest new paradigms of access. Social experience has also changed and takes on new forms molded by technology. Content sharing and production from users are some of the trends that globally influence how we relate to audiovisual content.

Therefore the aim is to develop ways to access television content, that allow commenting and sharing, through multimodal annotations. These annotations include text, sketches, handwriting and images. Our solution provides users a way to watch and annotate television content, in real-time and in a collaborative environment. Using a mobile device, users can annotate content together with other users, while watching both content and annotations on a TV. These annotations can also be shared through social networks or saved on other platforms. Finally, the system also uses content provided by the users to search and link to television content.

Keywords: multimedia content, video, annotation, television, mobile, shared and social TV.

Resumo

As formas de consumo e produção de televisão estão a mudar de forma significativa, com os espectadores a afastar-se do modelo linear tradicional. Os diversos dispositivos para aceder a conteúdos contribuem para estas alterações e sugerem novos paradigmas de acesso. Também a experiência social se alterou e assume novas formas mediadas pela tecnologia. A partilha de conteúdos e a produção por parte dos utilizadores são algumas das tendências que influenciam globalmente a forma como nos relacionamos com conteúdos audiovisuais.

O objectivo é assim desenvolver formas de acesso a conteúdos de televisão que permitam a partilha e comentário, através de anotações multimodais. Estas anotações incluem texto, desenho (sketches) e imagens. A nossa solução irá permitir aos utilizadores visualizar e anotar conteúdo televisivo, em tempo real e em ambiente colaborativo. Através de dispositivos móveis, os utilizadores podem anotar conteúdo em conjunto com outros utilizadores, enquanto visualizam o mesmo conteúdo e as anotações na televisão. Esta anotações podem ser partilhadas através das redes sociais ou guardadas em outras plataformas. Finalmente, o sistema também utiliza conteúdo disponibilizado pelo utilizador de forma a procurar e estabelecer uma ligação com conteúdos televisivos.

Palavras-chave: Contéudos multimédia, vídeo, anotação, televisão, mobile, televisão partilhada e social.

Contents

List of Figures

Listings

1

Introduction

The television experience is changing everyday. Long gone are the days where people stood by a television and just watched a broadcasted program. New technology and new ways of consuming and sharing content affected not only television but also the social experience. Sharing, commenting and receiving feedback became an essential part of our daily lives. Being such a common thing nowadays we would expect it to be quite simple. Instead, we see ourselves sometimes struggling to find what we need and to keep track of all the different content being posted in all the different sharing platforms. The act of sharing by itself is relatively simple, as there are several platforms and social networks at our disposal for that matter. However, finding exactly what we want, and making that content accessible to those we want, independently of where we are or what devices we have at our disposal, is a whole different problem. To solve this problem we must begin by understanding the importance of annotations.

An annotation is something that is added to a document, photo or video, with an intent of improving organization, memory, search and sharing. An annotation can be plain text, an audio note, a tag or even a drawing. The main motivations for annotating are organization and communication [AN07]. Annotations can be useful for organizing content, for oneself or for others, which then can also improve searching. Annotations can also give context and help someone recall where a video, or photo, was taken and when. Focusing on the social part, annotating enhances the sharing experience by facilitating the search of content by others and thus easily exposing what we intent to [AN07]. It also gives context to someone who has never seen the content.

Since television content sharing and commenting is relatively new and there is much to improve, the objective is thus to develop new ways that make possible for a user to share, comment and annotate that content.

1.1 Problem Description and Context

As previously mentioned, there is much to be done in the television content sharing area. Social experience and content sharing have continuously been improved but mainly for the web, so there is a need to bring that experience to television.

Television is a social experience, as people like to comment and share what they watched with others [WUMCKB11]. Providing means for users to share fragments of television content, and to enrich them, enhances this experience. The idea of creating personal bookmarks and sharing fragments of video with family and friends seems to be something that users are looking for, and that they will probably use if it is available for the television as well [CBGJKS08]. Making annotations and sharing content through the TV can be a hard task for now, so it is also important to think of ways and what technology to use for making this possible.

This project aims to create a system that allows users to share, annotate and retrieve television content. The proposed system will be divided into three components with the following objectives:

- Multimodal annotation of video, focusing on television content;
- Collaborative annotation within groups of users;
- Integration with social networks by sharing content and annotations;
- Offering a social TV experience to the users.

1.2 Presented Solution

The aim is to provide a system that allows for sharing television content and commenting through multimodal annotations. The idea is to have a collaborative and interactive environment where several users annotate and watch the same media content.

For annotating content, the user will be able to retrieve a frame or a fragment of the video being watched to their smartphone and then proceed to making annotations. These annotations can be free text, drawings, handwriting writing with the use of a pen, links to other videos or photos, and possibly audio notes. Once the user has finished, the annotations will be stored in the system. When annotating the usage of a pen is preferred, since it provides a more natural interaction for the user being very similar to human-paper-pen interaction.

Semi-automatic annotation, using image-based matching techniques, will also be considered. For example, if a user is watching a documentary about Paris and has been there before on a trip, he/she could add their own photos to that content. To achieve this the user provides their photos, or other videos, to match with the video content. If necessary the user can also add other files manually to complement the results.

The technology used to develop this solution is a smarthphone with the Android system and a television set. Currently we are using the Samsung Galaxy Note as it has support for ink based writing. The system was developed in C++ using the openFrameworks platform [Of], which is a library dedicated to creative programming. The openCV (Open Source Computer Vision) [Ocv] library was used for image processing and image-based matching.

1.3 Contributions

The main contributions of this thesis are:

- **Multimodal Annotation for the TV**: A model and a mobile application for annotation of television content, either with text, handwriting, drawing and links to other files;
- **Collaborative Annotation**: A system that manages annotations from several users on several videos or on the same video;
- **Television Experience and Content Viewing**: A system for presenting on a TV, or other visual display units, the content with the respective annotations;
- **Integration with Social Networks**: A system that is integrated with social networks and cloud storage services, allowing to share and save annotations on other platforms.
- **Video Search based on Image Matching Techniques**: A system for searching videos using images or photographs belonging to the user. For matching images with videos, image matching techniques will be used.
- **User Evaluation**: Usability tests of the system to evaluate both the interface and the application itself. After evaluating the system through a questionnaire and by gathering suggestions from the users, an analysis of the results will follow.

1.4 Document Structure

This document is structured in six chapters: introduction, related work, television content annotation, prototype development, evaluation and conclusion and future work.

The first chapter, **Introduction**, presents an overview of the dissertation, where several issues will be addressed such as context, problem description, proposed solution and the expected contributions. The second chapter, **Related Work**, is dedicated to systems related to this thesis whose features or techniques are relevant to our solution. The Related Work chapter focuses on multimedia annotation systems, television applications and television content annotation and sharing, and proximity interaction between devices and the environment. The third chapter, **Television Content Annotation**, describes

the implemented solution, first by defining the concept and the system's requirements and then presenting the interface and functionalities that fulfill those requirements, as well as the technologies used. Chapter four, **Prototype Development**, explains in detail how the solution was implemented, presenting the system's architecture and details regarding the implementation of the main functionalities. The fifth chapter, **Evaluation**, analyses the results obtained from the evaluation of the solution. Finally the last chapter, **Conclusion and Future Work**, critiques and comments the work developed for this thesis and includes possible improvements outlined as future work.

2

Related Work

This chapter presents the necessary concepts and techniques for the development of our solution. In the following sections several systems and algorithms related to this thesis will be presented. This chapter is divided into four main sections. The first section is dedicated to algorithms used for image matching and object recognition. The second section will cover systems for television and television content annotation. The third section will present systems that provide tools for video and photo annotation. The last section addresses collaborative annotation systems and proximity interaction between devices and the environment.

2.1 Image Matching and Object Recognition

In the 1990s, there was a burst of activity in the computer vision field relating recognition and motion. However, it was only in the past decade that techniques based on features for object recognition emerged [Sze10]. A feature can be defined as being an "interesting" part of an image. There are different kinds of image features, also called keypoints or interest points, such as corners and ridges. Another important class of features are edges, which can be useful for indicating object boundaries [Sze10]. The following figure (2.1) shows keypoints detected on a image and each circle represents a point that was considered to be relevant. Notice that the points found do a rough outline of the image and are more predominant around the boundaries of the monument.

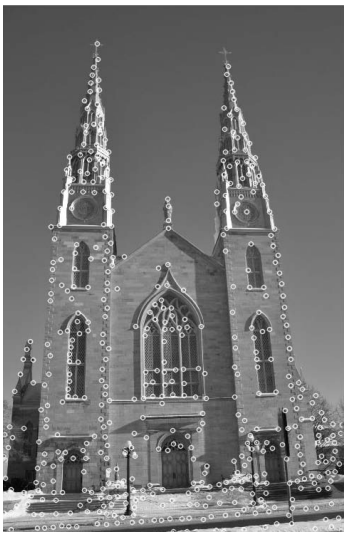

Figure 2.1: Features or keypoints detected on an image [Lag11]

Feature detection and matching is useful for object recognition and image-based matching, as well as other fields of computer vision. By using this kind of techniques it is possible to discover if, for example, two pictures match, *i.e.*, if they were taken in the same place or even if they were both taken during the night. The first step to accomplish this is to detect features of an image, as illustrated in the figure above. Sometimes detecting features produces too many keypoints and there are several algorithms to eliminate "bad" points, that involve imposing a minimum distance between two interest points. The second step is to describe features by extracting the descriptors of each feature. A descriptor describes different characteristics of a feature, such as color, texture and shape. Having the descriptors from each feature from both images we can now match them, comparing the feature descriptors in the first image to all the feature descriptors in the second image. The best matches are the results of the matching of the two images. The following figure (2.2) shows the match results and each match is a line that links a point to its corresponding image point on the other image.

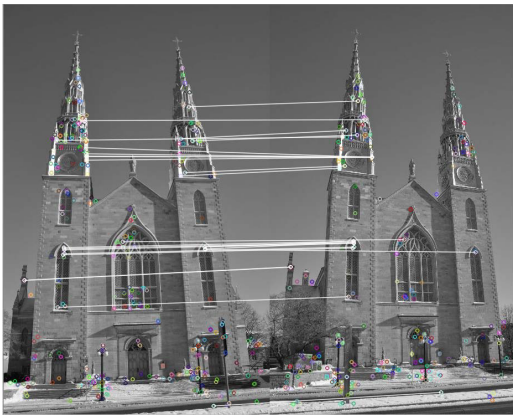

Figure 2.2: Image matching results [Lag11]

Since matching content of images and video will be a part of this thesis, there is a need to explore different techniques related to this subject. There are several feature detectors and in this section three of them will be discussed: SIFT [Low99], SURF [BTG06], and FAST [RD06].

2.1.1 SIFT

Scale Invariant Feature Transform (SIFT), developed by David Lowe [Low99], is an algorithm created for image matching and object recognition. The SIFT descriptor is invariant to translations, rotations and scaling transformations. It is also robust to changes in illumination, noise, and minor changes in viewpoint. In addition, SIFT features are highly distinctive and generally allow for a correct object identification with low probability of mismatch.

The SIFT algorithm consists of four key stages: scale-space extrema detection, keypoint localization, orientation assignment and keypoint descriptor. The first stage is where the keypoints, or points of interest, are detected. Scale-space extrema detection produces many keypoints candidates and some of them are not desirable. So, the next stage is used to eliminate keypoints that have low contrast (therefore sensitive to noise) or are badly localized alongside an edge. In the orientation assignment step, each keypoint is assigned one or more orientations based on local image gradient direction and thus achieving invariance to rotation. In the final stage, a descriptor vector is computed for each keypoint.

2.1.2 SURF

Speeded-Up Robust Features (SURF), presented by Herbert Bay *et al.* [BTG06], is similar to SIFT. However, SURF is faster than SIFT, thus enabling real-time applications such as tracking and object recognition. The algorithm is composed of three main steps: interest point detection, interest point description and feature matching. Like the SIFT method, the first two steps rely on scale-space representation and first and second order differential operators. These operators are speeded-up in SURF which leads to faster results.

2.1.3 FAST

Features from Accelerated Segment Test (FAST) is high-speed feature detector presented by Rosten and Drummond [RD06]. FAST is a corner detector algorithm based on the AST (Accelerated Segment Test) and its main contribution is speed. This is achieved by analyzing the fewest pixels possible, which reduces the detector's ability to average out noise. Thus, the results are not always as good as other algorithms. As SURF, FAST was also developed to work in real-time.

2.1.4 Discussion

Of the three algorithms, SIFT is the most robust and the slowest. SURF and FAST present not as good results as SIFT but they provide speed. SURF is more appropriate for real-time systems, so it will be used in our solution. However, the possibility of using SIFT or FAST will not be discarded completely.

2.2 Television Content Annotation and Applications

Television always involved a social component, as people like to comment what they watched with other people and share their experience [WUMCKB11]. Interactive television came to address this issue and to make the television experience more than a variety of uncontrollable streamed programs. The first generation of interactive television brought several interactive web-based models, that focused on the interests of content producers and could only be experienced in an individual way. The new generation focuses on the role of the user and to make interactive television a shared experience. The objective is to make possible for a user to communicate with others while watching TV, leave notes and comments, and share fragments of the content being watched [WUMCKB11].

In this chapter, several systems that implement this concept will be presented. The first system is WeOnTV [AAPN09], that allows viewers to watch the same channel together and to chat. Then, AMOPA [KKKRBHE11], which was designed for the purpose of annotating and organizing television content. Also, the EPG [OMAHT09], a system for creating virtual channels and making recommendations, using both the TV and the PC. Finally, the BluTV [SJBdLVL12], which is a platform for set-top boxes, where other widgets and applications can be installed on top of it.

2.2.1 WeOnTV

WeOnTV [AAPN09] is a social TV application that was developed on a staging platform of PT Inovação IPTV infrastructure. This application relies on the features of Instant Messaging (IM) and allows users to know what the other users are viewing, to recommend channels and to chat in multiple formats. It runs on a regular IPTV set-top box and offers a variety of social features, such as to personalize the content of messages and organize them by categories (family, sports, love, etc.), integration with a public IM service, multiple simultaneous IM sessions, privacy management and TV channel recommendation.

The application has two modes: "TV mode" and "Chat mode". In *TV mode* the primary aspect is the TV content but the user can still be aware of updates and alerts from the IM service. Messages received by other users are displayed at the bottom of the screen. These messages can include questions, to which the application offers predefined answers, and channel recommendations. When a user accepts the recommendation made by a friend, the viewed channel will automatically change to the recommended one. In this mode, the user can also know how many of his friends are on-line and which is the most viewed channel.

Figure 2.3: WeOnTV *TV mode* [AAPN09]

If a user accepts an invitation to "go to chat" from a friend, or if he chooses to, he will be redirected to *Chat mode*. Here, the TV content is secondary and the social features of WeOnTV take the leading part. The user is able to manage friends and conversations and to change his or hers status.

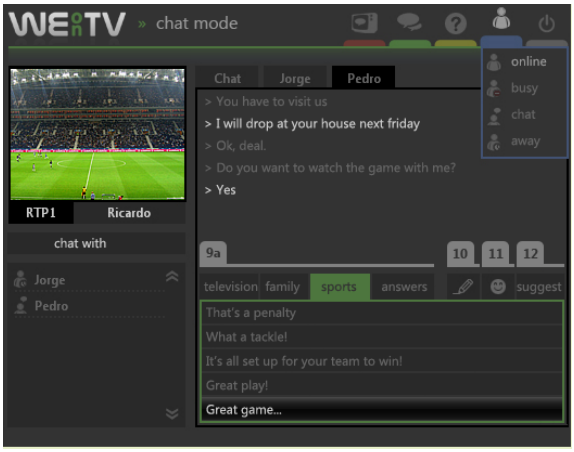

Figure 2.4: WeOnTV *Chat mode* [AAPN09]

2.2.2 Automated Moving Picture Annotator (AMOPA)

As part of the sachsMedia research initiative, that focuses on enterprises that produce TV content, the developers created a framework that facilitates the process of video production and distribution. As human annotation is very time-consuming and the existing computer-automated annotation lacks quality, this framework aims to find balance between both problems.

The framework AMOPA [KKKRBHE11] uses automatic content extraction and annotation to support editorial work and the construction and maintenance of archives. Automatic annotation is achieved by extracting and indexing predefined objects in the video files. Besides object and face recognition through conventional methods, they use

an OCR system and other techniques to find relevant text in the video materials. Speaker change recognition methods are also implemented in this framework. The feature extraction algorithms can be individually configured so that they can match the requirements of each user/client.

Mobile annotation was also considered by the authors as a way to make annotations on the user end [KKKRBHE11; KBE10]. Users' lack of time and concentration, as well as the difficulty in describing the content and using the adequate keywords, created the need to find a solution. The idea was to create a concept for a mobile application, for transferring and publishing videos that were annotated by the user itself. And thus the problem of meaningful tags arises. The solution was to divide tags by categories like "Who?","How?","Where?","What?" and "When?". Tests conducted by the authors revealed that users prefer tags based on emotions, meaning that the most used category was the "How?" category. Having videos tagged into different categories by the previous annotations of other users enables filtering and better search results in future searches and browsing.

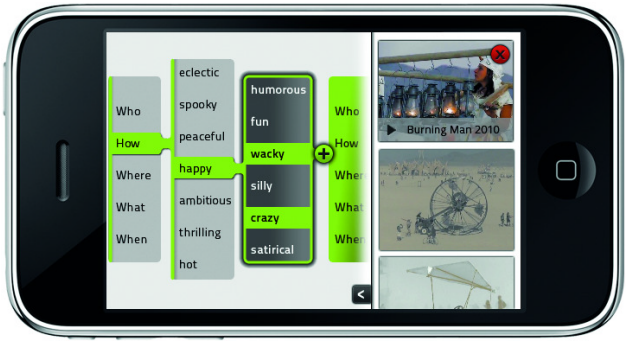

Figure 2.5: Selecting video clips using emotion tags [KKKRBHE11]

2.2.3 Electronic Program Guide (EPG)

The EPG [OMAHT09] concept allows users to browse and create media content with the TV and the PC. The content may come from different sources such as a connected external device, other devices with WIFI connectivity, on-line content and broadcasts. Besides the normal interactive TV services, the EPG offers social networking features for the TV environment, such as recommendation features. As the users can both access and share their content, or UGC (user generated content), the authors thought that adapting the interface to support such features would not work. So they assumed that the users would perform their settings using their computers.

To make the consumption of UGC on the TV possible, the user can create "virtual channels". These channels contain the results of a search made by the user. When in a web portal, the user performs a search for all the available content. Then a new channel will be created with the results (videos) of that search. These "virtual channels" can then be watched in a TV.

Figure 2.6: Virtual channel creation on the PC [OMAHT09]

The user is also able to share the TV content with friends. Users can publish a short extract of what they are watching and that extract is automatically published on a blog, where then other users can add it to their channels and watch it later on. A user can also rate content, and by doing so is recommending it to other users. When a video is highly recommended, a message will be automatically posted on the user's blog. This recommendation feature helps users to find content that they will be very likely interested on.

Figure 2.7: Recommendation on the EPG [OMAHT09]

2.2.4 BluTV

The BluTV [SJBdLVL12] is a platform that has been used to develop applications for the Brazilian Terrestrial Digital Television System. Building applications for Digital Television can be difficult for people that have interest on television systems but do not have the necessary knowledge and do not dominate the existing software tools to build those applications. The BluTV solves that problem by allowing the development of interactive television applications.

The BluTV was used for creating an interactive service guide that is composed by nine applications, fully functional and tested. The available applications cover different areas such as information and news, health and education.

In the viewer side, the BluTV also contains a set of tools, such as a player, for receiving

data and to process and present data information. The interactive service guide can be embedded in set-top boxes and thus reaching the viewers' homes. New applications can also be developed and added to the guide.

Figure 2.8: BluTVMediaPlayer in a set-top box [SJBdLVL12]

2.2.5 Discussion

Comparing the four systems, they serve different purposes but have some characteristics in common. WeOnTV and EPG, both have channel recommendation, although it is achieved in distinct ways. EPG allows for content recommendation, and then other users can watch the content later, while WeOnTV allows for a user to recommend content being broadcasted at the moment. On EPG, users can share content but that content can only be videos found in a previous search. AMOPA is not entirely television driven. Although its main purpose is to organize, search, and annotate television content, that can only be done on a PC. Of all four, AMOPA is the only one that uses object and face recognition techniques. The authors of AMOPA also address the problem of mobile annotation with the usage of tags, but only with the objective of organizing content and to encourage users to annotate. The BluTV is completely different from the other mentioned systems, but has an important role because it is a system made for set-top boxes and thus being relevant for this work.

Each system mentioned above has a characteristic or feature that is relevant to our proposed system. WeOnTV contributes with the possibility for several users to watch and chat about a current broadcasted program. AMOPA is useful for organizing and annotating large collections of video files. The EPG system allows for recommendation and sharing of content and also integrates those features with the web. Finally, the BluTV contributes with its integration on set-top boxes.

2.3 Video and Photo Annotation

As previously mentioned, an annotation is content that a user attaches to a document, video or picture, thus enhancing comprehension and interpretation of that content [GC-CGIGCP04]. Annotations are mostly used for organization and communication purposes [AN07]. Organization can be for both the user and others, and it also improves searching. Regarding communication, annotations can be useful for sharing content and to give context to others who are not familiar with the content [AN07]. The most common forms of annotation, for videos and photos, are free text, tags, semantic information, handwriting and audio.

Regarding video annotation, there is MPEG-7 [Mpe], which is a multimedia content description standard. MPEG-7 improves metadata management and allows for faster and more efficient searching. The description is separated from the video content, but they are related.

There are several systems for multimedia annotation, providing the user with a tool for creating annotations, searching and organizing content. This section will only focus on systems that provide these features, where the user makes their own annotations without collaborative annotation. Collaborative annotation will be addressed in the following section. The TKB system [Val11] focus on the capture and annotation of dance videos. In TKB, the annotation can be done in real-time, with a delay or added in post-production. The Family Video Archive [AGL03] is a system that provides the user an environment to manage and annotate his own videos. Finally, Caliph & Emir [Lux09] are two tools that allow for annotation and searching collections of photos.

2.3.1 TKB

The TKB (Transmedia Knowledge Base) project was envisioned to document, annotate and support creation of contemporary dance choreography [Val11; CVSAaFC11]. For that purpose, a tool that supports the capture and multi-modal real-time annotation of video was created.

This system supports the capture of two video streams from two different sources. It is possible to display both streams simultaneously or one at a time. It is possible to annotate on top of the video streams with different types of annotations.

In TKB, there are four different kinds of annotations - digital ink, text/hyperlinks, audio notes, and "marks" (bookmarks). The type of annotation that stands out from other annotation systems is the bookmark concept. Each "mark" is represented by an icon and it is associated with a keyword. After creating annotations, the user can then edit, view or delete them.

The user can annotate video in three different modes: *continuous*, *delayed* or *suspended*. On *continuous* mode, video capture is visualized synchronously with the live scene and the annotations will fade away with time. As the name suggests, in the *delayed* mode

what is being presented to the user has delay relatively to the live scene. Annotations will also fade with time, as in *continuous* mode. Finally, in *suspended* mode, the video capture will be suspended and annotations will be associated with the current frame.

Figure 2.9: TKB in *suspended* mode. On the left it is presented the current video capture. On the right is represented the selected frame for future annotation. [Val11]

This system also provides a timeline view that is composed by two main tracks, the video stream track and the annotation track. The annotation track splits into several tracks, each one corresponding to a different type of annotation.

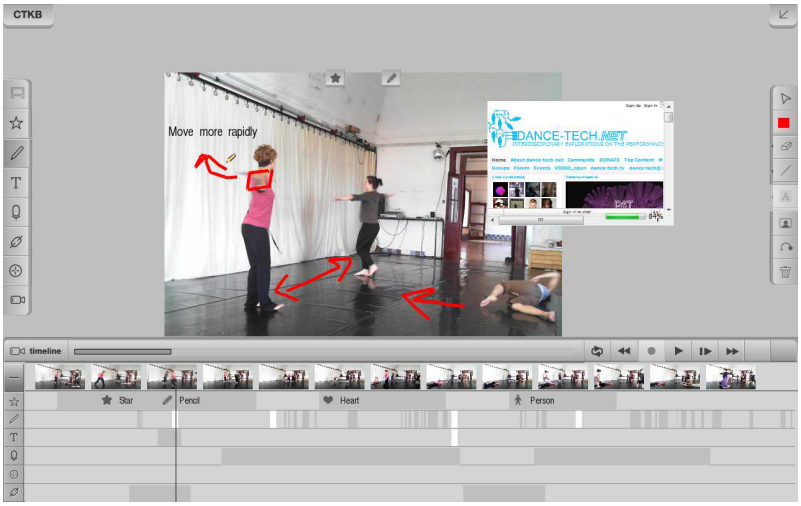

Figure 2.10: TKB: timeline view and the current video capture with different types of annotations [Val11].

2.3.2 The Family Video Archive

The Family Video Archive [AGL03] is a system for management, annotation and browsing large collections of home made videos. When a video is added to the system, it is

possible to watch it and control it like in a normal video player. There is also a timeline which is used to indicate the boundaries of a scene. A scene can be defined as a part of the video that is meaningful to the user and can be manipulated separately from the rest of the video. This system only supports the annotation of the current viewed scene. There are three different kinds of annotations - date, free text and metadata tags. The tags are created by the user and are organized into a hierarchy. The user is responsible for creating this hierarchy and can divide the tags into categories and subcategories. These tags can be used as search parameters.

Figure 2.11: The Family Video Archive annotation interface [AGL03]

This system also provides automatic annotation. When a scene is tagged with a tag from a certain category, all the tags from the level bellow that tag will be automatically associated with that scene too. Also when a user inserts a free text annotation, the system runs a matching algorithm over the text to provide a list of possible tags for that scene. The free text annotation can also be used to suggest new tags for the hierarchy.

2.3.3 Caliph & Emir

Caliph & Emir [Lux09] are two different tools resultant of the same project and they provide a user interface for MPEG-7 photo annotation and retrieval.

Caliph supports annotation of digital images and extraction of metadata and low level features (color layout, scalable color, edge histogram). The system has different types of annotations such as free text, structured textual content, shapes and image semantics. The structured textual content is divided into categories, such as *how*, *when* or *what*. All the semantic information about the image is represented through a graph. Each node represents an object, location, state, agent or concept, and each edge represents a relationship between those nodes. The user can add, delete and change the nodes and edges

when annotating an image. This information is stored using MPEG-7 XML.

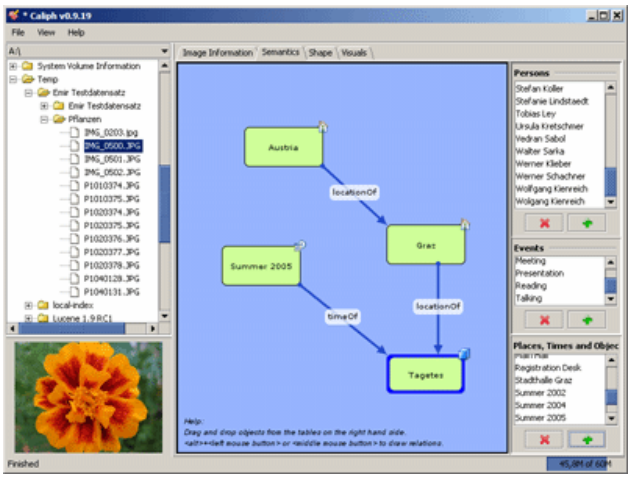

Figure 2.12: Caliph: editor for MPEG-7 semantic descriptors [Lux09]

Emir allows to search in MPEG-7 documents and focuses on the information created by Caliph. It is possible to search by keywords, retrieve semantic descriptors, and retrieve content-based images using low-level features.

2.3.4 Discussion

Although The Family Video Archive and Caliph have different targets, one is for video annotation and the other for photo annotation, they have some similarities. Although both support annotation by tags, in Caliph the tags are organized in a structured text description, with categories such as *how* or *when*. TKB also supports textual annotations, among others, but the main focus is the handwritten annotations. Of the three systems, TKB is the only one that allows real-time annotations.

In this section, only three systems were mentioned since video and photo annotation per se are not the main focus of this thesis. However, there are other systems that were taken into account, such as [GCCGIGCP04] and [DE06].

2.4 Collaborative Annotation and Proximity Interaction

As mentioned, annotation can help improve organization, memory and communication. The systems addressed in the previous chapter fully fulfill these intents except for communication. Although annotation by itself can help sharing and bringing context to others, the user still had to make their annotations in an individual way and could only share their work through other platforms.

A collaborative environment provides the capability to share and discuss information, which can lead to better results. Annotation can also be collaborative, where users annotate together, in real-time, on the same or different content. Hence, systems that provide such environment are of great interest to different communities such as educational,

media or scientific [SHK03]. In the following section, we will present CoVidA [ZWLS12], a collaborative interface for video annotations, Vannotea [SHK03], a system for real-time annotation and discussion of video content, and the CWaCTool [JGCP10], that allows users to share annotations synchronously in a P2P network.

In this chapter, besides collaborative annotation, it will also be discussed proximity interaction. In our daily lives, our actions are binded to how we interpret spatial relationships [BMG10]. However, this kind of interaction is still rarely used to manage our relationship with digital devices. Devices with knowledge of their surroundings, such as people, objects and other devices, can bring new forms of interaction and improve communication and sharing.

2.4.1 CWaCTool

The *Watch-and-Comment* (WaC) paradigm proposes to capture user-made multimodal annotations while they watch and comment a video. The results are interactive videos, that mashup the original videos with annotations. The tool CWaCTool [JGCP10] is an extension of that paradigm with the *Context Aware Peer-to-Peer Architecture* model, thus allowing users to annotate on a P2P network and share those annotations synchronously.

This tool allows context sharing, such as the user personal preferences, collaborative digital ink annotations, access to video streaming (from Youtube), audio notes, text notes, chat, P2P groups and integration with social networks. There are three possible types of annotations - digital ink, text and voice. Collaborative mode is not available for voice annotations.

The CWaCTool allows users to visualize and comment on Youtube videos. The user can both see their own comments and the comments made by users in the CWaCTool network. There is also integration with the social networks, namely with the Orkut network. This allows CWaCTool to suggest new contacts, prioritize suggestions from friends and recommend groups.

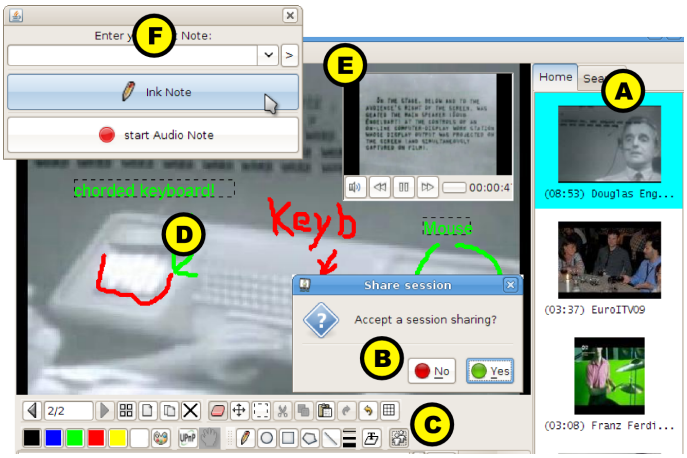

Figure 2.13: CWaCTool ink annotation mode. User can share videos (A), accept invitations (B) and make annotations (D) [JGCP10]

2.4.2 Vannotea

Vannotea [SHK03] is a system that enables real-time collaborative indexing, browsing, annotation and discussion of video content. Users can work collaboratively as a group to segment, browse and annotate a particular video of interest.

The system is divided into four components: a search and retrieval database, an annotation database, an application server and a MPEG-2 video streaming server. The search and retrieval database enables the search and browsing of video files and only stores objective and reliable metadata, provided by the video itself and by trained catalogers. The annotation database is used to store personal and subjective metadata, such as free text annotations made by users. The application server is responsible for coordinating events between clients using the same server-application. In this case, users in a collaborative group are connected to the same server and use an application called FilmEd to view and annotate their videos. In this system, at any given time there must be a client that was chosen to be the client-master. The server is responsible for replicating the events of the client-master through the rest of the clients, thus creating a collaborative environment.

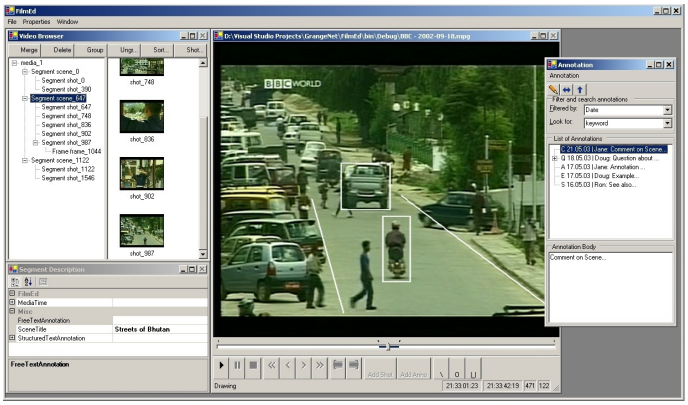

Figure 2.14: The *FilmEd* application is used by the system so the users can annotate, view and search videos [SHK03]

The annotation system allows annotations of segments, shots, frames and specified regions within frames. Users can make and share their own annotations and can browse for annotations made by other users. It is possible to see by who and when each annotation was made.

2.4.3 CoVidA

CoVidA [ZWLS12] is a pen-based collaborative interface for video annotations. This system allows users to annotate objects in images with keywords, and provides an environment for collaborative teamwork in only one device. It is possible for multiple persons to work on one video or on different videos. Manipulating videos and selecting annotations can be made through several predefined touch gestures.

Figure 2.15: CoVidA: collaborative video annotation [ZWLS12]

The users use a digital-pen for outlining objects and for writing annotation terms. The authors defined two types of annotations, *frame annotation* and *object annotation*. A *frame annotation* is associated with the whole frame and an *object annotation* is associated with the outline drawn by the user. The system can automatically distinguish between both types of annotation. Later on the user can edit or delete the annotations and can search them using handwriting. The annotation information is stored in XML and can be reused by other applications or on other videos.

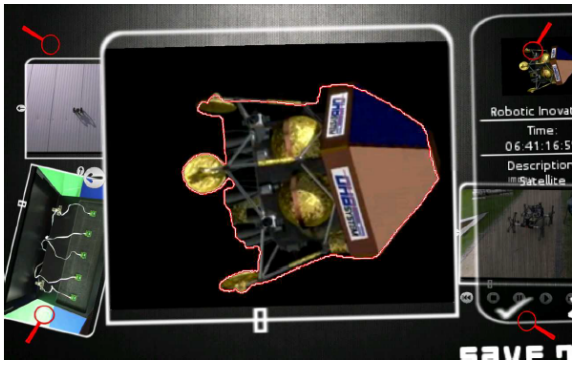

Figure 2.16: CoVidA: outlining the shape of an object with a pen [ZWLS12]

2.4.4 Proxemic Interaction

Proxemics is the way we interact with the world by interpreting spatial relationships [BMG10; Mar11; GMBDMW11]. Devices with proxemic interaction are supposed to have knowledge of nearby people and other devices, and to accomplish that they need to be aware of their *position*, *identity*, *movement* and *orientation*.

Position can be described in two ways: absolute or relative. For the absolute position we have to have a fixed point in space. On the other hand, relative position can be defined as the spacial relationship between two entities that do not necessarily have to be fixed, so there is no fixed point. *Orientation* is simply the direction that an entity is facing. *Movement* is the change of position and orientation over time. The *identity* property it is the ability to differentiate entities in the space.

For an interactive system with interaction like this, there is a need to distinguish what is fixed (walls, windows, chairs) and what is not (people). So the authors define two kinds of features: *fixed features* and *semifixed features* [BMG10]. *Fixed features* are fixed aspects like walls and *semifixed features* include objects that are normally fixed but that can change over time, like chairs and tables.

There are different features that can be added to this concept of proxemic interaction. The authors discuss the use of *mobile tokens* [BMG10], which are objects that explicitly interact with the system, like a mobile phone being used as a pointer. Other valuable feature is recognizing different people, which brings us back to the concept of *identity*. Thus, it is possible to have notion of history, which allows a user to continue activities that were previously started by himself, and also personalization and security.

The *proxemic media player application* is the example that is used to demonstrate this kind of interaction. There are different kind of interactions. First, when someone enters the room at position (a') the display is activated. Then when the person moves closer (b') to the display the video previews will continuously shrink. When the person is finally very close to the surface (c') they can select a video directly by touching the surface. When the person moves away to position (d'), the selected video track starts playing.

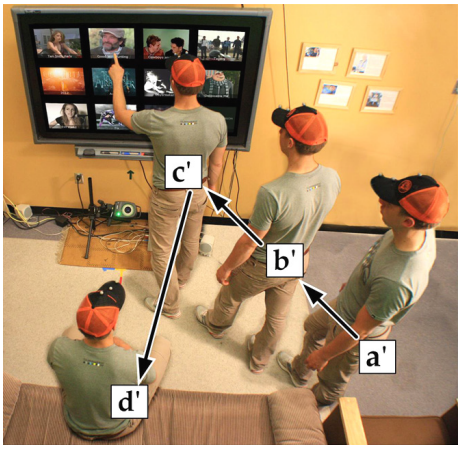

Figure 2.17: Proxemic Media Player: different responses depending on the user position [BMG10]

2.4.5 Discussion

The CWaCTool and Vannotea provide the users a collaborative environment where they can annotate videos and share those annotations synchronously. Both systems enable real-time annotation, browsing and discussion of video content.

Vannotea stores different types of annotations in different databases. The system has one database for storing video files and reliable metadata, and another database for storing annotations. Thus, this concept of storage can be relevant to our system since video files and annotations will be stored separately.

Of all presented systems, CWaCTool is the most similar to our solution. Besides providing real-time annotation and sharing on a collaborative environment, it also has the concept of groups and integration with social networks. The CWaCTool also offers digital ink and text annotations. However, in this system users annotate on a P2P network and can only access the system through a PC. CWaCTool focuses on video streaming, coming from sources like Youtube, contrarily to our system which focuses on television content and user-generated content.

CoVidA is very useful for a work environment providing collaboration on the same device. However, the users have to be at the same location of the device, which is not desirable for our system. Nevertheless, CoVidA's difference between frame annotation and object annotation can have some relevance to this work.

The proxemics interaction is also significant to our system, since it explores different possibilities of interaction between the smartphone and the television, and thus could be implemented in a future work.

3

Television Content Annotation

The study of related systems and techniques, such as those presented in chapter two, provided a better understanding of this particular field and the motivation behind it. Also, new skills and knowledge were obtained and new ideas emerged, thus creating means for developing the proposed system. This chapter presents the concepts and technologies used for the development of the system, as well as the main requirements.

The first section of this chapter focuses on the context of the problem. To solve this problem the system has to fulfill several requirements, which is the subject of the next section. Finally, all the necessary procedures to achieve such requirements will be exposed in the last sections. These procedures will include the technologies used for developing the solution and both the system's functionalities and a description of the interface.

3.1 Concept

The goal of this project is to create tools that allow users to annotate television content in a collaborative way. The proposed system provides support for annotation of video content by multiple users at the same time, re-watching both the user annotations as well as other users' annotations, and sharing and searching content.

Providing means for annotation of television content gives users a more complete experience and enriches the content itself. The TV experience can be enhanced by allowing the user to annotate together with their friends and family, or co-workers, in the same room. Each one of them can add content to the video currently playing and watch not only what they add but what others add as well. The user can also share the annotations through social networks, thus allowing to share content with other people on the web. Imagine, for example, a group a friends watching the same movie together and

adding funny comments to a specific part of the movie and then publish that content on Facebook or Tumblr. The system also provides a way for users to add more meaningful annotations. For example, a football coach who watches the games of his team or games of his opponents and takes notes, which can later be used to improve his team's tactics.

As the aim of our solution is to create a system that users can use to annotate television content, our system will be divided into two main parts: the video player on the TV side and a mobile application, for users to annotate. For the annotation, we chose a smartphone, a device that nowadays most people possess and that provides a more natural way for users to draw and add content to videos. Using a smartphone also gives more independence to the users, as each user can annotate independently from the others and walk freely around the room.

This system is different from other presented systems, because it provides users a platform for annotating television content while watching it on a TV. Unlike other systems, users fully annotate using a smartphone together with other users on the same room. It also integrates social features, such as sharing annotations on social networks.

3.2 Requirements

The requirements are defined by features and conditions imposed by users and the project team, that must be fulfilled so the solution may be adequate and closer to what is desired. Some features and conditions were defined at the beginning of the project, others were adapted and added throughout the development according to circumstances found along the way. In this chapter, two types of requirements will be presented: design and functional. Design requirements are related to certain characteristics and rules that the interface must oblige. Functional requirements are the operations and features that the solution must implement.

3.2.1 Design Requirements

The interface must be user-friendly and respect the principles of usability. The system is to be used by any kind of user, so having an interface that is easy to learn, well organized and can easily be recalled is of extreme importance. The design of the interface was developed considering two different levels of human-computer interaction: Hardware (physical interfaces - smartphone and TV) and Software (virtual interface). These types of physical interfaces promote the use of digital pens, touch and even motion triggered features.

The main requirement for the mobile application interface is to be designed specifically for smartphones, namely those with the Android platform, that possess a larger screen and promote the use of a digital pen. The size, spacing and other features of each element are according to the design principles for Android applications. The features must be divided into different sections that can be easily accessed through the mobile

application. There must a section where the user can annotate, another section where the annotations can be browsed and replayed and yet another section for searching content. The drawing environment and its tools must follow the standard of other drawing interfaces and promote the use of a digital pen.

For the TV interface all the elements must be designed to fit large screens and also to adapt to different screen sizes. As the main feature of the TV interface is to play videos, it is to important to have an interface that follows the standard of video players. Being the video the most important element on the screen, the video must occupy the majority of the TV interface area. The interface must also contain an area for the player controls, the video timeline and another area for displaying the image annotations.

3.2.2 Functional Requirements

The system's main requirement is the ability to annotate video collaboratively. Collaborative annotation is a feature that poses different challenges. Firstly, the system must allow collaborative but independent annotation, i.e., not sharing the same device. Since the users will be sharing the same room, such as the living room in their houses that do not have support for big devices (figure 2.15), or even be in different locations apart from each other, it is important that they can work independently from one another. Secondly, it is necessary to coordinate events, giving the users the feeling that they are working together.

As for types of annotations, they must be useful, easy to learn and take advantage of the mobile platform. Also, annotations must be stored apart from videos and apart from other users' annotations, so the original content is not modified, less storage space is used and each user can re-watch his own annotations or others' separately.

As mentioned before, the system must also provide the users the possibility to re-watch their annotations in an easy and straightforward way. When replaying annotations, being able to access them easily or access specific moments is also of importance.

Finally, since the system aims to provide the users a social TV experience, there must be some integration with social networks, offering the users the opportunity to share annotations with friends, family and the rest of the world.

3.3 Solution

The tools created for this solution are designed to support collaborative annotation of television content, browsing annotations and searching related content. In order to meet the requirements mentioned in the previous section, a graphical interface for the TV, a server to receive the clients' requests and a mobile application for annotating, will be created. Together they allow for video annotation in a collaborative way, giving the user a new and social TV experience. This section presents our solution in response to the

requirements, describing the technologies used and the system's interface and functionalities.

3.3.1 Technologies

The software technologies used for developing the solution were the openFrameworks platform for developing the player and related functionalities, the OpenCV library for image processing, the POCO library for creating the server, the Android Development Tools and the Android SDK for creating the mobile application, and the S Pen SDK for creating the drawing environment on the mobile application side.

The openFrameworks platform is an open source C++ toolkit designed to aid developers in the creative process by providing a simple framework. This framework integrates different libraries such as OpenGL, QuickTime and OpenCV, resulting on a powerful tool for accessing and manipulating audio, video, image and text.

OpenCV is an open source library for computer vision and machine learning. This library provides several algorithms for recognizing faces, identifying objects, processing images, tracking objects and much more.

POCO is an open source C++ library for building network-based applications and it is included in the openFrameworks package.

The Android SDK includes several tools and API libraries necessary for developing applications for Android. These tools together with the Android Development Tools (ADT), a plugin for the Eclipse IDE that extends the IDE's capacities, provide an integrated environment for creating Android applications.

The S Pen SDK, available by Samsung, provides a set of tools for developing applications that take use of the S Pen (input tool similar to conventional pens but more advanced, supplied with the Galaxy Note series devices) features, such as pen pressure and hover events. It also provides additional features such as a drawing environment with pen, eraser and text property settings. The version used for the drawing environment of our mobile application is the 2.3, released in February 2013.

The hardware technology for developing and testing the mobile application is a Samsung Galaxy Note. The specifications for this smartphone are a 1.4GHz Dual Core processor, 16GB of internal memory, a 5.3 WXGA (1280 x 800) screen, USB 2.0, Wi-Fi connectivity, a standard battery Li-on 2500 mAh, S Pen features and the Android 4.1 (Jelly Bean) platform. For testing the player it was used a Samsung ME46A LED display with a 1920 x 1080 resolution and a 46" Touch Overlay.

The annotations are stored in the XML format without resorting to a database. All videos, photos, images, fonts, annotations and other configurations are stored in disk, on the server side.

3.3.2 Interface

As mentioned before, the solution is divided into two main parts and for each one of them an interface was developed. In this section, the two interfaces will be presented: (1) the player interface for the TV and (2) the interface for the mobile application.

3.3.2.1 Mobile Application Interface

To solve the problem of each user adding their own content independently, a mobile application was created. The mobile application interface was built for smartphones with the Android platform and provides the means for controlling the system. The users can choose what to watch and annotate through the mobile application.

The application is divided into three main sections, as seen in figure 3.1 (from top to bottom):

1. **Play Video and Annotate**: in this section the user is able to create a new session, after choosing a video to annotate, or join an ongoing session, thus annotating collaboratively with another user. The current video being played and the annotations will be displayed on the TV interface.
2. **Playback Video and Annotations**: here the user can browse through previous sessions and re-watch them on the TV player. The chosen video will be played from the beginning and the annotations previously added will also be displayed at the appropriate moment.
3. **Search Videos**: this section corresponds to a different kind of search from the one users are used to. Instead of searching videos with keywords, the user can search videos using their own photos. The search will return videos that are related to the selected picture or photo.

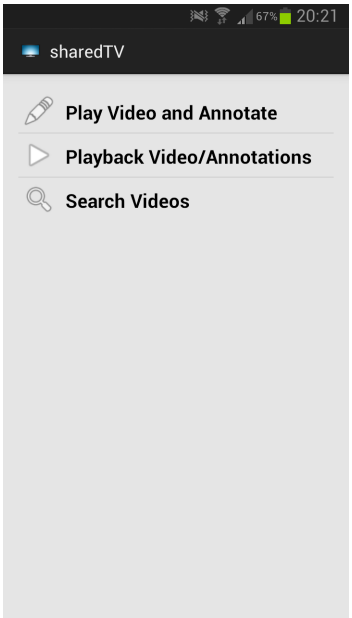

Figure 3.1: Mobile application: main menu

In both section 1 and section 2, the user begins by choosing a video. The available videos are shown as in figure 3.2, in a single choice list.

In the first section (1), after selecting a video the application will lead the user to a drawing environment where is possible to annotate content. The drawing environment consists of a canvas and a set of tools, as shown in figure 3.4. On the left side of the canvas are the tools for drawing purposes which, from top to bottom, are the following: brush/pen, eraser, text, color picker, clear, undo, redo and save. Clicking twice on one of the first four tools (brush, eraser, text, color picker) will open the settings window corresponding to the selected tool. The *clear* button cleans the canvas completely, the *undo* button undoes the last canvas' action and the *redo* button redoes the last action. The *save* button will save all annotations. The application's *Action Bar* [1] contains additional options such as adding attachments, sharing annotations and getting frames from the current video playing.

In the second section (2), after selecting a video the user can browse through the previous sessions and select one for re-watching. The sessions are presented as shown in figure 3.3 and each session has a name, a date and the user to whom the annotations belong. Sessions can be filtered using the top right button. It is possible to see all annotations or only the annotations belonging to the current user.

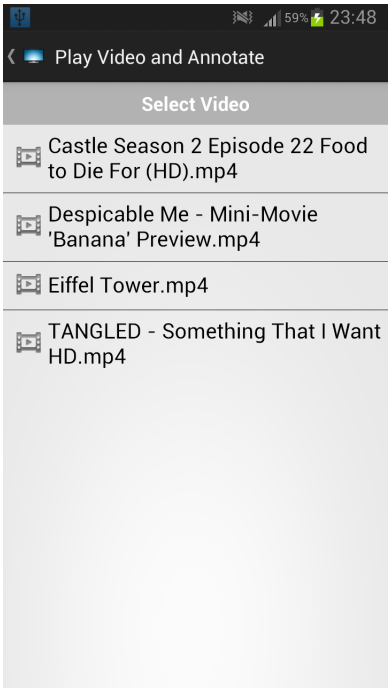

Figure 3.2: Mobile application: selecting a video to annotate

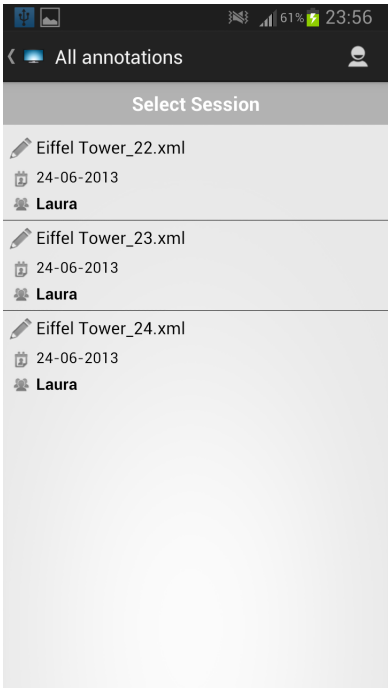

Figure 3.3: Mobile application: selecting a session for playback

[1] See http://developer.android.com/design/patterns/actionbar.html

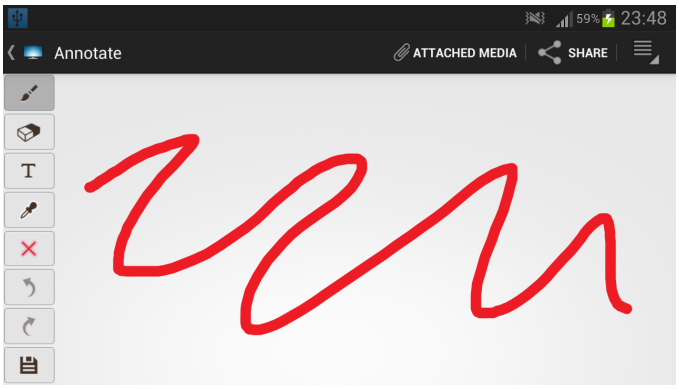

Figure 3.4: Mobile application: drawing environment when annotating a video

Finally, in the third section (3) the user can search videos using photos or pictures from the phone's *Gallery*. As shown in figure 3.5, this section contains a button for adding a new picture and an area for previewing the select picture. There are two possible types of search: search all videos or search only the select ones. When the last option is chosen a dialog will appear so the user can choose the videos. As this type of search is unconventional, a help button was also added (top right), with a brief explanation of what is expected of this kind of search. The results of the search will be displayed as a simple single choice list similar to the one on figure 3.2.

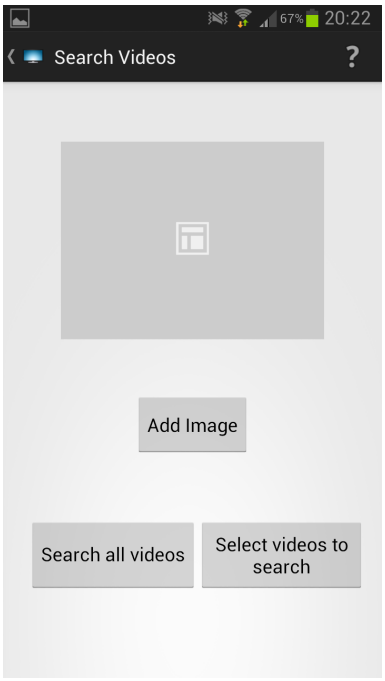

Figure 3.5: Mobile application: search videos

3.3.2.2 Player Interface

The player interface is designed for the TV and it is responsible for displaying the current video playing, the annotations being added at the moment, a timeline and the control buttons. This interface changes according to the current mode being used. The system has two different modes for the player: (1) Play mode and (2) Playback mode.

Figure 3.6: Player in *Playback mode*

In *Play Mode*, the majority of the interface's area is occupied by the video display. Regardless of the screen size, the area assigned for displaying the video is always 80% of the total width and 75% of the total height. Below the video display there is a timeline, which tracks the progress of the video. The timeline is clickable, allowing the user to easily jump from one position of the video to another. For controlling the video, as in any other video player, there are control buttons, located at the bottom of the interface. It is possible to pause, play, fast forward, rewind and stop the video. If the video is stopped, clicking *play* will play the video from the beginning. The annotations will be displayed in the same area as the video, scaled to appear in approximately the same place where the user drew them before. However, this only applies for ink and text annotations. Image annotations will be displayed in a different place, as they will appear to the right of the video. Each image annotation has exactly the same size, 100 pixels for width and 100 pixels for height, and it is followed by a caption that describes which user has sent that specific image.

In *Playback Mode*, the interface is almost identical to the one mentioned above, with the exception of the timeline and a set of labels. In this mode, the timeline will contain several markers that indicate the locations of the annotations. Each marker is placed at a specific

time in the timeline. That time corresponds to the time at which the user has added an annotation. The color of the marker changes according to the type of annotation: red for ink, blue for text and green for images. The labels for each color will be indicated at the bottom right of the interface, as shown in figure 3.13.

3.3.3 Functionalities

In response to the requirements, the solution must have a set of functionalities that satisfy those requirements and solve the problems that were described in previous sections. This chapter depicts the main functionalities of the system, such as types of annotation, sharing content through social networks and searching content.

3.3.3.1 Types of Annotation

Taking advantage of the drawing environment of the mobile application, the digital pen and other content the user may possess in his phone, three types of annotations were created: digital ink, text and image. The following section presents the three types of annotation, explaining the concept behind them and their ways of insertion.

Digital Ink: A digital ink annotation is a path of ink with time and space dimensions. The digital ink can be used for drawing or writing on the video, using an input pen or simply the user's finger. The digital ink annotation is the quickest to insert and it is where the user can be more creative. To add an ink annotation the user must choose the *pen* button on the tool bar and then start drawing on the canvas. Each ink annotation has a timestamp that will correspond to specific a time on the video. The moment the user releases the pen or finger and an ink annotation is added, meaning when the ink path is completed, a timestamp is also added. The timestamp corresponds to real time and only when the annotations are saved (pressing the *save* button), will the timestamp be converted to the actual video time. Thus, each ink annotation will be associated to a specific moment in the video.

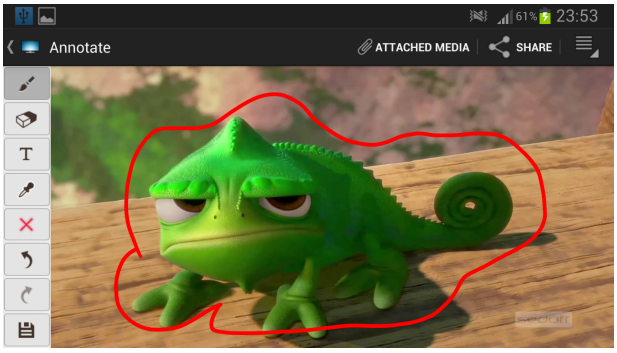

Figure 3.7: Mobile application: ink annotation

Text: Text annotations can be useful for adding more perceptible content or longer and

detailed comments. Inserting annotations in the form of text is advantageous as it provides better readability for users, either in *Play* mode or for future reading in *Playback* mode. To add a text annotation the user must press the *text* button on the tool bar and then press the desired location on the canvas. After tapping the canvas, a virtual keyboard and a text area will appear at the same location the user touched before. The keyboard also has handwriting recognition, which allows the user to use the digital pen. When the user has finished writing the annotation, touching anywhere outside the text area will complete the annotation. Text annotations also have timestamps, that are added after the user has completed the annotation and converted to video time when the *save* button is pressed. When a text annotation is pressed on, the text area, the keyboard and a delete button will appear, allowing the user to edit or remove the annotation.

Figure 3.8: Mobile application: text annotation

Image/Photo: Nowadays, smartphones come bearing a camera and it is common for people to possess several photos in their phones or even images they have downloaded from the web. Video content can be enriched by adding other types of content like photographs or some other kind of image. Ergo, it is appropriate to have images as a form of annotation, not only for enriching the content but also for sharing photos with other users. To add an image annotation the user must press the *attached media* button in the *action bar*. A pop-up window will appear and the user can add one or more images by pressing the *add image* button. It is also possible to delete more than one image by selecting them from the list and pressing the *delete* button. After closing the pop-up window, the images will be added as annotations. Similarly to the other annotations, image annotations also have timestamps that are added when the *save* button is pressed.

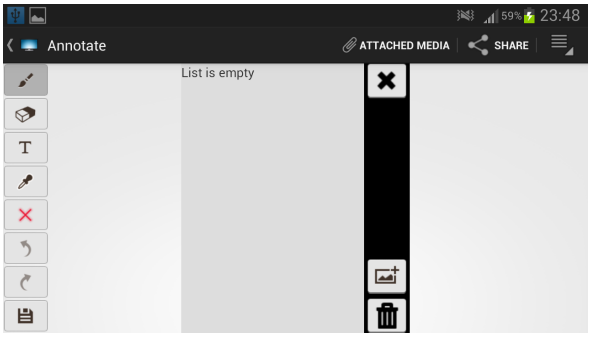

Figure 3.9: Mobile application: dialog to add new images

3.3.3.2 Video Annotation (*Play Mode*)

The main feature of the system is the possibility to annotate videos. The mobile application was designed to support that function and to allow users to annotate independently from one another.

Previously in the interface chapter, we could see that the mobile interface possesses a drawing environment, which is used for annotation purposes. Using the mobile application, the user can annotate videos by choosing the *Play Video and Annotate* option in the main menu. After selecting a video to annotate and after the video starts playing on the player, the drawing environment will appear. The tool bar has different types of annotation for the user to choose from. When the user has finished some annotation, pressing the *save* button will save the annotations and thereafter the annotations will appear on the player, as shown in figure 3.10. Each annotation will appear only for a limited time. After a certain time the annotation will start to fade out until it completely disappears.

If more than one user is annotating, all annotations from all users will appear on the player. Text and digital ink annotations will appear as they were created, with no different colors or other detail to differentiate one user from the others. However, image annotations will have a caption indicating the user that has added it.

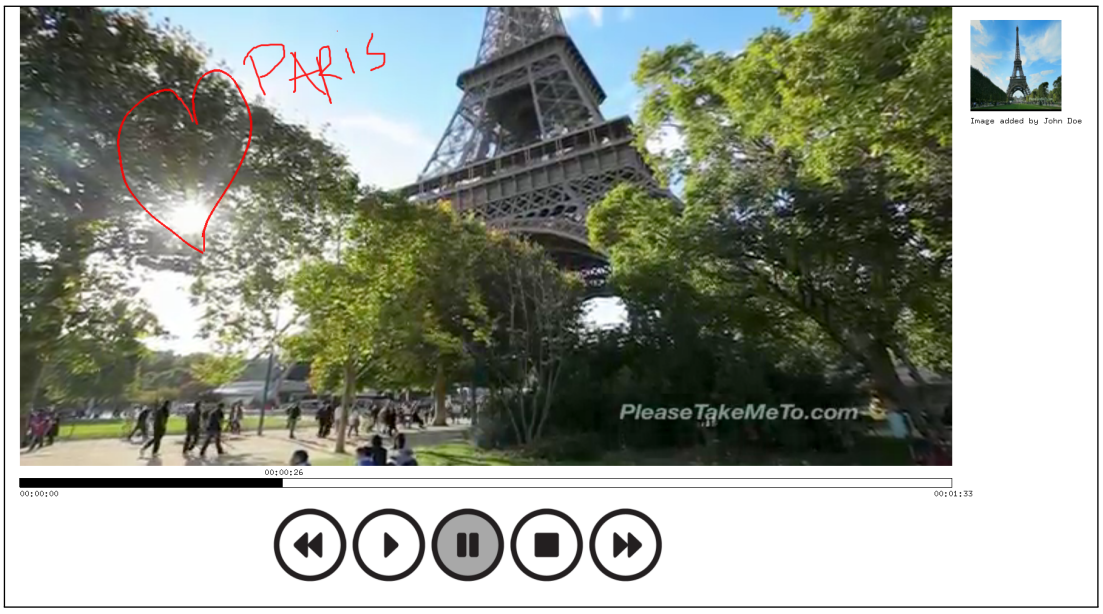

Figure 3.10: Player: ink and image annotations being displayed on the player interface

In the drawing environment, the canvas for adding ink and text annotations is completely white. Having a canvas that is completely white may bring some difficulties for users, such as not knowing exactly where to draw. To solve this problem two options were added: getting the current frame of the video being played or getting frames continuously, emulating a video stream. The user can select either option by pressing the *overflow* button, located at *Action Bar*'s far right, as shown in figure 3.11. When the *Get*

Current Frame option is selected, the frame being currently displayed in the player will be used to fill in the canvas. Only one frame is sent and now the user knows exactly where to draw. Selecting *Get Frames* option will continuously fill the canvas with frames, thus simulating a streaming event. However, this last option is not very usable as it is not fluid, not allowing the user to draw properly. These features can also have other purposes, that will be covered later on, in chapter 3.3.3.5.

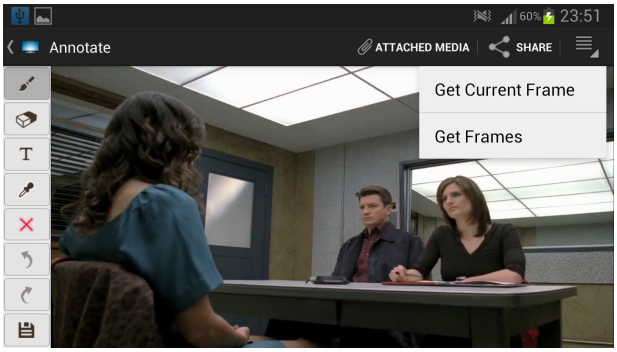

Figure 3.11: Mobile application: canvas filled with the current frame

3.3.3.3 Collaborative Annotation

This solution aims for annotation of television content, while involving the new concept of social TV. Therefore, the system also provides collaborative annotation, where users can annotate together in the same room while watching the same content.

If a user is already annotating a video, another user may want to join and annotate it too. While using the mobile application, after selecting *Play Video and Annotate* in the main menu, a dialog will appear if another user has a ongoing session. The dialog offers the user a choice, to join or not the ongoing session. If the user chooses to join, the drawing environment will appear and both users will be annotating the same video.

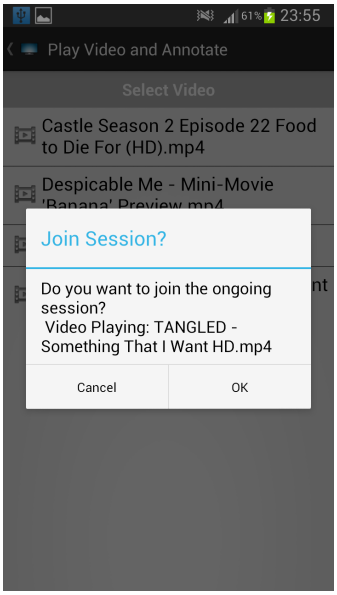

Figure 3.12: Mobile application: dialog for joining current session

As mentioned in the previous section, all annotations from all users will appear on the player and there will be no distinction between them, besides the image annotations that have a caption. So, for example, user A will not know explicitly if an ink annotation belongs to user B or user C.

Although it seems that both users are sharing the same session, the annotations are saved separately. Later, the user can replay his own annotations or annotations belonging to other users, which will be the topic of the next section.

3.3.3.4 Replaying Annotations (*Playback Mode*)

A user may want to replay a session either to review his own annotations or to watch annotations from other users. It is possible to replay a session by choosing the *Playback Video/Annotations* option, in the mobile application's main menu. The annotations are organized first by video and then by user/session. After selecting a video, a list containing all the annotations for that video will appear. Each entry on the list is a session belonging to some user. A session is defined by a video, a session number, a date and a user. After choosing a session, the user can watch the replay on the player.

As mentioned before in section 3.3.2.2, the player changes in *Playback Mode*. Each annotation is properly marked on the timeline and each marker has a color according to the type of annotation, as shown in 3.13. There are three different types of markers for each different type of annotation: red for digital ink, blue for text and green for images. Having markers with this color system allows for an easier navigation, as the user can skip to where annotations exist or to where certain types of annotations are. The annotations will appear at the time they were created and will only appear for a certain amount of time, as in *Play Mode*.

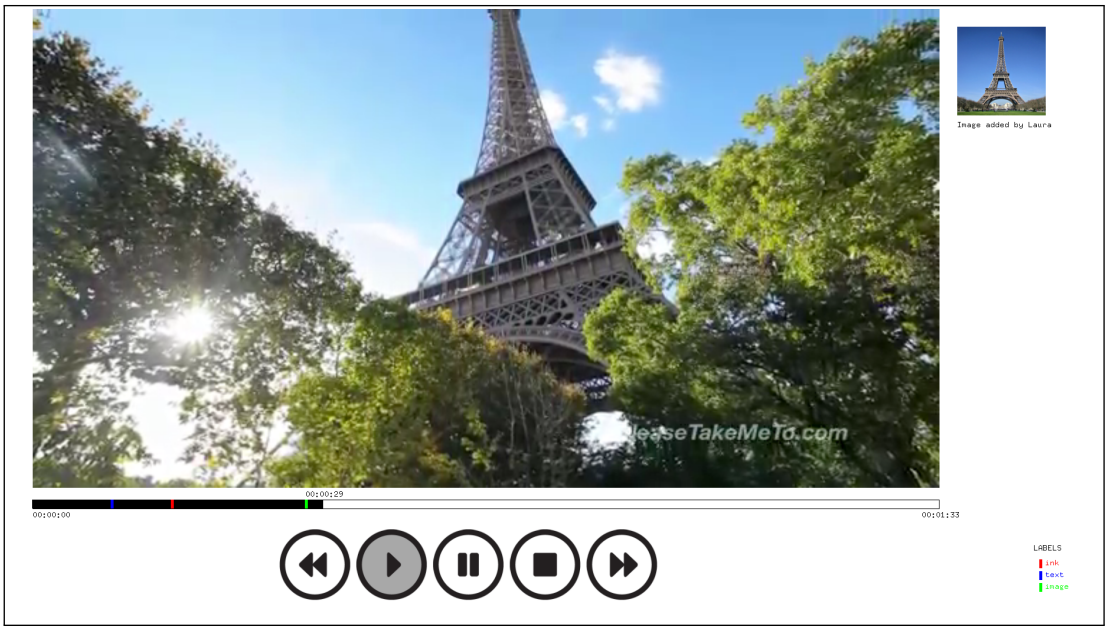

Figure 3.13: Player in *Playback Mode*

3.3.3.5 Sharing and Social Networks

Taking advantage of the mobile application, the system also has a sharing function and integration with social networks. The user can share an annotation through a social network or even send it by email. As mentioned in section 4.3, the *Get Current Frame* option in the drawing environment could have other purposes besides improving drawing accuracy. When a user asks for a frame and annotates over it, the *share* button becomes active. Pressing the *share* button, on the *Action Bar*, pops up a dialog for the user to choose an application that supports sharing a picture. All the applications that support sharing or sending a picture and that are currently installed on the user's phone, will appear as a possible option. For example, if the user has installed Facebook, Google+ or Dropbox, they can share or save that picture on that platform. When the *share* button is used, the annotation will also be saved on the phone's gallery, for future use.

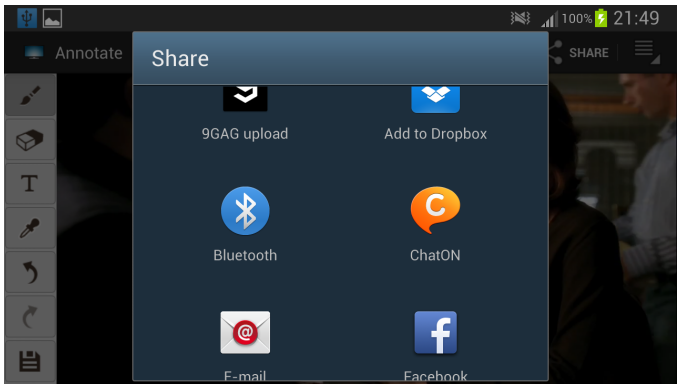

Figure 3.14: Mobile Application: share dialog window, after pressing the *share* button

3.3.3.6 Related Content Search

As mentioned earlier in this section, users usually have photos or other pictures in their phones and in this system they are used as a form of annotation, thus establishing a connection with the video content. Taking advantage of this connection, our system provides a feature that allows users to search videos using images that may or may not be related to that content. For example, if a user has photos of a trip to Paris they may want to watch movies that take place in Paris, or may want to annotate a video of Paris, leaving photos and notes for his friends, or for other users to watch, that may be interested in visiting Paris as well.

In the mobile application main menu, clicking in the *Search Videos* option will lead the user to the search section. First, the user must choose a picture by clicking the *add image* button. After selecting an image, the image will be displayed on the preview area, otherwise the preview area will remain "blank", as shown in figure 3.5. Only when an image is chosen can the user proceed to search. The system provides two types of search: search in all videos or search only in the selected videos. In the first type of search, the image is matched with all videos, *i.e.*, the image is compared with all videos, one by one,

and if a video is related to the image, that video is added as a search result. In the second type of search the process is identical, with the exception of the number of videos that are searched for. Only the videos that were selected by the user will be used on the search and compared with the image. After clicking in the corresponding search button, a dialog will appear, informing that the search is in progress and that the user must wait. When the search is completed a list is presented, containing the search results or empty if there are no results.

By clicking on a video from the results' list the user can watch the selected video. The video will be played on the player and the timeline will contain special markers. The markers indicate the frames, or moments, that are directly related to the picture that was used for the search. For example, if the user chose a photo from the Eiffel Tower and the tower appears in the movie, the moments where the Eiffel Tower appears will be marked with markers. Thus, the user can easily jump to moments that may be more interesting to him.

4

Prototype Development

The purpose of this chapter is to specify the functionalities of the system in a more technical and detailed way. In addition, this chapter also describes the system's architecture and its components.

4.1 Architecture

The system has a client-server architecture, illustrated in figure 4.1. As mentioned in chapter 3, the system is divided into three parts: the mobile application, the server and the player for the TV.

The server was created using the POCO library and it is a web server, which receives requests from clients that are using the mobile application. Web servers communicate using the HTTP protocol and are responsible for mapping the path component of a URL into files and programs. Therefore, when clients request for a certain URL, they will be given the content that was previously mapped to that address. In this system the URL's are used to send information back to the client, to store files in the server or to initiate some action. For example, the URL `www.example.com/get_videos` will return a list of the available videos stored in the server. To each URL, a handler is assigned to deal with the request and the server can have as many handlers as URL's. The POCO server has a special architecture that revolves around handlers. As illustrated in figure 4.2, the server has a Handler Factory that is responsible for assigning to each URL a handler, so each action or request can be treated separately in their respective handler. The web server is also responsible for saving content so all annotations, pictures, videos and configuration files are stored in the server side.

The mobile application was created for the Android platform, namely for devices with the *Jelly Bean* version. The drawing environment, which includes the canvas and the tool bar functions, was created using the S Pen SDK library. The mobile application is the client in this architecture and it communicates with the server through HTTP requests, which are usually GET or POST requests and in most cases the information sent between client and server is in the XML format. Only images and frames are sent in the form of bytes.

The player was created using the openFrameworks platform and it is located on the server side. The server and the player are connected, *i.e.*, in this version of the system the player is integrated with the server instead of being two completely independent modules that communicate as the server communicates with the mobile application. When the client sends a request for a specific action, such as "play video *X*", the server will then call a specific function on the player side, that corresponds to that action.

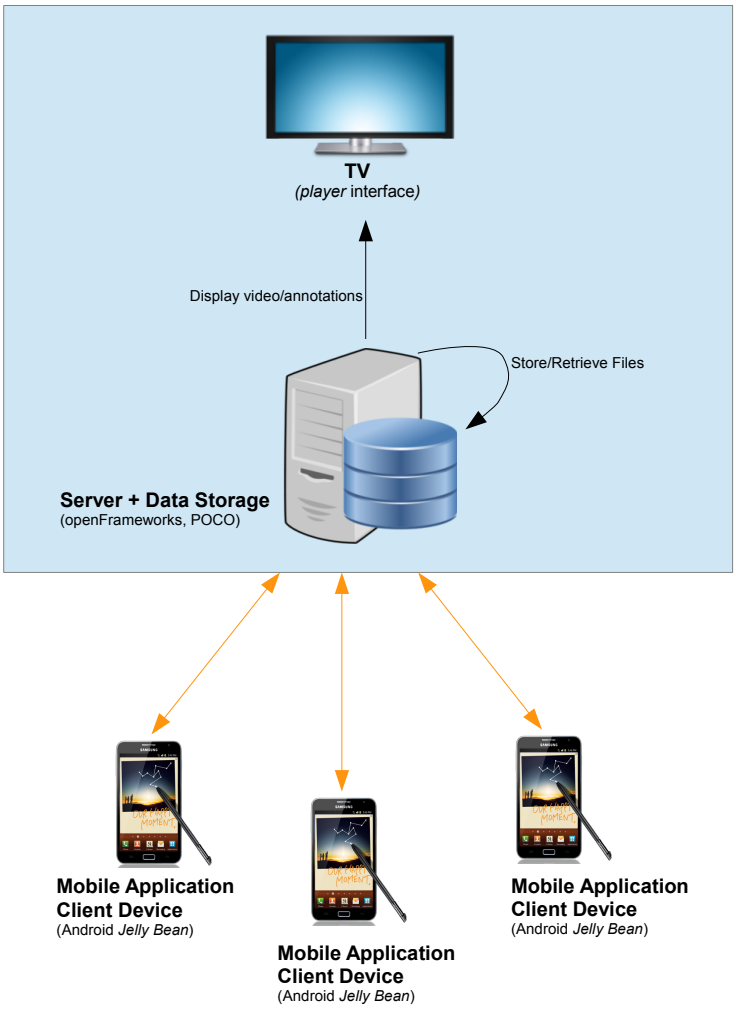

Figure 4.1: System architecture

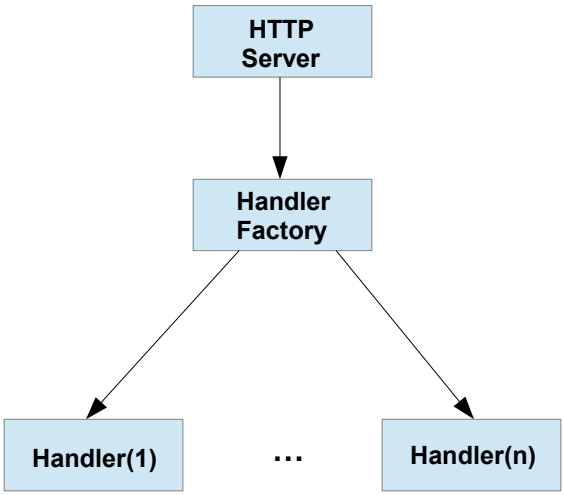

Figure 4.2: Server Architecture

An Android application is divided into five main components, which are: activity, intent, service, broadcast receiver and content provider. Each component is a different way for the system to access an application and each one has a different role and a distinct lifecyle.

An *activity* is a component that represents a single screen with a user interface and each activity is independent of the others. An application usually has several activities. A *service* is a component for running operations in the background and does not possess an interface. If the user switches to another activity or application, the service will continue to run. A *content provider* is responsible for managing a shared set of application data and provides a uniform interface to access data. Through the *content provider*, other applications can also access and modify the data. The *broadcast receiver* component responds to system-wide broadcast announcements. The broadcast can originate from the system or from applications, with announcements informing, for example, that the battery is low or that a picture finished downloading. The *intent* is not considered an application component, according to the Android development guide, but plays a fundamental role in the communication between other components. An *intent* is an asynchronous message that activates a component. Three of the components can be activated using intents: activities, services and broadcast receivers. For activities and services, an intent defines an action to perform, whereas for broadcast receivers the intent is the announcement to be broadcasted.

The openFrameworks platform has several layers of libraries and code, as shown in figure 4.3. The program starts at `main.cpp`, which is responsible for creating a new instance of the class *testApp*. The *testApp* class, named by convention, inherits the properties of *ofBaseApp*, which is the main class of openFrameworks and contains various event driven functions. The `testApp.cpp` is the file that contains the application's code. This

code can resort to a set of classes, built on top of the base libraries, that are available for manipulating video, audio, image, graphics, and others.

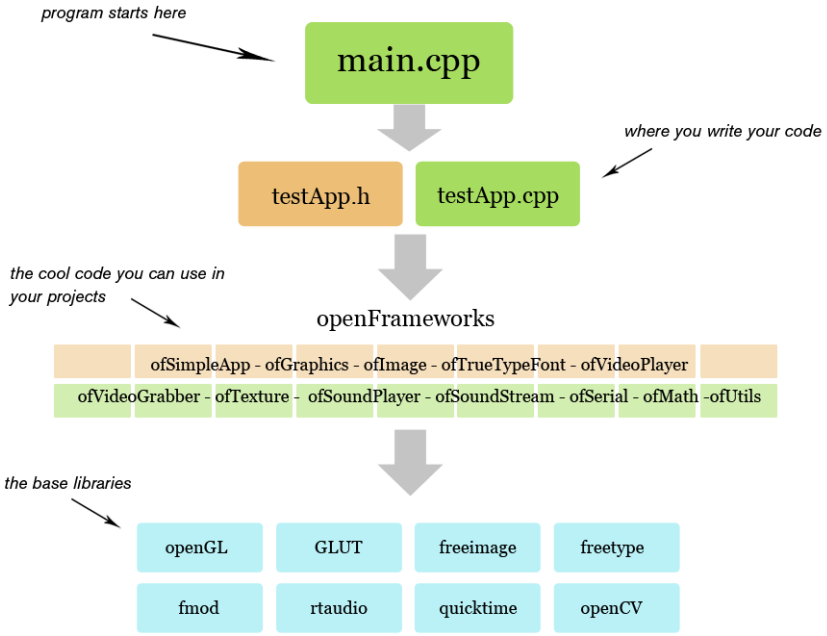

Figure 4.3: Structure of an openFrameworks application [Ofs]

An openFrameworks application must implement a set of functions, that include: `setup()`, `update()` and `draw()`. These functions are implemented in the `testApp.cpp` class. The `setup()` function is called only once, when the application starts. This function is used for initializing variables and to call settings that do not require being continuously updated. The `update()` function is continuously called, but it cannot be used for drawing purposes. This function is called before `draw()`. The `draw()` function is where the code for drawing the application's graphics is set. There are other functions that are also implemented in the `testApp.cpp` class, that are responsible for handling events, such as mouse and keyboard events.

4.2 Types of Annotation

Each type of annotation has different characteristics and thus for each type a different class was assigned, containing specific information to that type. The annotations also have common attributes and therefore each class inherits from a common class *Annotation*, as shown is figure 4.4. Ergo, each type of annotation is defined by its own attributes, enclosed in their specific class, and also by common attributes inherited from the superclass or base class. This relationship between classes, in a hierarchy form, simplifies the insertion of new types of annotations, as code from the base class can be reused.

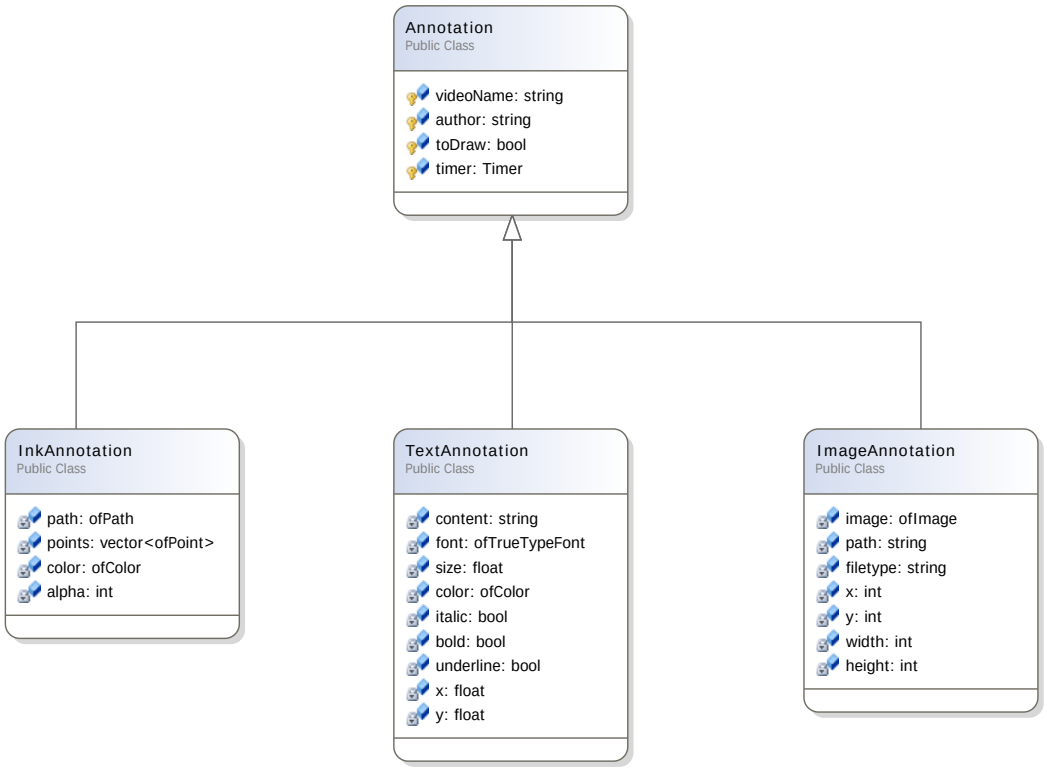

Figure 4.4: Annotations class diagram

An ink annotation comprises the line coordinates and formatting proprieties, such as color. To each drawn line corresponds an ink annotation. A line consists of a set of points with coordinates x and y, which are saved in the vector *points*. Each position of the vector contains an *ofPoint* object, that holds the point's coordinates. For drawing purposes, instead of using the vector and drawing point by point, a different approach was used. The class *ofPath* offers a simpler way to draw lines and has several useful methods available, such as `setColor()`, `setStrokeWidth()` or `draw()`. The object *path* is used for drawing the ink annotation and it contains the points that are also stored in the vector. The color of the line is specified by the user during the annotation, but the line width has a default value.

In *Play* or *Playback* mode, the ink annotations appear for a limited time and will fade away until they disappear. The fade effect is achieved by increasing the transparency over time. During the time the annotation is being drawn, the alpha value is continuously being decreased until the line disappears. Other types of annotations also have a time limit, but will not fade away as the ink annotations.

Text annotations also have formatting proprieties, such as font type, font size and color. The text inserted by the user is stored in the variable *content*. The boolean variables *italic*, *bold* and *underline* indicate if those proprieties are turned off or on. The *ofTrueTypeFont* class provides means for manipulating text settings, such as font type. For drawing purposes and for manipulating text settings an *ofTrueTypeFont* object is used.

An image annotation contains an image that is stored in disk, on the server side. The image is loaded to an object of the *ofImage* class and has a predefined width, height and anchor point (x and y). All images are drawn on the right side of the video being displayed on the player and are arranged from top to bottom. The *ofImage* class offers several methods for manipulating images and its pixels, such as `ofLoadImage()`, `draw()` and `getPixels()`.

The common attributes to all types of annotations include the video name and the user that they belong to. The *Annotation* class also has an object from the *Timer* class. The *Timer* class is a custom class that represents a timer and it is used for managing the time in which the annotations are displayed. The *toDraw* variable was also created for dealing with the time limit problem and indicates whether the annotation should be drawn or not.

4.3 Play Mode

To annotate, the user starts by joining an ongoing session or starting a new one. As mentioned before, when an annotation is inserted a timestamp will be added, indicating the time of insertion. The timestamp is in real time, *i.e.*, if an annotation was inserted at 3pm, the corresponding timestamp will be "15:00:00". Ink and text annotations receive a timestamp at the moment of insertion, whilst image annotations only receive it when the user presses the *save* button. When the *save* button is pressed, all annotations inserted on canvas and attached images will be sent to the server. The annotations are sent in the XML format and the XML file will contain not only content, such as text from a text annotation, but also formatting proprieties and other details of importance (see section 4.5). On the server side, before displaying the annotations on the player, the timestamps are converted to video time. If an annotation has a timestamp with the value "15:00:00" and has arrived at the server at "15:00:10", it indicates that the insertion occurred ten seconds ago. Therefore, if the current video's elapsed time is "01:30:00" the annotation timestamp will be converted to "01:29:50". The timestamp is then converted to seconds and saved to the XML file.

After converting all timestamps, the annotations are ready to be displayed. The annotations that have been sent will be presented on the player and for each one of them a timer will be initialized. During ten seconds, which is the default time, the annotations will be continuously drawn. After that time, the timer stops and the annotations will disappear from the screen.

The timestamp conversion process occurs when the canvas is blank. If the user requests a frame, by choosing the *Get Current Frame* option, all annotations will be associated with that frame.

4.4 Playback Mode

To replay a session and re-watch all annotations added to a video the user must select a session. Remember that a session has a number, an user, a video and a date associated to it. Each session is stored in a XML file that contains all the annotations belonging to that session. When a session is chosen, the system must read the annotations from the correspondent file and mark them conveniently on the timeline. Here is where the timestamps are actively used. Having the exact time in which the annotations were added, we can easily mark the annotations on the timeline.

In *Playback* mode, the timer also has an active role. When the video approaches a moment where an annotation was inserted, the timer is initialized. As in *Play* mode, the timer is set to ten seconds and after that time the annotations will disappear.

4.5 Stored Information

As mentioned in earlier chapters, the data is stored in the server side and it is separated into different folders. Videos, shots (see section 4.6), pictures, fonts, annotations and system files are all stored in the server, but in this section we will focus on the structure of the generated data, such as the XML files created for each session.

A session XML file follows as listed in 4.1 and the annotations are stored inside the tag *annotations*, that comprises one or more *annotation* tags, one for each annotation. The *annotations* tag has the following attributes: *number*, *videoName*, *videoFull*, *author*, *session* and *date*. The *number* attribute contains the total number of annotations for that session. The *videoName* attribute contains the video name, whilst the *videoFull* contains the video full name, including the extension. The *author* attribute corresponds to the user that created the annotations. The *session* attribute contains the session number and *date* contains the date when the session was created.

An *annotation* tag has always the following attributes: type, onFrame, time, timeInVideo. The *type* attribute specifies which is the type of the annotation: ink, text or media. The *onFrame* attribute indicates whether the annotation is linked to a specific frame or not. The *time* attribute has the original timestamp, whilst the *timeInVideo* contains the converted time, the actual video time.

Listing 4.1: Session XML

```
1 <annotations number="1" videoName="Eiffel␣Tower" videoFull="Eiffel␣Tower.mp4"
      author="John␣Doe" session="0" date="30-07-2013">
2   <annotation type="type" onFrame="false" time="00:00:00" timeInVideo="0.0"/>
3 </annotations>
```

An ink annotation is described in XML as shown in the listing 4.2. The *annotation* tag contains the proprieties of the ink annotation. The *color* tag contains the chosen color for the annotation in the *Integer* format. The path coordinates are stored inside the *path* tag, which is followed by *point* tags, that contain the x and y coordinates.

Listing 4.2: Ink annotation XML

```
<annotation type="ink" onFrame="false" time="18:56:08" timeInVideo="18.6">
  <color>-65536</color>
  <path>
    <point x="891.661" y="169.62712"/>
    <point x="891.661" y="171.08342"/>
    <point x="891.661" y="197.11848"/>
    <point x="892.7457" y="232.51692"/>
    <point x="895.032" y="273.97437"/>
    <point x="901.2341" y="311.8889"/>
    <point x="907.2033" y="339.77493"/>
    <point x="907.9322" y="341.69492"/>
  </path>
</annotation>
```

The text annotation is represented in XML as shown in the listing 4.3. As in the ink annotation, the color for the text annotation is also stored in the *color* tag. The content of the annotation, *i.e.*, the text, is stored inside the *content* tag. The anchor point is stored inside the tags x and y. The font size is stored in the *size* tag. Finally, the font style is stored inside the *styles* tag, followed by *style* tags that indicate font styles, such as bold, italic or underline. If the *styles* tag is empty then the font style is normal.

Listing 4.3: Text annotation XML

```
<annotation type="text" onFrame="false" time="18:30:25" timeInVideo="8.0">
  <x>100.0</x>
  <y>100.0</y>
  <color>-65536</color>
  <content>Hello!</content>
  <size>10.0</size>
  <styles>
    <style>italic</style>
    <style>underline</style>
  </styles>
</annotation>
```

An image annotation is described in XML as shown in the listing 4.4. An image annotation only has two tags: name and filetype. The *name* tag contains the name of the file, in this case, the name of the image file. The *filetype* tag stores the file type.

Listing 4.4: Image annotation XML

```
<annotation type="media" onFrame="false" time="00:01:33" timeInVideo="38.3">
  <name>tower.jpg</name>
  <filetype>jpg</filetype>
</annotation>
```

4.6 Image Matching

As previously mentioned in section 3.3.3.6, the system provides a video search using pictures from the user's mobile phone. The image chosen by the user is matched with videos stored in the server side. One by one the videos are matched with the image and if a video is relevant it will be included in the search results.

Before matching videos with the picture, first the videos need to be processed. Initially, each video is decomposed into parts or frames. Each frame is called a shot and it is saved on disk like an ordinary image. As the system deals with large videos, like movies or television series, it was decided that from hundred to hundred frames, a frame is saved. When extracted, the frame is stored in a matrix and later on is converted to an *ofImage* object and saved on disk. The following code presents the extraction of frames from a video, as shown in listing 4.5.

Listing 4.5: Capturing and saving frames

```
cv::VideoCapture * capture = new cv::VideoCapture();
capture->open(filepath);

int count = 0;
while(capture->grab())
{
  if(count%100 == 0)
  {
    cv::Mat frame;
    if(capture->retrieve(frame,0))
      saveShot(frame,filename,count);
  }
  count++;
}
capture->release();
```

Before actually saving a frame, we need to extract the frame descriptor. The code for extracting descriptors is presented in the listing 4.6. To extract descriptors a SURF feature detector and a SURF descriptor extractor were used. Remember that keypoints are points of interest in an image, such as corners, and a descriptor describes a feature point, either through color or shape, among others. The detector detects the image keypoints (or feature points) and stores them in a vector of keypoints. Then the descriptor extractor extracts descriptors (feature descriptors) and stores them in a matrix. Later, the descriptor is saved to a XML file and stored on disk. Saving descriptors and frames prevents having to extract them again, thus improving the response time. Besides extracting descriptors from the videos' frames, a descriptor is also extracted from the image used in the search.

Listing 4.6: Extracting descriptor from frame

```
cv::Mat descriptor;
vector<cv::KeyPoint> keypoints;
int minHessian = 400;
cv::SurfFeatureDetector detector( minHessian );

detector.detect(matImage, keypoints);
descriptorExtractor->compute(matImage, keypoints, descriptor);
```

Having the frames and their respective descriptors, we can proceed to matching each frame from each movie to the image. The code for matching a frame with an image is presented in the listing 4.7. For matching, a Flann Based descriptor matcher was used. The matcher matches the descriptor of the image to the descriptor of the shot or frame. The results from the match are stored in a vector of *DMatch* objects. Each vector position contains the result from the match between two keypoint descriptors, including the distance between points.

If the number of matches is equal or superior to a minimum value, then there are sufficient matches to consider a possible match between a frame and the image. Next, we need to eliminate matches that are too distant from each other. The formula to calculate the distance threshold is

$$thresholdDist = \frac{\sum_{i=1}^{n} a_i}{2n} \tag{4.1}$$

where n is the number of matches and a_i is the *ith* match.

If a match distance is inferior to the distance threshold, then it is dismissed. Finally, if the number of the remaining matches is above a default threshold, the frame is considered to have relevance and it is added to the search results. If a video has at least one frame that matched the image, the video will appear as a result to the user.

Listing 4.7: Matching descriptors

```
vector<cv::DMatch> matches;
descriptorMatcher->match(image_descriptor, shotDescriptor, matches);

int countMatches = 0;
if(matches.size() >= minOfMatches)
{
  float total = 0;
  float thresholdDist = 0;

  for (size_t i = 0; i < matches.size(); i++)
  {
    total += matches[i].distance;
  }

  thresholdDist = total/(matches.size()*2);

  for( int i = 0; i < matches.size(); i++ )
  {
```

```
19      if(matches[i].distance < thresholdDist){
20          countMatches++;
21      }
22    }
23  }
```

5

Evaluation

This chapter describes the evaluation process, where the implemented solution was evaluated and tested through usability tests. The tests involved different types of users, some with experience with touch devices and drawing applications and others with few or no experience with such applications. To assess the usability, utility and overall satisfaction, a questionnaire was created and given to the participants to be filled in.

Two tests were conducted, the first in July (preliminary tests) and the second in September (final tests), both at Faculdade de Ciências e Tecnologia with students and teachers from the faculty. The tests consisted of a set of tasks that the users must perform, followed by the questionnaire filling. During the evaluation tests a Samsung Galaxy Note and a Samsung Galaxy S III were used for the mobile application and a Samsung ME46A LED display with a 46" Touch Overlay for displaying the player.

5.1 Preliminary Tests

The preliminary tests conducted in July aimed to determine the system's usability, specifically if the the platform is easy to learn, easy to recall and the overall satisfaction, and to receive feedback from the users concerning errors and other features that could complement and improve the solution. This first test gathered several suggestions and ideas from the users, that resulted in improvements on the existing features and in new features as well.

5.1.1 Description

The preliminary tests were divided into two separate days, the first test taking place on the 25th of July and the second on the 29th, both at Faculdade de Ciências e Tecnologia. The participants were students and teachers from the faculty and they had at their disposal the smartphones mentioned above, a Galaxy Note and a Galaxy S III. The tests were conducted in a room with a television for displaying the player and the users had to perform several tasks. There was no script for the test but the users received a brief explanation regarding the concept and the objectives of the system, and had to perform a minimum amount of tasks, such as: annotate a video using all types of annotation, re-watch a session and search videos. The system was tested with two users at a time, for the purpose of evaluating the collaborative annotation feature. The evaluation of the solution was accomplished through a questionnaire and observation, to assess issues and difficulties experienced by the users.

The users also had to evaluate another project from a doctoral student, regarding television content annotation through motion capture. This other application uses the Kinect device from Microsoft to capture the user's movements and instead of using a smartphone and a pen to annotate, the user annotates through predefined gestures. After testing both applications, the users had to answer some questions about whether they would choose one application or the other, regarding usability and recommendation to other users.

5.1.2 Observation Analysis

During the course of the tests, the users showed only some difficulties while using the application and also suggested new features to improve it. The difficulties were mostly related to saving annotations and replaying the desired session.

Having to explicitly save the annotations was not always clear while testing the application. Having to click the *save* button to save annotations and have them appear on the player was an issue for some users, because they were expecting that the annotations would be automatically displayed on the screen.

Earlier in the application design, to erase ink annotations on the canvas, there were only two options: the eraser and the *undo* button. The tests revealed that users disliked having to erase their annotations with either option, because it was time consuming. Thus, the possibility to clear all the canvas at once was suggested and became a feature on the final version of the system.

All the types of annotations were well received by the users and there were no difficulties to add any of the types to the videos. Of the three types, the most used and enjoyed was the digital ink. Some users had some problems while tracing the path due to the canvas being initially blank and thus not knowing exactly were to draw. Once the *Get Frame* option was discovered the problems disappeared.

The playback feature was highly appreciated by the users and there were only some

suggestions regarding it. Even if two users are annotating the same content together, their annotations are saved separately and they can only be replayed separately as well. Some users suggested that it would be useful if it were possible to re-watch annotations from different users at the same time.

Finally, an issue experienced by most users was the long response time of the search functionality. When searching through videos with higher quality and a longer duration, the response time was significantly long, leaving users waiting for results. This was the main issue pointed out by the users.

Some suggestions, such as the clear option, were implemented while others, though taken into to account, could not be developed within the project duration due to time and resources limitations. The final test was scheduled for September and includes the new features and improvements suggested by the users.

5.1.3 Questionnaire and Analysis of the Results

In order to collect data regarding usability, utility and overall satisfaction, an anonymous questionnaire was created and handed over to the users to fill in after concluding the test. The questionnaire is divided into two main parts: one concerning the user profile and the other regarding the evaluation of the solution. The first part aims to summarize the user profile, namely their age, gender and familiarity with technologies and drawing interfaces. The second part aims to evaluate the solution through the rating of the system's features for a quantitative analysis and through suggestions and commentaries for a qualitative analysis. As the users were Portuguese native speakers, the questionnaire was written in Portuguese. The results obtained from the questionnaire are presented in greater detail in Appendix B.

5.1.3.1 User

In the preliminary tests ten users were surveyed, of which 30% are female and 70% are male (C.2), with ages between 22 and 38, being the predominant age group between 22 and 23 years, as shown in C.1. All participants said to be comfortable with new technologies (C.3) and 50% said they had tried drawing interfaces before (C.4), being tablets (50%) and smartphones (30%) the most common devices (C.5).

5.1.3.2 Usability

Regarding usability, most users said the application was easy to learn and easy to interact with, as shown in charts C.7 and 5.1. As for the mobile application, most users said the content is well organized and that the transitions between sections are fluid, as shown in charts C.9 and C.10. However, opinions were divided in question CI3 (5.2) concerning the application time response, where 30% absolutely considered the application to respond quickly and 20% said otherwise.

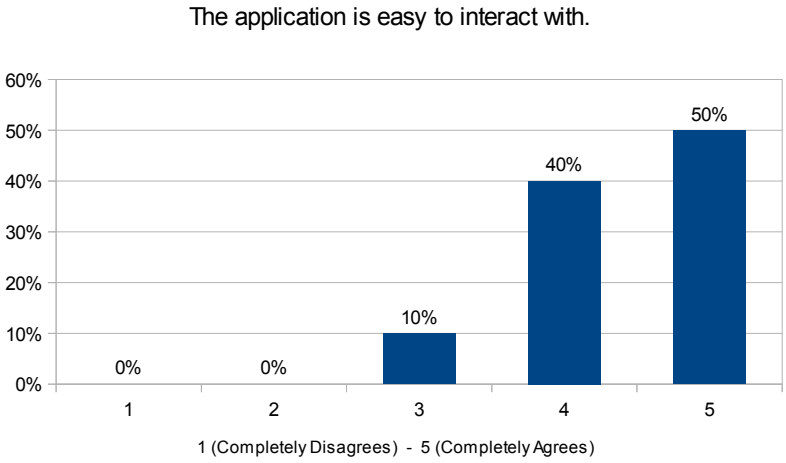

Figure 5.1: Question CI1 : The application is easy to interact with.

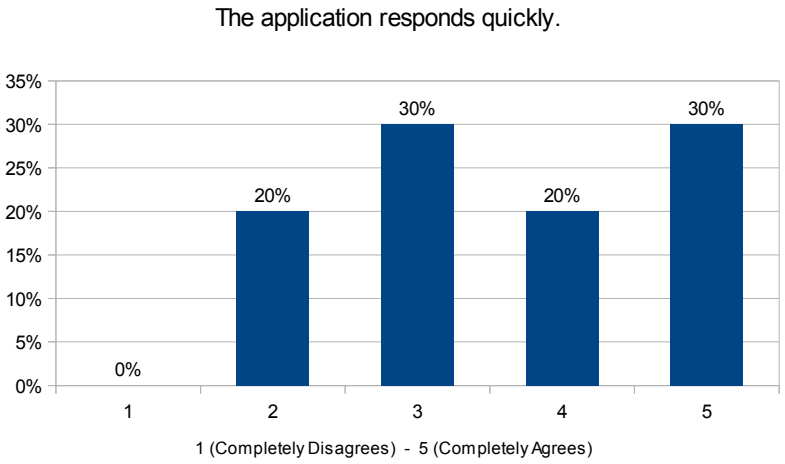

Figure 5.2: Question CI3 : The application responds quickly.

5.1.3.3 Annotation

In the annotation section, 70% of users said that the tools in the drawing environment are easily recognizable and that the settings for each tool can be easily accessed, as shown in charts C.12 and C.13. As observed during the tests, most users said that the need to save the annotations was not clear, so they could be displayed on the player, as seen in figure 5.3. Most users considered the possibility to attach pictures and to request a frame to be clear, as shown in charts C.15 and C.16. As for drawing with or without a pen, most users stated that is easy to draw in either situation but it is easier with the aid of a pen, as shown in charts C.17 and C.18. 50% said to be very satisfied with this section and 50% said to be satisfied.

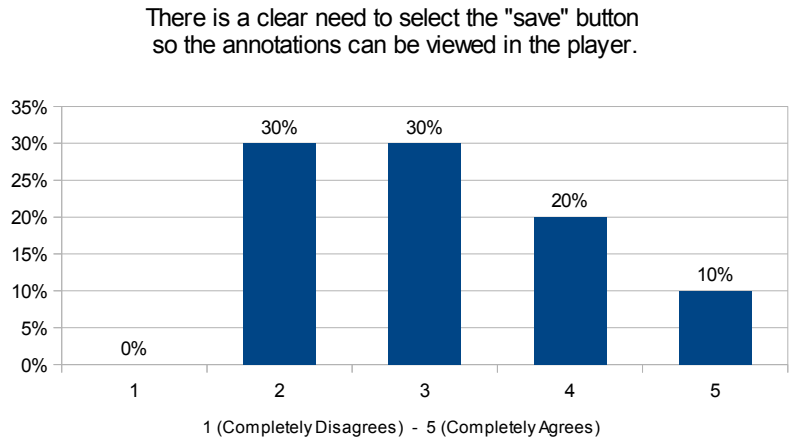

Figure 5.3: Question CII 1c) : There is a clear need to select the "save" button, so the annotations can be viewed in the player.

5.1.3.4 Playback Mode

In the playback mode section, 80% completely agreed that being able to review their annotations is important and 80% said being able to review annotations from other users is also important, as shown in C.20 and C.21. Most users said it to be easy to distinguish their annotations from others and that they could easily access a specific session in the mobile application, as shown in C.22 and C.23. 60% completely agreed that it is easy to access a specific annotation (ink, text or image) on the player, in playback mode. 60% of users said to be very satisfied with this section.

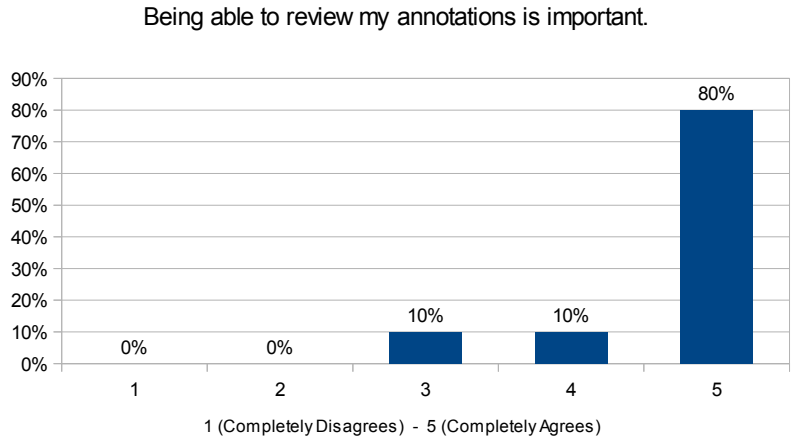

Figure 5.4: Question CII 2a) : Being able to review my annotations is important.

5.1.3.5 Video Search

In the video search section, 80% considered that searching for videos related to their pictures is useful, as shown in C.27. Most users said the two possible types of search are clear and that the instructions provided are sufficient, as shown in C.30 and C.31. As observed during the tests, most users said the waiting time for results is too long, as

shown in 5.5. There where different opinions regarding the search results, where some users said that the results were strongly related to the picture they had chosen and others said otherwise, as shown in 5.6. This contrasting data from question CII 3d) was probably due to the picture the user selected or even to the selection of videos used for testing. 50% of users said to be satisfied with this section.

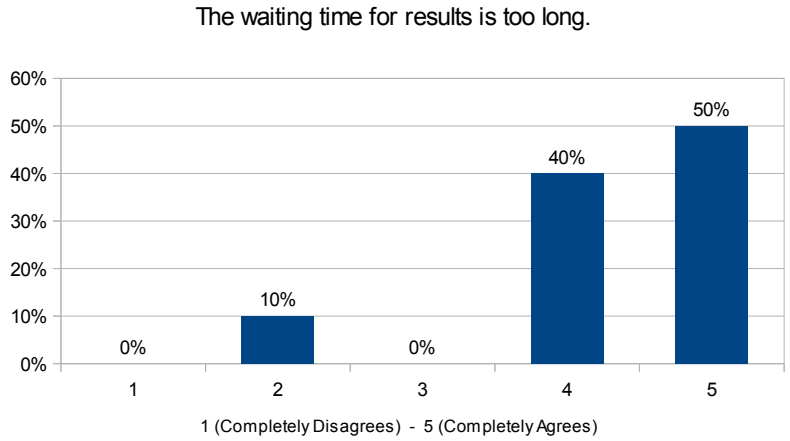

Figure 5.5: Question CII 3c) : The waiting time for results is too long.

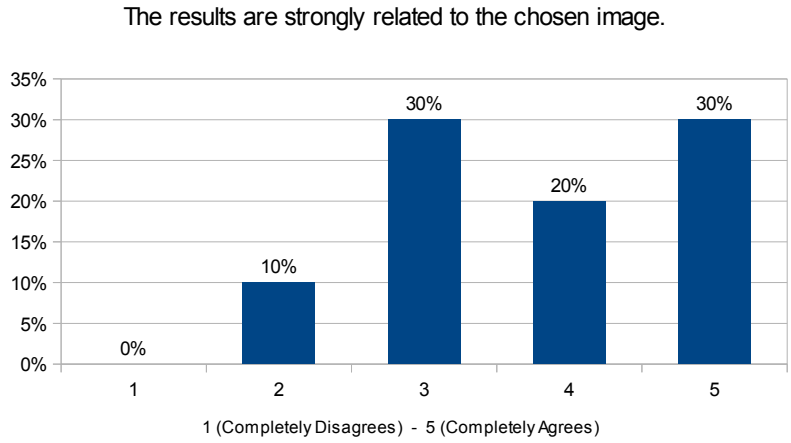

Figure 5.6: Question CII 3d) : The results are strongly related to the chosen image.

5.1.3.6 Player

In the player section, most users said that the video occupies an adequate portion of the screen and that the functions of each button for controlling the video are easily identifiable, as shown in C.33 and C.35. Most users also said that the amount of time the annotations remained visible is appropriate, as shown in 5.7. All users said that the markers on playback mode and the images are visible (C.38 and C.37) and most users said the text on the player is legible (C.36). The labels in the playback mode were considered appropriate by most users, as shown in C.39. 60% of users said to be very satisfied with this section.

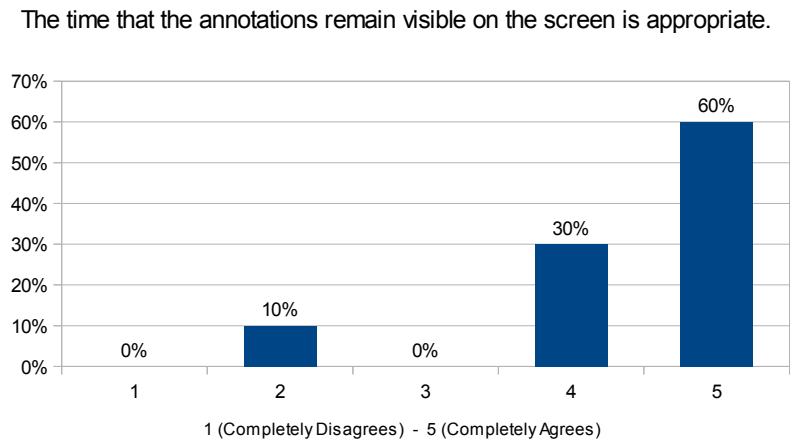

Figure 5.7: Question CII 4b) : The time that the annotations remain visible on the screen is appropriate.

5.1.3.7 Overall Satisfaction

The application was said to provide the necessary tools for video annotation by most users (C.41). The majority of users also considered the application to be very useful (C.42). Although users said using a pen facilitates the drawing process, 50% said that using a pen for annotating is completely unessential (5.8). The font size and font type where considered to be adequate by most users, as shown in C.44 and C.45. A fair amount of users said they required help to navigate the application (C.46), but all of them said they would recommend the application to others (C.47). 50% of users said to be satisfied with the application and 50% said to be very satisfied, as shown in figure 5.9.

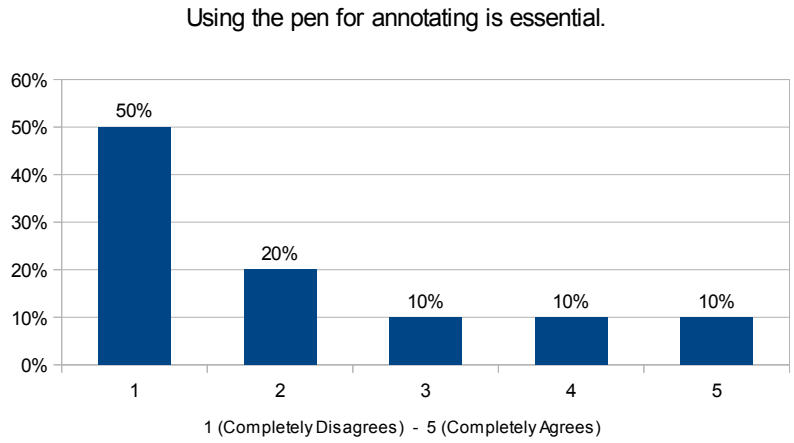

Figure 5.8: Question CIII 3 : Using the pen for annotating is essential.

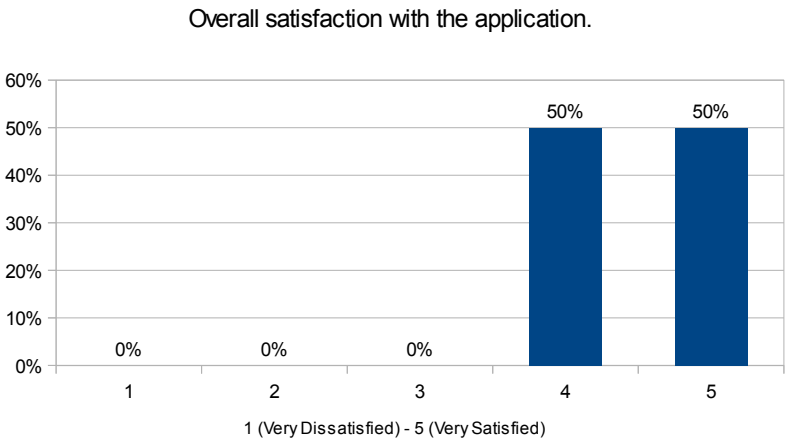

Figure 5.9: Question CIII 8 : Overall satisfaction with the application.

5.1.3.8 Kinect vs Mobile

After testing both applications, the users also had to respond some questions to compare the application that used the Kinect and our application. 70% of users said both applications were easy to use, as shown in B.49. 40% of users said the Kinect application was more straightforward and 40% of users said otherwise, choosing the mobile application (B.50). 50% of users said the Kinect application to be more appealing, as shown in B.51. Finally, 50% of users said they would prefer the mobile application (5.10), but 70% said they would recommend both applications to a friend (B.52).

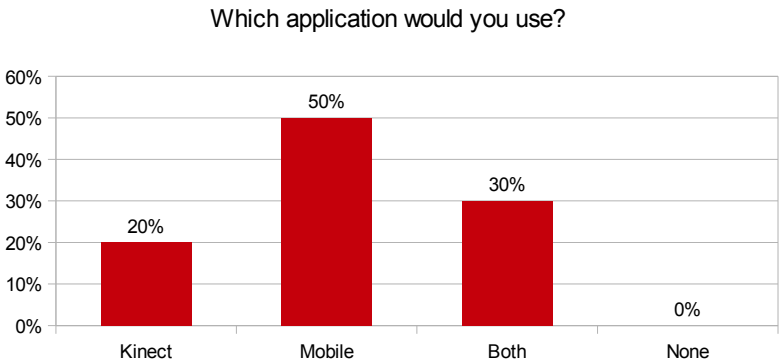

Figure 5.10: Question Q30 (from the comparison questionnaire) : Which application would you use?

5.2 Final Tests

The final tests conducted in September also aimed to determine the system's usability and overall satisfaction with the application. Unlike the preliminary tests, the final tests focused only on this solution and the collected data will be compared with the data from the previous tests. Suggestions from the participants were taken into account, however

no new features were added.

5.2.1 Description

The final tests were divided into two separate days, the first test taking place on the 17th of September and the second on the 18th, both at Faculdade de Ciências e Tecnologia. The participants were students and teachers from the faculty and they had at their disposal the same devices as in the previous tests. The tests were conducted in the same conditions as the previous ones, but this time only one user tested at a time. There was no script for the test, but the users receive a brief explanation about the concept of the system and had to perform a minimum amount of tasks, such as: annotate a video using all types of annotation, share an annotation, re-watch a session and search videos. The evaluation of the solution was accomplished through a questionnaire, which was exactly the same that was given in the preliminary tests.

5.2.2 Questionnaire and Analysis of the Results

The questionnaire is equal to the one delivered to the users on the preliminary tests. The following sections describe the results obtained from the final tests. The results obtained from the questionnaire are presented in greater detail in Appendix C.

5.2.2.1 User

In the final tests eight users were surveyed, of which 87,5% are male and 12,5% are female, with ages between 22 and 26, being the predominant age 23 with 50% of the users. All participants said to be comfortable with new technologies and 62,5% said they had tried drawing interfaces before, being smartphones (62,5%) and tablets (37,5%) the most common devices.

5.2.2.2 Usability

In a scale from 1 (completely disagrees) to 5 (completely agrees), 62,5% of users answered with 4 and 37,5% with 5, concluding that the application was easy to interact with (5.11). Using the same scale, when asked if the application was easy to learn users agreed, where 50% of users answered with 4 and 50% with 5. In question CI 3. concerning the application's time response, the data diverged (5.12), where 12,5% answered with 2, 50% with 3, 25% with 4 and 12,5% with 5. This different responses, both in the preliminary and final tests, are probably due to problems experienced with the wi-fi connectivity and even the possibility that the users included the search time response with the overall time response of the application. Again with the same scale, users said that the application is well organized (50% with 4 and 50% with 5) and that the transitions between sections on the mobile application are fluid (12,5% with 4 and 87,5% with 5).

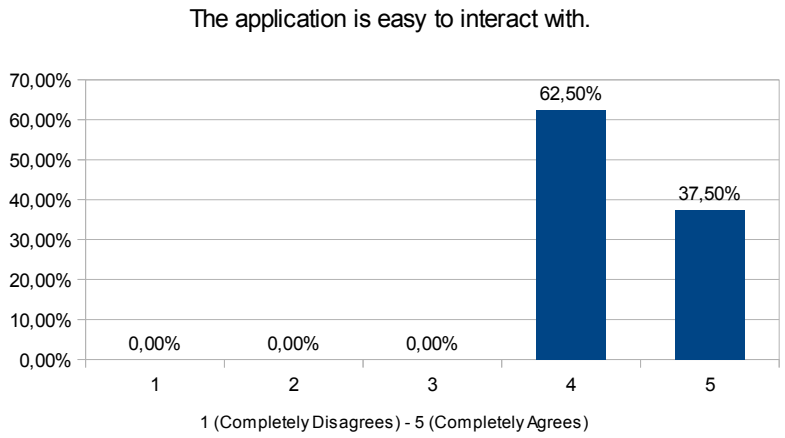

Figure 5.11: Question CI1 : The application is easy to interact with.

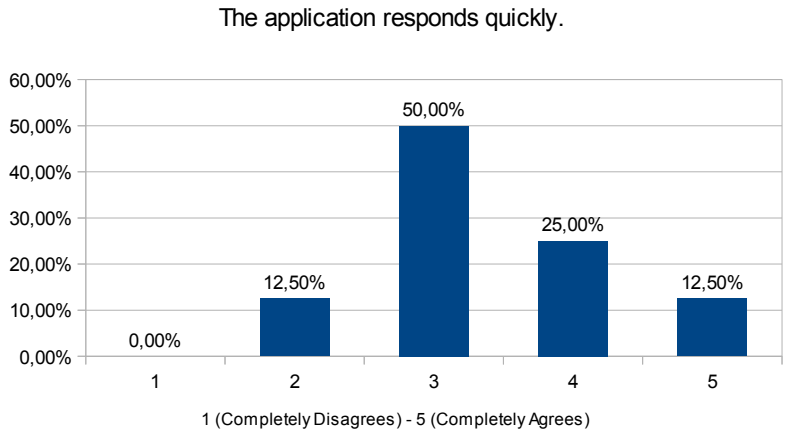

Figure 5.12: Question CI3 : The application responds quickly.

5.2.2.3 Annotation

In the annotation section, most users said that the tools in the drawing environment are easily recognizable where, in a scale from 1 (completely disagrees) to 5 (completely agrees), 25% answered with 4 and 50% with 5. Users also said that the settings for each tool can be easily accessed, where 37,5% answered with 4 and 50% with 5. As in the preliminary tests, users said that the need to save the annotations was not clear (5.13), where again with the same scale, 37,5% answered with 2 and 37,5% with 3 and only 25% with 4. Users considered the possibility to attach pictures (50% with 4 and 37,5% with 5) and to request a frame (37,5% with 4 and 62,5% with 5) to be clear. As for drawing with or without a pen, users said that is easy to draw in either situation, where 87,5% completely agreed to be easy to draw with a pen and 75% also completely agreed to be easy to draw without a pen. 50% of users said they were satisfied with this section and 37,5% said to be very satisfied.

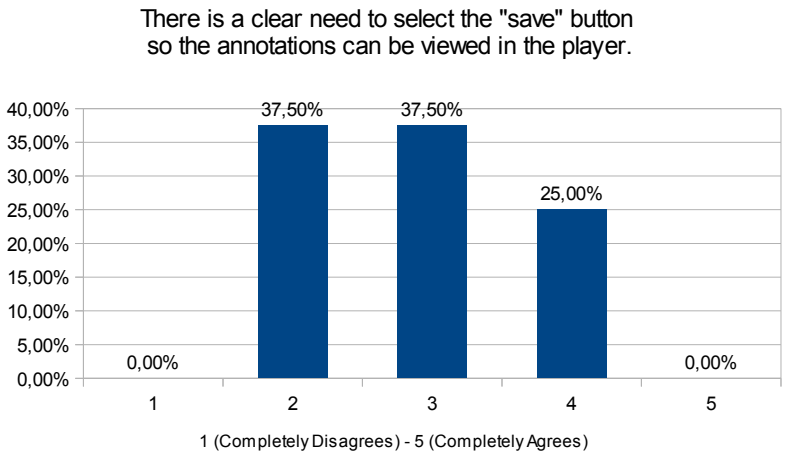

Figure 5.13: Question CII 1c) : There is a clear need to select the "save" button, so the annotations can be viewed in the player.

5.2.2.4 Playback Mode

In the playback mode section, 75% said they completely agreed that being able to review their annotations is important (5.14) and 62,5% said being able to review annotations from other users is also important. 87,5% of users said they completely agreed on being easy to distinguish their annotations from others and 75% also completely agrees that accessing a specific session on the mobile application is easy. Most users said to be easy to access a specific type of annotation (ink, text or image) on the player, after selecting a session on playback mode (62,5% answered with 5 and 25% with 4, in a scale of 1 (completely disagrees) to 5 (completely agrees)). 37,5% of users said to be satisfied with this section and 50% to be very satisfied.

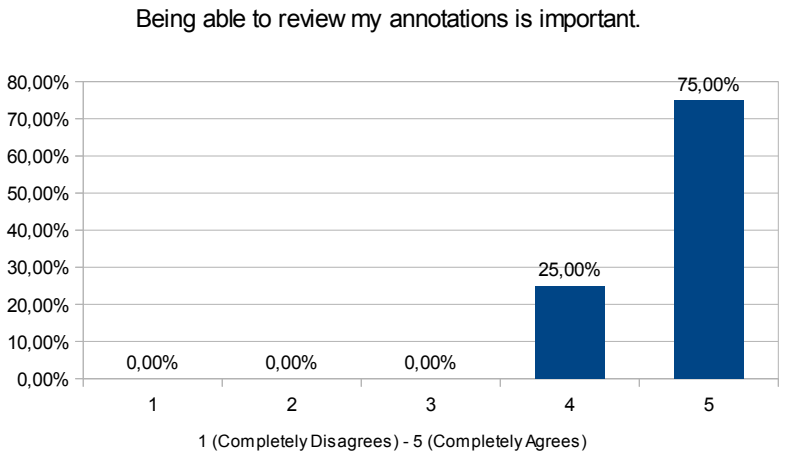

Figure 5.14: Question CII 2a) : Being able to review my annotations is important.

5.2.2.5 Video Search

In the video search section, 75% completely agreed that searching for videos related to their pictures is useful. The objective of this section was considered to be clear (50% answered with 5 and 50% with 4, in a scale of 1 (completely disagrees) to 5 (completely agrees)) and the two possible types of search were also clear (87,5% answered with 5 and 12,5% with 4.). All users said that the provided instructions are sufficient for understanding the search. Most users agreed that the waiting time for results is too long (5.15), with 37,5% answering with 5, 50% with 4 and 12,5% with 3, in a scale of 1 (completely disagrees) to 5 (completely agrees). As in the preliminary tests, there were different opinions regarding the search results, where some users said that the results were related to the picture they had chosen and others said otherwise (25% answered with 2, 37,5% with 3 and 37,5% with 4), as shown in 5.16. Again this contrasting data from question CII 3d) was probably due to the picture the users selected or the selection of videos used for testing. 62,5% of users said to be satisfied with this section and 25% to be very satisfied.

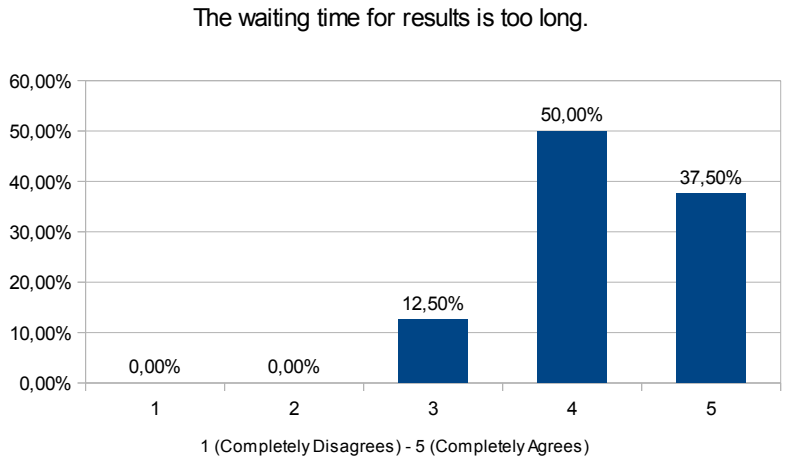

Figure 5.15: Question CII 3c) : The waiting time for results is too long.

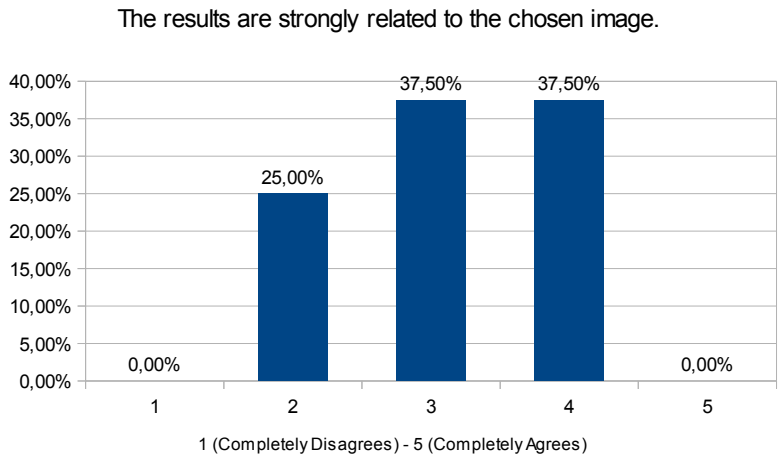

Figure 5.16: Question CII 3d) : The results are strongly related to the chosen image.

5.2.2.6 Player

In the player section, 75% of users completely agreed that the video occupies an adequate portion of the screen and 87,5% of users completely agreed that the functions of each button for controlling the video are easily identifiable. 75% of users completely agreed that the amount of time the annotations remained visible is appropriate (5.17). All users said that the markers on playback mode and the images are visible and 75% of users completely agreed that the marker labels are adequate. The question regarding whether the texts on the player are legible or not had divergent data. In a scale from 1 (completely disagrees) to 5 (completely agrees), 12,5% answered with 2, 12,5% with 3, 25% with 4 and 50% with 5. These results from question CII 4d), are probably due to some problems experienced with fonts during the tests, that did not allow some users to properly read the text annotations. 37,5% of users were satisfied with this section and 62,5% were very satisfied.

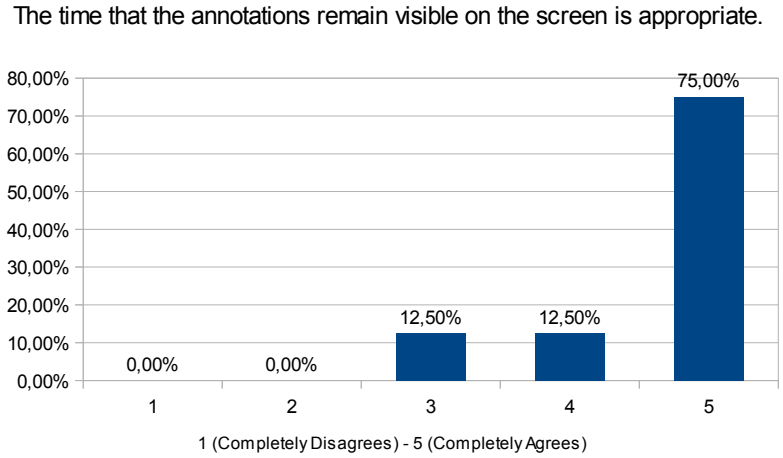

Figure 5.17: Question CII 4b) : The time that the annotations remain visible on the screen is appropriate.

5.2.2.7 Overall Satisfaction

The application was said to provide the necessary tools for video annotation by most users, where 87,5% answered with 5 in a scale of 1 (completely disagrees) to 5 (completely agrees). Most users considered the application to be useful, where 25% answered with 5, 50% with 4 and 25% with 3, in a scale of 1 (low) to 5 (high). Again, despite being said that using a pen facilitates the drawing process, most of users said that using a pen for annotating is completely unessential (5.18). Most users completely agreed that the font size is adequate (62,5% answered with 4 and 37,5% with 5) and that the font type is also adequate (50% answered with 4 and 50% with 5). Most users disagreed with help being necessary for navigating the application (25% answered with 1 and 50% with 2). Most users said they would recommend this application to others and 75% said to be satisfied with the application (5.19).

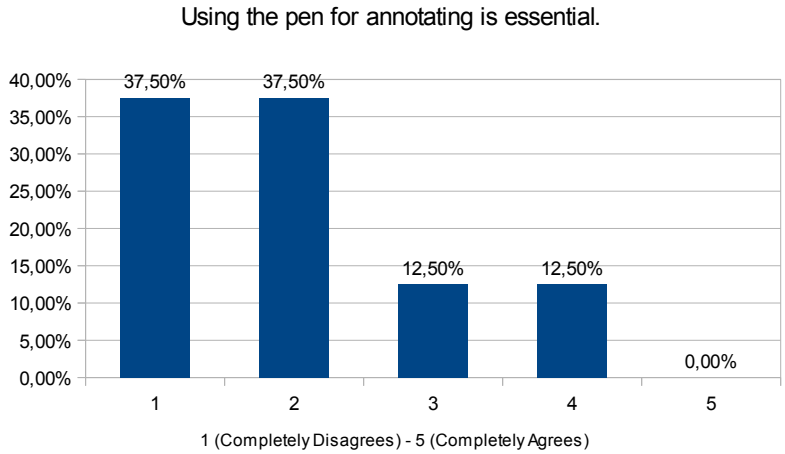

Figure 5.18: Question CIII 3 : Using the pen for annotating is essential.

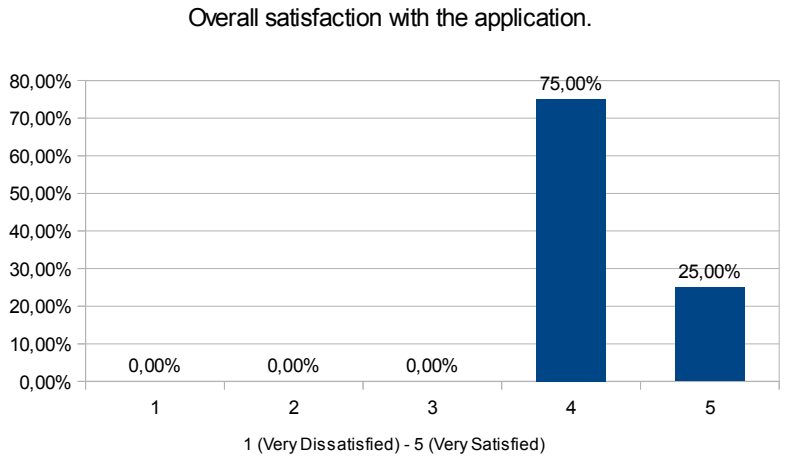

Figure 5.19: Question CIII 8 : Overall satisfaction with the application.

5.3 Preliminary Tests vs Final Tests

The final and the preliminary tests had some discrepancies but overall the results were similar and were as expected. The majority of the users were male and the predominant age group was between 22 and 23 years in both tests. Most users said they had tried drawing interfaces before, being smartphones and tablets the most common devices.

Regarding usability, in both tests most users considered the application to be easy to interact with and easy to learn, as well as the content being well organized and the transitions between sections being fluid in the mobile application. In both tests the question with divergent data was CI3; however in the final tests, the answers tended more to one value (3), indicating the users considered the application to respond not quickly but also not slow. In the preliminary tests, users did not experienced connectivity issues unlike the final tests, but in both tests some users might have included the search response time in the global response time, which may explain these results.

In the annotation section, the tools and their respective settings were easily recognizable and accessed in both tests. The possibility to attach pictures and request frames was also clear in both tests. As expected, the results were similar regarding the need to save annotations in the preliminary tests and in the final tests. Users were expecting that the annotations would be saved automatically, but once they realize otherwise they would automatically try to find a way to save them. The users considered to be easy to draw either with or without a pen.

In the playback section, most users considered important being able to review their and others' annotations in both tests. Most users also agreed on being easy to distinguish their annotations from others' and that they could easily access a specific type of annotation or a specific session.

The video search feature was considered to be useful by most users in both tests. Users agreed that the instructions provided for this section were sufficient and that the two search modes were clear. In both tests, users agreed on the waiting time for results being too long but opinions were divided regarding the search results. The waiting time in the search mode is affected by different factors, such as the quality of the videos, the quality of the image and the duration of the videos. The higher the quality of the images the longer it takes to process and match them, thus increasing the processing time. The longer the duration of the videos the higher the number of frames that need to be processed, thus increasing the processing time as well. All these factors, including the number of videos the user selects for search, contribute to a longer waiting time which explains the results of question CII 3c). Regarding the search results, there are also different factors that can affect the outcome of a search. The user could have selected an image that had no relation whatsoever with the videos saved on the server and so the search will return no results or results that do not match the expectation of the user. Besides this possible scenario, there can also be problems with the matching algorithm, which can be fallible and does not always produces good results.

In the player section, most users considered that the video occupies an adequate portion of the screen and that the functions of each button for controlling the video are easily identifiable. Users agreed that the markers and the images are visible and that the labels for the markers are adequate. The time that the annotations remain visible on the screen was considered to be appropriate by most users in both tests. Only question CII 4d) had different results, where most users considered the texts to be visible in the preliminary tests and in the final tests opinions were more divided. The divergent data collected from later tests was probably due to problems experienced during the tests regarding fonts, as mentioned previously.

The application was said to provide the necessary tools for video annotation by most users in both tests. Opinions diverged slightly in the final tests regarding the application's utility, however the majority of users considered the application to be useful in both occasions. The pen was considered to be unessential in both tests by most users. This result was unexpected, since from beginning the idea was having the user using

a pen for annotating, as it provides a more natural feel and it aids with precision. In the preliminary tests, a fair amount of users said that help was needed for navigating the application, however in the final tests most users said otherwise. This situation was probably due to the fact that on the first test users tested two at a time and in the final test users tested alone. Evaluating the system by themselves probably allowed users to concentrate more and being less distracted by their fellow colleagues.

Overall users said to be satisfied or very satisfied with the application and most of them said they would recommend it to others.

6

Conclusion and Future Work

The following chapter will present a brief analysis concerning the work accomplished with this dissertation, as well as suggestions and ideas for improving and continuing the implemented solution.

6.1 Conclusion

This dissertation introduces a collaborative video annotator for annotating and sharing television content. For the prototype, a mobile application and a video player interface were developed, for smartphones and television respectively.

The solution provides means for annotating video collaboratively with digital ink, text and images as annotations. Different users sharing the same physical space can work collaboratively or cooperatively to annotate the same content, using either types of annotation to add notes, leave comments and attach pictures. To achieve this goal, each user annotates using the mobile application while watching the video on the TV. The application offers a platform for annotation, where a canvas is provided for drawing, adding text or attaching images. Meanwhile, the TV interface displays the video of choice and the annotations that are being currently added by users sharing the same session. These annotations can also persist outside the application context, since the solution provides integration with social networks and cloud storage services, allowing users to share and save annotations on other platforms.

Reviewing annotations can be very useful, either after an annotation session to review their own annotations or simply for browsing other users' annotations, and thus this also became a feature of our solution. While re-watching a session, annotations can be easily accessed to, being properly marked on a timeline for that purpose.

Being images a form of annotation and thus sharing a connection with the video content, it was considered to be interesting the possibility to search videos using pictures. Either photographs taken by the owner of the smartphone or images downloaded from the web can be used to search content. Thus, if someone is interested in watching a movie that takes place in a city that they are eager to visit they could do such a search, using an image of a famous monument or museum to search for the desired video.

The system was regularly tested throughout the development process and underwent usability tests, which allowed for improvements on the existing features and introducing new features as well. The tests were conducted on the Faculdade de Ciências e Tecnologia by different types of users, where they had to perform several tasks and answer a questionnaire. The results were positive and showed that the users enjoyed the application and that they would use it again and recommend it to their friends. The possibility to rewatch annotations was highly appreciated and the search using images was considered to be an interesting feature.

There are still many improvements and functionalities that could be implemented, which is the subject of the following section. Aside from the developed work during the course of this dissertation, writing a paper for submitting in a human-machine interaction or multimedia international conference is also planned.

6.2 Future Work

In the future, the solution should be redesigned to fully support collaborative annotation to allow both users that share the same space and users in different locations, to share the same session and annotate the same content. Instead of having a local system, the architecture will be distributed to support this type of feature.

To improve the playback mode, annotations could be accessed individually while rewatching a session. Instead of only selecting a session and watching all annotations, the user could select a specific annotation or browse through all annotations using the mobile application. It would also be interesting to edit or delete annotations.

At this time, there are only three types of annotations: ink, text and image. These types of annotations could be extended to other types, such as audio, video or links.

To improve the video control, the main functions for controlling a video, such as *play* and *pause*, could be implemented on the mobile side. Thus, the user could remotely control the video without having to go to the player on the TV to do such a task.

Regarding the video search, the matching algorithm could also be improved, thus decreasing the long waiting time for results to a more reasonable time.

Finally, we believe that our solution contributed to this field and could be further improved and so the writing of a paper, or papers, is in order and is in schedule for the months that follow the delivery of this document.

Bibliography

[AGL03] G. D. Abowd, M. Gauger, and A. Lachenmann. "The Family Video Archive: an annotation and browsing environment for home movies". In: *MIR '03 Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*. Berkeley, California, USA: ACM Press, 2003, pp. 1–8. ISBN: 1-58113-778-8.

[AAPN09] J. F. Abreu, P. Almeida, R. Pinto, and V. Nobre. "Implementation of Social Features Over Regular IPTV STB". In: *EuroITV '09 Proceedings of the seventh European conference on European interactive television conference*. Leuven, Belgium: ACM Press, 2009, pp. 29–32. ISBN: 978-1-60558-340-2.

[AN07] M. Ames and M. Naaman. "Why we tag: motivations for annotation in mobile and online media". In: *CHI '07 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. San Jose, CA, USA: ACM Press, 2007, pp. 971–980. ISBN: 978-1-59593-593-9.

[And] *Android application components*. http://www.edureka.in/blog/android-interview-questions-answers-for-beginners/. Last Access: August 2013. 2012.

[BMG10] T. Ballendat, N. Marquardt, and S. Greenberg. "Proxemic Interaction: Designing for a Proximity and Orientation-Aware Environment". In: *ITS '10 ACM International Conference on Interactive Tabletops and Surfaces*. Saarbrücken, Germany: ACM Press, 2010, pp. 121–130. ISBN: 978-1-4503-0399-6.

[BTG06] H. Bay, T. Tuytelaars, and L. V. Gool. "SURF: speeded up robust features". In: *ECCV'06 Proceedings of the 9th European conference on Computer Vision*. Springer-Verlag, 2006, pp. 404–417. ISBN: 3-540-33832-2 978-3-540-33832-1.

[CVSAaFC11] D. Cabral, J. a. Valente, J. a. Silva, U. Aragão, C. Fernandes, and N. Correia. "A creation-tool for contemporary dance using multimodal video annotation". In: *Proceedings of the 19th ACM international conference on Multimedia*. MM '11. Scottsdale, Arizona, USA: ACM, 2011, pp. 905–908. ISBN: 978-1-4503-0616-4. DOI: 10.1145/2072298.2071899. URL: http://doi.acm.org/10.1145/2072298.2071899.

[CBGJKS08] P. Cesar, D. C. Bulterman, D. Geerts, J. Jansen, H. Knoche, and W. Seager. "Enhancing social sharing of videos: fragment, annotate, enrich, and share". In: *Proceedings of the 16th ACM international conference on Multimedia*. MM '08. Vancouver, British Columbia, Canada: ACM, 2008, pp. 11–20. ISBN: 978-1-60558-303-7. DOI: 10.1145/1459359.1459362. URL: http://doi.acm.org/10.1145/1459359.1459362.

[DE06] N. Diakopoulos and I. Essa. "Videotater: an approach for pen-based digital video segmentation and tagging". In: *Proceedings of the 19th annual ACM symposium on User interface software and technology*. UIST '06. Montreux, Switzerland: ACM, 2006, pp. 221–224. ISBN: 1-59593-313-1. DOI: 10.1145/1166253.1166287. URL: http://doi.acm.org/10.1145/1166253.1166287.

[GCCGIGCP04] R. Goularte, R. G. Cattelan, J. A. Camacho-Guerrero, V. R. Inácio Jr., and M. da Graça C. Pimentel. "Interactive multimedia annotations: enriching and extending content". In: *Proceedings of the 2004 ACM symposium on Document engineering*. DocEng '04. Milwaukee, Wisconsin, USA: ACM, 2004, pp. 84–86. ISBN: 1-58113-938-1. DOI: 10.1145/1030397.1030414. URL: http://doi.acm.org/10.1145/1030397.1030414.

[GMBDMW11] S. Greenberg, N. Marquardt, T. Ballendat, R. Diaz-Marino, and M. Wang. "Proxemic interactions: the new ubicomp?" In: *ACM Interactions* 18.1 (Jan. 2011), pp. 42–50. ISSN: 1072-5520. DOI: 10.1145/1897239.1897250. URL: http://doi.acm.org/10.1145/1897239.1897250.

[JGCP10] R. F. Júnior and M. da Graça Campos Pimentel. *Um modelo arquitetural para captura e uso de informações de contexto em sistemas de anotações de vídeo*. Instituto de Ciências Matemáticas e da Computação da Universidade de São Paulo. Available at: http://www2.icmc.usp.br/~posgrad/geral/artigos2010/Artigo_Roberto_Faga_Junior. 2010.

[KBE10] R. Knauf, A. Berger, and M. Eibl. "Constraints and simplification for a better mobile video annotation and content customization process".

In: *Workshops of the 8th European Conference on Interactive TV and Video, EuroITV 2010*. 2010.

[KKKRBHE11] R. Knauf, J. Kürsten, A. Kurze, M. Ritter, A. Berger, S. Heinich, and M. Eibl. "Produce. Annotate. Archive. Repurpose – Accelerating the Composition and Metadata Accumulation of TV Content". In: *AIEMPro '11 Proceedings of the 2011 ACM international workshop on Automated media analysis and production for novel TV services*. Scottsdale, Arizona, USA: ACM Press, 2011, pp. 31–36. ISBN: 978-1-4503-0988-2.

[Lag11] R. Laganière. *OpenCV 2 Computer Vision Application Programming Cookbook*. recommended: for OpenCV support. Packt Pub Limited, 2011. ISBN: 1849513244.

[Low99] D. G. Lowe. "Object Recognition from Local Scale-Invariant Features". In: *ICCV '99 Proceedings of the International Conference on Computer Vision*. Vol. 2. Corfu, Greece: IEEE Computer Society Washington, 1999, pp. 1150–1157. ISBN: 0-7695-0164-8.

[Lux09] M. Lux. "Caliph & Emir: MPEG-7 photo annotation and retrieval". In: *MM '09 Proceedings of the 17th ACM international conference on Multimedia*. Beijing, China: ACM Press, 2009, pp. 925–926. ISBN: 978-1-60558-608-3.

[Mar11] N. Marquardt. "Proxemic interactions in ubiquitous computing ecologies". In: *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA '11*. ACM Press, 2011, p. 1033. ISBN: 9781450302685. DOI: 10.1145/1979742.1979691. URL: http://portal.acm.org/citation.cfm?doid=1979742.1979691.

[Mpe] *MPEG-7 overview (version 10)*. http://mpeg.chiariglione.org/standards/mpeg-7. Last Access: January 2013. 2004.

[OMAHT09] M. Obrist, C. Moser, D. Alliez, T. Holocher, and M. Tscheligi. "Connecting TV & PC: an In-Situ Field Evaluation of an Unified Electronic Program Guide Concept". In: *EuroITV '09 Proceedings of the seventh European conference on European interactive television conference*. Leuven, Belgium: ACM Press, 2009, pp. 91–100. ISBN: 978-1-60558-340-2.

[Ocv] *OpenCV*. http://opencv.org/. Last Access: September 2013.

[Of] *openFrameworks, an open source C++ toolkit for creative coding*. http://www.openframeworks.cc/. Last Access: September 2013.

[Ofs] *openFrameworks application structure*. http://wiki.openframeworks.cc/index.php?title=OF_code_structure_%28image%29. Last Access: August 2013. 2008.

[RD06] E. Rosten and T. Drummond. "Machine learning for high-speed corner detection". In: *ECCV'06 Proceedings of the 9th European conference on Computer Vision*. Springer-Verlag, 2006, pp. 430–443. ISBN: 3-540-33832-2 978-3-540-33832-1.

[SJBdLVL12] J. B. dos Santos Junior, K. P. Braga, D. de Souza, J. P. L. Lopes, G. Vilela, and D. Lima. "Exploring the Interactivity For Citizenship Applications in the Brazilian Digital Television System". In: *EATIS '12: Proceedings of the 6th Conference of the Euro American Conference on Telematics and Information Systems*. Valencia, Spain: ACM Press, 2012, pp. 143–148. ISBN: 978-1-4503-1012-3.

[SHK03] R. Schroeter, J. Hunter, and D. Kosovic. "Vannotea: A collaborative video indexing, annotation and discussion system for broadband networks". In: *K-CAP 2003 Workshop on Knowledge Markup and Semantic Annotation*. Sanibel, FL, U.S.A.: ACM Press, 2003.

[Sze10] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.

[Val11] J. G. Valente. "Sistema multimodal para captura e anotação de vídeo". MSc in Computer Science. Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, 2011. URL: http://hdl.handle.net/10362/6633.

[WUMCKB11] D. Williams, M. F. Ursu, J. Meenowa, P. Cesar, I. Kegel, and K. Bergström. "Video mediated social interaction between groups: System requirements and technology challenges". In: *Telematics and Informatics* 28.4 (2011. Elsevier Limited), pp. 251 –270. ISSN: 0736-5853. DOI: 10.1016/j.tele.2010.11.001. URL: http://www.sciencedirect.com/science/article/pii/S0736585310000730.

[ZWLS12] T. Zimmermann, M. Weber, M. Liwicki, and D. Stricker. "CoVidA: pen-based collaborative video annotation". In: *VIGTA '12 Proceedings of the 1st International Workshop on Visual Interfaces for Ground Truth Collection in Computer Vision Applications*. Capri, Italy: ACM Press, 2012. ISBN: 978-1-4503-1405-3.

Questionnaire

The model of the questionnaire used for the evaluation is presented below.

Questionnaire – Television Content Sharing and Annotation

A. User Data

Age: ____

Gender: M [ ] F [ ]

B. User's Past Experience

I. Are you comfortable with new technologies?

Yes [ ] No [ ]

II. Have you ever tried drawing oriented interfaces?

Yes [ ] No [ ]

III. If **yes**, in which device?

1. Tablet [ ]
2. Smartphone [ ]
3. Interactive Table [ ]
4. Interactive Board [ ]
5. Other: _____

C. Application

I. Usability

1. The application is easy to interact with.

   Completely disagree 1 2 3 4 5 Completely agree

2. The application is easy to learn.

   Completely disagree 1 2 3 4 5 Completely agree

3. The application responds quickly.

   Completely disagree 1 2 3 4 5 Completely agree

4. In the mobile application, the content is well organized.

Completely disagree 1 2 3 4 5 Completely agree

5. In the mobile application, the transition between sections is fluid.

Completely disagree 1 2 3 4 5 Completely agree

6. The buttons have an adequate size.

Completely disagree 1 2 3 4 5 Completely agree

II. Functionalities

1. Annotation

a) The tools are easily recognizable.

Completely disagree 1 2 3 4 5 Completely agree

b) The settings of each tool are easily accessible.

Completely disagree 1 2 3 4 5 Completely agree

c) There is a clear need to select the "save" button, so the annotations can be viewed in the player.

Completely disagree 1 2 3 4 5 Completely agree

d) The ability to attach images as annotations is clear.

Completely disagree 1 2 3 4 5 Completely agree

e) The possibility of getting the current frame of the video, for annotating (and share it on social networks), is clear.

Completely disagree 1 2 3 4 5 Completely agree

f) How do you assess the difficulty of drawing in the application with a pen?

Difficult 1 2 3 4 5 Easy

g) And without a pen?

Difficult 1 2 3 4 5 Easy

h) Satisfaction with this section.

Very dissatisfied 1 2 3 4 5 Very satisfied

2. *Playback* Mode

a) Being able to review my annotations is important.

Completely disagree 1 2 3 4 5 Completely agree

b) Being able to review other users' annotations is important.

Completely disagree 1 2 3 4 5 Completely agree

c) It is easy to distinguish my annotations from other users' annotations.

Completely disagree 1 2 3 4 5 Completely agree

d) It is easy to access the desired annotations/sessions in the mobile application.

Completely disagree 1 2 3 4 5 Completely agree

e) It is easy to access a specific annotation (text, ink, image), from the session previously chosen, in the player.

Completely disagree 1 2 3 4 5 Completely agree

f) Satisfaction with this section.

Very dissatisfied 1 2 3 4 5 Very satisfied

3. *Video Search*

a) The objective of this section is clear.

Completely disagree 1 2 3 4 5 Completely agree

b) Search for videos related to my pictures is useful.

Completely disagree 1 2 3 4 5 Completely agree

c) The waiting time for results is too long.

Completely disagree 1 2 3 4 5 Completely agree

d) The results are strongly related to the chosen image.

Completely disagree 1 2 3 4 5 Completely agree

e) The two different types of search are clear.

Completely disagree 1 2 3 4 5 Completely agree

f) The instructions on how to perform the video search are sufficient.

Completely disagree 1 2 3 4 5 Completely agree

g) Satisfaction with this section.

Very dissatisfied 1 2 3 4 5 Very satisfied

4. Player

a) The video occupies an adequate portion of the screen.

Completely disagree 1 2 3 4 5 Completely agree

b) The time that the annotations remain visible on the screen is appropriate.

Completely disagree 1 2 3 4 5 Completely agree

c) The functions of each button are easily identifiable.

Completely disagree 1 2 3 4 5 Completely agree

d) The texts are legible.

Completely disagree 1 2 3 4 5 Completely agree

e) The images are visible.

Completely disagree 1 2 3 4 5 Completely agree

f) The annotation markers on the timeline are visible (*playback* mode).

Completely disagree 1 2 3 4 5 Completely agree

g) The labels are appropriate (*playback* mode).

Completely disagree 1 2 3 4 5 Completely agree

h) Satisfaction with this section.

Very dissatisfied 1 2 3 4 5 Very satisfied

III. Overall Satisfaction

1. The application provides the necessary tools for video annotation.

   Completely disagree 1 2 3 4 5 Completely agree

2. Rate the application's utility.

   Low 1 2 3 4 5 High

3. Using the pen for annotating is essential.

   Completely disagree 1 2 3 4 5 Completely agree

4. The font size is (in the mobile application):

   Inadequate 1 2 3 4 5 Adequate

5. The font type is (in the mobile application):

   Inadequate 1 2 3 4 5 Adequate

6. Help is necessary for navigating in the application.

   Completely disagree 1 2 3 4 5 Completely agree

7. I recommend using this application.

   Completely disagree 1 2 3 4 5 Completely agree

8. Overall satisfaction with the application.

   Very dissatisfied 1 2 3 4 5 Very satisfied

Suggestions/Observations (optional):

Thank you!

B

Questionnaire Results from the Preliminary Tests

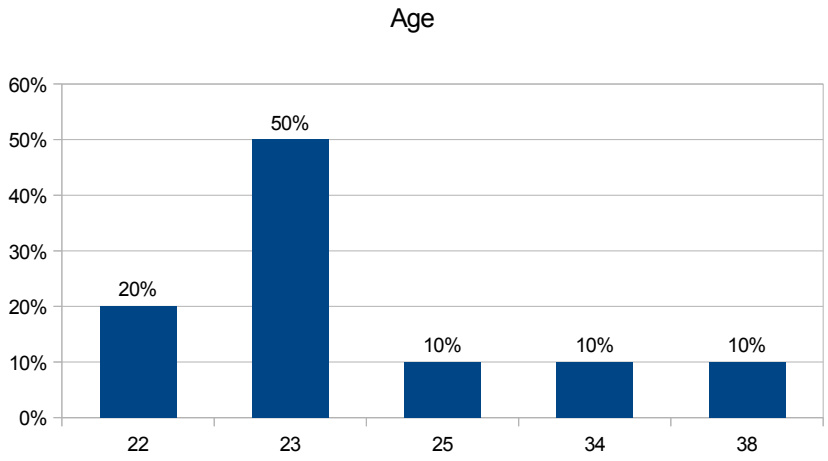

Figure B.1: Question A - Age

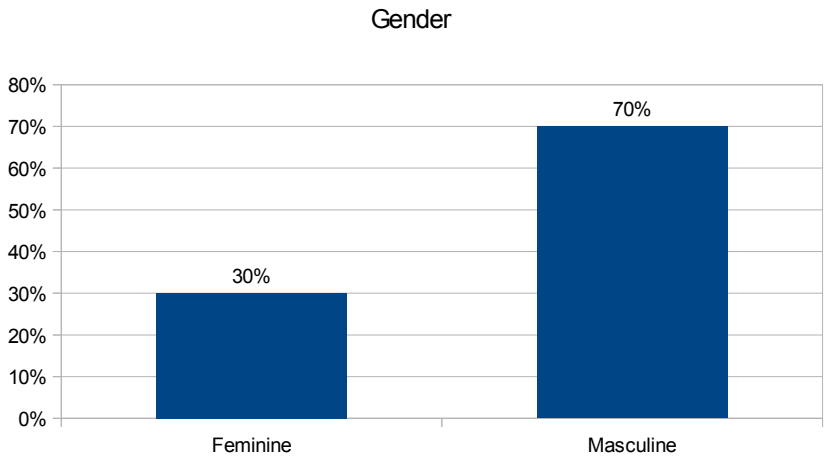

Figure B.2: Question A - Gender

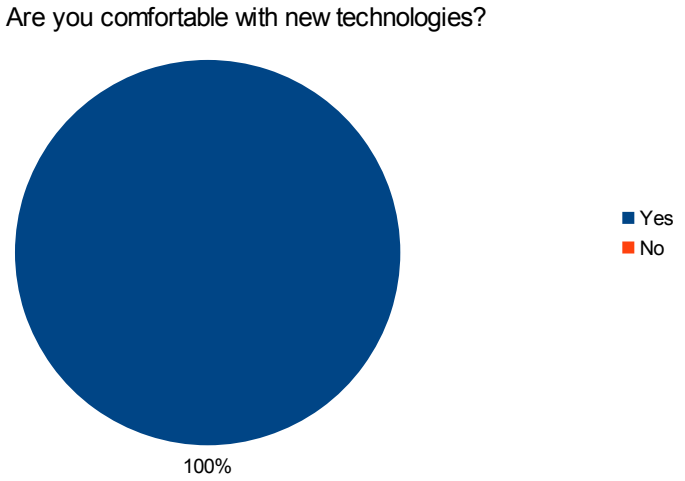

Figure B.3: Question BI: Are you comfortable with new technologies?

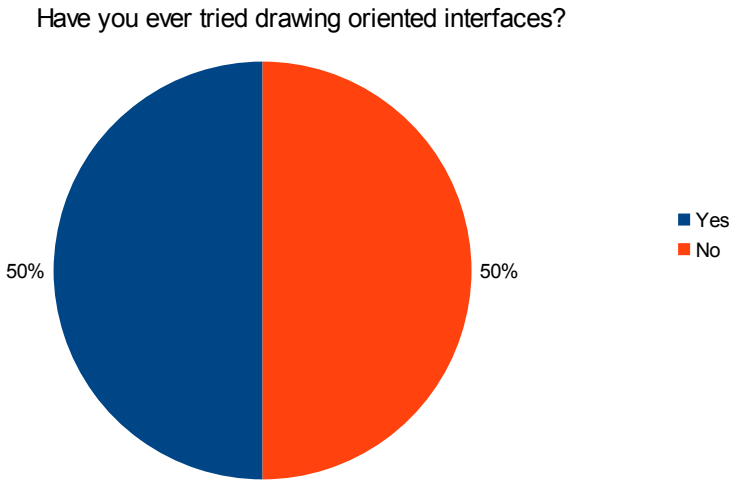

Figure B.4: Question BII: Have you ever tried drawing oriented interfaces?

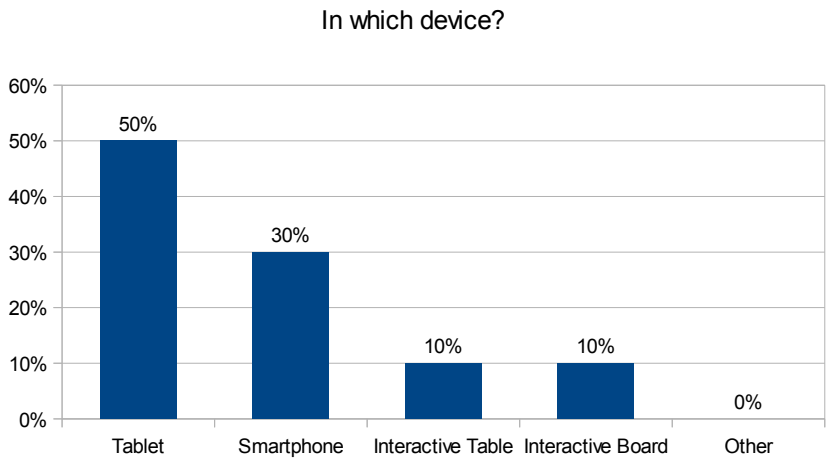

Figure B.5: Question BIII: If yes, in which device?

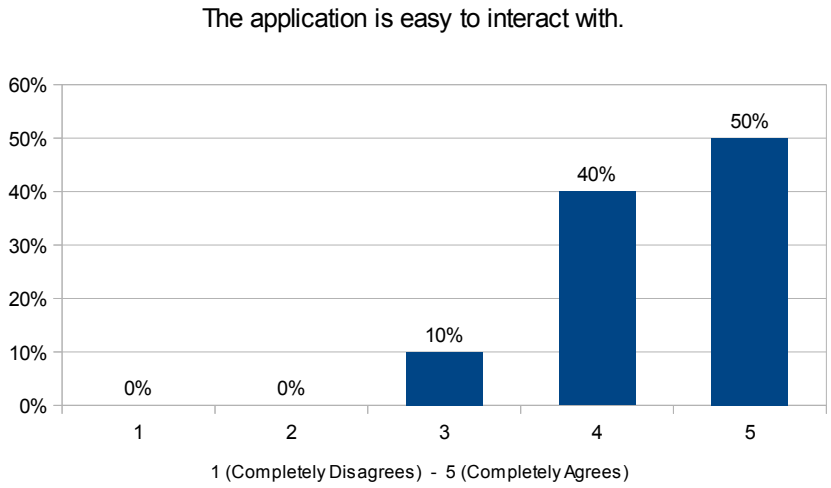

Figure B.6: Question CI1: The application is easy to interact with.

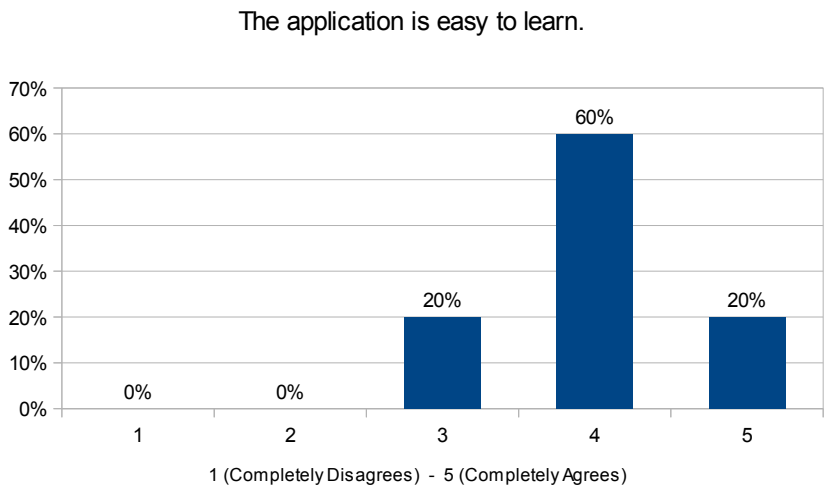

Figure B.7: Question CI2: The application is easy to learn.

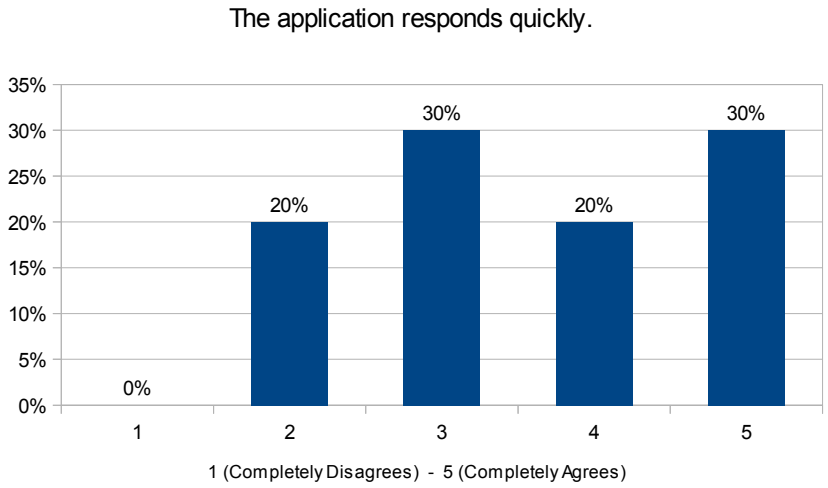

Figure B.8: Question CI3: The application responds quickly.

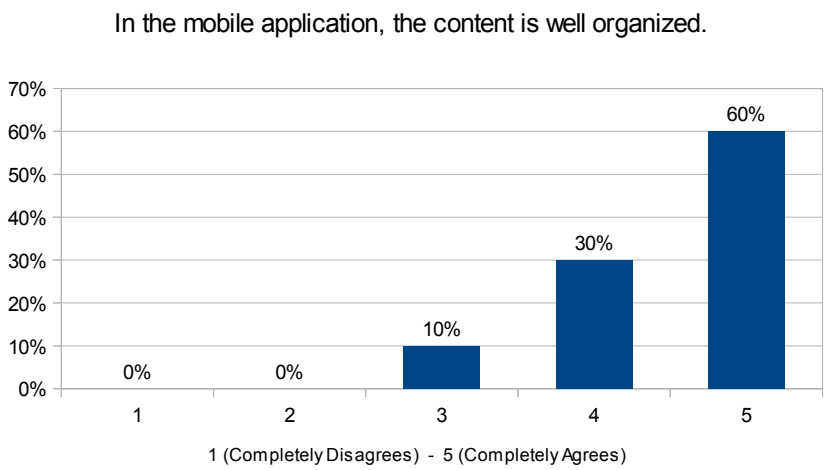

Figure B.9: Question CI4: In the mobile application, the content is well organized.

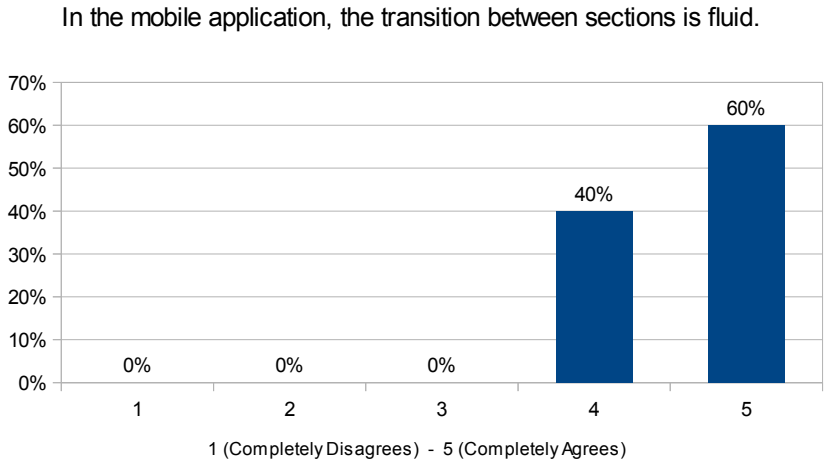

Figure B.10: Question CI5: In the mobile application, the transition between sections is fluid.

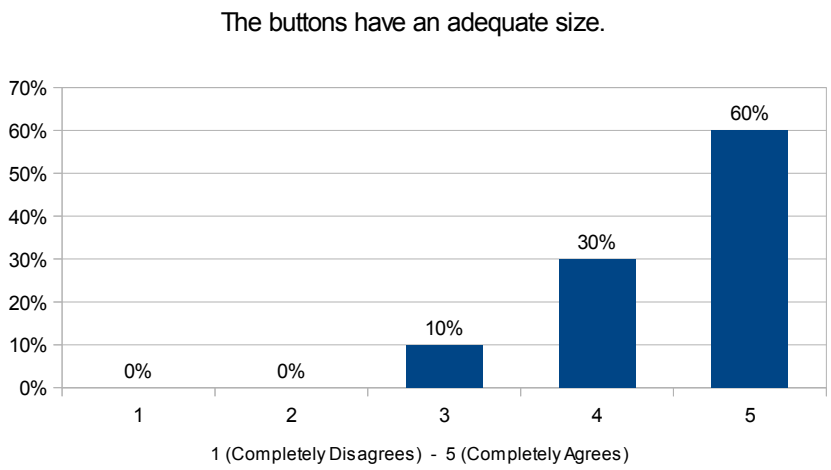

Figure B.11: Question CI6: The buttons have an adequate size.

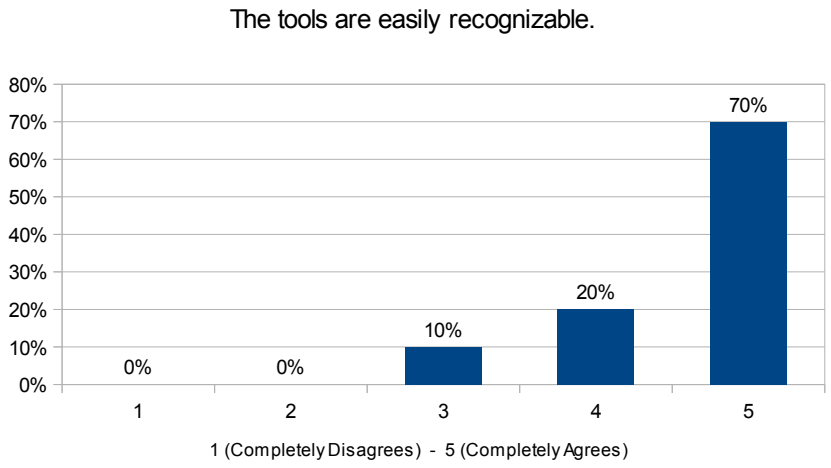

Figure B.12: Question CII 1a) : The tools are easily recognizable.

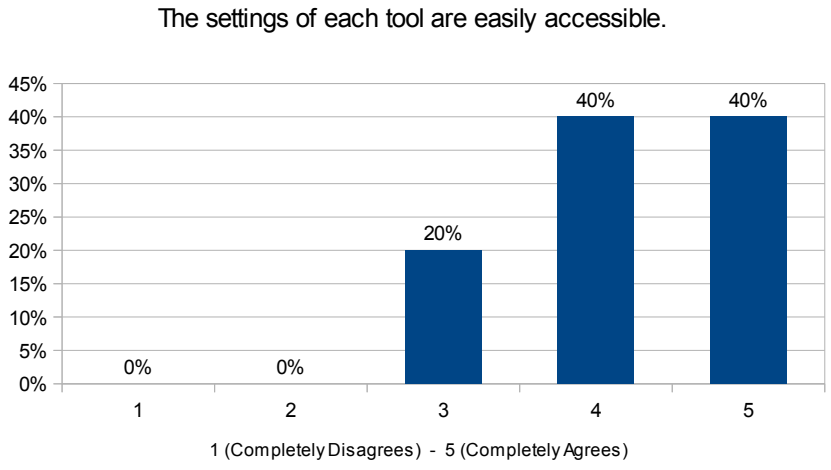

Figure B.13: Question CII 1b) : The settings of each tool are easily accessible.

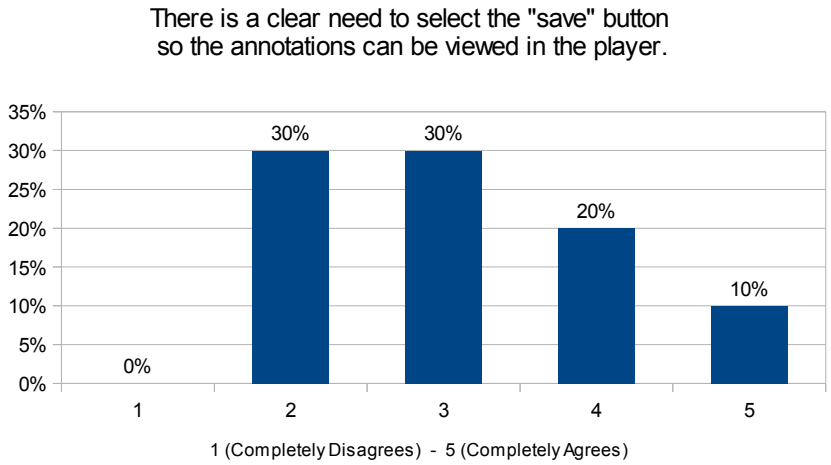

Figure B.14: Question CII 1c) : There is a clear need to select the "save" button, so the annotations can be viewed in the player.

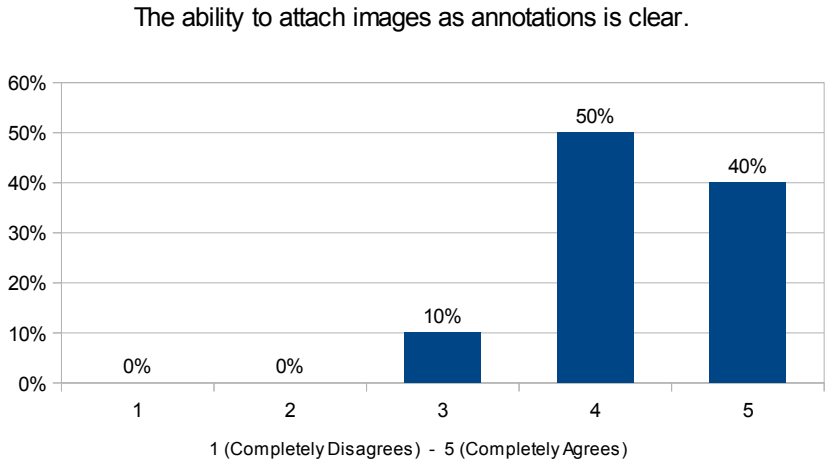

Figure B.15: Question CII 1d) : The ability to attach images as annotations is clear.

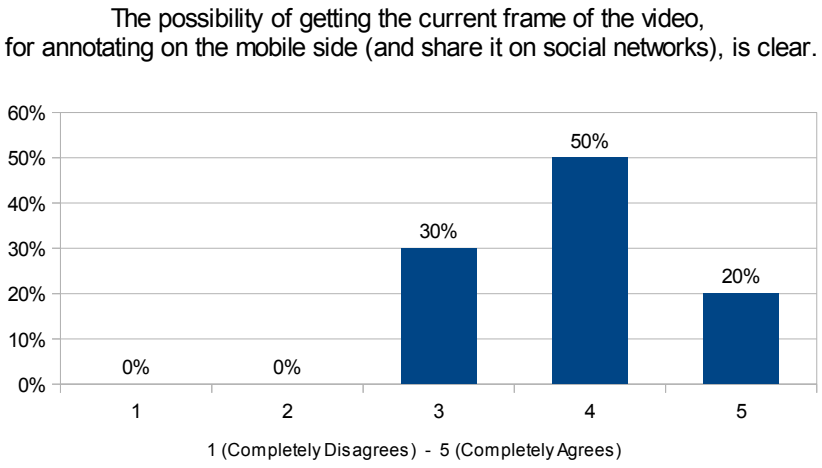

Figure B.16: Question CII 1e) : The possibility of getting the current frame of the video, for annotating (and share it on social networks), is clear.

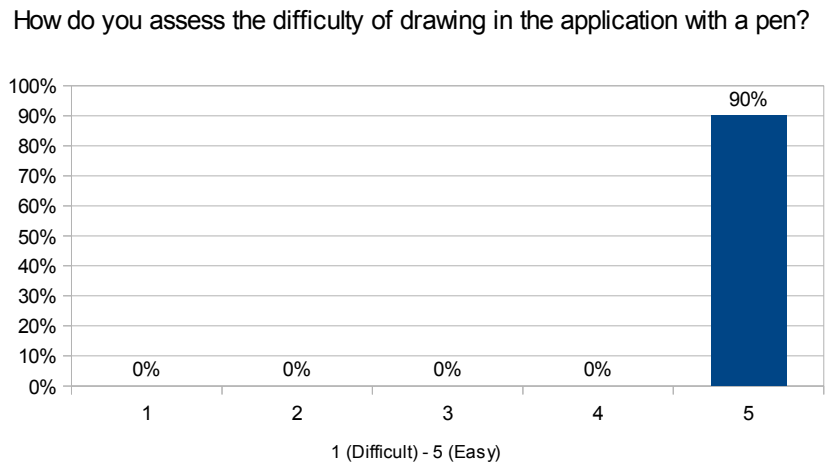

Figure B.17: Question CII 1f) : How do you assess the difficulty of drawing in the application with a pen?

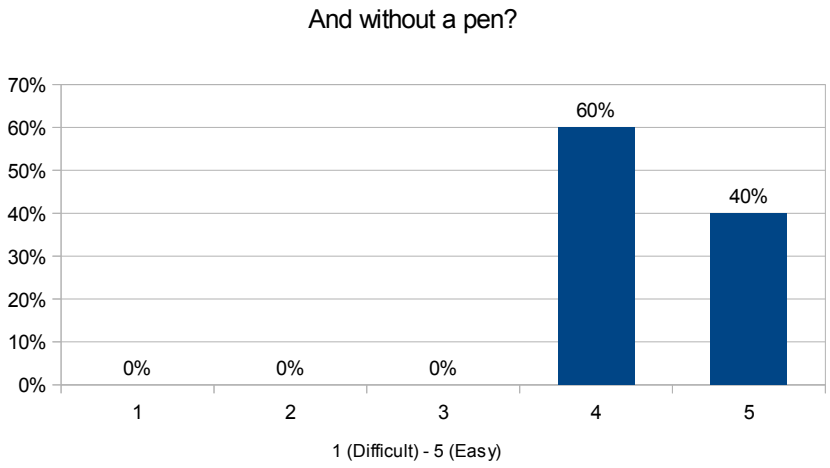

Figure B.18: Question CII 1g) : And without a pen?

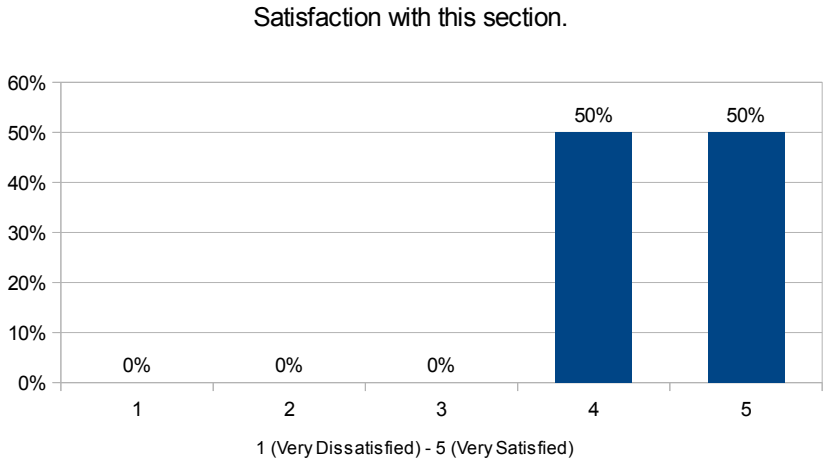

Figure B.19: Question CII 1h) : Satisfaction with this section.

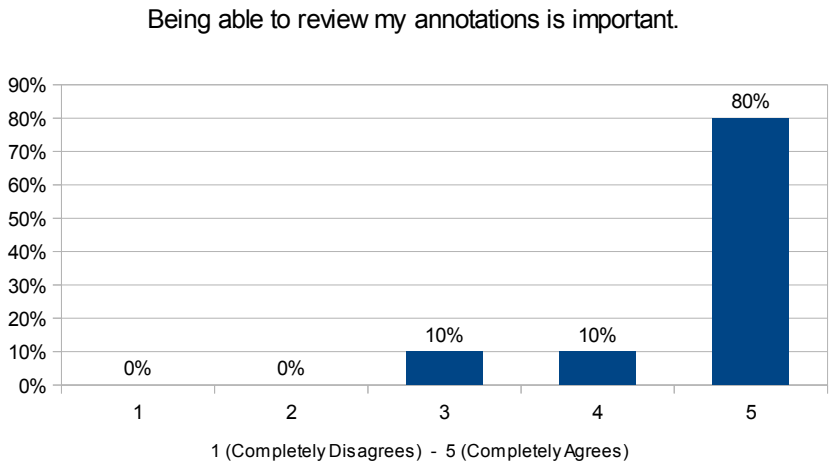

Figure B.20: Question CII 2a) : Being able to review my annotations is important.

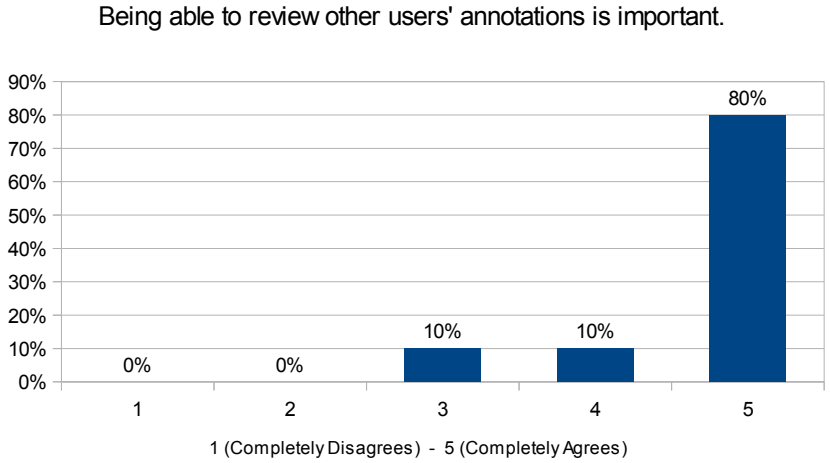

Figure B.21: Question CII 2b) : Being able to review other users' annotations is important.

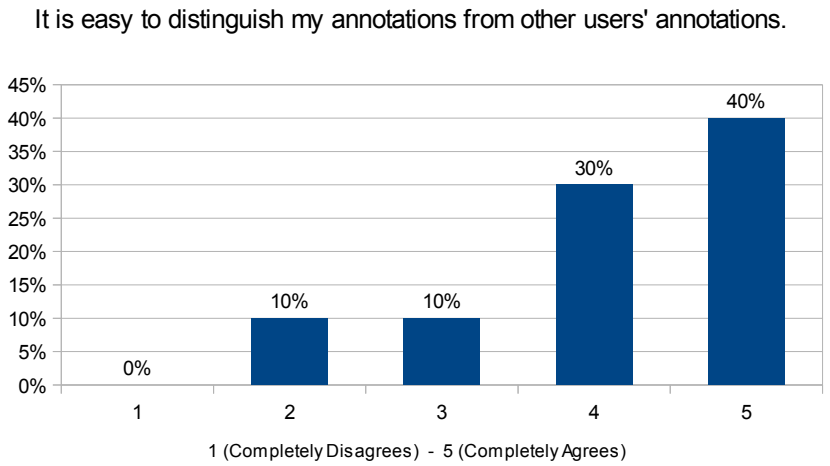

Figure B.22: Question CII 2c) : It is easy to distinguish my annotations from other users' annotations.

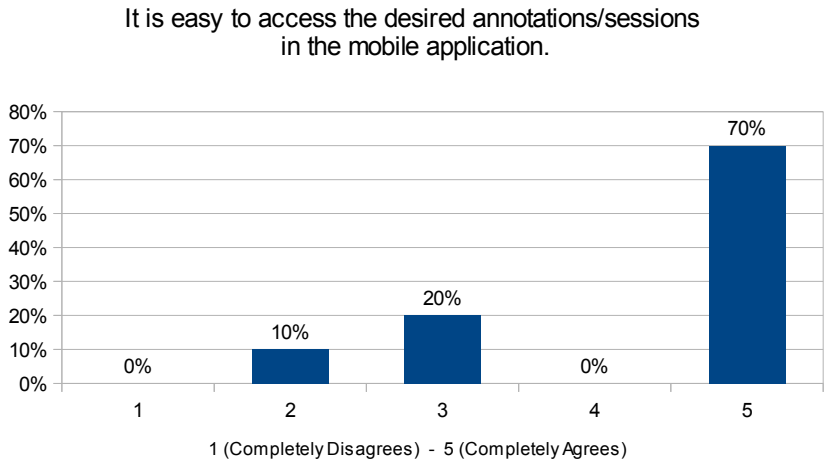

Figure B.23: Question CII 2d) : It is easy to access the desired annotations/sessions in the mobile application.

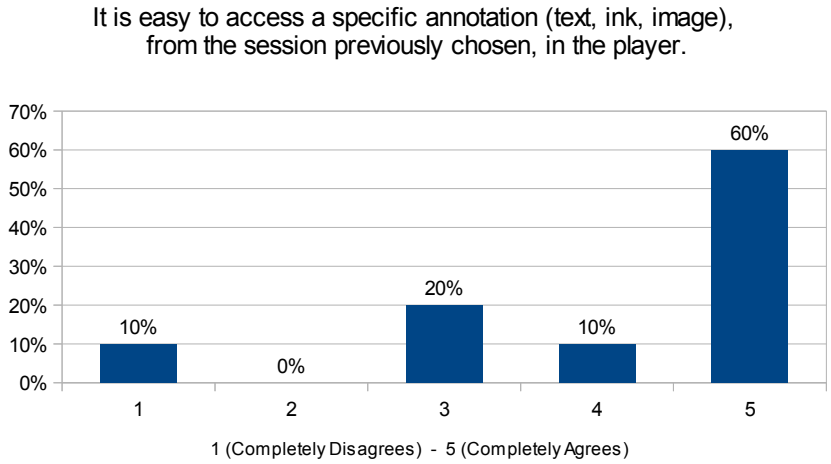

Figure B.24: Question CII 2e) : It is easy to access a specific annotation (text, ink, image), from the session previously chosen, in the player.

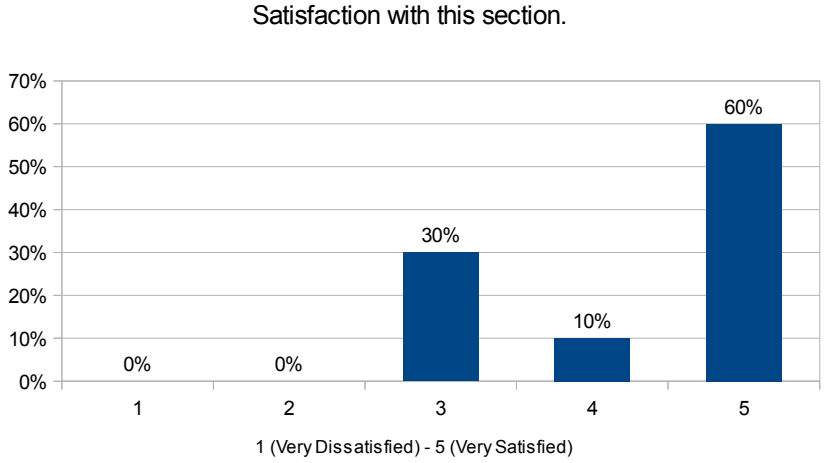

Figure B.25: Question CII 2f) : Satisfaction with this section.

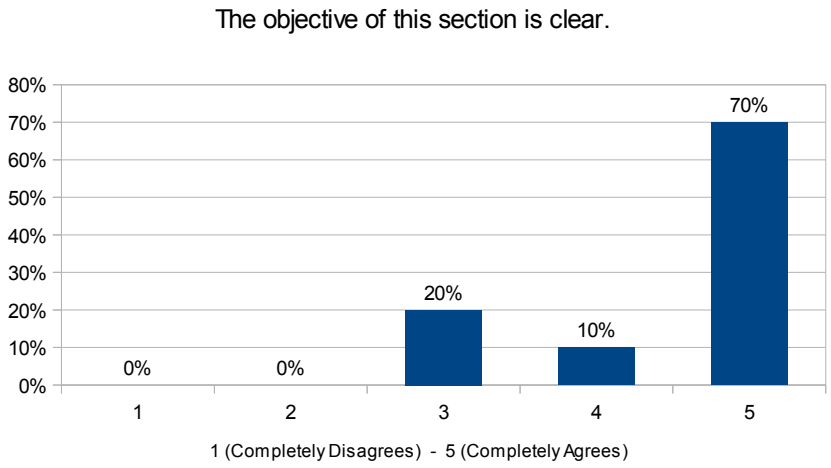

Figure B.26: Question CII 3a) : The objective of this section is clear.

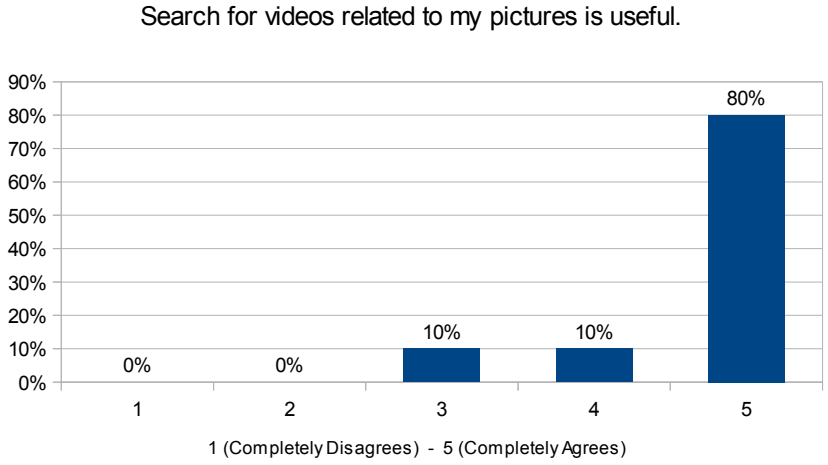

Figure B.27: Question CII 3b) : Search for videos related to my pictures is useful.

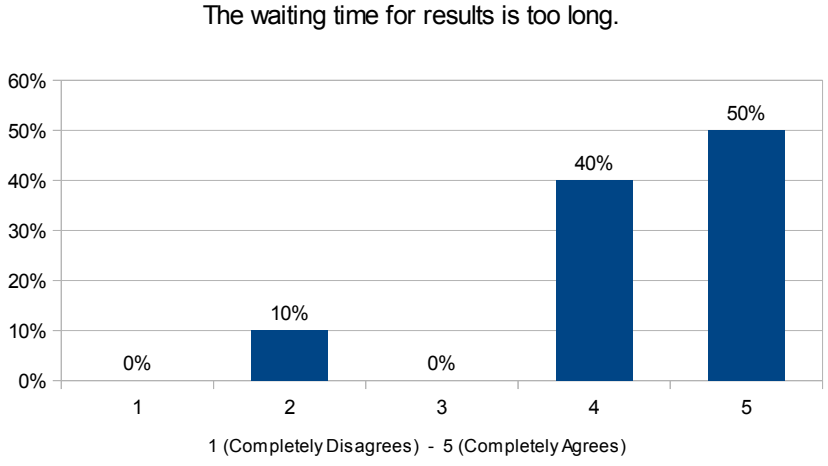

Figure B.28: Question CII 3c) : The waiting time for results is too long.

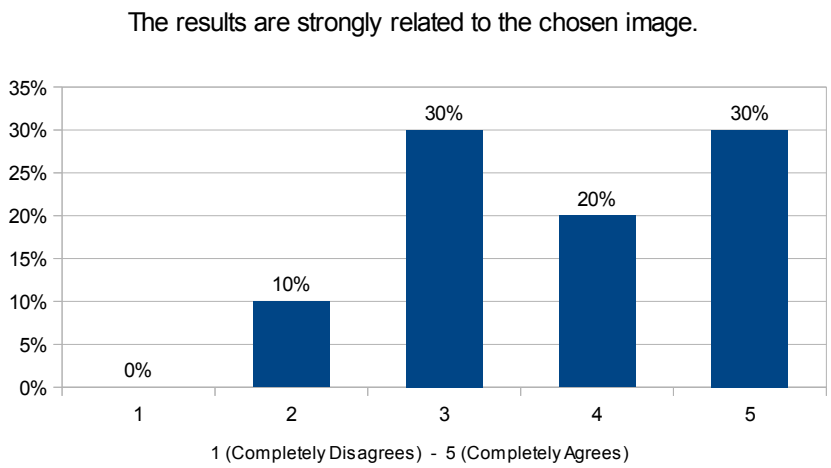

Figure B.29: Question CII 3d) : The results are strongly related to the chosen image.

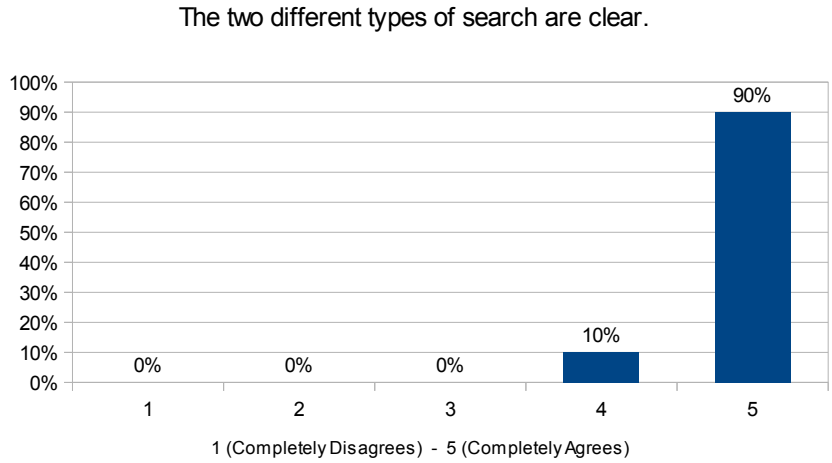

Figure B.30: Question CII 3e) : The two different types of search are clear.

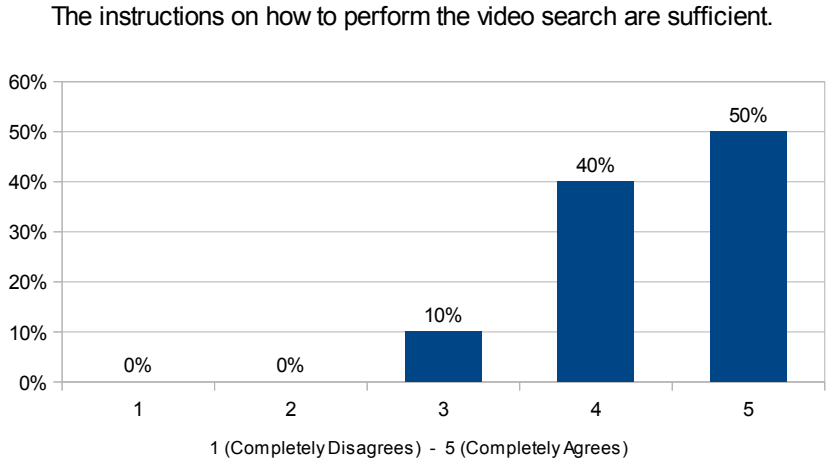

Figure B.31: Question CII 3f) : The instructions on how to perform the video search are sufficient.

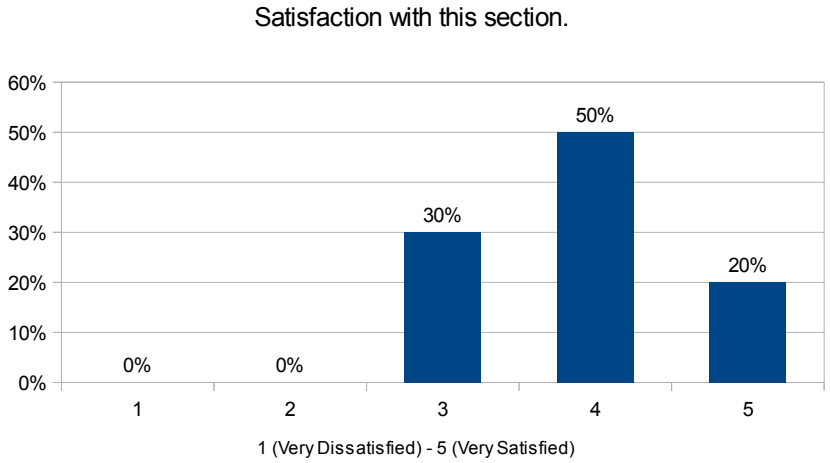

Figure B.32: Question CII 3g) : Satisfaction with this section.

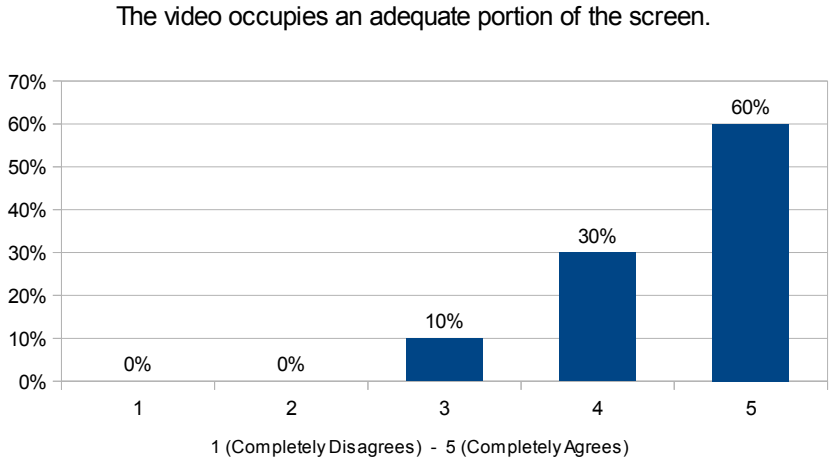

Figure B.33: Question CII 4a) : The video occupies an adequate portion of the screen.

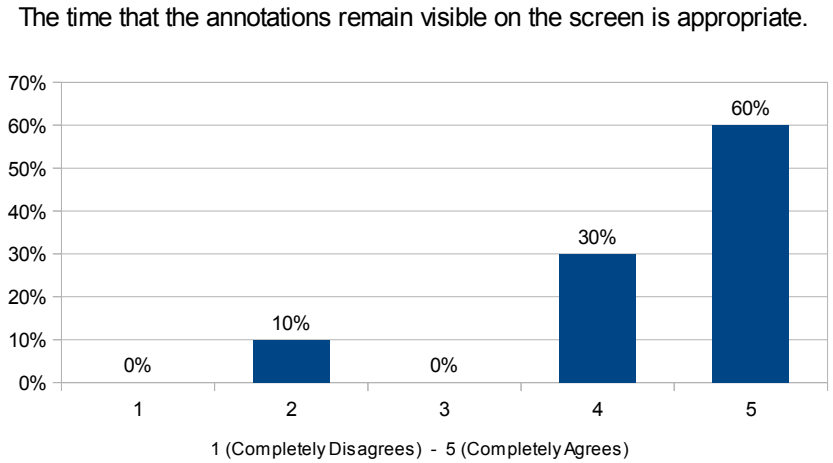

Figure B.34: Question CII 4b) : The time that the annotations remain visible on the screen is appropriate.

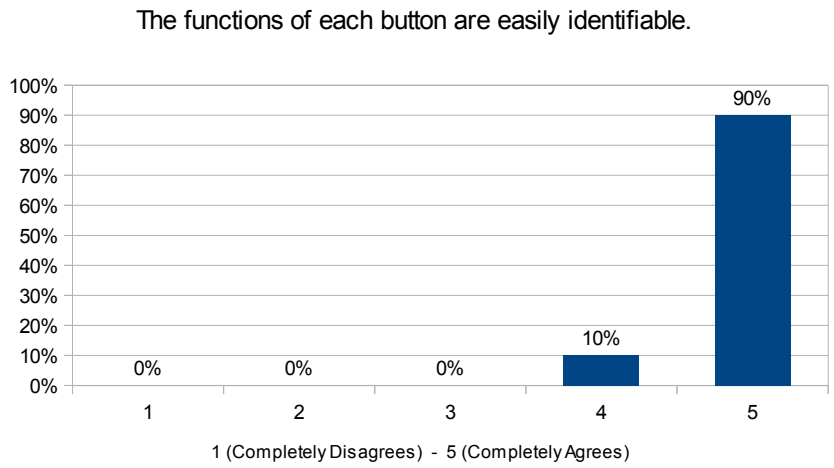

Figure B.35: Question CII 4c) : The functions of each button are easily identifiable.

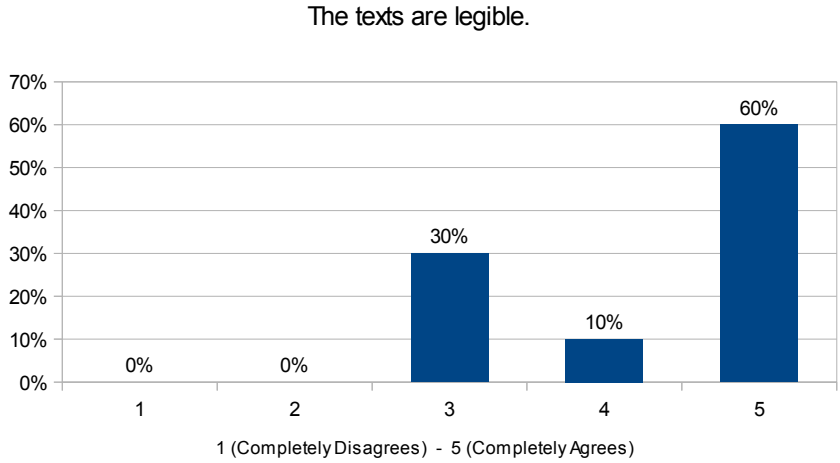

Figure B.36: Question CII 4d) : The texts are legible.

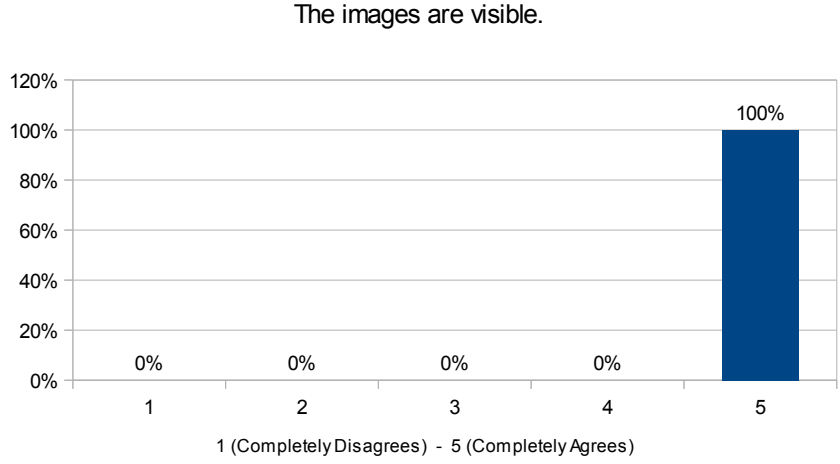

Figure B.37: Question CII 4e) : The images are visible.

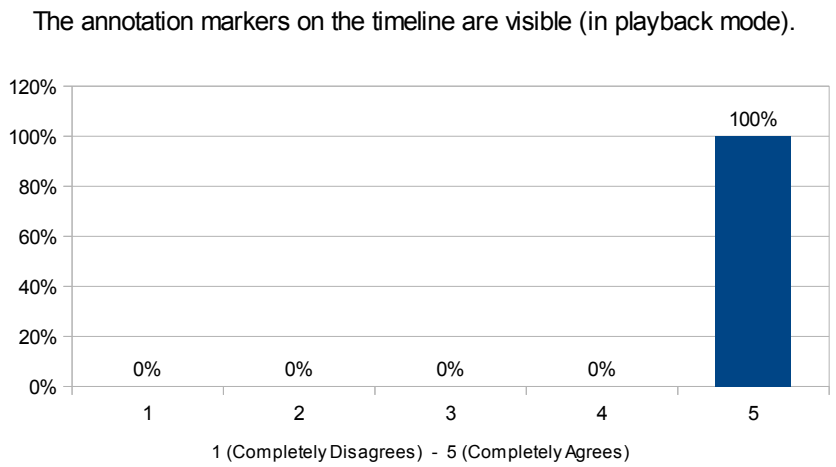

Figure B.38: Question CII 4f) : The annotation markers on the timeline are visible (playback mode).

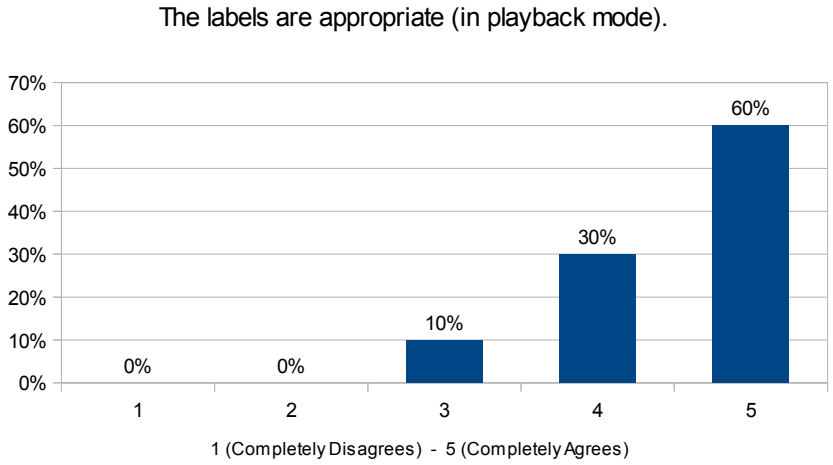

Figure B.39: Question CII 4g) : The labels are appropriate (playback mode).

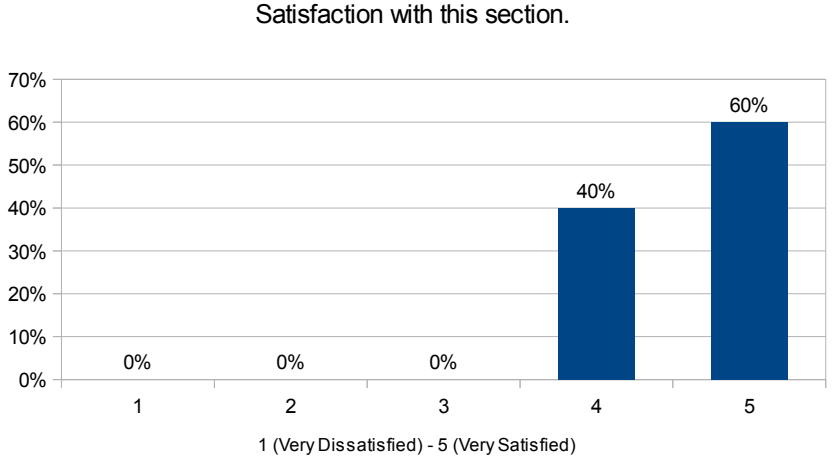

Figure B.40: Question CII 4h) : Satisfaction with this section.

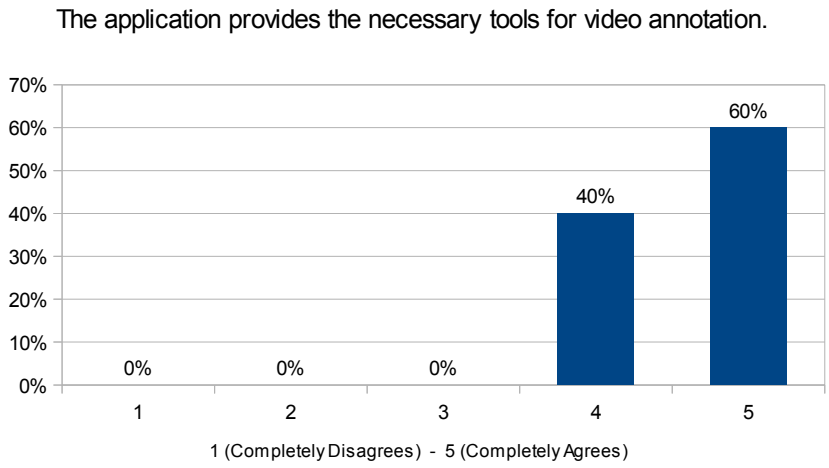

Figure B.41: Question CIII 1 : The application provides the necessary tools for video annotation.

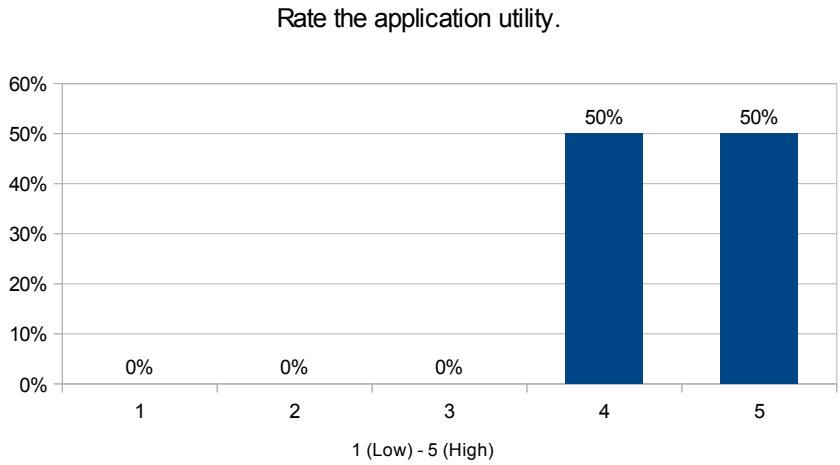

Figure B.42: Question CIII 2 : Rate the application's utility.

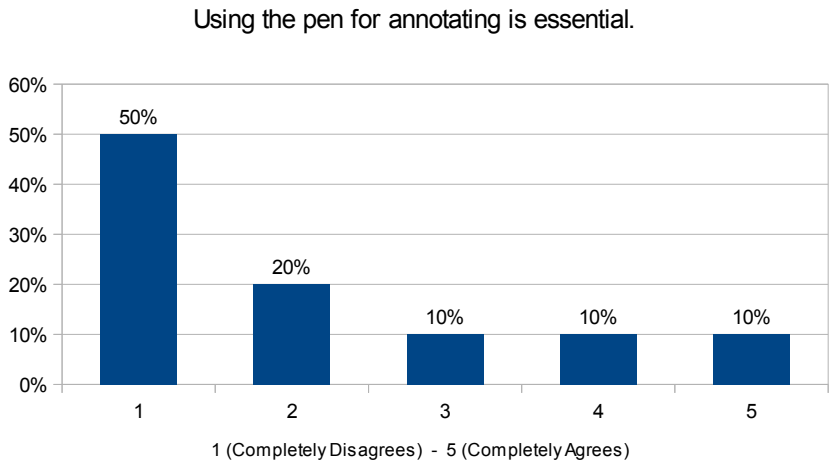

Figure B.43: Question CIII 3 : Using the pen for annotating is essential.

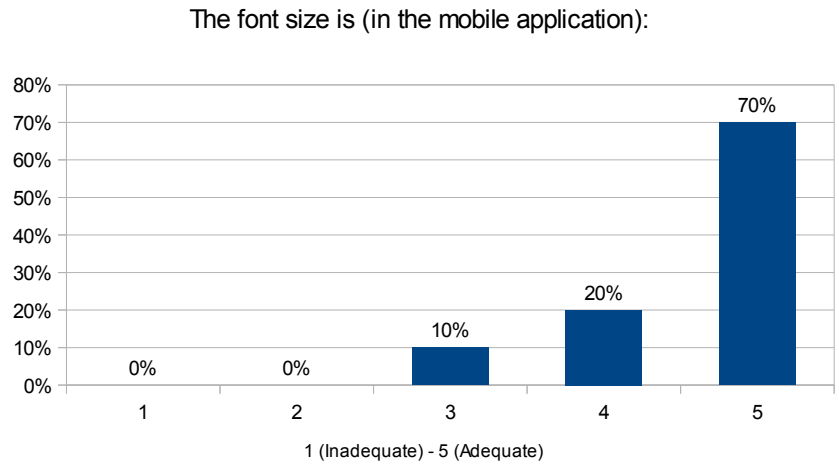

Figure B.44: Question CIII 4 : The font size is (in the mobile application)

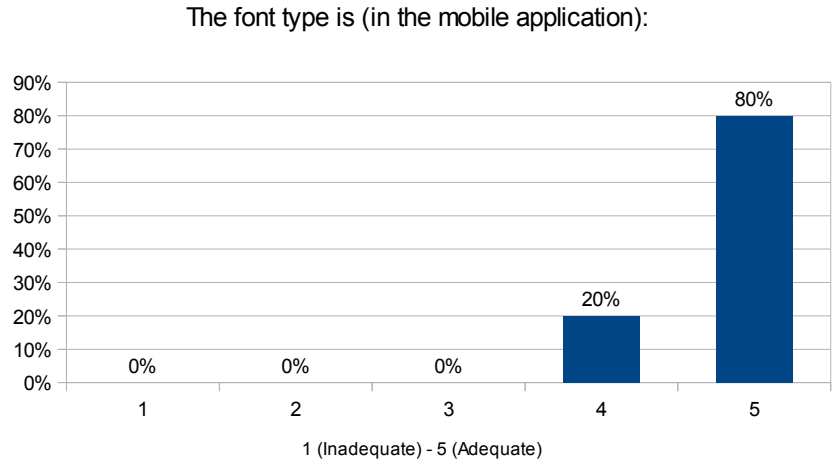

Figure B.45: Question CIII 5 : The font type is (in the mobile application)

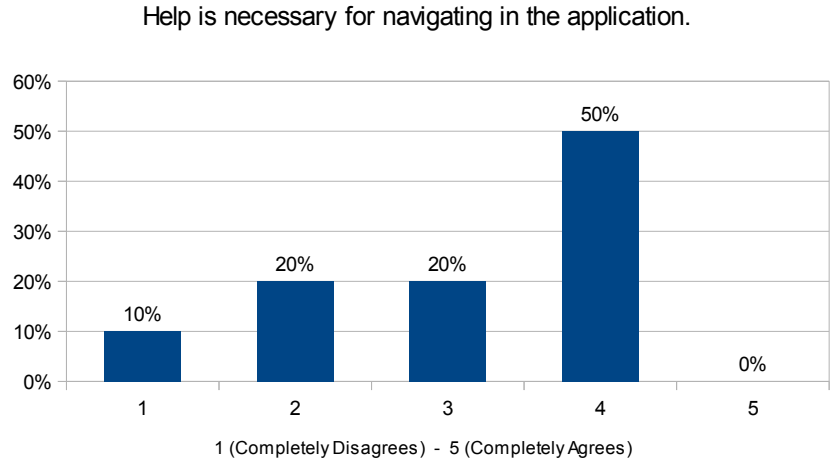

Figure B.46: Question CIII 6 : Help is necessary for navigating in the application.

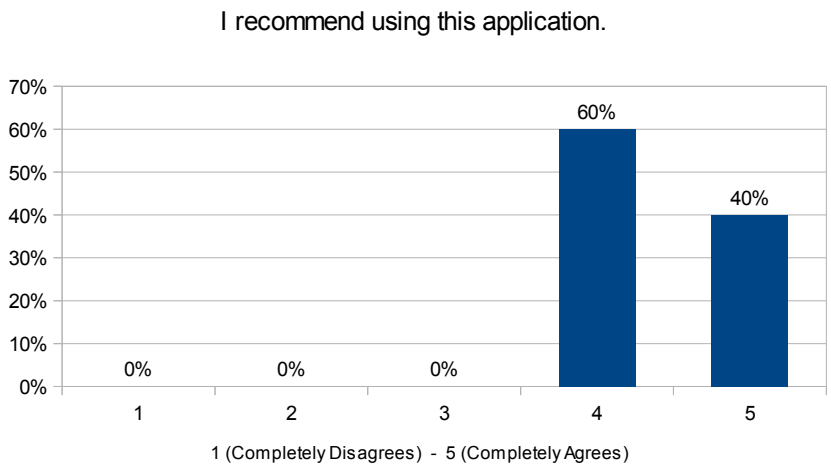

Figure B.47: Question CIII 7 : I recommend using this application.

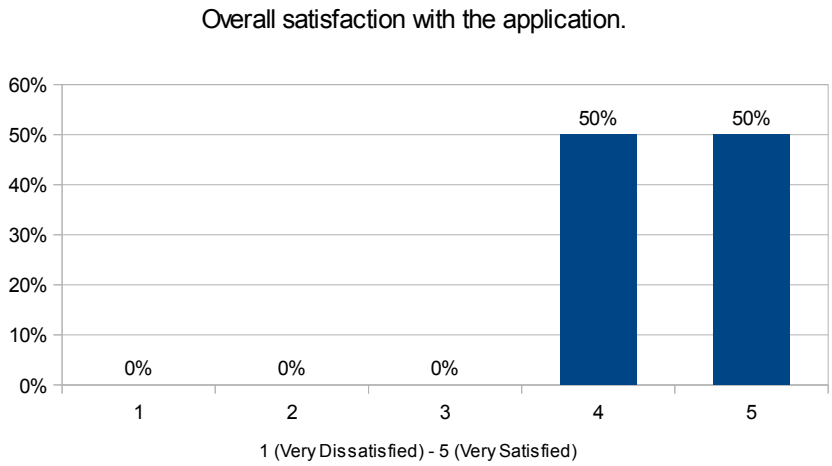

Figure B.48: Question CIII 8 : Overall satisfaction with the application.

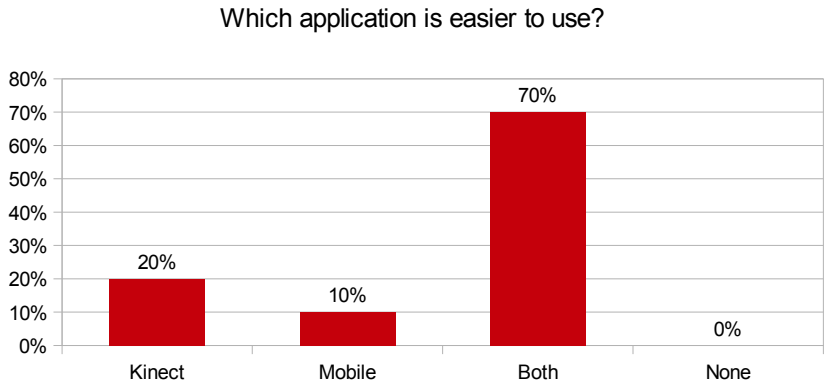

Figure B.49: Question Q26 (from the comparison questionnaire) : Which application is easier to use?

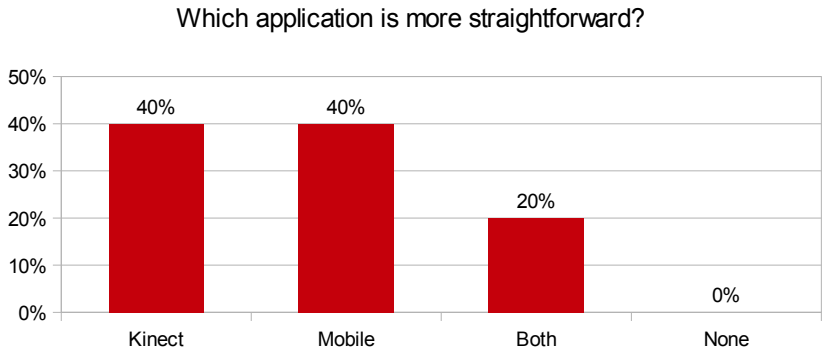

Figure B.50: Question Q27 (from the comparison questionnaire) : Which application is more straightforward?

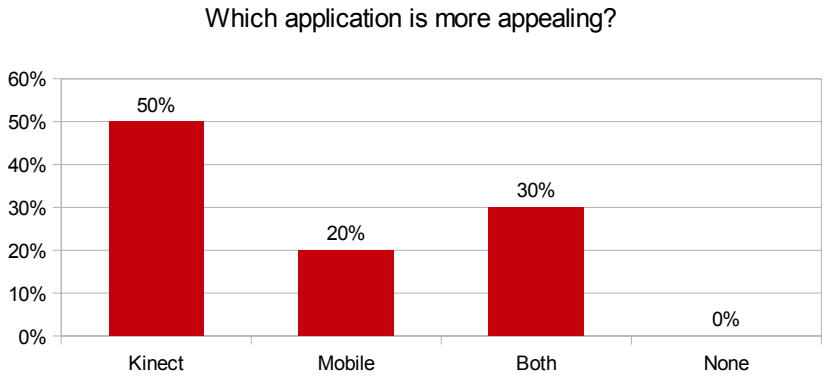

Figure B.51: Question Q28 (from the comparison questionnaire) : Which application is more appealing?

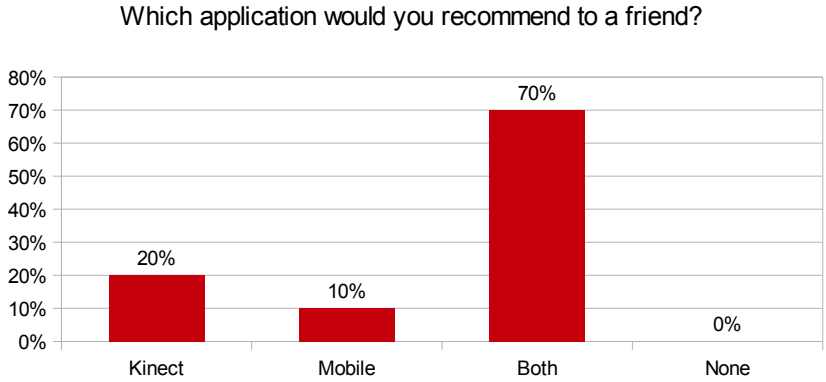

Figure B.52: Question Q29 (from the comparison questionnaire) : Which application would you recommend to a friend?

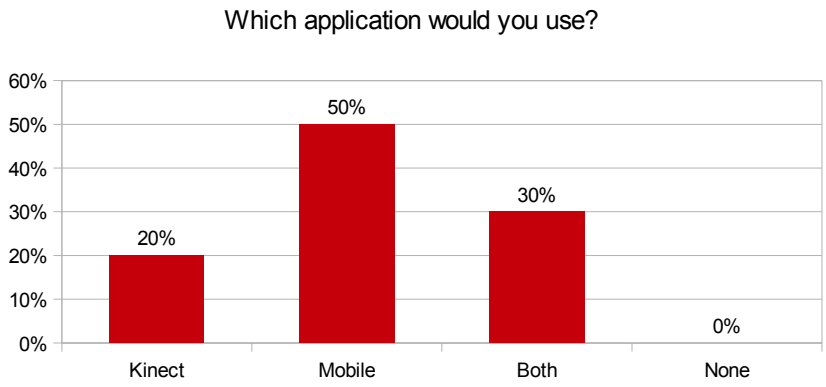

Figure B.53: Question Q30 (from the comparison questionnaire) : Which application would you use?

C

Questionnaire Results from the Final Tests

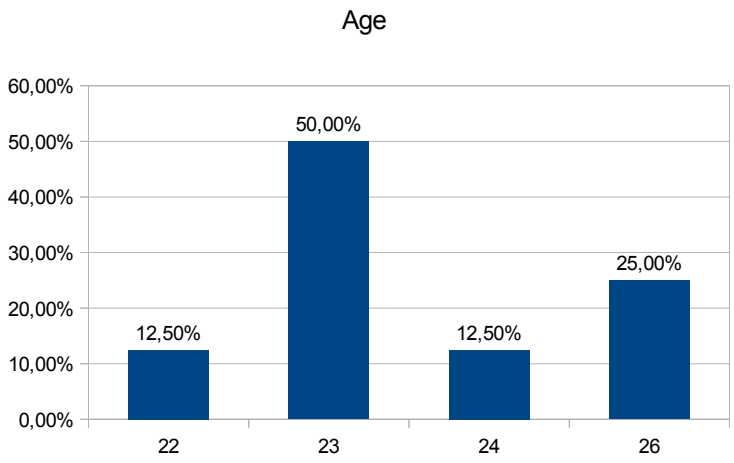

Figure C.1: Question A - Age

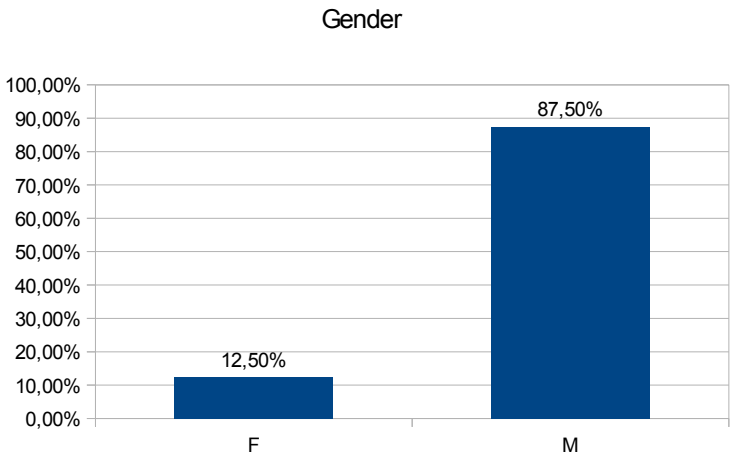

Figure C.2: Question A - Gender

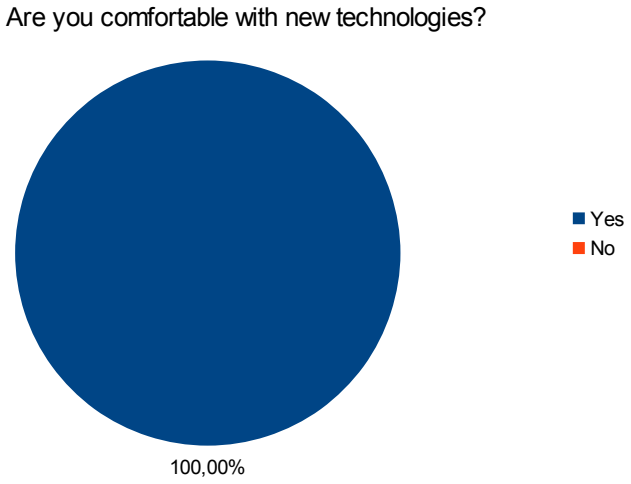

Figure C.3: Question BI: Are you comfortable with new technologies?

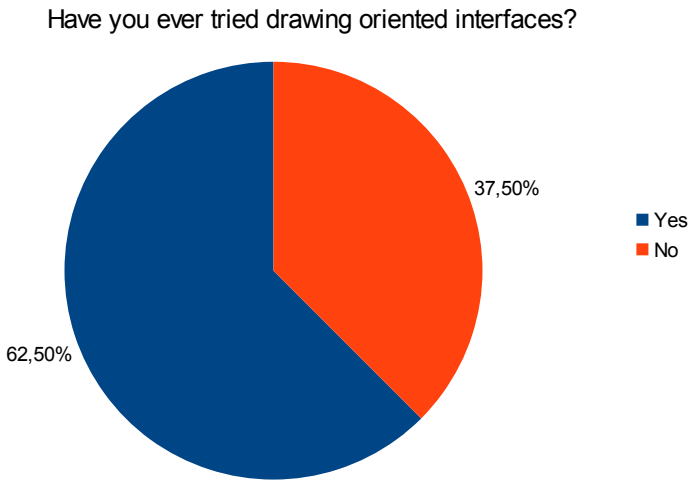

Figure C.4: Question BII: Have you ever tried drawing oriented interfaces?

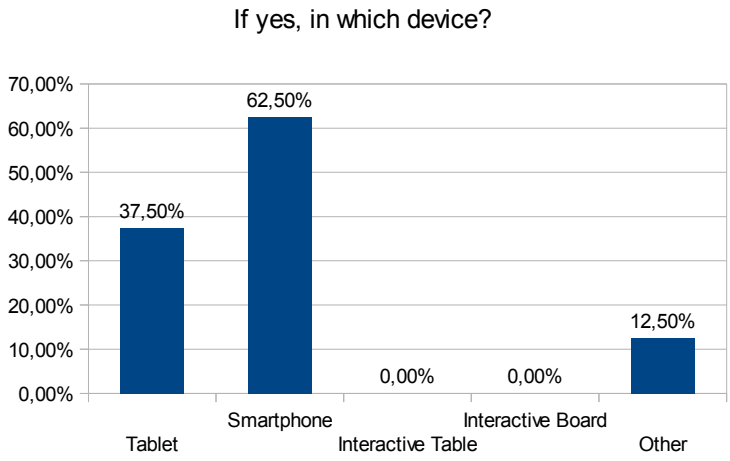

Figure C.5: Question BIII: If yes, in which device?

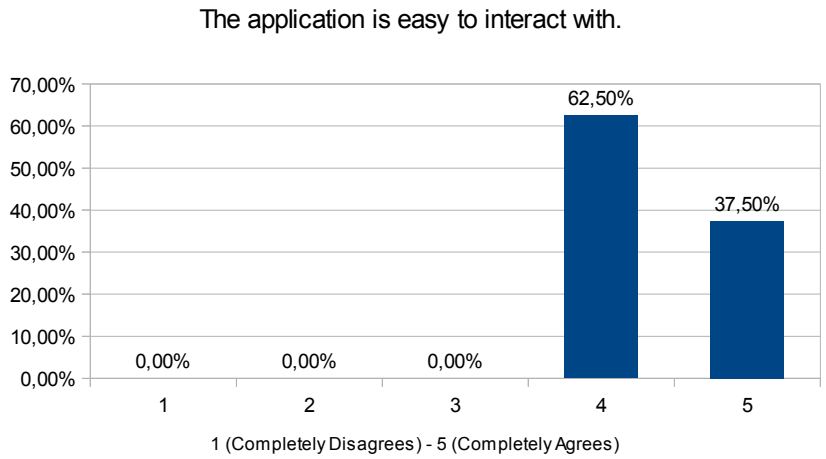

Figure C.6: Question CI1: The application is easy to interact with.

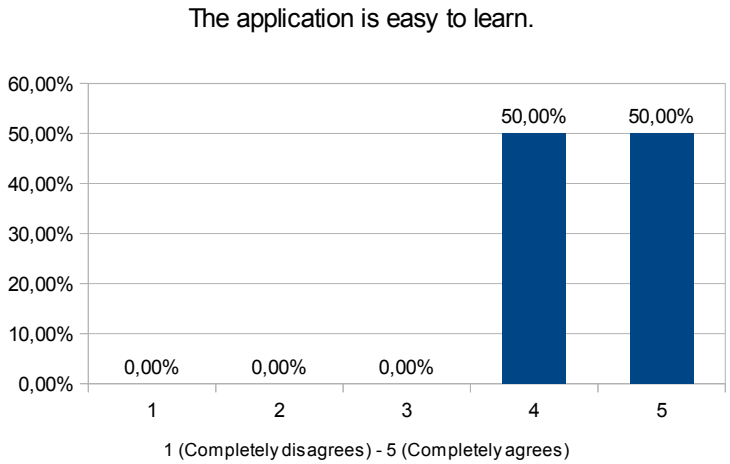

Figure C.7: Question CI2: The application is easy to learn.

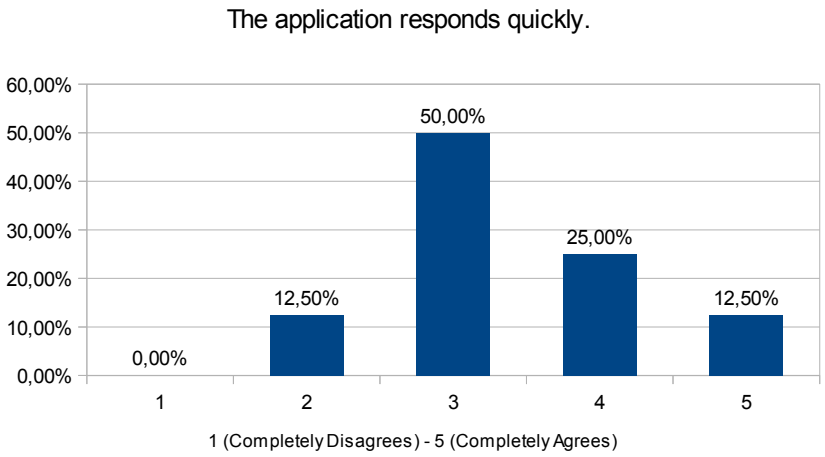

Figure C.8: Question CI3: The application responds quickly.

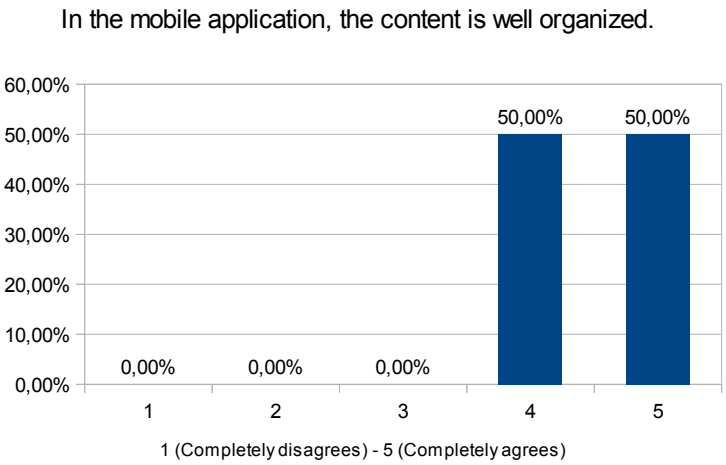

Figure C.9: Question CI4: In the mobile application, the content is well organized.

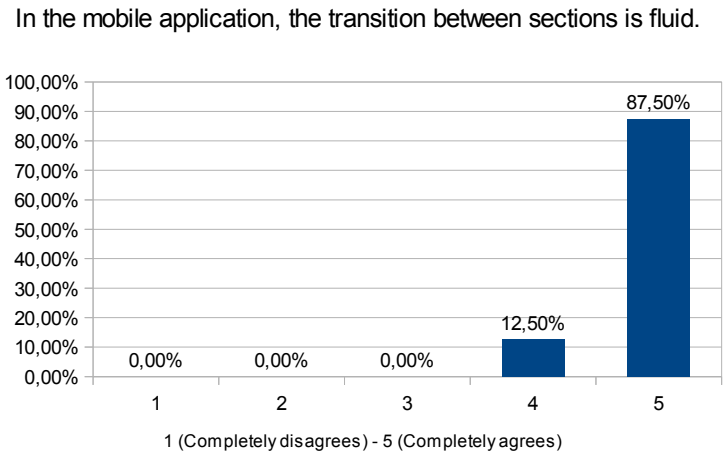

Figure C.10: Question CI5: In the mobile application, the transition between sections is fluid.

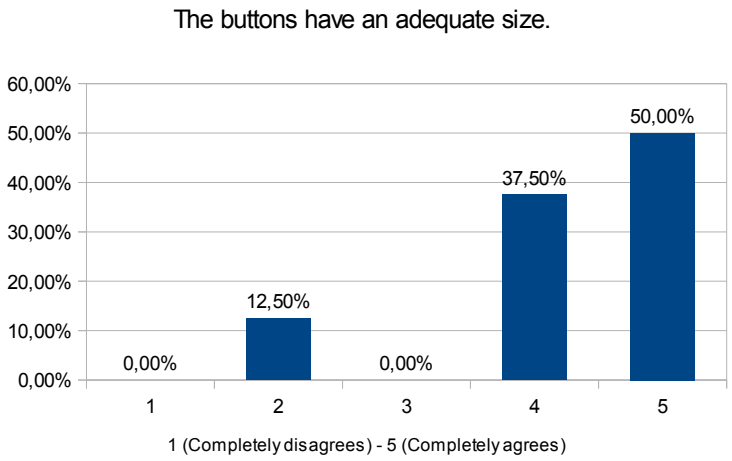

Figure C.11: Question CI6: The buttons have an adequate size.

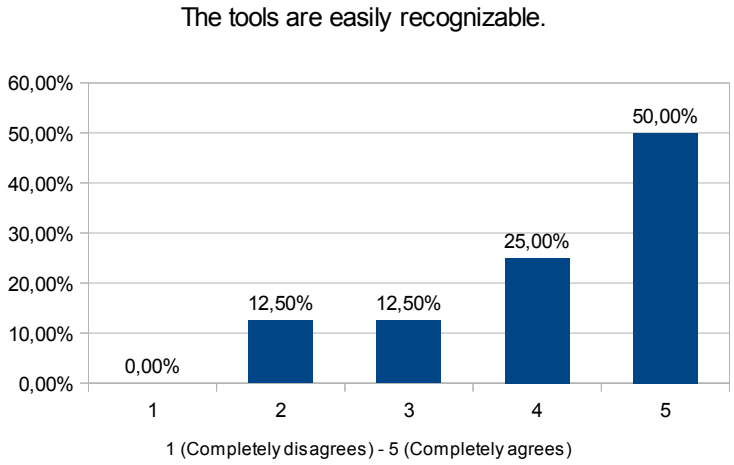

Figure C.12: Question CII 1a) : The tools are easily recognizable.

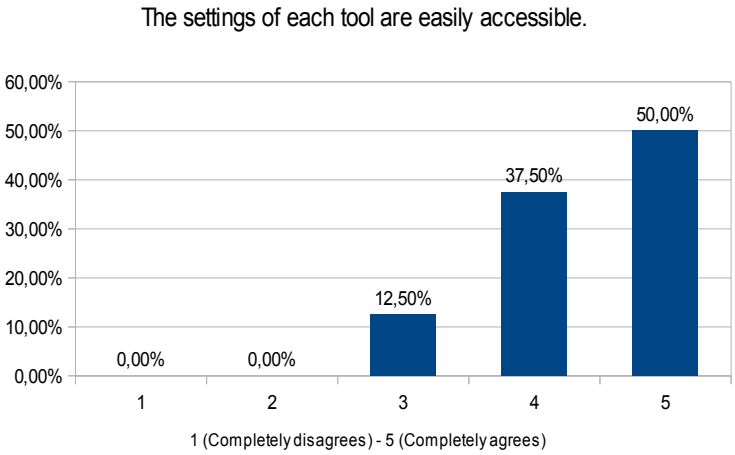

Figure C.13: Question CII 1b) : The settings of each tool are easily accessible.

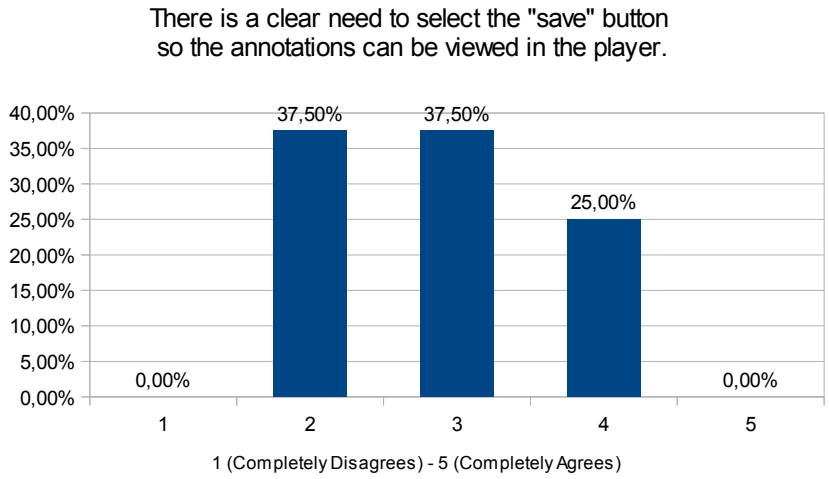

Figure C.14: Question CII 1c) : There is a clear need to select the "save" button, so the annotations can be viewed in the player.

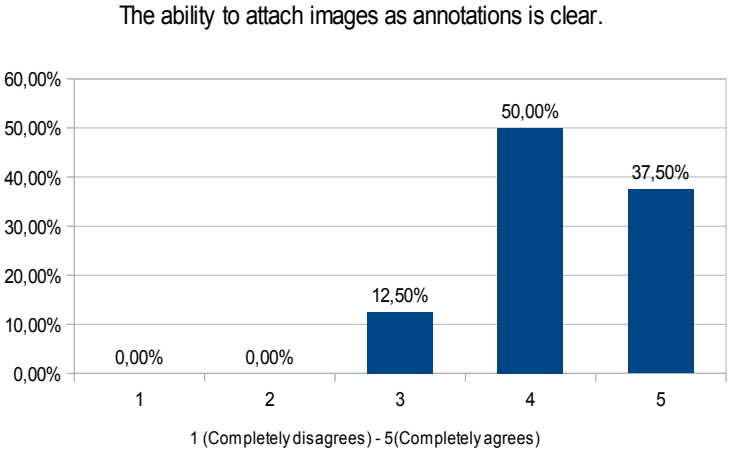

Figure C.15: Question CII 1d) : The ability to attach images as annotations is clear.

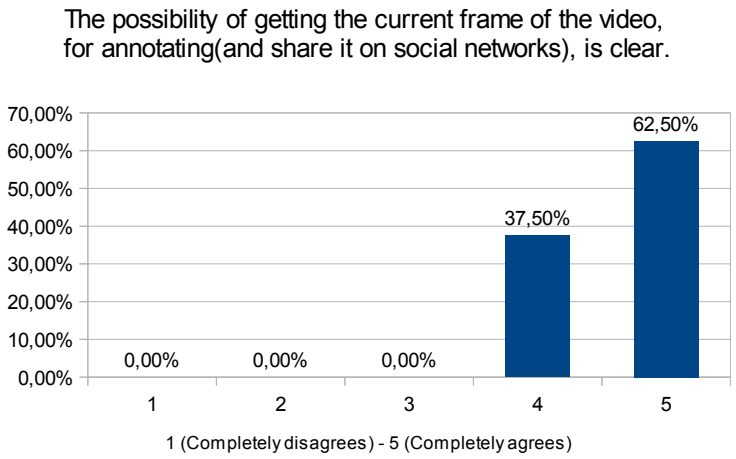

Figure C.16: Question CII 1e) : The possibility of getting the current frame of the video, for annotating (and share it on social networks), is clear.

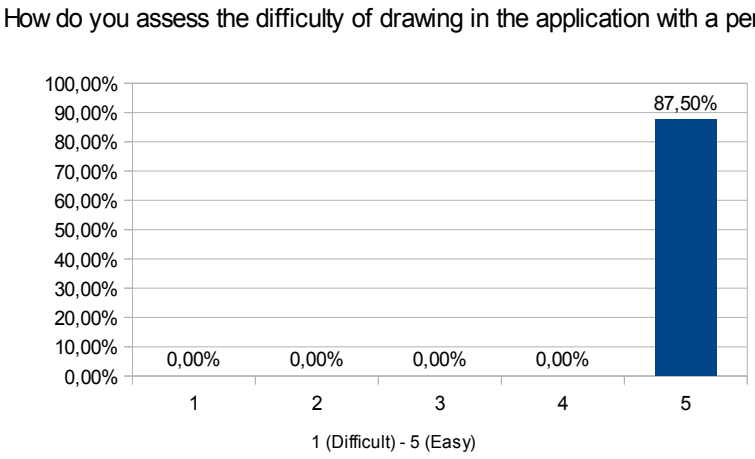

Figure C.17: Question CII 1f) : How do you assess the difficulty of drawing in the application with a pen?

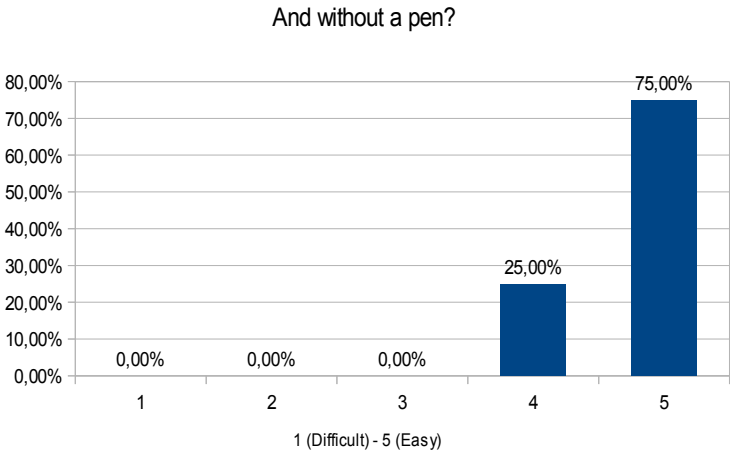

Figure C.18: Question CII 1g) : And without a pen?

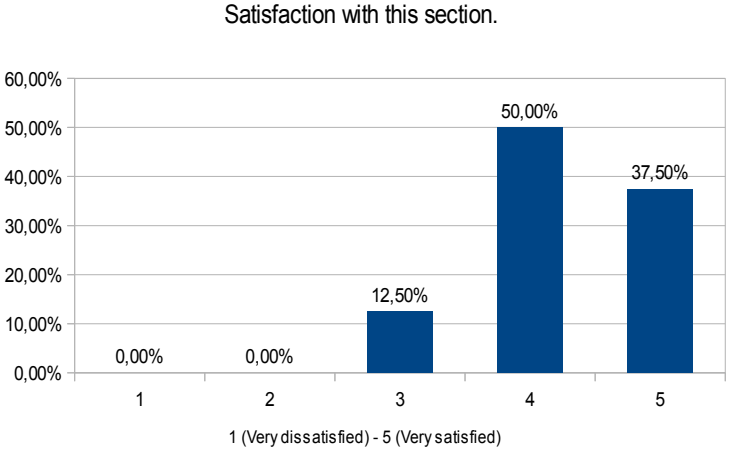

Figure C.19: Question CII 1h) : Satisfaction with this section.

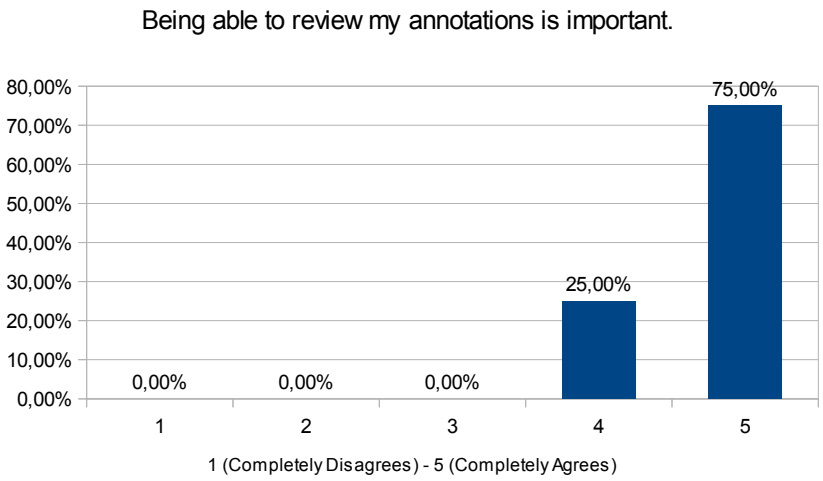

Figure C.20: Question CII 2a) : Being able to review my annotations is important.

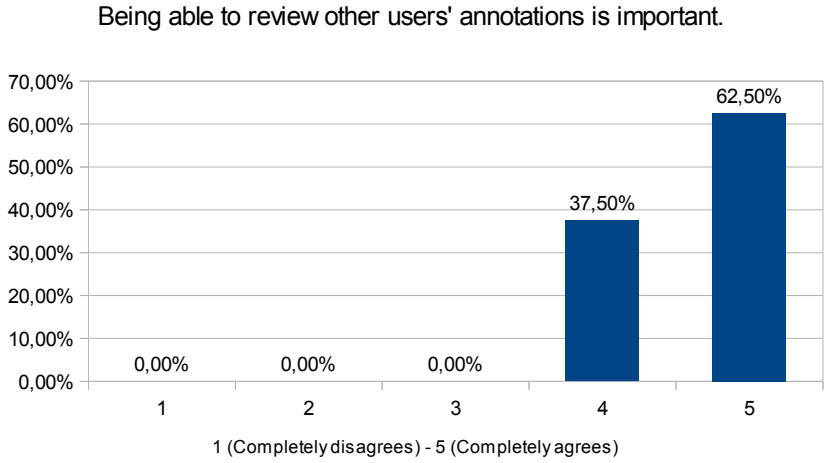

Figure C.21: Question CII 2b) : Being able to review other users' annotations is important.

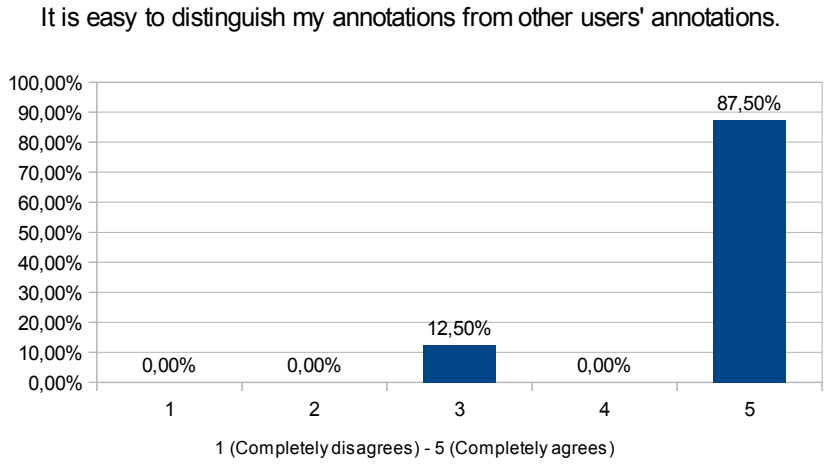

Figure C.22: Question CII 2c) : It is easy to distinguish my annotations from other users' annotations.

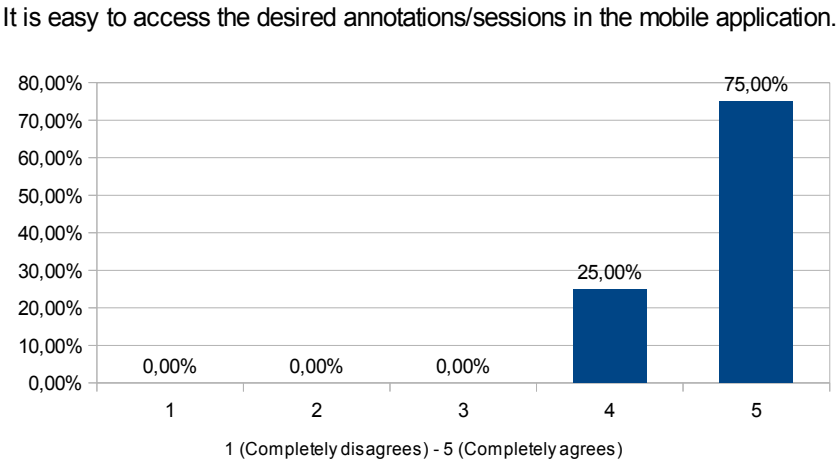

Figure C.23: Question CII 2d) : It is easy to access the desired annotations/sessions in the mobile application.

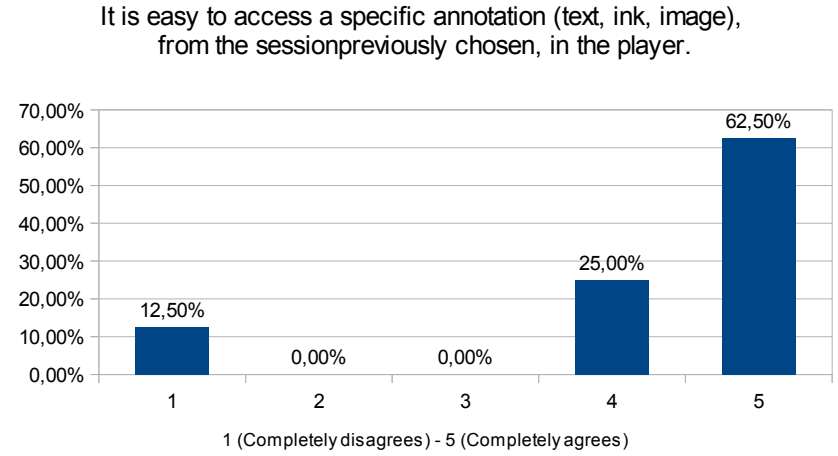

Figure C.24: Question CII 2e) : It is easy to access a specific annotation (text, ink, image), from the session previously chosen, in the player.

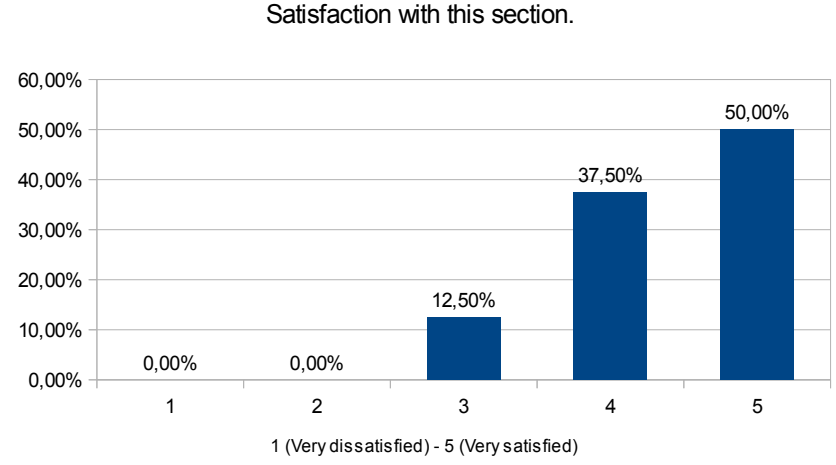

Figure C.25: Question CII 2f) : Satisfaction with this section.

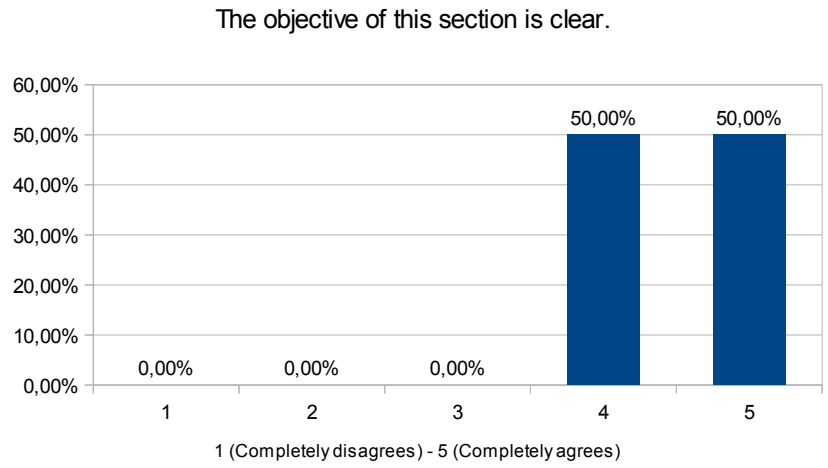

Figure C.26: Question CII 3a) : The objective of this section is clear.

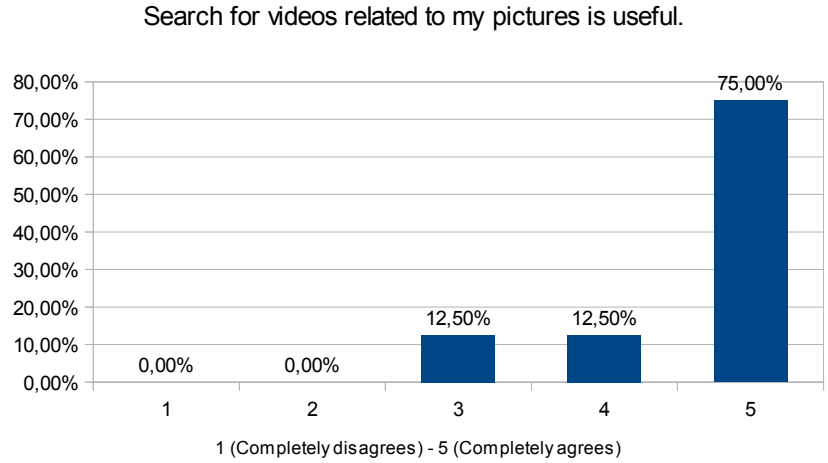

Figure C.27: Question CII 3b) : Search for videos related to my pictures is useful.

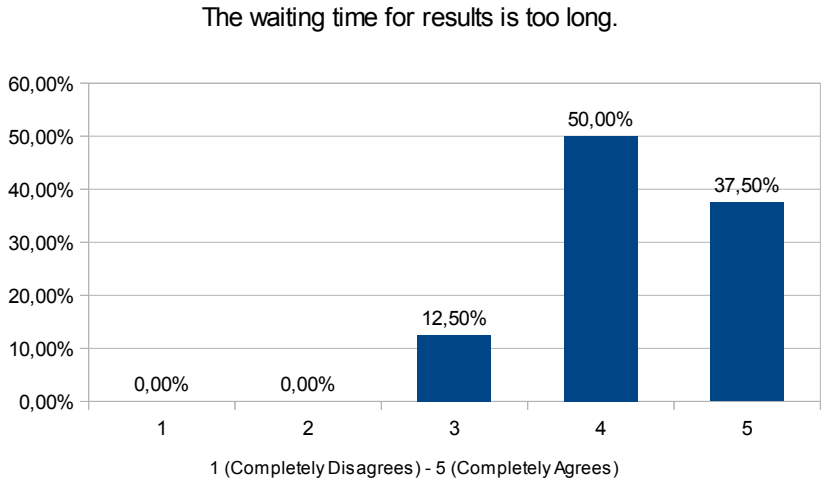

Figure C.28: Question CII 3c) : The waiting time for results is too long.

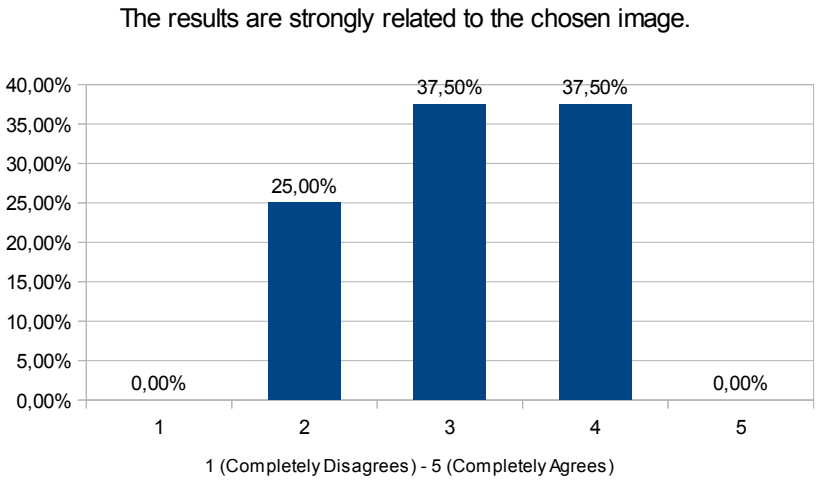

Figure C.29: Question CII 3d) : The results are strongly related to the chosen image.

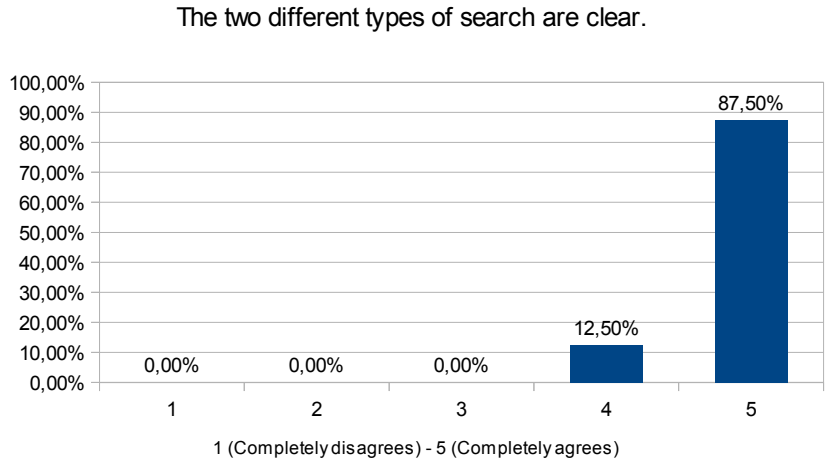

Figure C.30: Question CII 3e) : The two different types of search are clear.

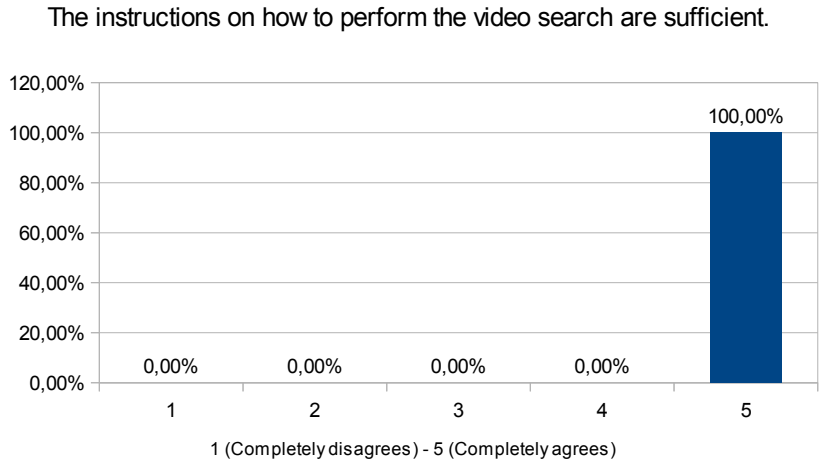

Figure C.31: Question CII 3f) : The instructions on how to perform the video search are sufficient.

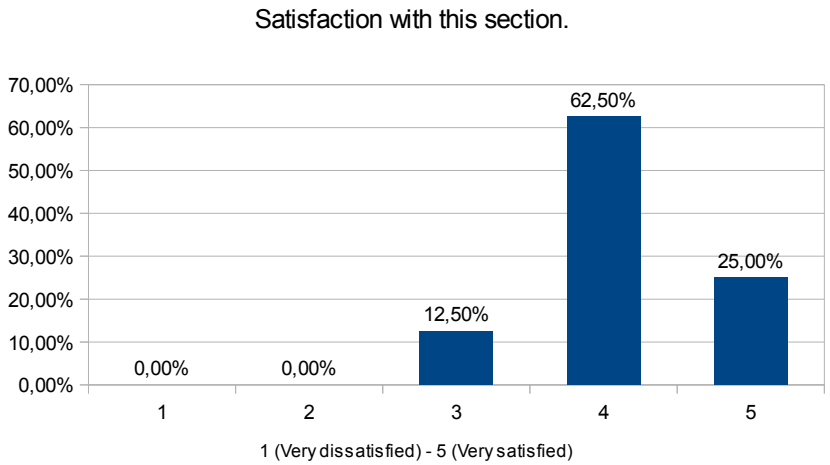

Figure C.32: Question CII 3g) : Satisfaction with this section.

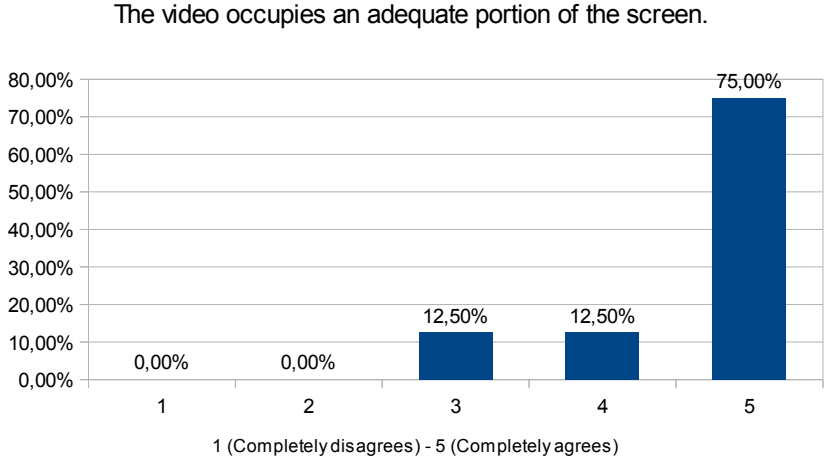

Figure C.33: Question CII 4a) : The video occupies an adequate portion of the screen.

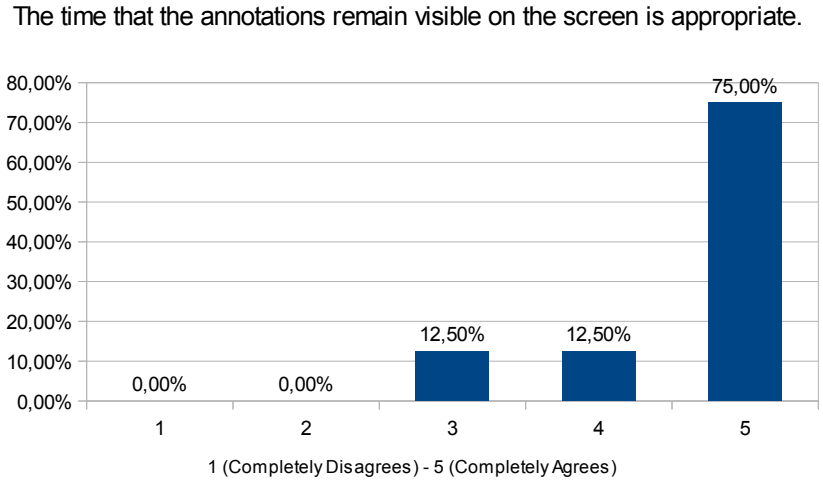

Figure C.34: Question CII 4b) : The time that the annotations remain visible on the screen is appropriate.

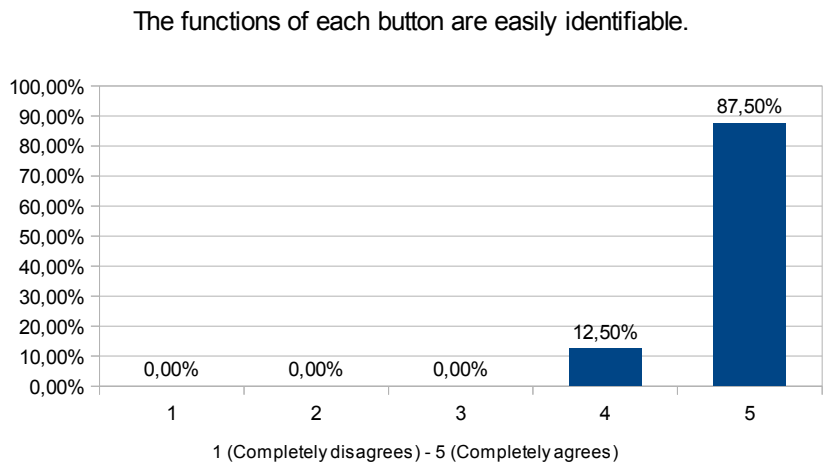

Figure C.35: Question CII 4c) : The functions of each button are easily identifiable.

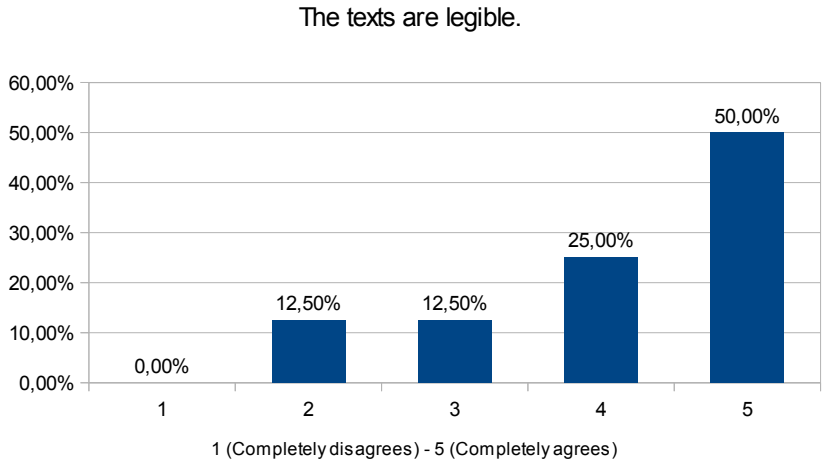

Figure C.36: Question CII 4d) : The texts are legible.

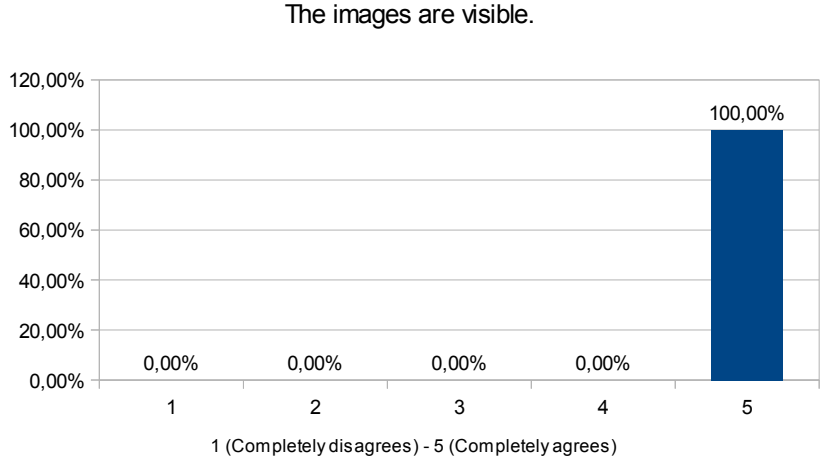

Figure C.37: Question CII 4e) : The images are visible.

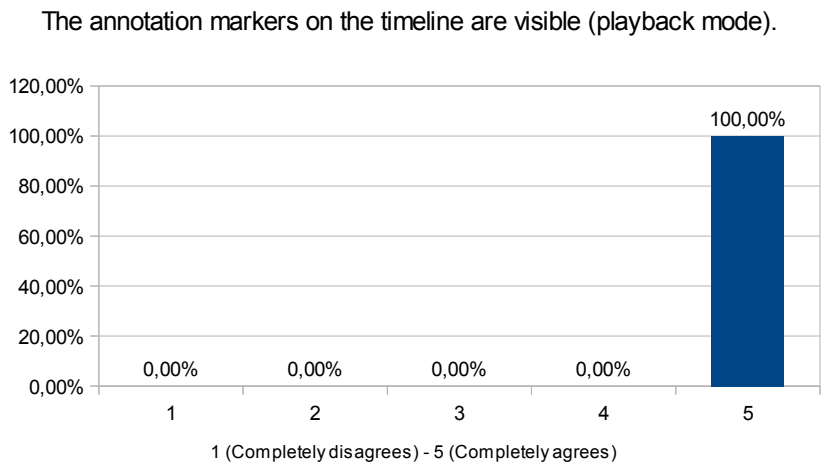

Figure C.38: Question CII 4f) : The annotation markers on the timeline are visible (playback mode).

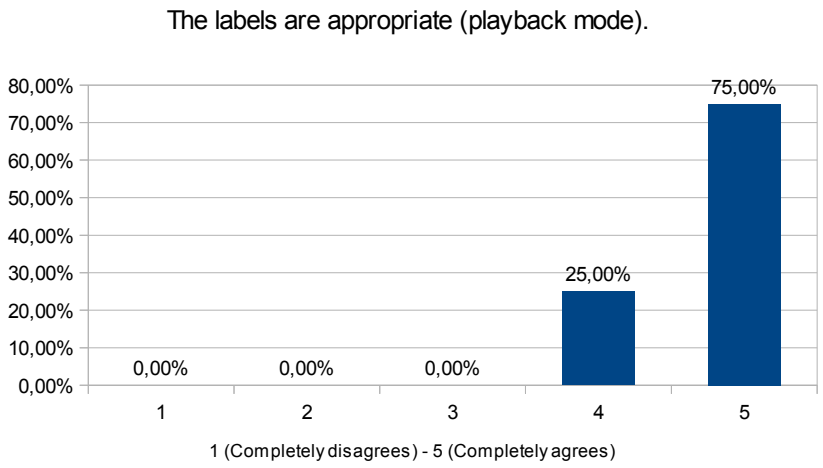

Figure C.39: Question CII 4g) : The labels are appropriate (playback mode).

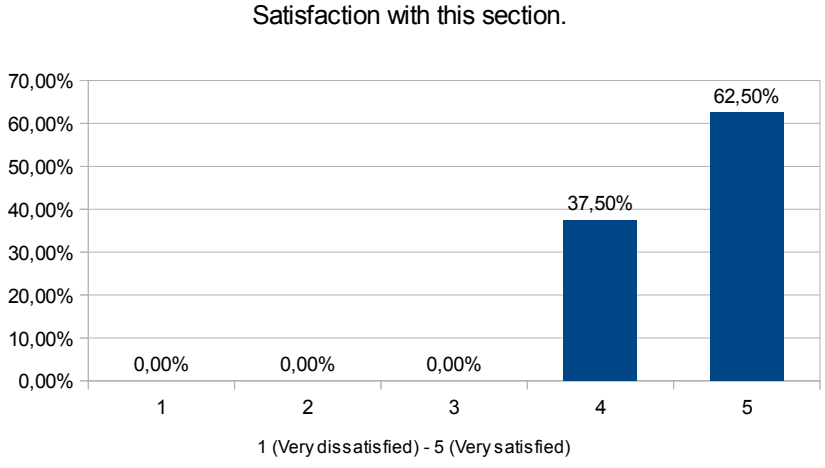

Figure C.40: Question CII 4h) : Satisfaction with this section.

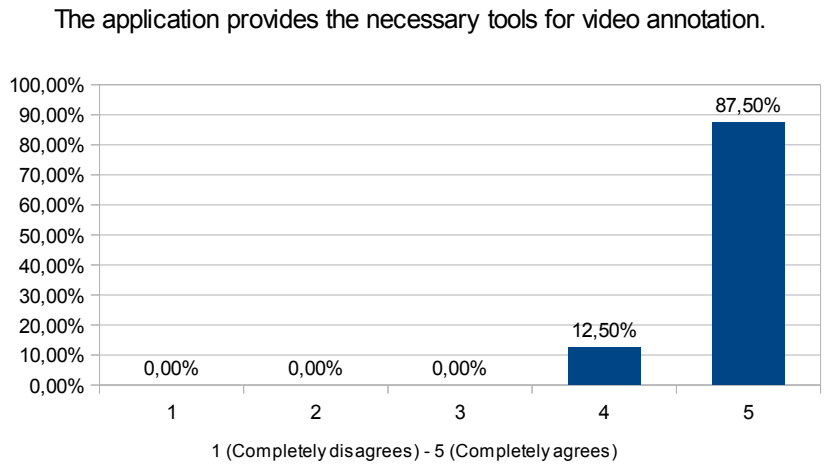

Figure C.41: Question CIII 1 : The application provides the necessary tools for video annotation.

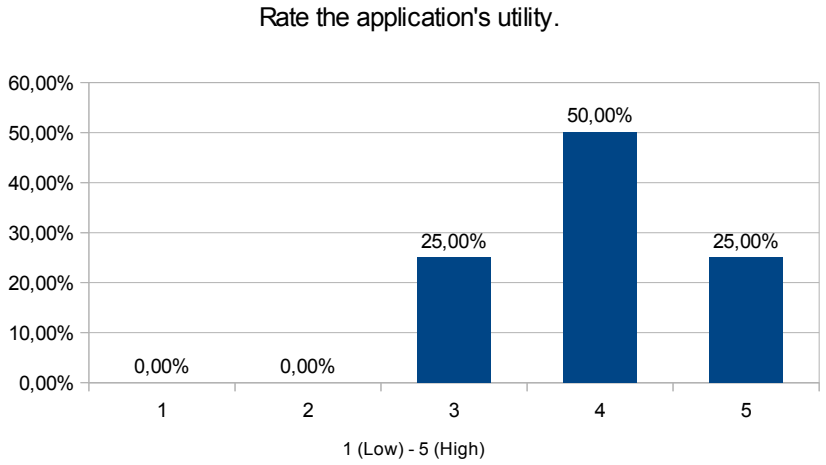

Figure C.42: Question CIII 2 : Rate the application's utility.

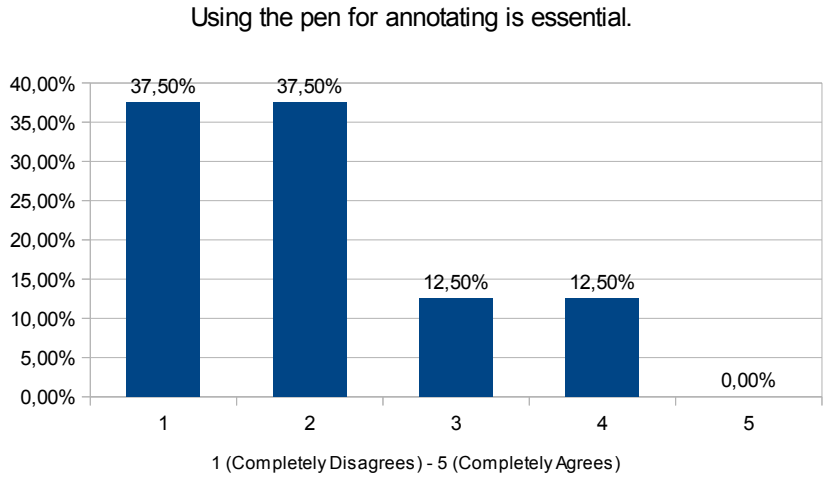

Figure C.43: Question CIII 3 : Using the pen for annotating is essential.

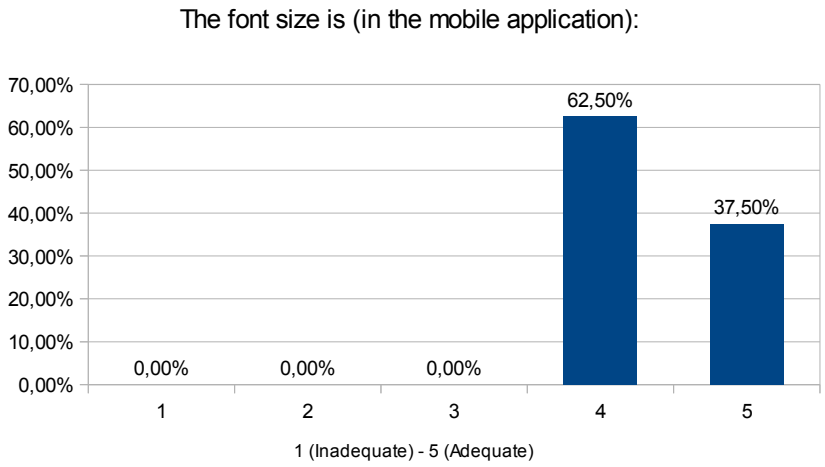

Figure C.44: Question CIII 4 : The font size is (in the mobile application)

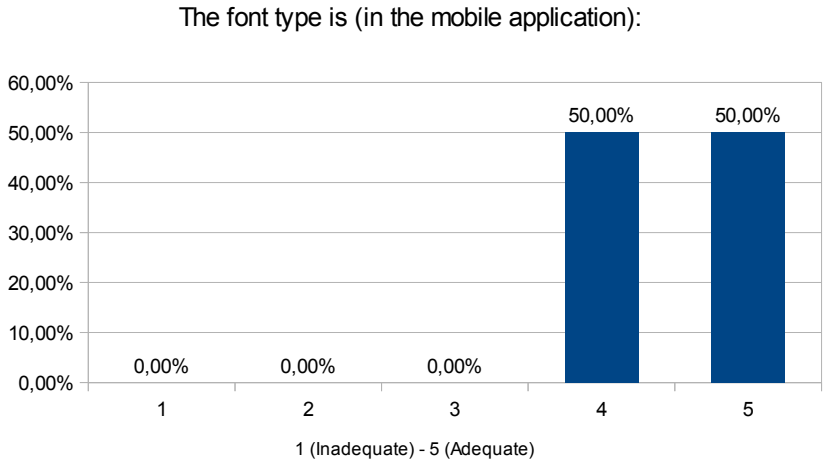

Figure C.45: Question CIII 5 : The font type is (in the mobile application)

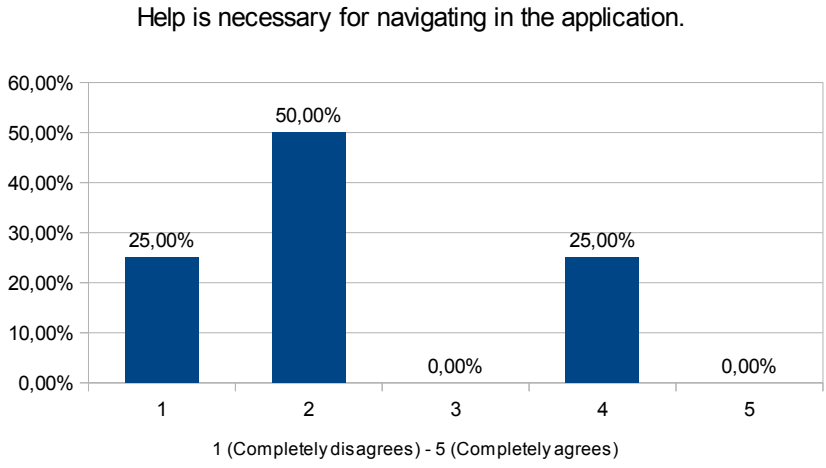

Figure C.46: Question CIII 6 : Help is necessary for navigating in the application.

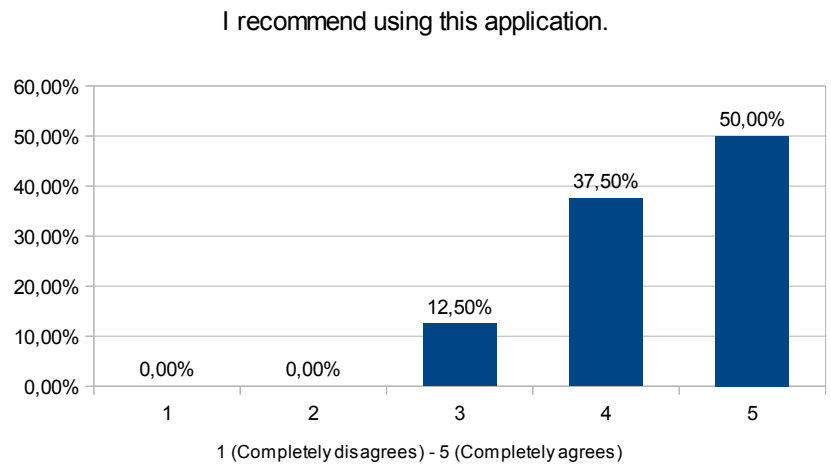

Figure C.47: Question CIII 7 : I recommend using this application.

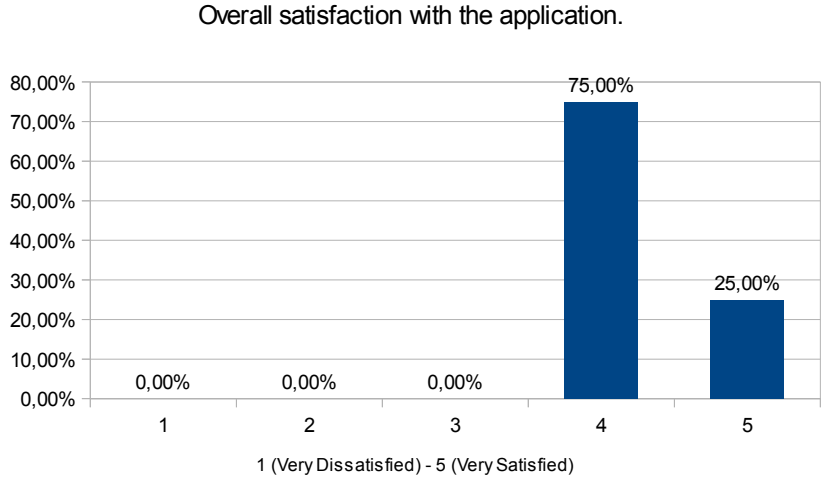

Figure C.48: Question CIII 8 : Overall satisfaction with the application.