Research on Tablet

Tablet Apps for Supporting Research Activity

Li Wai Lun (1009615163)
Mo Lim Kwan Pakey (1009618374)

Supervised By
Prof. LYU Rung Tsong Michael

©2011 The Chinese University of Hong Kong

The Chinese University of Hong Kong holds the copyright of this thesis.
Any person(s) intending to use a part or whole of the materials in the thesis in a proposed publication must seek copyright release from the University.
Abstract: To use the tablet devices for supporting Research Activities, we may need to install a number of apps to perform various tasks, such as paper/reference library searching, PDF reader & annotation tool, file sharing, and group communication client. In this project, we would like to develop an application to help in supporting research activities in CSE Department. The final product would include the following three features: paper or reference management, research activity log and communication. A server is required to be used as paper/reference repository, research activity synchronization and instant messaging.
# Table of Contents

1. Introduction .......................................................................................................................... 7  
   1.1 Motivation ......................................................................................................................... 7  
   1.2 Objective .......................................................................................................................... 9  
      1.2.1 Make reading paper simply ....................................................................................... 9  
      1.2.2 Improve efficiency ................................................................................................. 10  
      1.2.3 Exchange ideas easily ............................................................................................. 11  
      1.2.4 Minimize workload of managing resources ............................................................ 12  
   1.3 Project Overview ............................................................................................................ 13  
      1.3.1 Research on relevant topics ...................................................................................... 13  
      1.3.2 Design of the whole system ..................................................................................... 15  
      1.3.2 Division of works into two phases ......................................................................... 17  
2. Background .......................................................................................................................... 19  
   2.1 Raising population of tablet device .............................................................................. 19  
   2.2 Lack of academic application for tablets ....................................................................... 20  
   2.3 Reading paper takes the advantage of tablets ............................................................... 21  
   2.4 More features beyond reading ....................................................................................... 22  
3. Research on relevant topics ................................................................................................. 23  
   3.1 Format of research paper : what can be shown to user? .............................................. 23  
   3.2 Format of PDF files : what PDF does? ........................................................................... 25  
   3.3 Occurrence of the phrases : how to make the paper readable for computer? .............. 27  
      3.4 More features can be utilized : how to achieve more? ............................................... 29  
   3.5 Tools help to build the app: Any API available? ......................................................... 32  
4. Overall Design ..................................................................................................................... 34  
   4.1 System Architecture ....................................................................................................... 34  
   4.2 Module Description ........................................................................................................ 36  
      4.2.1 Paper Management ................................................................................................. 36  
      4.2.2 Communication ....................................................................................................... 37  
      4.2.3 Research Log ........................................................................................................... 38  
   4.3 Functional Specification ................................................................................................. 39  
      4.3.1 File Exploring ......................................................................................................... 39  
      4.3.2 Paper Parsing .......................................................................................................... 39  
      4.3.3 Referencing ............................................................................................................ 39  
      4.3.4 Library Renewal ...................................................................................................... 40  
      4.3.5 Posting Comments ................................................................................................. 40
8.3.1. Sharpen the performance of the parser .................................................. 110
8.3.2. Implement advanced features ................................................................. 110
8.3.3. Improve the interface ............................................................................ 110
9. Acknowledgement ......................................................................................... 111
10. Reference ...................................................................................................... 112
Appendix : Research on Paper Format .............................................................. 113
1. Introduction

1.1 Motivation

Tablet devices gain greater and greater importance around the world. The portable design attracts users for reading information and watching videos. Although there are still limitations of tablet devices and it is not possible to replace the traditional personal computer, the tablets can be further utilized in many aspects.

![Figure 1.1 Samsung Galaxy Tab](image1)

![Figure 1.2 Using tablet to reading e-book](image2)

Seeing the advantages of the portability and the ease of reading books, the tablets should be promoted to perform academic objectives. We would like to introduce it to research activities. However, the design of tablet aims at the simplicity. As mentioned previously, the tablet is different from personal computer. Too complex tasks should not be implemented. So what can be done without a substantial keyboard? Yes, the task should be performed without large amount of input data. We noticed that the large screen of tablet eases the difficulties of reading books. Papers are important components of research activity, and that is what can be easily done without a substantial keyboard.
Purely reading paper seems an easy job. It seems far easier than e-books, because e-books may not available on the internet while huge resources of paper are easily “Googled” (since there is “Google scholar” http://scholar.google.com) on the web. Additionally, PDF reader is common installed in tablets.

Nevertheless, what we hope to do is not only giving a PDF reader to read paper. We would like to build up on the existing resources, and develop a tailor-made application for users to view paper easily. For example, are there ways to help us to view the paper easier? Paper seems like ordinary printings with long long pages of words, but they are indeed very different. References are especially for academic papers. Would the application improve the ease of viewing the reference? And though it is mainly used for viewing paper, we believe that more stuff can be completed with the assistance of the tablet. Extra featured are expected to achieve.

We hope that the product constructed can relieve much difficulties of viewing paper and help in research activity. We envisage this product can truly reach the objectives with the rapid popularization of the tablets,
1.2 **Objective**

Our final project is hoped to develop a tablet application to fulfill the following four objectives.

### 1.2.1. Make reading paper simply

Reading paper is quite a simple task. However, it is sometimes quite annoying – especially for reading papers. Let’s think of the situation of flipping a hardcopy of a paper. Browsing the pages one by one, looking at the dense lines of words and several pieces of images and diagrams, everything runs smoothly. However, some not special but common nagging digits start to annoy us quickly and frequently. The digits within the [ ] bracket show its prevalence in the whole passage, if you would like to know what is that means, you should flip to the last page to have a note – usually you know nothing because the line tells you a string of unknown article names and strange names of researcher – to have a look. Repeat the procedure for 10 times or above – see? Yes, it is annoying!

The above case seldom appears in daily life indeed, since we are not common to refer to reference in daily reading. Nonetheless, the reference weighs heavily in the scholars’ writing. Tones of unknown phrases and findings may be shown and these are all valuable to be referred.
Of course, the problem is easily solved using the PC by opening the file twice. The problem can also be addressed on the hardcopy by tearing the pages out. But we would like to make everything simpler, to view the reference or annotation effortlessly by just one click. If the tool is powerful enough, we expect the tool can directly show us the referred articles too. All these helps make things simpler.

1.2.2. Improve efficiency

Obviously, the tablet devices aim to help saving much more time. We can treasure our time by doing vivid tasks anytime and anywhere – and what the users focusing is not just “what they can do”, but “how they can complete their tasks simply and efficiently”. For the word “simple”, we would like to define the word as doing much more things with minimal work done on it. We try to achieve this “simple” which has been analyzed in the last paragraph.

So what is “efficient” mean? To make it more materialized, we make it related to time consuming. Time is definitely our major concern, and finish the tasks in a simple way is one way to increase the efficiency. Besides, efficiency requires strong supports behind simplicity. Strong, powerful and well-designed database and libraries can greatly devote to the enhancement of the efficiency, which let you locate what you need within the minimal time spend.

In fact, efficiently cannot be easily measured. Comparing to the effectiveness, we may quantify some targets to see whether it can be completed fully or partially, or in how much percentage it completed, while efficiency is heavily depends on many different tiny stuffs. We hope to achieve this by having research on the methodology of reading papers and design our system carefully.
1.2.3. Exchange ideas easily

Research papers, to a certain extent, are the concepts of spreading the themes and theories – or we name it “ideas” – around the world. Research activity involves a great scale of exchanging ideas in principle, in order to invoke much more new inventions. Sharing papers are what we can commonly do nowadays, yet would there be rooms to improve it? Can the communication be further enhanced? If yes, then how?

Building upon reading research papers, we may add features to further boost the circulation of ideas. We should have experienced such kind of situation that we would have some questions or challenges to raise, or we would have our own ideas or different point of views concerning certain point described about the topic. These should all be considered as academic sharing of ideas and they may stimulate us to rethink deeply. Sparks invoked may contribute to new ideas too.
1.2.4. Minimize workload of managing resources

Information is flooding, especially on the Internet. Millions and millions of information floods when we look for a certain topic. Perhaps it helps us to include as much as information as can, while we often spend quite a lot time for filtering and locating what we actually need. Duplication of information always occurs, which prevents us from extracting the pieces we want.

To avoid this, good management may address the problem. If the system is clever and mature enough to manage paper neatly and avoid the repeats, workload of researchers can be minimized. Much more time can be spent to concentrate on the research topic. All these help are more than just saving time of the researchers and scholars – these also save the resources of our computer to achieve green computing in some sense.
1.3 Project Overview

To achieve our objectives, we should implement our project steps by steps. This report will discuss each part clearly, and here will give a whole picture of our project.

1.3.1 Research on relevant topics

Chapter 2.2 gives some analysis of our survey which has been conducted before starting our project with coding. Following are the area we would like to study:

A) **Format of research paper: what can be shown to user?**

Understanding the skeleton of papers makes the filter of information run smoothly. Some parts are essential to identify the uniqueness of the paper and building index of the paper library, just like the title and authors of a paper. But would there be more information can be easily recognized by the server, so as to improve the accuracy of storing paper or showing useful information to the users?
B) **Format of PDF files: what PDF does?**

It is not surprising that most papers are saved as PDF format. Printing as PDF prevents it from editing or mix-up due to careless mistakes and make it portable to various computer working environment and operating system. Of course, we are not trying to analyze “why PDF is used”, but “what PDF does”. Format of a paper is readable for human beings, while PDF format is known by computer. Studying more and getting familiarized to this format would lead us to better development of the application and the buildup of server.

![Figure 1.7 A PDF file](image)

C) **Occurrence of the phrases: how to make the paper readable for computer?**

Besides understanding the paper format, the phrases used by the papers are important for teaching the system too. Keep in mind that papers are designed to be read by human. Languages of people, unlike the languages of computers, can vary within a certain range. Different wordings (even with “syntax errors”) may tell the same ideas. Analysis of the occurrence of some key phrases improves the accuracy for computer to locate the data.
D) **More features can be utilized: how to achieve more?**

Not just referencing is significant; we should investigate more characteristics to enrich the application and accomplish our aims. For example, are there any communication features have been implemented in the tablet that can be made good use of to facilitate the exchange of ideas? Or are there any open source software can be imported to help the development runs smoothly? Just have a research to find out the answers.

E) **Tools help to build the app: any API available?**

Building an application on the tablet devices is not easy. Android platform, to a certain extent, is quite a new platform to us. It is quite hard for us to start to learn to write a PDF viewer from knowing nothing. We would like to find out if there is any APIs can help us to build our application smoothly. After all, learn to build a PDF viewer is not our ultimate goal. We would like to save time so as to run more towards our goals.

### 1.3.2 Design of the whole system

Chapter 3 concludes the overall design of our system. Our system includes server side and client side development, that both sides heavy workload to handle.

For the server side, it has been developed using PHP web server language. The server side aims to do every parsing and analysis on the PDF files and manage the database. PHP is chosen because PHP has strong libraries for reading the PDF files. This make the analysis on the parsing paper runs smoothly.
For the client side, our application has been implemented on the Android platform. Originally it is expected to be implemented on the iOS. Yet, due to the limitation of hardware resource that we do not have iOS development platform, we give up the iOS and switch to Android platform. The Android application is built by Java language and the development is comparatively portable to the application runs on iOS. Reiterate that our focus for once – we would like to make research activity simpler on tablet devices but not on MACs only.

To ensure the progress of the project, the project is roughly divided into two parts. One student focuses on client side and the other student focuses on the server side.

There three main modules to achieve our goals:

1) Paper management
2) Communication
3) Research Log

Further explanations on design and implementation are included in Chapter 3 and 4.
1.3.2 Division of works into two phases

To make the tasks clearly, we divided the works into two phases. We have tried to conduct a comprehensive survey and implement some key features of the applications during the first phase. In the second phase, we would aim at increase the accuracy of the application and make fine tunes to improve the efficiency.

![Phase 1 Diagram]

*Figure 1.10 Works in phase one*
Figure 1.11 Works in phase two
2. Background

2.1. Raising population of tablet device

When it comes to tablet devices, they are seldom used in academic aims. Chasing back to the time when Steve Jobs presented his “truly magical product” – iPad[1], he defined it with many functions: “browsing the Web, sending e-mail, sharing photos, watching videos, listening to music, playing games and reading e-Books”. Since this most popular tablet in the world released, we almost found most of users treat it just as a smartphone with a larger screen. Surely most of features of the tablets, not matter iPad or Android tablets, can be found on the iPhones or Android smartphones. The main differences between the tablets and smartphones are the calling and SMS functions and the screen size.
However, we thought that these portable devices with a larger screen size should not be treated like an alternatives of smartphone. They are truly different, and we should fully utilize the differences. Once we noticed what Steve Jobs presented, one point should be focused – the reading e-Books function of the iPad. As one of experienced smartphone users, this point woke us up. Among the abovementioned features, we are comforted to perform most of them. We may browse the web for dealing with some urgent stuff or finding emergent information, yet we seldom use the phone to read books. Of course, reading books is possible on phone devices, while the small size screen is not preferable compared with the tablet. And this is what we looking for – the absolute difference between tablet devices and the small touchscreen gadget. On recognizing the difference, we decided to develop a tool to make good use of it.

2.2. Lack of academic application for tablets

We would like to introduce the tablet devices to academic aspects. Take a look at the Android market. Click on the category tab of “Books and References” – the one seems related to academics research – and check if there is anything helps in research or paper referencing. However, we would get a list of e-books or e-book readers. Even we click onto the tab of “education”, we also get similar results. How about we type “research” and some relevant terms? Most of the results show application related to business or stocks. One result of “research paper” catches our eyes – well, the app “Writing Your Research Paper” seems just an E-book, not an app help me to writing the
paper.

The case may not such extreme indeed. If we search “research Android App” on Google, meaningful results can still be found. However, there are still few meaningful results. The two applications we found is just like a database and libraries for us to search for materials [2, 3]. There is lack of strong and useful applications for supporting activity.

2.3. **Reading paper takes the advantage of tablets**

As mentioned in the motivation part, complex task should be avoided violating the original concept of the tablet design. This tells reasonably that nowadays the application available only provide the function of searching papers or research resources for users.

Taking of advantages that tablet can read books and keeping the origin of tablet, we would like to develop an application related to reading research paper. Reading paper seems quite a simple task. It seems far simple than e-books. Books may not be available on Internet, while there are great resources of paper on the web.
Yet when we read papers, no matter the hardcopy or softcopy on the tablet, we find it annoying to clip the pages over and over, to grab the reference and chase where the ideas come from. The clipping maybe not the great problem, but the referencing can surely be done better and better. Somehow, just clipping to the last page we just know more about a list of strange titles and authors. What we want is the paper! And that is what we looking for – a paper reader with automatically referenced and linked to others.

Figure 2.4 The large screen of tablet is good for reading books and papers

2.4. More features beyond reading

If a platform is designed for reading research paper, then could it be further extend to a research tool? Quite more than reading paper and required little data input. We would like to improve and sharpen the product in order to realize a helpful research tools that truly cater the needs of researchers. This goal prompts us to think more, that will be further clarified in the research part and the design part.
3. Research on relevant topics

Research tools are not rare in personal computer. What we are now seeking for is not just software of a research tool, but an application specifically catering the aim of reading paper and using the tablet devices. Yet, Scholars’ writings, though have been quite restricted in the writing formats, vary much because they are written by human. To make the paper more manageable, we should first start to see the difference and tell the computer what parts to look at. There are four main topics we have conducted relevant surveys.

3.1. Format of research paper: what can be shown to user?

To start with, we should clearly define what the research paper is. Somehow our product may not powerful enough to recognize all PDF files without a standard. Quoted from Wikipedia: "Research paper, (also called scholarly paper), which is published in academic journals and contains original research results or reviews existing results" [4]. Under such definition, we try to restrict the paper format fulfill the rules of the publication of the academic journals.

Firstly, generalizing on the ideas of the some reference page of the University of California [5] and the Rice University [6] from USA, the research paper have to include the title, author(s) (and maybe the contact information), the abstract and the references. Of course there are introduction and contents, but abstract is enough for achieving our goals.
Second, we conduct a rough survey on a pool of research papers. The pool of papers contains 169 PDF files and 149 of them fulfill the definition of a scholarly paper. We analyze the position of title and author, the appearance of abstract and keywords, and how the reference show. Nearly 95% of the papers have an abstract, and 90% of them start with the word “Abstract”. The keywords appear in 50% of the papers. Almost 95% of the references indexed by the [ ], while few of them using other format or with no index (just show a list of references). Some of research papers consist of unstandardized format, such as author goes before title, or in the form of report. 15% of them start with a cover page which is not a standard.

To conclude, over 80% of the scholar papers in the research pool follow the format that includes title, authors, abstract, introduction, content and references. Though it may not be the real statistics of all the papers available on the web, it is worth to have a try in parsing papers in the format investigated.
3.2. **Format of PDF files : what PDF does?**

According to Wikipedia: "Portable Document Format (PDF) is an open standard for document exchange. This file format, created by Adobe Systems in 1993, is used for representing documents in a manner independent of application software, hardware, and operating systems."[7]

**Structure of PDF files**

The PDF files consists of eight primitive objects including Boolean values, numbers, strings, names, arrays, dictionaries, streams, and the null objects[8]. PDF contains everything as objects, to standardize different data printed by various files originally. Take text as one of the examples. Adobe creates an element for the text and makes it as stream [7]. The primitive type “streams” is used for storing a large amount of data. The element records not only the text wordings, but also the font size, the font family, etc. PDF also records the layers of each object.

**Metadata**

There is one more significant feature of PDF files. There is metadata of PDF files. The metadata records much key fields of the title, author and update dates. The metadata seems to be powerful, which could help us a lot in completing or goals. PDFmeat [9] is a powerful open-source tool to acquire the metadata of PDF Files and we have done a research on the metadata using the PDFmeat. Though there are few errors, over 70% of papers are correctly located.
**PDFmeat**

PDFmeat stands for PDF Metadata Acquisition Tool [9]. The tool provides great support for analyzing the metadata in PDF header and managing the PDF files. It aims at convert the metadata to BibTex. To run the program, PDFmeat provides a user-friendly GUI which embedded in the Firefox browser. On click it will generate BibTex data to catch the metadata of PDF file. However, the GUI is not feasible for realize our aim. We would like to have the CLI program to run and catch data. We have done many searching and refer to the official page for many times but we still can access the PDFmeat successfully. Unless we can achieve it successfully, we are impossible to use the metadata.

**PHP Library - PDFlib**

There were many share libraries available for PHP, and some of them related to PDF files. Nonetheless, most of them are aimed to create PDF files rather than parsing / reading the PDF files. Filtering those useless, we focus on the PDFlib. [10] The PDFlib get various value like font size, position in X and Y, page numbers etc. [11] All these features shall be taken at note for implementation.

*Figure 2.5 Icon of PDFmeat*  
*Figure 2.6 Icon of PDFlib*  
*Figure 2.7 Icon of Xpdf*
Xpdf – pdftotext

Though PDFlib can read the PDF files, most of the functions build-in are expected to be used in creating or editing PDF files. Like the functions in PDFlib, these is no any function can directly extract the text of the existing PDF. Parsing PDF text may be not relied on the PHP library. Nonetheless, there are other open-source programs help to extract the text in the PDF, like PDFBox [12] and Xpdf [13]. We would first implement Xpdf for experiment.

3.3. Occurrence of the phrases: how to make the paper readable for computer?

To study how to parsing the paper, we analyze the phrase occurred in the paper which may make the paper more understandable to the computer.

Title

95% of the papers do not have the “title” word to indicate the title of the paper. Usually the font size of the title is the largest word appeared in the whole file. Few of the titles appear after the author name. Titles may not the first line of the paper, because there may be headers of the page or some further information.

Author

There are a huge number of papers which are written or collaborated by more than two authors. Besides the name of authors, more than a half of papers we surveyed contain a line of information and contact information about the authors. For those do not show author information at the top of the paper, the information may be added at the end of the paper (after the appearance of references).
**Abstract**

More than 80% of abstract is written following the presence of the “Abstract” word. Some of them are written directly after the name / the contact information of the authors. When the abstract ends, introduction or keywords will be presented.

**References**

Nearly all the papers have the “References” or “Bibliography”, while there is a small variation between paper and papers. Statistic shows 90% of references have shown in the index form [ ]. The rest shows without [ ] but numbers only, and some of them without indexing. Those lists of references without indexing will not be considered in this project because of the meaninglessness.

**Keywords**

Presence of keyword is not a necessary part of a standard scholarly paper. Yet, after consideration, we would like to check if there is any presence of the keywords column. If there is, the keywords will be recorded in our database to assist the searching of paper. The keywords usually appear with the presence of different noticing words like “keywords”, “index terms” or “general terms”.
3.4. More features can be utilized: how to achieve more?

We have done some searching in further exploration on the functionality of the application. We tried to start with our objectives set, and some sparkles invoked.

To improve the efficiency?

As undergraduate students, we have never got a chance to perform a formal and comprehensive research on any topic. However, we believe that every time we do research and investigation, we need plenty of information and bibliography to support what we are saying. Quite annoying scenario arouses – when we try to look for some new information, we read and find some unknown words and terms. Then we try to make it clear, and we search for other pages of information and more and more terms will come out. Just like some kind of recursion. Finally, after we open for hundreds pages of reference, we almost forget what we asking for!

Thus, we find it will be valuable to have a research log for each user. User information are recorded in our database and users have to login to use the application. Every time they read a paper, there are their own logs to save the record. Even though they go though from one paper to another for hundreds of times, they can still find what they originally look for and can more handily filter what they need.
To exchange ideas?

When it comes to communication, it is not surprised to think of the idea of instant chatting. Chatting or messaging helps for better discussion on a topic, while some views are worth arguing. It seems reasonably, but impractical indeed. Firstly, due to absence of substantial keyword, the application is designed for as less input as possible. Discussing on instance messages requires a certain amount of input data. Moreover, the main advantage of a tablet is the portability. Serious discussion seems not meaningful on the railway journey, or if it does, phone calling would be more practical and efficiency.

We want some features are not redundant but useful. Does no instant messaging mean there is no need towards exchange of ideas? Definitely not. Ideas and views are aroused during the reading of other’s writing, and we may give some response to the writer, or share what we think to a specific point. Yes! The comments.

Adding comments onto the paper, when someone has his own view, his can put it down with few sentences. The comments will be saved when anyone reopen the file. Every users can comment and view what others thinking by just one click. Exchange of ideas can hence be achieved, handily.
To minimize workload of managing resources?

Papers are our main characters of the show. We cannot implement an application without papers. We are actually not worried about functionality and the implementations, because once the algorithm and functions work successfully, it seldom have bugs. (Of course further analyze and maintenance is needed to improve the accuracy.)

However, even if the application is precisely implemented with perfect algorithms, the renewal of paper should be carried out everlastingly.

Could the paper library enriches and renews itself automatically? We raise this question and have a draft answer. We may embed a web browser into our application. When the user cannot find what he wants in our existing database, he can open the web browser to look for new information. If he would like to download a new paper, the library would save it add do checking and parsing. If there is duplication, the new file will be abandoned. Otherwise, the paper will be parsed for every data requires and a new entry will be added. With this feature, the maintenance and management by human being can be greatly reduced.

We noticed that there is a http://scholar.google.com/ which is a strong library of scholarly papers. We put this back and will look if we can take this advantage in the Project Phase 2.
3.5. **Tools help to build the app: Any API available?**

To alleviate our workload, we choose to use the API of PDF viewer. We have found several in the Internet, such as ‘APV - Android PDF Viewer’, ‘vudroid’, ebookdroid' and ‘apdfviewer’. We have tested their performance by reading different types of PDF files.

<table>
<thead>
<tr>
<th></th>
<th>APV</th>
<th>vudroid</th>
<th>ebookdroid</th>
<th>apdfviewer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only text</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Only image</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Both text and image</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Small file size</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Very large file size</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

(1 is the best, 4 is the worse)

We have tested the four APIs by five different types of PDF files, one is only containing texts, one is only containing images, one is containing both texts and images, one is of small file size and the final one is of very large file size.
**APV - Android PDF Viewer**

‘APV’ has the best overall performance, and does best when reading PDF containing both texts and images with small file size.

**vudroid**

‘vudroid’ does worst when reading PDF with only images and of very large file size. Its overall performance is the worst among the four APIs.

**ebookdroid**

‘ebookdroid’ has very good performance when reading PDF with only texts and of very large file size. However it does not perform good when reading images.

**apdfviewer**

‘apdfviewer’ does the best when reading images, however, it has the worst performance when reading texts.

**Conclusion**

Although ‘ebookdroid’ does better than ‘APV’ when reading PDF with texts only and of very large file size, a paper contains both texts and images, and its file size would not be very large, we believe that ‘APV’ is the best API to read papers, and we choose to use ‘APV’ in our project.
4. Overall Design

4.1. System Architecture

Our system consists of server side and client side for better communication and management, and for better collaboration with the database.

Figure 4.1 Architecture diagram on server base
Figure 4.2 Architecture diagram on tablet base
4.2. Module Description

To make the tasks simpler, we have divided our works into three different modules.

4.2.1. Paper Management

Paper management plays a great role in our project. On one hand, well paper management helps researchers and scholars to locate their target resources and information easily. If the library is strong enough, the researchers can save much more time and improve the research efficiency. Just say, how to guarantee each paper is unique? Even when we type some keywords onto anyone of search engine, a list of repetitive results is not surprising. Yet most of the search engine is not clever enough to ease out those duplicated results. Extracting the information cost much time when we try to perform search. Therefore, we would try to alleviate the problem of duplication – try to store only one file despite different file name of same paper, or save the same paper only if the version is different.

On the other, the available paper resources online are vivid. Of course it is impossible to grab all papers from the internet once the application released, yet it is our ultimate goal to keep as much file as possible in our library. The tool may not powerful and intelligent enough to download all files online automatically, but we hope to achieve the task with the help of the users. The application on tablet embeds a web browser and let the users to search on web. When a paper is downloaded, the app will raise a signal and send the link of the paper to the server. After comparing the paper newly downloaded is unique and has not been saved on the server, a new
file will be included and the source path is recorded to avoid duplication in the future. The library can hence expanded without heavy maintenance and involving a great amount of manpower.

Moreover, assisting the users to view the paper easily is one of our main objectives. To achieve this, other than show the references in just one click, we would also like to introduce some linkage between papers. While the user takes a look at the reference of some specific points, a hyperlink is provided to link the reference paper and the user can therefore have further investigation on the related topics. If the idea can truly be realized, it can surely minimize much more time for the user to do searching and collecting.

4.2.2. Communication

The concept of exchange of ideas has been discussed previously. We expect no any instant messaging because we hope to avoid inputting great amount of data without a substantial keyword, and we believe the user would relied on other more effective and efficient communication method rather than using our application.

We will let the use to post his comment onto the paper. The comment can be added onto anywhere of the paper. There will be a button or a flag for identification of the existence of the comment. Every other user can view the comment and can give response to the point too. The users who have echoed their ideas before would receive a notice that someone posted response. These stimulate the discussion by a small scale of input data. Since all the comment will be saved, the discussion can be shared to many users and the circulation of ideas can be achieved.
4.2.3. Research Log

Researchers face a huge obstacle that has been noticed for many times before: the huge amount of information flooding into our mind. When we go through for a referenced idea, we flip over the pages for the first term. To understand the meaning of a theory or a term, we shall then refer to one another. We flip the pages over and over, and for many times after we almost forgot what we are reading at first.

The research log shows everything that each user has done. Every single paper that his has read is record and sorted by the opening time. This allows the user to chase back what his is originally looking for. Bookmarking will be considered as one of the features. If the user finds some papers that are truly inspiring, the user can put a star on it. Then his can reopen it by referring to the log.

We would like to make the research as strong as possible. Even the time the user posts a comment on a specific paper would be record. And if the comment posted has been replied by someone else, the user would be told by the research log too.
4.3. **Functional Specification**

This section will show general functions that the system can perform.

4.3.1. **File Exploring**

User can click on the file explorer to open the research paper that he needs. According to the interface design, after the user choose the file and the paper will be shown on the right hand side. A searching function will be available for user to search keywords, title or authors.

4.3.2. **Paper Parsing**

Once the paper is added into the server, the paper shall be parsed in detail and all data shall be recorded in the database for better management. All fields will be parsed once to check if it is duplicated with existing papers. If the paper is new, all fields of data will be stored in the database for future uses.

4.3.3. **Referencing**

The newly added paper will be parsed to check if the reference index available. If yes, a list of references will be parsed into the database. The reference index throughout the whole passage will be highlighted and linked and user can refer to that point by a click. No matter the paper have indexed reference, a checking and linking procedure will be performed on the references in the list. If a piece of reference points to another paper that exists in our database, a linkage will be built for a faster referencing.
4.3.4. Library Renewal

A basic paper managing panel should be provided for uploading paper onto the server. The panel is monitored by administration and the uploading process is manual. We would like to introduce a browser to be embedded within the application. Even if the introduction is fail, we still hope the application can be called automatically when a PDF file downloaded. The application will first check if it is a scholar paper. The paper will be parsed and uploaded to the server. All these are expected to be implemented with manual control.

4.3.5. Posting Comments

User can post comment onto the paper. Different actions can be done, such as viewing, deleting and replying. The important stage is how can those have commented or responded to the specific point can be informed to extend the discussion. This part will be collaborated with the user research log functionality.

4.3.6. User Login

The application requires user to login. User would have his or her own profile for other users to refer to. Only little basic information is needed, like the full name and simple description of the user.
4.3.7. User Log

Each user has his user log for recording the history of browsing each paper so as to make the process of referencing run easily. Bookmarks are allowed. The log also contains the history of posting comments and notifies the user if there is anyone responded to the comments he posted.

4.4. Development Platform

**Server side**

The server is built by php5.3.6 with Ubuntu. Due to the research related to the PDF, there is PDFlib in PHP to read PDF. PDFlib supports only the version newer than PHP 5.0.Xpdf and PDFBox are also need to be installed in the server for parsing the text in the PDF.

As for the pdfmeat, since we still cannot implement the CLI successfully, we would like to implement the project without it in phase1. If the self-implemented result is not satisfied, the pdfmeat will be introduced in phase2.

**Client side**

Since the Android platform is finally selected by the end of September, the latest version of Android at that time was chosen. Our project will be implemented in the Android 3.1 (the version which Samsung Galaxy tab 10.1 implemented [14]).
5. Detail Design

This chapter describes the design of the whole system in details. The shows all tiny features we would like to implement. In this part, we would state clearly whether the features will be implemented in phase 1 or 2 of the project.

5.1. Module 1 : Paper Management

This module performs four main functions.

5.1.1. File Exploring

The first major concern is on opening files.

Show list of paper (Phase 1)

The user interface is composed by two parts. The sliding door on the left hand side is a file explorer. The list shows all the papers available on the server. In the first phase we will show the file name only. In the second phase we may consider to show the paper with title or author.
**Additional feature (Phase 2)**

On the first click the abstract of the paper will appear. The paper will be shown on the second click.

*Figure 5.1 Basic UI design on the application*
Show paper (Phase 1)

On clicking the file listed on the explorer, the paper will be shown on the right hand side. The PDF file should be allowed to zoom in or zoom out. It allows the user to perform searching on the content of the file. Directly going to a certain page is also allowed.
**Sort of paper (Phase 2)**

For locating the file accurately, there should be a number of sorting methods provided. For example, it can be sorted by title, authors or creating time.

![Image of UI design of sorting papers](image-url)

*Figure 5.3 UI design of sorting papers*
Search for paper (Phase 2)

The searching function should also be available. This makes the process of opening the correct paper run faster and more efficient. The searching will echo the keywords in title, authors, key terms and abstracts of the paper. When the list of searching is shown, abstract will follow the title and authors to provide sufficient information to users.

Figure 5.4 UI design of searching papers
5.1.2. Paper Parsing

Paper requires parsing and saving data in database.

Get information needed (Phase 1)
Based on the result of our studies, we need several fields of paper: title, authors, abstract, keywords (if any) and references. The title, authors name, abstract and key terms are all help to identify every single paper. The title and author play important roles in display the stream of papers while the abstract and keywords help in searching efficiently. References are parsed for achieving better referencing.

Check duplication (Phase 2)
During the parsing, we would like to compare the original file name and title with the others in the database. If we found the title is duplicated, we will further compare the authors and abstract. If the file is truly duplicated, it will be abandoned. However, we would try to see if the version is different or few modifications are applied. In this case, we would like to store two or more versions for reference.

Check the linkage (Phase 2)
When a file is being parsed, it should be compared with the reference list of the other papers. If the references of other paper refer to this paper, they should be recorded and shown to the users.
5.1.3. Referencing

*Show the reference (Phase 1)*

The indexes appear throughout the paper will be highlighted. When one of the indexes is clicked, a box will be prompted and show the piece of reference.

![Figure 5.5 UI design of showing reference list](image-url)
When one of the button of reference is clicked, a box will be prompted and show the piece of reference.

![Figure 5.6 UI design of showing reference list](image)

**Build linkage to other paper (Phase 2)**

If linkage is found during parsing, the result would be recorded in the database. One the linking reference is prompted out, a hyperlink is built and the user can check what it refers to on clicking.
5.1.4. library renewal

**Manage files manually (Phase 1)**
A panel is designed for the administrator to perform the paper management handily and efficiently. The panel allows the admin to upload files, to control the parsing procedure manually and to delete files. The panel helps the developer for further enhancement by showing all fields and entries to ease viewing the parsing performance.

**Embed a browser (Phase 2)**
A browser is suggested to be embedded within the application, so as to make the further search more handily. When downloading PDF is invoked during the searching in the browser, the application can automatically chase the file downloaded.
**Update automatically (Phase 2)**

When application noticed there is PDF downloading relevant to certain papers, it would first check if the PDF is a scholarly paper. The paper will be grabbed to the server and undergo ordinary parsing, and hence the paper library can be further extended.
5.2. Module 2: Communication

5.2.1 posting comments

*Add comments (Phase 1)*

Every user can add their comment at anywhere of the PDF file. When adding the comments, the user ID and the time will be recorded.

![Figure 5.8 UI design of showing comments](image-url)
Show comments (Phase 1)

When a user added a comment, a flag / a button will be shown on the position he added. When the flag is clicked, a dialogue box will show and all the content of the discussion can be viewed. The discussion will be stored in the database and whenever any user opens the same paper, the comments will stay there and every one can view.

Reply comments (Phase 1)

On the dialogue box showing the details of a discussion, a reply button is available for others to keep the discussion.

Delete comments (Phase 1)

For better management and avoiding misunderstanding in discussion, editing comment is not allowed. However, deleting comment is allowed. If the head topic is deleted, all the replies will be deleted. Yet if the reply is deleted, the topic will still keep.

Notify user of new replies (Phase 2)

The users who have participated in the discussion should all be notified when a new reply responded to the discussion is posted. This could further promote the exchange of ideas. This requires collaboration with the development of research log.
5.3. Module 3: Research Log

5.3.1. User accounts

*Login to system (Phase 1)*
Before the user use the application, he should login to the page. Unauthorized login is not allowed to use the application. This ensures every user can perform all the features smoothly.

![Figure 5.9 UI design of login](image)

*Figure 5.9 UI design of login*
Edit user profile (Phase 2)

There is a simple and basic profile for each user, just to make things clear and help the users to know each other. The profile may include the name, description, and contact information.

5.3.2. Research log

View history of reading (Phase 2)

The reading log will refresh every time the user open a paper to read. The entries of the log will be sorted by the time happened in descending order. For better resources management, about 100 entries will be recorded for each user.

Figure 5.10 UI design of research log
**Bookmark paper (Phase 2)**
If a paper is found interesting or valuable for researching, the user can bookmark the papers for future reference. They can found their bookmarks in the bookmark tab of the research log.

**View commenting activities (Phase 2)**
The log will show the all the commenting activities that the user performed, including both adding comments and replying. The commenting time, title of paper will be shown.

**Show notification of new reply (Phase 2)**
While the discussion the user participated has been renewed, the user should be notified by referring the log.
5.4. **Database design**

![ER diagram](image)

*Figure 5.11 ER diagram*
6. Implementation

In the chapter, we are going to describe the implementation of features we implemented in project phase 1 only. For every module, we will describe the implementation method for each function in both client side and server side respectively.


6.1.1. File Exploring

*Show list of paper*

A) **SERVER SIDE** –
The server calls functions in PHP to open the directory “paper/” which store all the files. It returns all the files except the current directory path and parent directory path.

```
- Pseudo code -

IF (open the directory successfully)
{
    WHILE (there is file)
    {
        GET every file within the directory
        PUT the file into the json array   }
    CLOSE the directory       }
RETURN the json array
```
Note that the file name is changed into paper ID during the uploading of paper. Original file name will be kept in the database, for the use of checking duplication in the future.

The expected result will contain a list of file names and will be encoded as a JSON array.

```
```

**B) CLIENT SIDE** –

After login validates, the client device sends request to the server and gets the list of file names. It will show the list in the file explorer in transparent panel on the left hand side one by one.

```
IF (login validate)
{
    SEND request to web server for the list of papers
    GET the result from web server
    SHOW the list of papers in the transparent panel
}
```
Although the file name is changed into paper ID, the list will show the title of the papers obtained from the database through the web server.

Figure 6.1 Screen capture of the paper list
**Show paper**

**A) SERVER SIDE –**
The server saves the PDF files by changing the name to paper ID. This can avoid crashing due to the presence of special characters in file name.

**B) CLIENT SIDE –**
When the user selects a paper, the client device will download the paper through web server and save it in a temp folder. Then the client device will show the paper with some functions, including zooming, finding texts and going to pages. Besides, it will also show the buttons for reference and previous comments in the paper after downloading the paper, these two features will be talked about later.

---

**- Pseudo code -**

```plaintext
IF (select paper)
{
  SEND request to web server to download the paper
  SEND request to web server for the references and previous comments
  GET the result from web server
  SHOW paper
  SHOW buttons for reference and comments
}
```
The three functions, including zooming, finding texts and going to pages, are the built-in functions of the API of PDF viewer we have used. We have studied how they work and are going to make some changes to them, so as to make it more suitable to our design.

For the zooming functions, the user can zoom in, zoom out or zoom to fit the width of paper. These zooming functions are implemented by pressing three buttons in the bottom of the screen, or double tapping the screen.
Final Year Project (2011-2012)
LYU1102 Research on Tablet: Tablet Apps for Supporting Research Activity     Fall 2011

- Pseudo code -

IF (press button of zooming in or double tap)

{  CALCULATE the new paper size

    SHOW the new view

}

IF (press button of zooming out)

{  CALCULATE the new paper size

    SHOW the new view

}

IF (press button of zooming to fit the width)

{  CALCULATE the new paper size

    SHOW the new view

}

Vision-based Pedestrian Detection: will Ants Help?

M. Bertozzi, A. Broggi, A. Fascioli, and P. Lombardi

Abstract—This work presents the vision-based system for detecting pedestrians in real environments implemented on the ARGO prototype vehicle developed by the University of Parma. The system is aimed at the localization of pedestrians in various poses, positions and clothing, and is not limited to moving people.

Initially, attentive vision techniques relying on the search for specific characteristics of pedestrians such as vertical symmetry and strong presence of edges, allow to select interesting regions likely to contain pedestrians. Then, such candidates areas are validated verifying the actual presence of pedestrians by means of a shape detection technique based on the application of autonomous agents.

Keywords—pedestrian detection, machine vision, autonomous agents, symmetry, ants.

I. INTRODUCTION

The detection of pedestrians is a mandatory requirement for future driving assistance systems, since having the capability of avoiding crashes with pedestrians is essential for effectively aiding the driver in urban environments.

This work presents the vision-based system for pedestrians implemented on the ARGO vehicle. ARGO is an

Figure 6.3 Screen capture of zooming
For the function of finding texts, the user can find the texts in the paper after pressing the button of finding texts. The texts found by the user will then be bounced three red lines. Besides, the user can choose to find the previous and next result. If this texts finding is finished, the user can hide all the result, and the view is returned to the normal view.

- Pseudo code -

IF (choose finding text)

{  ASK the user to input the text for finding

    FIND the text in the paper

    SHOW the results in the paper

    WHILE (showing paper with results of finding)

        {  IF (choose previous result of finding)

            {   MOVE the view to the previous result   }  

            IF (choose next result of finding)

            {   MOVE the view to the next result   }  

            IF (choose hiding the results of finding)

            {   HIDE all the results of finding

                BREAK   }   }   }
Research on Tablet: Tablet Apps for Supporting Research Activity

Figure 6.4 Screen capture of finding texts

Figure 6.5 Screen capture of result of finding texts
The last function is going to a certain page. The user can go to the first page, last page or a certain page entered.

- **Pseudo code** -

IF (choose going to certain page)

{ ASK the user to input the page number

SHOW the paper at that page

}  

IF (choose going to first page)

{ SHOW the paper at the first page }

IF (choose going to the last page)

{ SHOW the paper at the last page }

---

Figure 6.6 Screen capture of result of going to page
6.1.2. Paper Parsing

Get information needed

A) SERVER SIDE –
To begin with, once the paper is uploaded into the server, the server will call the “pdftotext” program (developed by Xpdf [13]) to print the text elements into the text file. A text file will be generated and will be used to parse the specific field of the paper.

In the section, the result of our survey conducted in chapter 3 should be set as reference. To implement the parsing, we summed up for several methods of parse:

1) Search for the pattern of the target information.
2) Search for the keywords of the target information
3) Search for the start of the target, ie. what is expected to be written before the target.
4) Search for the end of the target, ie. what is expect to appear after the end of the target.

One or more above methods will be used in parsing different fields.

Now we would like to show the implementation one by one.
Title -

Using the result analyzed, there are seldom the “title” term to indicate the title. Also, the titles may not be first line of the paper and they may stay in more than one line. Extracting the first line or some lines are not possible, because it would be more difficult for the computer to identify whether the line following is the authors’ name or just some technical proper noun.

Based on these, we try to get the font size of all the text elements and compare them (implementation method 1 – pattern recognition). We can quite sure that the font size of title is different from the header and the author’s information. We found that the title is usually in the biggest size. Moreover, to make sure that the size got is the title’s one, few more rules should be set. For example, we compare the size of the fonts on the first page only.

- Pseudo code -

WHILE (is on page one)

{ GET font size of each line

GET the largest font size

GET the line number(s) of the largest font appears}

OPEN the txt file

COPY the text in the line(s) of the largest font appears

UPDATE the database with title got

RETURN the title found
The title of the paper will be return in expect.
Author:

According to the statistics, most of the papers are written by two or more authors. The authors’ names are usually grouped in one line right after the title. There are often two to five lines descriptions and contact information follows the authors’ names. We found that it is an extremely complex task to search for the next line of authors’ names, because there is no any standard and pattern for the description. The description may include the title of the authors or the organization they belong to. Therefore, the implementation method 4 (search for the end) is not feasible. The lack of uniformity is also found in the authors’ name, and hence the recognition of pattern for the names is not possible too.

Finally, we decided to make the method simple. We would like to simply parse the line after the title (implementation method 3 – search for start) because most of cases the authors’ names appear right after the title.

- Pseudo code -

```
WHILE (title is found)
{
    GET the next line AS authors
}
UPDATE the database with authors got
RETURN authors
```
Camera calibration and 3D reconstruction from a single view based on scene constraints

Guanghui Wang a,b,*, Hung-Tat Tsui a, Zhanyi Hu b, Fuchao Wu b

*Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, P.R. China
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, P.R. China
Received 30 January 2004; revised manuscript received 27 July 2004; accepted 30 July 2004

The authors have focused on the problem of camera calibration and 3D reconstruction from a single view of structured scene. It is well known from computer vision that for an imaging system, the intrinsic parameters of a camera can be obtained from the vanishing points of three collinear points.
Abstract

Usually the abstract is written after the appearance of the “Abstract” word, with reference to the survey result. Therefore it is easy to locate where the abstract starts (implementation method 2 – keywords location). However, sometimes the abstract are in more than a paragraph. To make sure the scope of abstract can be extract correctly, we may also check where is the end point of the abstract. Usually the introduction, sometimes the keywords column, follows the abstract. Therefore the scope ends before the line which contains the “keywords” or “introduction/summary” (implementation skill 4 – search for the end).

Pseudo code

WHILE (a line is parsed)
{
  IF (“Abstract” is found)
  {
    GET every line AS abstract
    UNTIL “Keywords” or “Introduction” appears }
}

UPDATE the database with abstract got

RETURN abstract in a json
Since the abstract may consist of several lines or paragraph, it will be encoded as json array to ensure correct data transmission.

```
{ "Abstract": "This paper mainly focuses on the problem of camera calibration. A camera can be obtained from the vanishing points of three mutually orthogonal planes defined by two of the vanishing directions in the structured scene, with known length ratio in space. The constraint obtained from one image, and the camera can be recovered under the four independent assumptions of collinearity, orthogonality, and scene constraints. Thus, a scene structure can be reconstructed from the literature, validate our proposed methods. © 2004 Elsevier B.V." }
```
**Keywords**

If there is any keyword, most of them appear after several terms indicate the keywords. The terms indicate keywords varies, like “Keywords”, “key terms”, “Index terms” and “General terms”. All of them should be checked. Similar to the case of “Abstract”, the keywords can be found by locating the specific terms and end by the presence of “Introduction” (implementation methods 2 & 4). More to note that, each keyword or key phrase are usually separated by “,” or “;”.

- **Pseudo code** -

```plaintext
WHILE (a line is parsed)
{
  IF (“Keywords” is found)
  {
    GET Keywords SPLIT BY “,” or “;”
    UNTIL “Introduction” appears }
}
UPDATE the database with Keywords got
RETURN Keywords in json
```
Since there will be a list of keywords, it will be encoded as json array to ensure correct data transmission.

{"Camera calibration", "3D reconstruction", "Absolute conic", "Circular points", "Single view modeling"}
References -
The references appear 100% after the word “REFERENCES” / “BIBLIOGRAPHY”. According to survey, 90% of references have shown in the index form [ ]. Some shows in the form “1.” while the rest have only the reference items without index. In phase 1, we would like to parse the type of [ ] and those without indexes.

After locating the “References”, we will check if “[“ appears. If “[“ appears and is followed by an index (sometimes may not only digits but also alphabets) and “]”, we record the index with relevant reference item. If there is no “[“ appears within 10 alphabets, we record each line as one reference item.

- Pseudo code -

WHILE (a line is parsed)
{   IF (“References” / “Bibliography” is found)
    {   IF (“[“ is found within 10 alphabets]
        {   GET index within [ ]
            GET relevant reference item }
        ELSE
            {   GET a line AS a reference item } }
    }
UPDATE the database with References got
RETURN References in json
Geometry Group of the University of Oxford is developing the test images. The work is supported by the Hong Kong RGC Grant CUHK 4378/02E, the National Natural Science Foundation of China under grant no. 60175009 and the National High Technology Development program of China under grant no. 2002AA135110.

References


Result

6.1.3. Referencing

Show the reference

A) SERVER SIDE –
To assist the search of reference indexes and reduce the workload of the client side, the server will parse the whole paper and get a list of reference indexes throughout the passage. For example:

![Figure 6.7 Photo showing the references inside the paper](image)

Then the [1,2],[22,23] and [3] will be parse into the reference appearance list. The list will be returned to the client directly when client request.
B) CLIENT SIDE –

The client device will first ask the web server to send the list of references to it. The list of references will be shown in the transparent panel in the left hand side of the screen. Furthermore, the user can click on the buttons of references on the paper to show what the reference is. After clicking, the information of the reference will be shown.

# Pseudo code

SHOW the list of references in the transparent panel

IF (choose one of reference buttons)

{ SHOW the information of that reference }

Figure 6.9 Screen capture of reference list
6.1.4. Library renewal

Manage the library manual

A) SERVER SIDE –

During the early development, a viewable managing GUI platform is provided for the administrator to use.

There is a drop box for the admin to drag the PDF file and upload it. Once the file is dropped and successfully uploaded, a new row will be inserted at the top. The txt file will be automatically generated for parsing. The newly uploaded file will be parsed too. On the panel, admin can delete the paper too.
6.2. Module 2: Communication

6.2.1. Posting comments

Add comments

A) SERVER SIDE –
On adding the comment, the client sends the user ID, paper ID, page number, positions and the comment to the server. Server will automatically generate a comment ID and the update time for inserting record into database.

B) CLIENT SIDE –
The user can add comments on the paper by long-pressing on the screen. The client device will then create a comment button at that location, and ask the user to input the comment. After the user input the comment, the client device will send some information, including location of the comment, content of the comment, paper ID, user ID, etc., to the web server to update the database.

- Pseudo code -

IF (add comment)
{
  CHECK the location
  ASK the user to input the comment
  CREATE a new button for comment
  SEND the information to the web server
}
I. INTRODUCTION

The detection of pedestrians is a mandatory requirement for future driving assistance systems, since having the capability of avoiding crashes with pedestrians is essential for effectively aiding the driver in urban environments.

This work presents the vision-based system for detecting pedestrians implemented on the ARGO vehicle. ARGO is an experimental autonomous vehicle developed by the University of Parma, equipped with a vision system, a processing engine, and automatic steering capabilities [1]. The main target of the ARGO Project is the development of an active safety system which can also act as an automatic pilot for a standard road vehicle.

Vision-based pedestrian detection in outdoor scenes is a challenging task even in the case of a stationary camera. In fact, pedestrians usually wear different clothes with varying colors that, sometimes, are barely distinguishable from the background (this is particularly true when processing black and white images). Moreover, pedestrians can wear or carry items like bags, umbrellas, and many others, which give a broad variability to their shape.

When the vision system is installed on-board of a moving vehicle additional problems must be faced, since the observer's ego-motion entails additional motion in the background. An algorithm is used to verify the presence of pedestrians in these areas through an innovative application of a shape detection technique. This method allows the identification of pedestrians in various poses, positions and clothing, and is not limited to walking people.

This paper is organized as follows. Section 2 introduces the attentive algorithm aimed at the determination of the areas of interests, and shows its results. Section 3 describes the agent-based algorithm used for the validation of the candidates and presents the preliminary results of this shape detection technique. Section 4 ends the paper with some final remarks.

II. ATTENTIVE VISION

As a first processing step, attentive vision techniques are applied to concentrate the analysis on specific regions of interest only. In fact, the aim of the low-level part of the processing is the focusing on potential candidate areas to be further examined at a higher-level stage in the following steps.

The areas considered as candidate are rectangular bounding boxes which:

A) SERVER SIDE –

On deleting the comment, the client sends the paper ID, page number and position as request. Server will check return a list of usernames, comments and comment IDs from database to the client.

Figure 6.10 Screen capture of adding comment

Show comments
B) CLIENT SIDE –

Before showing a paper, the client device will ask the web server for the previous comments of that paper. The client device will then create the comment button at the corresponding location, and load the previous comment content. The user can read the comment added by himself or others. Besides, the comment buttons are translucent so that not to block the text of the paper.

- Pseudo code -

SHOW paper

CREATE buttons for all previous comments

IF (click on a comment button)

{  SHOW the previous comment contents

    SHOW the button of reply and delete  }

Figure 6.10 Screen capture of showing comment
**Reply comments**

A) **SERVER SIDE** –
The client sends the user ID, paper ID, page number, positions and the comment to the server. By checking the database, server will recognize that there is a discussion invoked on the same point before. It will hence add the comment with an additional entry “ReplyTo”, which records the comment ID the new comment replies to.

B) **CLIENT SIDE** –
User can reply on any comment added by himself or others. After replying, the message will be sent to the web server to update the database.

- **Pseudo code** -

    IF (click reply button)
    {
        ASK user to input the reply message
        SEND the information to the web server
    }
**Delete comments**

*A) SERVER SIDE* –

The client sends the comment ID to the server. With reference to the database, if the CID is the first comment invoking the whole discussion, all the comments of the same discussion will be deleted. Yet, only the comment of that CID will be deleted. Meanwhile, the other comment which have “ReplyTo” points the deleted CID with be redirected to the previous comment that the reply to point to.

*B) CLIENT SIDE* –

User can delete his own comments and replies, the information will be sent to the web server to update the database.

- Pseudo code -

```
IF (delete comment)
{
    SEND the information to the web server
}

IF (delete reply)
{
    SEND the information to the web server
}
```
6.3. Module 3: Research Log

6.3.1. User accounts

Login to system

A) SERVER SIDE –
When the user logins to system, the client sends the user ID and password to the server. The server will check if the login succeeds, the user name will be returned to the client.

B) CLIENT SIDE –
User is required to login to use this application, which is required to input the login ID and password, the client devices will then send these information to the web server to check if the login validates. If it succeeds, the user is able to choose a paper to read and use the application normally, otherwise, an error message about login failure will be shown, and the client device asks the user to input the information again.
- Pseudo code -

IF (open the application)
{
    Ask the user to input login ID and password
    Send the login ID and password to web server
    GET the result from the web server
    IF (validate)
    {
        ALLOW the user to use the application
    }
    Else
    {
        SHOW error message
        ASK the user to login again
    }
}
If the user want to exit the application, the client device will ask the user to confirm his exiting. If the user confirms to exit, he will logout from the application and the application will close. Otherwise, the client device will return to the previous view and let the user continue to use.

Figure 6.12 Screen capture of login fail
**Pseudo code**

IF (choose exit)

{ ASK the user to confirm

   IF (confirm)

   { EXIT the application }

   ELSE

   { STAY on that page  }

}

*Figure 6.13 Screen capture of exiting*
7. Testing and Evaluation

In this section, we are going to show the testing result and evaluation on them. We will test the server and client side separately. Both of them focus on different aspects. For the server, the testing focuses on the parsing performance of the papers. While for the client, the testing focuses on the functional performance of the application in the tablet devices.

7.1. Server Side

7.1.1. Testing Methodology

To test the parsing performance, we modified the managing panel designed for the administrator to capture the data easily. With the panel, the data call be represented in the table form, so as to make everything clearly.

<table>
<thead>
<tr>
<th>Paper ID</th>
<th>Upload Time</th>
<th>Title OK</th>
<th>Author OK</th>
<th>Abstract OK</th>
<th>Keywords OK</th>
<th>Ref OK</th>
<th>Des</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>2011-12-20 20:42:11</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>22</td>
<td>2011-12-20 20:42:09</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>23</td>
<td>2011-12-20 20:42:07</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>24</td>
<td>2011-12-20 20:42:05</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>25</td>
<td>2011-12-20 20:42:04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>26</td>
<td>2011-12-20 20:42:03</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

![Admin Panel on web server](image)

Figure 7.1 Admin Panel on web server

There are eight columns designed for the testing.
**Data Viewing**

The paper file can also be viewed by clicking the Paper ID, while the "IsParsed" column provides a link to view the parsing results. The page shows all the entries that parsed into the database.

<table>
<thead>
<tr>
<th>Title</th>
<th>Camera calibration and 3D reconstruction from a single view based on scene constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authors</td>
<td>Guanghui Wang, b, *, Hong-Tat Tung, Zheqin Huo, Fuchao Wub</td>
</tr>
<tr>
<td>Abstract</td>
<td>This paper mainly focuses on the problem of camera calibration and 3D reconstruction.</td>
</tr>
<tr>
<td></td>
<td>Camera can be obtained from the vanishing points of three mutually orthogonal directions.</td>
</tr>
<tr>
<td></td>
<td>However, planes defined by two of the vanishing directions in the structured scenes.</td>
</tr>
<tr>
<td></td>
<td>It is proved in this paper that if equal length or with known length ratio in space,</td>
</tr>
<tr>
<td></td>
<td>the camera can be calibrated under the widely accepted assumption projection matrix.</td>
</tr>
<tr>
<td></td>
<td>With respect to a given world coordinate system. Furthermore, several methods and scene</td>
</tr>
<tr>
<td></td>
<td>constraints. Thus, a scene structure can be reconstructed by combining the planar patch.</td>
</tr>
<tr>
<td>Keywords</td>
<td>Camera calibration</td>
</tr>
<tr>
<td></td>
<td>3D reconstruction</td>
</tr>
<tr>
<td></td>
<td>Absolute conic</td>
</tr>
<tr>
<td></td>
<td>Circular points</td>
</tr>
<tr>
<td></td>
<td>Single view modeling</td>
</tr>
<tr>
<td>References</td>
<td>index = 1</td>
</tr>
<tr>
<td></td>
<td>Ref = R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge</td>
</tr>
<tr>
<td></td>
<td>University, 1997</td>
</tr>
<tr>
<td></td>
<td>index = 2</td>
</tr>
<tr>
<td></td>
<td>Ref = O.D. Faugeras, Three-dimensional computer vision: a geometric viewpoint, MIT Press</td>
</tr>
<tr>
<td></td>
<td>index = 3</td>
</tr>
<tr>
<td></td>
<td>Ref = Y. Horii, K. Anjo, K. Arai, Tour into the picture, in Proceedings of SIGGRAPH,</td>
</tr>
</tbody>
</table>

**Result Recording**

There are five columns available to mark the tick or cross for each entry. After checking the results, we mark down whether the results given are successful or not.

**Remarks Marking**

There is a "Des" to show the remarks. We check the results after parsing, then try to find out the reason of failure and mark it down. The remarks can be edited by clicking the pencil icon on the right.
7.1.2. Data Description

We have parsed twenty PDF files for testing the parsing performance. The following figure shows our results:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>2011-11-29 20:42:17</td>
<td>1</td>
<td>✔️</td>
<td>❌</td>
<td>❌</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>30</td>
<td>2011-11-29 20:42:09</td>
<td>1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>29</td>
<td>2011-11-29 20:42:07</td>
<td>1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>❌</td>
</tr>
<tr>
<td>28</td>
<td>2011-11-29 20:42:05</td>
<td>1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>27</td>
<td>2011-11-29 20:42:04</td>
<td>1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>26</td>
<td>2011-11-29 20:42:03</td>
<td>1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>25</td>
<td>2011-11-29 20:41:58</td>
<td>1</td>
<td>❌</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>24</td>
<td>2011-11-29 20:41:56</td>
<td>1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>❌</td>
</tr>
<tr>
<td>23</td>
<td>2011-11-29 20:41:55</td>
<td>1</td>
<td>❌</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>22</td>
<td>2011-11-29 20:41:54</td>
<td>1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>21</td>
<td>2011-11-29 20:41:53</td>
<td>1</td>
<td>❌</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>20</td>
<td>2011-11-29 20:41:52</td>
<td>1</td>
<td>❌</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>19</td>
<td>2011-11-29 20:41:50</td>
<td>1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>18</td>
<td>2011-11-29 20:41:49</td>
<td>1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>17</td>
<td>2011-11-29 20:41:48</td>
<td>1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>16</td>
<td>2011-11-29 20:41:47</td>
<td>1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>15</td>
<td>2011-11-29 20:41:46</td>
<td>1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>14</td>
<td>2011-11-29 20:41:45</td>
<td>1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>13</td>
<td>2011-11-29 20:41:43</td>
<td>1</td>
<td>❌</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>12</td>
<td>2011-11-29 20:41:42</td>
<td>1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

Percentage of successful TITLE Parsing = 15/20 = 75%
Percentage of successful AUTHOR Parsing = 5/20 = 25%
Percentage of successful ABSTRACT Parsing = 11/20 = 55%
Percentage of successful KEYWORDS Parsing = 3/20 = 85%
Percentage of successful REFERENCE Parsing = 7/20 = 35%
7.1.3. Evaluation

**Title**
The percentage of successful cases (75%) is satisfactory. Throughout the fail cases, most of them resulted in the malfunction of getting font size function in PHP. Within 20 papers, there are totally 4 papers with header goes before the title and only one of them success. The fail case returns the header as the title. Besides, one of failure (PID=25) resulted because of the wrong arrangement of the title and author. The author in this paper goes before the title. The paper with PID=13 fails because the “pdftotext” function does not support the UTF-8 encoding.

**Author**
The percentage of successful cases (25%) is disappointing. Part of the failure is caused because of the failure of parsing title (since our implementation method of the authors is parsing the “authors” right after the “title”). For the majority of the fail cases, the failures are resulted because of the format of PDF file. Take the following case as examples:

---

**P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints**

Zdenek Kalal  
University of Surrey  
Guildford, UK  
z.kalal@surrey.ac.uk

Jiri Matas  
Czech Technical University  
Prague, Czech Republic  
mataj@cmp.felk.cvut.cz

Krystian Mikolajczyk  
University of Surrey  
Guildford, UK  
mikolajczyk@surrey.ac.uk

---
In this case, the authors are parsed to plain text as:

**P-N Learning: Booststrapping Binary Classifiers by Structural Constraints**
Idenek Kalal University of Surrey Guildford, UK
z.kalal@surrey.ac.uk

Jiri Matas Czech Technical University Prague, Czech Republic
matas@cmp.felk.cvut.cz

Krystian Mikolajczyk University of Surrey Guildford, UK
k.mikolajczyk@surrey.ac.uk

And this comes with difficulties that have been mentioned in the previous chapters: there has no standard to capture the difference between the authors’ name and the descriptions like the authors’ titles and contact information.

There is another case with the authors’ descriptions. Let’s take a look at the following case:

**An HOG-LBP Human Detector with Partial Occlusion Handling**
Xiaoyu Wang* Tony X. Han* Shuicheng Yan†
Electrical and Computer Engineering Dept. Electrical and Computer Engineering Dept.
University of Missouri National University of Singapore
Columbia, MO 65211 Singapore 117576
xw9x9@missouri.edu hantx@missouri.edu eleyens@nus.edu.sg
And we have the following text parsed:

An HOG-LBP Human Detector with Partial Occlusion Handling
Xiaoyu Wang\textsuperscript{1,2} – Tony X. Han\textsuperscript{1,2} – Shuicheng Yan\textsuperscript{1,2} – Anjiao An\textsuperscript{3}.

Electrical and Computer Engineering Dept. Electrical and Computer Engineering Dept. University of Missouri National University of Singapore
Columbia, MO 55211 Singapore 11757

It is clearly that the new line character is not parsed after the end of the last author’s name. This results that the authors’ names are recorded with their titles.

**Abstract**

The percentage of successful cases (55%) is just fair. The most significant failure the abstract includes some part of content or the figure caption. The reason that makes the parsing of abstract unsuccessful is the malfunction of the parsing program converting the PDF file to plain text (pdftotext). One of the case is shown as follow:
The above case shows that the content of the nest column goes before the “Introduction”. This results in wrong parsing of the abstract.

There are still few problems of parsing. One is that some paper in report-like format that contains table of content would make the server cannot locate the end of the scope of abstract. Another problem is that some abstract are stated without the word “Abstract”.

Keywords

The percentage of successful cases (95%) is almost satisfied. One reason is that not every paper has the keywords column. Even if the paper has keywords part, the keywords are few that avoid the case happened in the abstract – the probability of wrong parsing into fragments is relatively. The failure case is because of the wrong encoding of the PDF file to text file.


References

The percentage of successful cases (35%) is not satisfied. The reason of failure is quite similar to the cases as shown previously. Most of the papers are divided into two columns for better spacing. Yet, the parser sometimes mistakenly parses the right column first and make the order of the text got mess. Some of the reference items is parsed before the term “reference” and hence results in missing items.

There are also some problems for those reference items stated without the [ ] index. Since the “[ ]” are missing, we must rely on the new line character to break the reference item into pieces. However the new line characters are not always successfully parsed. This makes parse the reference items fail.

7.2. Client Side

7.2.1. Evaluation

For application in the client devices, we do testing on every functions, including login, opening and showing a paper, functions about references, functions about comments, zooming, finding texts, going to certain page and exiting the application.
### Login

<table>
<thead>
<tr>
<th>Case</th>
<th>Expectation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1   Empty Login ID and Password</td>
<td>Error Message</td>
<td>Same</td>
</tr>
<tr>
<td>2   Wrong Login ID</td>
<td>Error Message</td>
<td>Same</td>
</tr>
<tr>
<td>3   Wrong Password</td>
<td>Error Message</td>
<td>Same</td>
</tr>
<tr>
<td>4   Correct Login ID and Password</td>
<td>Login Succeed</td>
<td>Same</td>
</tr>
</tbody>
</table>

### Open and Show a Paper

<table>
<thead>
<tr>
<th>Case</th>
<th>Expectation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1   Choose one paper</td>
<td>Open the paper</td>
<td>Same</td>
</tr>
<tr>
<td>2   Choose another paper while reading one</td>
<td>Close the original paper and open the new one</td>
<td>Same</td>
</tr>
<tr>
<td>3   Show every part of the paper when moving it</td>
<td>Show every part normally</td>
<td>The last page cannot be read normally, which suddenly flick to the previous page</td>
</tr>
</tbody>
</table>
**Reference**

<table>
<thead>
<tr>
<th>Case</th>
<th>Expectation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Show the reference list after opening a paper</td>
<td>Show the reference list in the transparent panel</td>
</tr>
<tr>
<td>2</td>
<td>Show the reference button after opening a paper</td>
<td>Show the reference button beside every references in the paper</td>
</tr>
<tr>
<td>3</td>
<td>Show the information of the reference by clicking reference button</td>
<td>Show the information of that reference in an alert box</td>
</tr>
</tbody>
</table>

**Comment**

<table>
<thead>
<tr>
<th>Case</th>
<th>Expectation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Show the previous comments after opening a paper</td>
<td>Show the buttons of comments at the correct positions</td>
</tr>
<tr>
<td>2</td>
<td>Add new comment within the paper</td>
<td>Ask the user to input the comment, and add a new button</td>
</tr>
<tr>
<td>3</td>
<td>Add new comment very close to a previous one</td>
<td>Unable to add a new comment, and read the content of the previous one</td>
</tr>
<tr>
<td></td>
<td>Add new comment</td>
<td>Unable to add a new comment</td>
</tr>
<tr>
<td>---</td>
<td>-------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>5</td>
<td>Show the comment content by clicking the comment button</td>
<td>Show the comment content besides the button without being out of screen</td>
</tr>
<tr>
<td>6</td>
<td>Hide the comment content by clicking the comment button</td>
<td>Hide the comment content, and make the button translucent</td>
</tr>
<tr>
<td>7</td>
<td>Reply comment</td>
<td>Show the reply at the bottom of the comment</td>
</tr>
<tr>
<td>8</td>
<td>Cancel replying a comment</td>
<td>No new reply will be added</td>
</tr>
<tr>
<td>9</td>
<td>Delete a comment</td>
<td>Delete that comment and all its replies, and also the button</td>
</tr>
<tr>
<td>10</td>
<td>Delete a reply</td>
<td>Delete that reply</td>
</tr>
<tr>
<td>Case</td>
<td>Expectation</td>
<td>Result</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>Click the zoom in button</td>
<td>The paper is zoomed in, show all the view with the new scale</td>
</tr>
<tr>
<td>2</td>
<td>Click the zoom out button</td>
<td>The paper is zoomed out, show all the view with the new scale</td>
</tr>
<tr>
<td>3</td>
<td>Click the zoom fit-in-width button</td>
<td>The paper is zoomed to fit the width of the paper, show all the view with the new scale</td>
</tr>
<tr>
<td>4</td>
<td>Double click the screen</td>
<td>The paper is zoomed in, show all the view with the new scale</td>
</tr>
</tbody>
</table>

Figure 7.3 Screen capture of original position of comment

Figure 7.4 Screen capture of position of comment after zooming
### Find Text

<table>
<thead>
<tr>
<th>Case</th>
<th>Expectation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty string</td>
<td>Nothing will be found</td>
<td>The application has an error</td>
</tr>
<tr>
<td>A text does not exist in the paper</td>
<td>Nothing will be found</td>
<td>Same</td>
</tr>
<tr>
<td>A text existing in the paper</td>
<td>The texts in the paper will be bounded by three red lines</td>
<td>Same</td>
</tr>
<tr>
<td>Choose previous result</td>
<td>Move the view to the location of the previous result</td>
<td>Same</td>
</tr>
<tr>
<td>Choose next result</td>
<td>Move the view to the location of the next result</td>
<td>Same</td>
</tr>
<tr>
<td>Choose hiding all the result</td>
<td>Return to the normal view without showing the result of finding</td>
<td>Same</td>
</tr>
</tbody>
</table>

### Go to Certain Page

<table>
<thead>
<tr>
<th>Case</th>
<th>Expectation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty String</td>
<td>Error Message</td>
<td>Same</td>
</tr>
<tr>
<td>A number which is out of bound</td>
<td>Error Message</td>
<td>Same</td>
</tr>
<tr>
<td>A number which is within the bounding</td>
<td>Move the view to that certain page</td>
<td>Same</td>
</tr>
<tr>
<td>Choose going to the first page</td>
<td>Move the view to the first page</td>
<td>Same</td>
</tr>
<tr>
<td>Choose going to the last page</td>
<td>Move the view to the last page</td>
<td>Same</td>
</tr>
</tbody>
</table>
### Exit the Application

<table>
<thead>
<tr>
<th>Case</th>
<th>Expectation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Click exit button</td>
<td>Ask the user to confirm</td>
<td>Same</td>
</tr>
<tr>
<td>2 Confirm exiting</td>
<td>Exiting the application</td>
<td>Same</td>
</tr>
<tr>
<td>3 Cancel exiting</td>
<td>Return to the view before</td>
<td>Same</td>
</tr>
</tbody>
</table>
8. Conclusion and Future Work

8.1. Difficulties and Suggested Solutions

We are going to summarize all the difficulties we have encountered during the whole working period in phase 1. This part will be divided into the server and client side.

8.1.1. Server Side

**Difficulty**

On summarizing, we have found there are quite a lot problems raised in the parsing functions. There are significant problems should be highlighted.

Firstly, we face great difficulties in parsing the authors’ name and information. It is not standardized and extremely various in format, which makes us hard to check where to start to record the author’s name and where to end. Thought there are also unstandardized problem in the other parsing fields, they can be addressed by doing a more comprehensive survey and gather the pattern. Yet it is nearly impossible to apply the same method onto the author’s part, because it is so varied that enlarging the sampling spare still cannot help.
Secondly, there is problem in the parser which we currently adopted. The performance of Xpdf pdftotext program varies so much that makes the parsing fails frequently. The uncertainty often occurs when the two column formatted paper is parsed. In fact the percentage of fail parsing of the two column paper is quite large.

Thirdly, the format of paper varies. Especially for those scholarly paper is so long that it is written in a report-like format. Such format usually comes with a table of content. The format of those papers is quite different from our original design parsing method. Such paper is not easy to be successfully parsed in our design.

**Suggested Solution**

This project requires huge amount of data and statistic to support. After all, the paper is written for human beings to read but not the computer. The variation in format is ordinary. To reduce the influence the variation bringing to our system, a larger sample space and more comprehensive research data are required.

Nonetheless, besides improving the research result, some solutions are suggested to make the progress run smoothly.
Firstly, to ease the parsing of the author field, the introduction of PDFmeat to our system should be seriously considered. In fact we have investigated that the PDFmeat is really powerful in acquiring the metadata and is in high precision in parsing the title, authors and year etc., as we mentioned in chapter 3. The reason why we have not implemented the PDFmeat in phase 1 is that we cannot access the CLI interface successfully, which prevent us from using it. After the a long term of trying to parse the data by ourselves, we think that it would be wiser to adopt the PDFmeat. We have to emphases again that our goals to make the management simpler, and high accuracy in data parsing help our goal very much.

Secondly, we would like to seek for other PDF parser to replace the Xpdf which is very unstable. If the plain text parser functions smoothly, most of the parsing problem can be solved with our originally designed implementation methods.

Thirdly, we thought it is not easy to solve the problem of the report-like formatting papers. In fact the report-like formatting paper should be parsed with a brand new implementation method different from what we are writing. We considered that parsing papers in such format should be suspended until most of the functions of the application are fully-developed, since the proportion of report-like formatting paper is relatively minor.
8.1.2. Client Side

**Difficulty**

It takes time to understand the design of the API since its design is very complicated. It results in being very hard to modify it to fit our project. Besides, the API has some bugs originally, which have to be fixed. For example, in our testing on the application on the client device, we have found that the API has a problem of reading the last page of a PDF file. These bugs are very difficult to be fixed due to its complicated design.

Another difficulty is about the calculation of the position of buttons which we need to add on the paper. Again, due to the complex design, the calculation of the view is not simple. We need to be very clear about how the API works and how it calculates the view of the paper. Some build-in functions like zooming will also affect the calculation, and make it much more intricacy for us to add other features to it, e.g. buttons of references and comments. Because of this problem, our calculation is something wrong that results in erroneous position of buttons after zooming, which we have mentioned in the part of testing on application on the client devices.
**Suggested Solution**

One of our problems is the API cannot work properly when viewing the last page of any papers. Owning to the last page of a paper is quiet important that it shows the references of the paper, we would work on it by finding out the original bugs in the API, and try to fix it.

In addition, we would be much more careful to calculate the views of different elements. By fully understanding the computing logic of the API, it is possible to make all the views in their corresponding positions. However, it needs a large amount of time to completely understand its complex design. We would spend more time on it to realize the complicated design.

Moreover, we have found some bugs during our testing on the application on the client devices. We would try our best to fix them. For the problems of adding comments, we would set some limitation so that the user cannot add comments at some invalid positions. For the case of inputting empty string to find texts, the issue can be solved by add a restriction to it, so that the application would not have a error.
8.2. Conclusion

Our objective of the project is to make things simpler. Tablet devices take rising importance in the age. Although it is impossible for the tablet to fully substitute the traditional laptops, its raising prevalence should be valued, and we believe that tablet device can be introduced to academic aspects.

Yet, it is not easy to develop the application. No matter how much researches we have done to assist the implementation of the whole project, many obstacles still appear during the progress made.

For the server side, we learnt much concerning the format of a formal scholarly paper so as to teach the computer to read the paper. Many parsing problem aroused from the execution. Some of them are solvable, while the others depend on the tools support. We hope that the parsing problem can be addressed by the introducing the PDFmeat and new PDF to plain text parser next semester.

For the client side, most of them are executing problem on the interface. Several solutions are suggested and hoped to alleviate the problems. There are still numerous points can be fine-tuned and we envisage the interface can be further improved.
8.3. Future Work

8.3.1. Sharpen the performance of the parser

It is suggested to bring new tools back to the parser, like PDFmeat and a new “PDFtoText” tool. We will keep making effort on enhancing the performance and accuracy of parsing in order to minimize the workload of maintenance in the future. The more accurate the system is, the more users show their confidence to the whole implementation of the system.

8.3.2. Implement advanced features

We have designed many features to achieve our objectives and some of them have not been implemented, such as the searching functions and the user log functions. Implementing advanced features will therefore be our major concern of phase 2.

8.3.3. Improve the interface

On developing the new functions, we will try to notice whether the existing interface design suits the user’s need or it is really user-friendly. A good user interface can greatly improve the efficiency of the working progress, and the interface is not easily quantified for assessment. Improving the interface should be achieved by gathering more experiences for users.
9. Acknowledgement

We would like to thank the supervisor of our project, Professor Michael Lyu. who gives us valuable advices for our project and remainders to present well. Professor Lyu arranges a meeting with us every week for 1 hour so as to keep our progress up. We are not sure whether the final product can be achieved perfectly, but we feel confident to overcome the obstacles with the help of Professor Lyu.

Besides, we would like to thank Mr.Edward Yan in VIEW Lab. He provides us with facilities and technical support when we are setting up the server and gives us many suggestions on how to make the application more feasible.
10. Reference


[2] Android application: “gScholarReader” :
http://www.androidzoom.com/android_applications/lifestyle/gscholarreader_r_mdcf.html

[3] Android and iPhone Apps for Selected Research Resources:
http://www.lincolnlibraries.org/databases/Android_and_iPhone_Apps_for_Selected_Research_Resources.htm

[4] “Research paper” – WIKIPEDIA :


[6] "Writing Research Papers", Rice University:

[7] "Portable Document Format", WIKIPEDIA :


[14] “SAMSUNG GALAXY Tab 10.1”:
## Appendix : Research on Paper Format

<table>
<thead>
<tr>
<th>No.</th>
<th>Title</th>
<th>No. of col.</th>
<th>Is title the 1st line</th>
<th>Presence of</th>
<th>Content after Ref.</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>authors</td>
<td>Key.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>contact</td>
<td>Ref.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Interactive Light Field Display from a Cluster of Projectors</td>
<td>2</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td>The GPU as Numerical Simulation Engine</td>
<td>2</td>
<td>Y(url)</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td>Keywords</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>3</td>
<td>- WRONG CODING -</td>
<td>1</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>4</td>
<td>Data Fusion Development Concepts within complex Surveillance Systems</td>
<td>2</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Keywords</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>5</td>
<td>Development and characterization of a high-spatial-temporal-resolution foot-sole-pressure measurement system</td>
<td>1</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td>PAGE NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>REPORT</td>
</tr>
<tr>
<td>6</td>
<td>Counting People in Crowds with a Real-Time Network of Simple Image Sensors</td>
<td>2</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>PAGE NO</td>
</tr>
<tr>
<td>7</td>
<td>HIDING PRIVACY INFORMATION IN VIDEO SURVEILLANCE SYSTEM</td>
<td>2</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>8</td>
<td>A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique</td>
<td>1</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Keywords</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PAGE NO</td>
</tr>
<tr>
<td>9</td>
<td>Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>INDEX TERMS</td>
<td>AUTHOR INFO</td>
</tr>
<tr>
<td>10</td>
<td>Disc In-dash CD Changer Deck</td>
<td>MERGE</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>1) AUTHOR INFO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Japanese</td>
</tr>
<tr>
<td>11</td>
<td>Simple and efficient hash-based verifiable mixing for remote electronic voting</td>
<td>Complex</td>
<td>?</td>
<td>Y</td>
<td>N</td>
<td>Keywords</td>
</tr>
<tr>
<td></td>
<td>Francesc Sebé a,*, Josep M. Miret a, Jordi Puiglòs a, Jordi Puiggall b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Complex Layout</td>
</tr>
<tr>
<td>12</td>
<td>Weighted Sampling for Large-Scale Boosting</td>
<td>1</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>13</td>
<td>Online learning of robust object detectors during unstable tracking</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>14</td>
<td>P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>15</td>
<td>FACE-TLD: TRACKING-LEARNING-DETECTION APPLIED TO FACES</td>
<td>MERGE</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>INDEX TERMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td>REMARKS</td>
</tr>
<tr>
<td>No.</td>
<td>Title</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------------------------------------------------------</td>
<td>-------</td>
<td>-----------</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>16</td>
<td>Forward-Backward Error: Automatic Detection of Tracking Failures</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>17</td>
<td>Robust Recognition of Specific Human Behaviors in Crowded Surveillance Video Sequences</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>Y</td>
<td>Y</td>
<td>?</td>
</tr>
<tr>
<td>18</td>
<td>Vanishing Point Detection without Any A Priori Information</td>
<td>2</td>
<td>Y(Header)</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>19</td>
<td>E-voting 14 PUBLISHED BY THE IEEE COMPUTER SOCIETY</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>Design and Implementation of Secure Internet Based Voting System with User Anonymity using Identity Based Encryption System</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>21</td>
<td>Counting people in crowded environments by fusion of shape and motion information</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>22</td>
<td>FAST HUMAN DETECTION USING MI-SVM AND A CASCADE OF HOG-LBP FEATURES</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>23</td>
<td>View-insensitive Gender Recognition Using Local Binary Patterns</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>24</td>
<td>An HOG-LBP Human Detector with Partial Occlusion Handling</td>
<td>MERGE</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>25</td>
<td>A mean-shift tracker: implementations in C++ and Hume</td>
<td>1</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>26</td>
<td>A probabilistic framework for real-time head shape detection and tracking</td>
<td>MERGE</td>
<td>PAGE</td>
<td>Y</td>
<td>N</td>
<td>WITH MANY DES</td>
</tr>
<tr>
<td>27</td>
<td>A Robust Algorithm for Characterizing Anisotropic Local Structures</td>
<td>1</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>29</td>
<td>Advanced Data Fusion for Airport Surveillance</td>
<td>MERGE</td>
<td>PAGE</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>30</td>
<td>Automatic E-Comic Content Adaptation</td>
<td>1</td>
<td>Y(Header)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>31</td>
<td>Background Cut</td>
<td>1</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>32</td>
<td>Homography-based 2D Visual Tracking and Servoing</td>
<td>2</td>
<td>AUTHOR</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>33</td>
<td>Bokode: Imperceptible Visual tags for Camera Based Interaction from a Distance</td>
<td>MERGE</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

114
<table>
<thead>
<tr>
<th></th>
<th>Title</th>
<th>Authors</th>
<th>Keyword(s)</th>
<th>Complex Layout</th>
<th>Page No</th>
<th>Coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>Vision-based Pedestrian Detection: will Ants Help?</td>
<td>MERGE</td>
<td>N  N  N  Y</td>
<td></td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>35</td>
<td>Computing commonsense</td>
<td>MERGE</td>
<td>N  N  N  DIRECT N</td>
<td>1</td>
<td>AUTHOR INFO</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>N/A</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>37</td>
<td>Shoes Customization Design Tools for the “Diabetic Foot”</td>
<td>1</td>
<td>Y(Header) Y Y Y Keywords</td>
<td>Y</td>
<td>FOOTER</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Real-time Volumetric Reconstruction and Tracking of Hands in a Desktop Environment</td>
<td>1</td>
<td>N      Y      N      Y   KEY WORDS 1. a</td>
<td>DIAGRAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>Camera calibration and 3D reconstruction from a single view based on scene constraints</td>
<td>MERGE HEADER</td>
<td>Y Y Y Keywords</td>
<td>R</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>Coercion-Resistant Electronic Elections</td>
<td>1</td>
<td>N      Y      Y      Y   KEY WORDS 1. a</td>
<td>MANY PAGES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Dynamic Depth of Field on Live Video Streams: A Stereo Solution</td>
<td>MERGE HEADER</td>
<td>N N Y Keywords</td>
<td>1. a</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>N/A</td>
<td>N/A</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>43</td>
<td>Footwear Fit Categorization</td>
<td>1</td>
<td>N      Y      N      DIRECT N Y</td>
<td>AUTHOR INFO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>Computational Re-Photography</td>
<td>2</td>
<td>N      Y      N      DIRECT GENERAL TERMS &amp; KEY WORDS R DIAGRAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>N/A</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>46</td>
<td>N/A</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>47</td>
<td>Constructing Implicit 3D Shape Models for Pose Estimation</td>
<td>MERGE</td>
<td>N      Y      N      Y      N      Y      N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>Footwear Fit Categorization</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>49</td>
<td>Yet another improvement over the Mu–Varadharajan e-voting protocol</td>
<td>? PAGE</td>
<td>Y Y Y Keywords</td>
<td>Y</td>
<td>AUTHOR INFO</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>- WRONG CODING -</td>
<td>1</td>
<td>N      N      Y      Y      Keywords</td>
<td>R</td>
<td>PAGE NO CODING</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>Histograms of Oriented Gradients for Human Detection</td>
<td>MERGE</td>
<td>N      Y      Y      Y      N      Y      N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>N/A</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>53</td>
<td>Data Mining for Surveillance Applications</td>
<td>1</td>
<td>N      Y      Y      Y      N      N      N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>ON THE APPLICATION OF A THEORY FOR MOBILE SYSTEMS TO BUSINESS PROCESS MANAGEMENT</td>
<td>1 PAGE Y Y Y N Y N</td>
<td>COMPLEX LAYOUT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>N/A</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>56</td>
<td>Robust Real-Time Visual Tracking using Pixel-Wise Posteriors</td>
<td>1</td>
<td>N      Y      Y      Y      N      1. a</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>MapReduce Online</td>
<td>MERGE</td>
<td>PAGE Y N Y N Y N</td>
<td>PAGE NO</td>
<td>COMPLEX LAYOUT</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>Efficient Indexing for Large Scale Visual Search</td>
<td>MERGE</td>
<td>N      Y      Y      Y      N      Y      N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page</td>
<td>Title</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------------------------------------------------</td>
<td>-------</td>
<td>------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>59</td>
<td>Efficient particle filtering using RANSAC with application to 3D face tracking</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>60</td>
<td>Enhancing the Accessibility for All of Digital Comic Books</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>61</td>
<td>Ensemble Tracking</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>62</td>
<td>Estimating the Number of People in Crowded Scenes by MID Based Foreground Segmentation and Head-shoulder Detection</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>63</td>
<td>Estimation of Number of People in Crowded Scenes Using Perspective Transformation</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>64</td>
<td>FaceSeg: Automatic Face Segmentation for Real-Time Video</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>65</td>
<td>Fast automatic single-view 3-d reconstruction of urban scenes</td>
<td>MERGE</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>66</td>
<td>Fast automatic single-view 3-d reconstruction of urban scenes</td>
<td>MERGE</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>67</td>
<td>- WRONG CODING -</td>
<td>1</td>
<td>Y(Header)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>68</td>
<td>N/A</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>69</td>
<td>A new fast track-fit algorithm based on broken lines</td>
<td>MERGE</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>70</td>
<td>From Video Sequences to Motion Panoramas</td>
<td>MERGE</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>71</td>
<td>- WRONG CODING -</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>72</td>
<td>View-dependent exploration of massive volumetric models on large-scale light field displays</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>73</td>
<td>Histograms of Oriented Gradients for Human Detection</td>
<td>MERGE</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>74</td>
<td>Hive – A Petabyte Scale Data Warehouse Using Hadoop</td>
<td>MERGE</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>75</td>
<td>N/A</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>76</td>
<td>Putting Objects in Perspective</td>
<td>MERGE</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>77</td>
<td>N/A</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>78</td>
<td>N/A</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>79</td>
<td>Homography Estimation</td>
<td>1</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>80</td>
<td>Homography Estimation</td>
<td>MERGE</td>
<td>Y(Header)</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>81</td>
<td>Human Detection Using Oriented Histograms of Flow and Appearance</td>
<td>1</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>82</td>
<td>N/A</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

116
<table>
<thead>
<tr>
<th>No.</th>
<th>Title</th>
<th>Code</th>
<th>Header</th>
<th>Title</th>
<th>Keywords</th>
<th>ACM</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>83</td>
<td>Image interpolation for virtual sports scenarios</td>
<td>MERGE</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>84</td>
<td>Image interpolation for virtual sports scenarios - WRONG CODING -</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>85</td>
<td>Indexing Key Positions between Multiple Videos</td>
<td>MERGE</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>86</td>
<td>Mining High-Dimensional Data for Information Fusion: A Database-Centric Approach</td>
<td>MERGE</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>