

SPATIALLY AWARE COMPUTING FOR NATURAL INTERACTION

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Amin Roudaki

In Partial Fulfillment
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Computer Science

April 2013

Fargo, North Dakota

North Dakota State University
Graduate School

Title

SPATIALLY AWARE COMPUTING
FOR NATURAL INTERACTION

By

Amin Roudaki

The Supervisory Committee certifies that this *disquisition* complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Jun Kong

Chair

Dr. Kendall Nygard

Dr. Changhui Yan

Dr. Jin Li

Approved:

5/31/2013

Date

Dr. Brian M. Slator

Department Chair

ABSTRACT

Spatial information refers to the location of an object in a physical or digital world. Besides, it also includes the relative position of an object related to other objects around it. In this dissertation, three systems are designed and developed. All of them apply spatial information in different fields. The ultimate goal is to increase the user friendliness and efficiency in those applications by utilizing spatial information. The first system is a novel Web page data extraction application, which takes advantage of 2D spatial information to discover structured records from a Web page. The extracted information is useful to re-organize the layout of a Web page to fit mobile browsing. The second application utilizes the 3D spatial information of a mobile device within a large paper-based workspace to implement interactive paper that combines the merits of paper documents and mobile devices. This application can overlay digital information on top of a paper document based on the location of a mobile device within a workspace. The third application further integrates 3D space information with sound detection to realize an automatic camera management system. This application automatically controls multiple cameras in a conference room, and creates an engaging video by intelligently switching camera shots among meeting participants based on their activities. Evaluations have been made on all three applications, and the results are promising. In summary, this dissertation comprehensively explores the usage of spatial information in various applications to improve the usability.

ACKNOWLEDGEMENTS

I would like to thank my major advisor Dr. Jon Kong for his continued support, help, and direction. I wish to convey my gratitude to Dr. Kendall Nygard, Dr. Changhui Yan, and Dr. Jin Li for being on my graduate committee.

TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	iv
LIST OF TABLES.....	viii
LIST OF FIGURES.....	ix
1. INTRODUCTION.....	1
2. DISCOVERY OF WEB PATTERNS: A GRAPH GRAMMAR APPROACH.....	3
2.1. Introduction.....	3
2.2. Web Patterns.....	7
2.3. Approach Overview.....	12
2.4. Graph Generation.....	17
2.5. Pattern Specification and Validation.....	22
2.6. Grammar Induction.....	27
2.7. Experiment.....	38
2.8. Related Work.....	45
2.9. Conclusion.....	49
2.10. References.....	51
3. A LOW-COST SPATIALLY-AWARE MOBILE INTERACTION DEVICE.....	58
3.1. Introduction.....	58

3.2. Related Work	61
3.3. Approach Overview	66
3.4. System Architecture	68
3.5. Position Calculator	72
3.6. Accuracy Evaluation	79
3.7. An Application – An Architectural Plan	80
3.8. Empirical Study	84
3.9. Data Analysis and Results	94
3.10. Discussion of Results	100
3.11. Conclusion	102
3.12. References	102
4. A LOW-COST AND INTELLIGENT CAMERA MANAGEMENT SYSTEM	109
4.1. Introduction	109
4.2. Related Work	111
4.3. Approach Overview	115
4.4. System Architecture	118
4.5. Environment Sensing	120
4.6. Video Sources	124
4.7. Virtual Director	127

4.8. An Empirical Study.....	131
4.9. Data Analysis.....	133
4.10. Conclusion and Future Work.....	137
4.11. References.....	138
APPENDIX. IRB APPROVAL.....	142

LIST OF TABLES

<u>Table</u>	<u>Page</u>
2-1. The summary of web patterns	9
2-2. Experimental results	41
2-3. The complexity of spatial graphs and processing time.....	43
3-1. The movement sensitivity result.....	80
3-2. Study variables	86
4-1. Overall production quality result.....	134
4-2. Overview/close-up shots.....	137

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1-1. Levels of spatial awareness for each project	2
2-1. Pattern 1	8
2-2. Pattern 2	8
2-3. Framework overview	13
2-4. Graph generation and optimization	14
2-5. Hash function determining relationships between an object size & threshold value	19
2-6. Recalls and precisions of distance calculation approaches	20
2-7. Different combinations of containers	21
2-8. The spatial graph grammar formalism.....	24
2-9. The graph grammar for pattern 1	26
2-10. The graph grammar for pattern 2.....	27
2-11. Extraction of product information from a spatial graph	27
2-12. A spatial graph.....	29
2-13. The grammar induction	31
2-14. A sample spatial graph	32
2-15. The text equivalent of Image1	32
2-16. The graph grammar for text pattern.....	36
2-17. A sample-based grammar editor.....	38
2-18. Comparison between HTML elements and spatial graph nodes	42

2-19. Far distance between text elements	44
3-1. PhoneLens interacting with a paper document.....	60
3-2. The system architecture of PhoneLens.....	69
3-3. The mobile phone holder with infrared LEDs.....	69
3-4. View conversion	73
3-5. Distance approximation.....	75
3-6. The calculation of the wiimote angle based on the center point	75
3-7. The calculation of the gap	76
3-8. The eight points for evaluation.....	77
3-9. Evaluating the moving direction in z-axis.....	80
3-10. The lighting layer.....	82
3-11. An arrow points to the annotation	82
3-12. Measuring distance	83
3-13. Study design	90
3-14. Comparison of average time spent on tasks 1, 1', 2 and 2'	94
3-15. Percentage of the rating on each characteristic for each method (Categorized)	98
4-1. The SmartCamera system components.....	119
4-2. Matching a sound source with a participant	121
4-3. The PTZ camera on a rail	125
4-4. Adjusting a camera based on the speaker	127
4-5. The virtual director activity diagram	131

1. INTRODUCTION

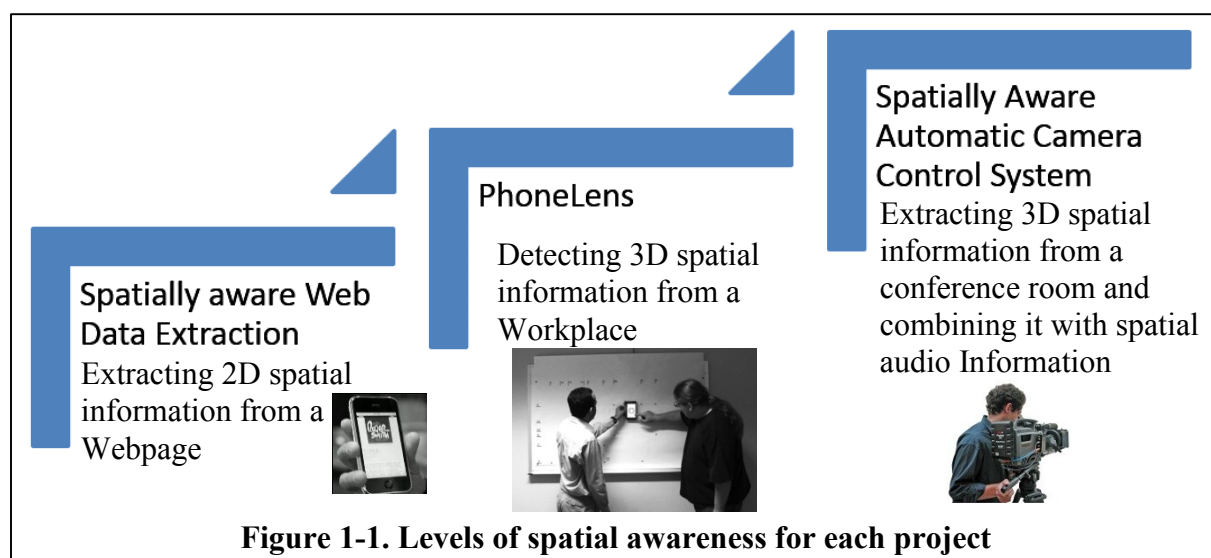
Humans use spatial information to understand relations among various objects. For example, when a person looks at a painting, the placement of different objects in the picture conveys a meaningful concept to him/her. Similarly, spatial information has been commonly used in a digital world. For example, the placement of various information blocks in a web page implies their semantic relations, e.g., semantically related blocks are in general placed in proximity. This observation leads us to explore the usage of spatial information in the human computer interaction to improve usability.

The two main communication channels between humans are speech and vision. However, most computer-based applications use Graphical User Interfaces (GUIs) to communicate with human users. A GUI presents the information mostly in 2D graphic, and a user can interact with the interface through keyboard and mouse. This interaction paradigm may reduce the usability of the communications between a human and a computer since it is not consistent with the natural communication methods that users are used to. With the fast development in hardware, natural user interfaces have been proposed, such as gestures based on multi-touch screens.

This dissertation focuses on using the spatial information to implement natural user interfaces, which allow users to access digital information without changing the normal communication methods in a physical world. Our main objective is to provide natural user interfaces so that users can focus on tasks instead of interface. Besides, we want to augment the physical world with the digital information and automate tedious tasks.

We have chosen three applications with different levels of spatial awareness. As shown on Figure 1-1, the first application extracts and analyzes spatial information from a Web page, i.e., a

2D workspace. In the second application, we implement an interface for augmented reality by detecting 3D spatial information in a paper-based workspace and accordingly overlaying digital information to a paper document. In third project, we combine both the 3D spatial information (i.e., the position of a person) and the audio information (i.e., the sound of a person) to implement a smart camera management system. In all of these projects, we explore the usage of spatial information to implement intelligent and natural user interfaces.



The remaining of the document is organized as follows. We have provided one chapter for each of three discussed systems. In each chapter first gave an introduction and then illustrated the system design. Afterward, included the related work and evaluation for each of the systems. Chapter 2 is on usage of a Spatial Graph Grammar to extract information from a Web page. Chapter 3 is on the PhoneLens system, which bring digital information using a mobile device on a traditional paper workspace. Chapter 4 presents the SmartCamera project, which enables full automation of video recording a meeting with multiple participants.

2. DISCOVERY OF WEB PATTERNS: A GRAPH GRAMMAR APPROACH

With the fast growth of World Wide Web, Web sites contain a large amount of information that makes it challenging to retrieve useful information. In order to facilitate information search with a high usability, the same type of information is in general displayed with a consistent layout (referred to as Web pattern) among different Web pages. The discovery of Web patterns can benefit many applications, such as enhancing Web browsing efficiency and extracting structured data. This paper presents a framework for specifying and discovering Web patterns based on graph grammars. In the framework, a common Web pattern is visually yet formally specified as a graph grammar. More specifically, a grammar induction engine complemented with an example-based grammar editor provides an interactive mechanism to semi-automatically define Web patterns. By abstracting a concrete Web page as a spatial graph that highlights the essential spatial relations between information objects, the discovery of instances of a Web pattern is implemented as a graph parsing process, according to a predefined graph grammar. We evaluated the framework on twenty-one e-commerce Web sites. The evaluation results are promising with a F1-score (i.e., a measure of a test's accuracy) of 97.49%.

2.1. Introduction

The Web provides a convenient and promising approach to accessing digital information. Meanwhile, wireless network and mobile devices make it possible to access information from anywhere at any time. However, exploring useful information on the Web becomes increasingly difficult as the volume of available information rapidly grow. In order to facilitate Web browsing, the same type of information is in general displayed with a consist layout among different Web pages [Zha04, Shn09, Kon12]. Those commonly accepted practices record successful designs, and

are referred to as Web patterns. The discovery and verification of Web patterns can benefit many applications, such as enhancing Web browsing efficiency and extracting structured Web data.

HTML, being the language to organize and present information on the Web, provides valuable clues on information structures within a Web page. For example, researchers have used HTML DOM structures to extract useful information from Web pages [Hog05, Pas09, Rei04]. However, HTML has been flexibly used to produce diverse DOM structures among different Web pages. For example, some Web designers may use tables to present tabular data while others use tables to divide the space into grids for page layouts. The diversity makes it challenging to discover Web patterns. The complexity of DOM structure further complicates the process of pattern discovery and verification. For example, even the DOM structure of Google homepage includes approximately 110 HTML tags.

Recently, layout-based analysis receives more and more attention [Che08, Sim05, Zhe07, Kon12] to reveal information organization. According to the usability principle in the Human-Computer Interaction, consistent layouts achieve a high usability [Shn09]. Accordingly, Web pages that include similar information in general are presented with a consistent layout even though their HTML source codes may be different. Therefore, layout based analysis addresses the diversity issue to a certain degree. However, existing layout-based approaches have limited applicability. For example, some approaches [Zhe07] require training while others are optimized for specific domains [Che08].

This paper presents a graph grammar based framework for specifying, discovering and verifying Web patterns. Due to the existence of common designs on the information representation [Zha04, Shn09, Kon12], Web patterns are defined according to the spatial properties among

information objects. Such a layout based Web pattern works with human's cognition of visual perception since human beings generally recognize and group information based on the spatial properties (such as size and location) among information objects [Che08]. Spatial relations in Web interfaces are essentially considered as a 2D property. Therefore, Web patterns can be precisely specified through graph grammar, which visually yet formally model structures and concepts in a 2-dimensional fashion [Roz97]. Graph grammar is also powerful in capturing structural variances among instances of a Web pattern through recursive computing. More specifically, a graph grammar is made of a set of grammatical rules (called *productions*). Each production defines local spatial relations among information objects while the complete graph grammar hierarchically glues those local spatial relations together. By formalizing a Web pattern as a graph grammar, the validation of instants of a Web pattern is implemented as a graph parsing process in a bottom-up fashion.

Since it is time consuming to manually summarize a Web pattern as a graph grammar, our framework supports a semi-automatic mechanism to increase the scalability and applicability. The discovery of Web patterns is realized through grammar induction. Briefly, a grammar induction algorithm automatically extracts the most common structure from sample Web pages and then represents the recognized structure hierarchically as a graph grammar. However, the grammar induction cannot always correctly recognize a Web pattern. Furthermore, the lack of domain knowledge in grammar induction may make an induced grammar hard to understand, due to program-generated names. In our framework, the grammar induction is complemented with a sample-based grammar editor. The grammar editor provides a graphic user interface, which allows users to edit or revise a graph grammar by directly manipulating information objects in the

screenshot of a Web page as a concrete example. Such a direct manipulation capability allows users to intuitively specify Web patterns by examples.

A Web pattern defines essential spatial relations among information objects. Therefore, given a Web page, it is necessary to extract atomic information objects from a Web page before pattern specification and validation. An important task in the framework is to abstract a concrete Web page as a *spatial graph*, where a node indicates an atomic information and an edge indicates a semantic relation between two information objects. Instead of using image processing techniques [Kon12], our approach recognizes information objects according to the dynamic DOM structure by rendering a Web page. The DOM-structure based recognition avoids the training required by technique, and is efficient to recognize basic information objects, such as texts, links or images.

In summary, this paper discusses a complete framework that supports the specification, discovery and validation of Web pattern. Our framework abstracts a Web page as a spatial graph that highlights the most essential spatial relations among atomic information objects. Based on the spatial graph, a Web pattern is visually specified as a graph grammar. The validation of instances of a Web pattern is then implemented through a graph parsing process along the line of the defined graph grammar. Instead of manually designing a graph grammar from scratch, we also propose an interactive procedure to specify Web patterns through a grammar-induction engine. We have evaluated the aforementioned framework on 21 Web sites to validate the instances of a Web

pattern. The results are promising, and the performance of our approach measured in terms of F1-Score¹ is better than the benchmark approach of MDR [Liu03].

2.2. Web Patterns

With the advent of Internet, successful designs on the Web have been summarized and recorded in various Web development guidelines, which promote standardization and encourage consistent layouts [Shn09]. By observing those guidelines, Web pages across Web sites of the same genre appear similar layouts. For example, closed related information is displayed in proximity and a title is placed above detailed contents. A consistent layout on a specific type of information is referred to as a Web pattern in our approach. The layout based specification on Web patterns addresses the issue of the irregular usage of DOM structures.

2.2.1. Patterns in e-commerce web sites

This paper selects E-commerce Web sites as a case study to go through our framework. The main contents in E-commerce Web sites are displaying various products. The well-structured nature of product information [Hui06, Wu06] makes it appropriate to investigate Web patterns. We have investigated 42 E-commerce Web sites, and summarized two patterns from those Web sites (refer to Table 2-1). In the following description, those two patterns are used as a case study to illustrate our framework throughout the paper.

¹ “In statistics, the F1 score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct results divided by the number of all returned results and r is the number of correct results divided by the number of results that should have been returned. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0.” [Wik10]

2.2.1.1. Pattern 1

As shown in Figure 2-1, Pattern 1 presents product information from left to right. A product picture is displayed at the left side, and other descriptive textual information is presented at the right side. The descriptive information includes several lines of texts, whose layout is organized from left to right and from top to bottom. Among those lines of texts, a title is represented as a hyperlink which allows users to access more detailed information.



Figure 2-1. Pattern 1

2.2.1.2. Pattern 2

As shown in Figure 2-2, Pattern 2 presents information vertically. Either a title or a product image is presented at the top, followed by a list of textual lines. The price is always listed at the bottom.

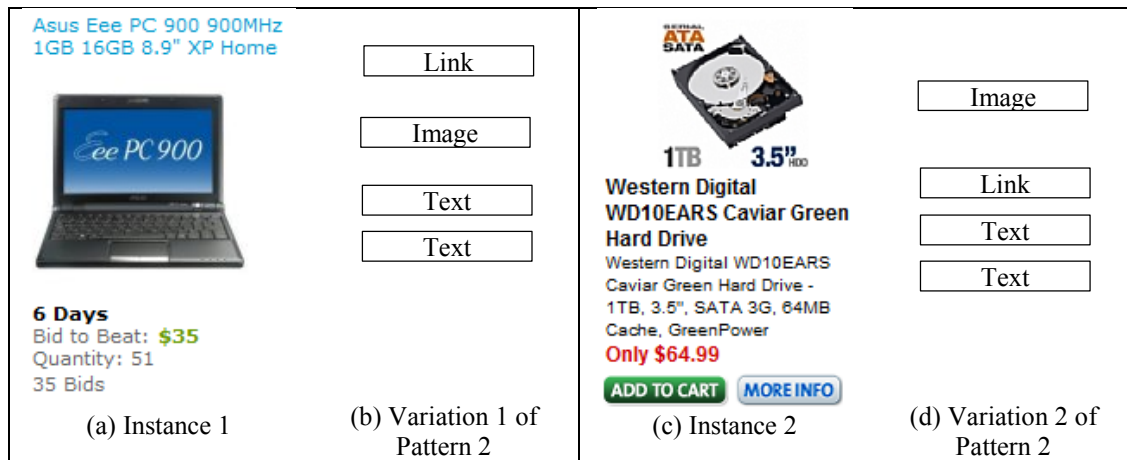


Figure 2-2. Pattern 2

In summary, each pattern prescribes the layout at a high level and records the essential spatial properties. Table 2-1 presents the list of Web sites we have evaluated. Especially, some Web sites apply two patterns together. The summary of Web patterns on product information validates the observation that a consistent layout on the same type of information is commonly used across different Web sites. In our approach, each pattern is formalized as a graph grammar, which implements a visual computing on the discovery and validation of Web patterns.

Table 2-1. The summary of web patterns

Domain Name	1	2	Domain Name	1	2
1 http://shopping.yahoo.com/	×		22 http://www.epicurious.com	×	
2 http://scistore.cambridgesoft.com/	×		23 http://www.cooking.com	×	×
3 http://shop.lycos.com	×		24 http://www.asiatravel.com		×
4 http://www.barnesandnoble.com	×	×	25 http://www.godaddy.com		×
5 http://www.borders.com	×		26 http://www.radioshack.com	×	
6 http://www.circuitcity.com	×	×	27 http://www.dell.com	×	×
7 http://www.compusa.com	×	×	28 http://www.macys.com		×
8 http://www.drugstore.com	×		29 http://www.buyflowersonline.com		×
9 http://www.ebay.com	×	×	30 http://www.buy.com	×	×
10 http://www.etoys.com		×	31 http://www.bestbuy.com	×	
11 http://www.kidsfootlocker.com		×	32 http://www.deals2buy.com	×	
12 http://www.kodak.com		×	33 http://www.6pm.com		×
13 http://www.newegg.com	×	×	34 http://www.bigdeal.com		×
14 http://www.nothingbutsoftware.com	×	×	35 http://www.shopzilla.com	×	
15 http://www.overstock.com		×	36 http://www.haggle.com		×
16 http://www.powells.com	×		37 http://www.shop.com		×
17 http://www.softwareoutlet.com		×	38 http://www.alibaba.com	×	
18 http://www.ubid.com		×	39 http://www.pricegrabber.com	×	
19 http://www.amazon.com		×	40 http://www.shopnbc.com		×
20 http://www.shopping.hp.com	×		41 http://www.shopstyle.com		×
21 http://www.qualityinks.com/		×	42 http://www.target.com		×

2.2.2. Usefulness of web patterns

Web patterns indicate commonly accepted standards to present and organize information on the Web. Therefore, a Web pattern provides structural and semantic hints underlying a Web page, which can be useful in many different applications.

- **Extraction of Structured Data.** With the dramatic increase of diversified information, it becomes increasingly difficult to explore useful information from the Internet. To efficiently discover knowledge from the big data on the Web, it is critical to extract meaningful contents from Web pages and organize extracted information in a structured format, i.e. *Web data extraction* [Kus00, Lae02]. One challenge is to design a generic algorithm that can efficiently discover how information is organized underlying a Web page. Recently, layout based extraction [Che08, Sim05, Zhe07] receives great attention, which is coherent with human's cognition of visual perception, since human generally recognize and group information based on the size and location of visual objects [Che08]. One use of Web patterns is the layout-based data extraction. A Web pattern records a common layout on a specific topic, and implies an intended information structure. One can first summarize a Web pattern from sample Web pages through a grammar induction engine. Then, a graph parser recognizes all instances of the recognized Web pattern. Each instance reveals an information block, which includes a set of information objects that are closely related to a single topic. Within each recognized instance, the user can further identify the semantics of each information object (such as title or price) based on spatial properties (such as position and size) and accordingly extract useful information.

- **Automatic Consistency Inspection.** Striving for consistency is one of the eight golden rules to design user-friendly interfaces [Shn09]. Therefore, it is a vital task to evaluate a Web page's conformance to Web design guidelines, i.e., consistency inspection. However, it is time consuming and error-prone to manually inspect such consistency. There are many different forms of consistency, and Web patterns can support automatic layout consistency inspection, which has various advantages, such as a reduction in cost and time [Ivo01]. Web designers can derive from Web design guidelines a Web pattern, which is formalized as a graph grammar through our framework. Then, given a Web page, a graph parser can recognize all instances of a Web pattern that satisfy required spatial properties according to Web design guidelines. In other words, the automatic consistency inspection is actually implemented through a graph parsing process. Any instance recognized by a graph parser indicates a consistent design. Otherwise, it means a violation of design guidelines.
- **Personalized Web Interfaces.** When viewing a Web page, users can have varied focuses. For example, about the description of a product, some users may first notice the price while others may first look at the brand. Due to diversified personal preferences, it is desirable to personalize Web designs. Web patterns can facilitate the personalization of Web layouts. More specifically, we can first recognize a Web pattern and analyze the semantics of information objects within Web pages of the same pattern. Then, an eye tracker, which can record an eye's moving path and the time that the eye stays on each position, is used to capture a user's browsing behavior. Based on user's browsing behavior and the semantics of information objects within a Web pattern, we can derive user's focus on the contents and accordingly personalize the interface.

2.3. Approach Overview

This paper discusses a toolset, i.e. *Visual Engineering for Web Patterns*, to specify and validate Web patterns. Our approach consists of three components as shown in Figure 2-3. The graph generation component abstracts a Web page as a spatial graph that simplifies the original Web page and highlights important semantic relations between recognized information objects. The graph generation proceeds in the following steps: (1) render a Web page on the screen, (2) recognize information objects and divide a Web page into different regions according to its DOM structure, (3) calculate semantic relations between recognized information objects based on the layout information, and finally (4) optimize the spatial graph. Based on spatial graphs, a Web pattern is visually specified in the form of a graph grammar. The interactive grammar design tool eases the effort of developing a graph grammar and improves the usability of our approach. Briefly speaking, a grammar induction engine automatically discovers a Web pattern from sample Web pages and summarizes it as a graph grammar, which can be further modified through a sample based grammar editor. The third component, i.e., the graph parser, parses the spatial graph of a Web page in the line of a graph grammar. The parsing process can efficiently identify information blocks that satisfy required spatial properties in a Web pattern. In the other words, the graph parser discovers and validates instances of a Web pattern according to a graph grammar.

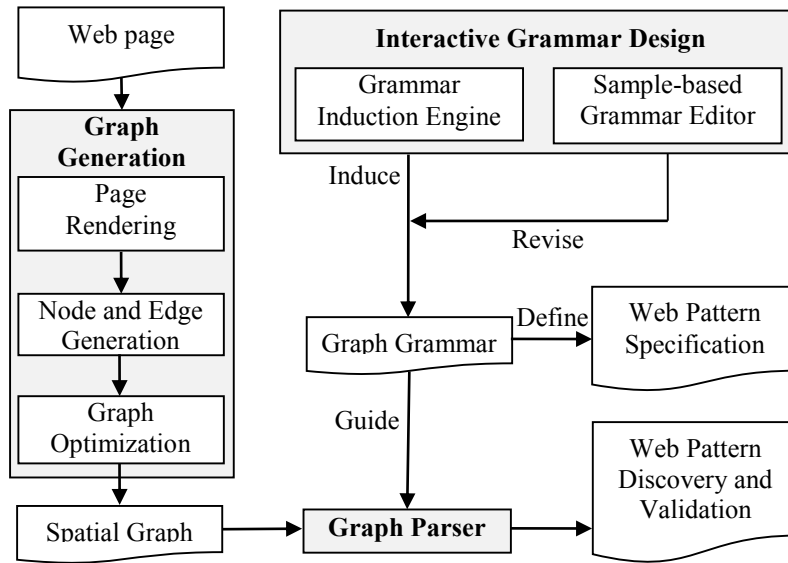


Figure 2-3. Framework overview

2.3.1. Graph generation

Graph generation is a critical step in our approach since it simplifies original Web pages and eliminates variations among different Web pages. More specifically, the graph generation process removes (1) style and layout elements, which do not include any real content, (2) advertisements and (3) menus in the border areas. The simplification effectively reduces the complexity of HTML pages and removes potential noises in the pattern recognition. The graph generation process proceeds in the following steps: Web page rendering, node and edge generation, and graph optimization.

The visual layout of a web page is determined by three variables, i.e. (1) the actual HTML source code that specifies the DOM structure of the page, (2) data items such as *text* and *picture* and (3) style sheets and client side scripts which are executed by a browser at runtime. Therefore, we can access all HTML elements, especially dynamic elements which are generated on the fly, only by actually rendering a Web page. Also, the page rendering determines the position and size

of each element. It is very likely that an HTML DOM Structure has some markup errors. According to [Che05] only about 5% of Web pages are “valid” in compliance with the HTML standard. In order to make the page rendering autonomously, our approach recovers a Web page from possible errors during the rendering process.

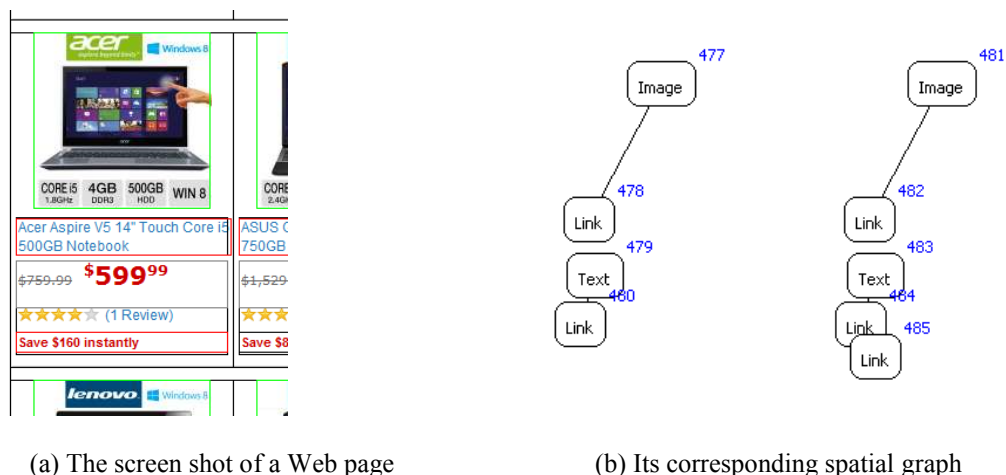


Figure 2-4. Graph generation and optimization

After achieving the dynamic and static HTML elements and their spatial properties, the second step generates a spatial graph in which a node represents an information object for data extraction and an edge indicates a close semantic relation between the pair of connecting nodes. For example, Figure 2-4 shows the screenshot of a Web page and its corresponding spatial graph. Instead of applying image processing to extract information objects, our approach recognizes information objects based on HTML tags. Contents enclosed within several adjacent HTML tags may be consolidated as a single atomic information object based on the DOM structure. Then, semantic relations are calculated among information objects. We have extensively investigated different Web sites and found that a small distance strongly indicates a close semantic relation between two objects. This observation is consistent with the Human Computer Interaction principle that closely related objects should be grouped together and placed in proximity [Shn09].

Considering different sizes of information objects, we proposed a novel approach to calculating distance in a 2D space. The last step in the graph generation is to optimize the generated spatial graph by removing noises (such as advertisements and menus). We can identify those noises based on their positions since they in general are placed in the border areas of a Web page. Another type of noise is repetitive small pictures.

2.3.2. Grammar design and pattern validation

A spatial graph simplifies its original Web page while keeping essential spatial properties among information objects. Based on spatial graphs, a Web pattern is visually specified through a graph grammar that defines computations in a multi-dimensional fashion through a set of *productions*. Each production consists of two parts: a left graph and a right graph, and the difference of which visually indicates the changes caused by a computation. In the pattern discovery and validation, the left graph in a production includes a composite information object, which is made of a set of atomic/composite information objects observing specified spatial relations in the right graph. Then, a complete graph grammar hierarchically integrates local spatial relations together to construct a Web pattern. Based on a graph grammar, a graph parser analyzes spatial properties in a spatial graph in the bottom-up fashion and recognizes in the spatial graph all substructures that are consistent with the graph grammar (i.e., recognizing instances of a Web pattern).

Instead of designing a graph grammar from scratch, we designed an interactive process to design a graph grammar visually and intuitively. Grammar induction [Ate06, Ate07] is an automatic process of generating a grammar from given example graphs, saving the effort of manually designing the grammar. The graph grammar induction engine used in our framework is

called VEGGIE [Ate06, Ate07] that extends the SUBDUE Grammar Learner [Jon03]. The induction process is based primarily on the idea of graph compression. Like popular file compression techniques, the compression process looks for common substructures within the graph, and compresses the graph by recording the substructure and replacing all instances by a marker. As substructures are captured, this process models a simple context-free grammar induction process. Like string-based data compression, graph-based data compression relies on a substructure matching technique to find instances within the graph. In addition to automatic grammar induction, a sample based grammar editor allows users to directly select one or more information objects in the Web page to make a production. This editor supports a direct manipulation on the grammar design that reduces the gap between a concrete Web pattern and an abstract graph grammar. With the help of this tool, even users without much training in graph grammars may design a graph grammar. In summary, a graph grammar brings the following benefits to define a Web pattern:

- A Web pattern defines essential spatial relations among objects at a high level. However, it is inevitable that there exist variations among instances of a Web pattern. For example, a pattern that displays a news story is to place a title on the top, followed by several paragraphs. Obviously, different news stories can have different numbers of paragraphs. A graph grammar is powerful to handle such variations by applying a production recursively.
- It is a one-time effort to design a graph grammar. Once a graph grammar is defined, it can be applied to different Web pages to validate instances of a defined Web pattern. Our approach decouples the specification of a Web pattern from pattern recognition.

Consequently, our framework can efficiently recognize and validate the evolving patterns by accordingly updating the graph grammar without changing source codes. Decoupling knowledge specification from the pattern validation process makes our approach more robust to handle pattern evolution and variations among different Web pages.

2.4. Graph Generation

The graph generation process first renders a web page and identifies all information objects (i.e. node generation) with their spatial properties (i.e. size and location). Afterward, it calculates the semantic relations between two nodes based on their distance. Finally, the generated spatial graph is optimized by removing useless sections and noises.

2.4.1. Node generation

Identifying information objects requires first loading and rendering a Web page, since client-side scripts are commonly used to generate dynamic contents in a Web page. We can access those dynamic contents only by actually rendering a Web page. Furthermore, spatial properties (i.e., the size and position) of each HTML element are calculated during the rendering process.

Contents are stored in three types of nodes, i.e. *image*, *text* and *link*. The contents enclosed in the `` or `<a>` tags are recognized as an *image* node or a *link* one, respectively. However, it is challenging to identify a *text* node since one complete sentence may be separated by several HTML tags and it is necessary to consolidate those information pieces together. For example, formatting and styling tags, such as ``, `
`, ``, ``, can divide a sentence into several pieces. In the graph generation, all those formatting and styling tags are removed and adjacent contents are consolidated as one single *text* node. This consolidation is consistent with human perception that people recognize the whole sentence as an atomic information object to

interpret a Web page. Furthermore, the consolidation reduces the number of nodes in the spatial graph and thus speeds up the parsing process. In addition to the above three nodes, we also introduce a *container* node, which implies a region within a Web page. A container node does not store real content. Instead, it is used to calculate semantic relations in the next step. Container nodes are derived from structural HTML elements, such as `<Table>` or `<Div>`.

2.4.2. Edge generation

After identifying atomic information objects as nodes, it is a critical to calculate semantic relations between information objects and use an edge to connect two nodes that are semantically related. In a two dimensional space, an information object can have an arbitrary spatial relation with adjacent nodes. A complete spatial parsing that analyzes different spatial properties in a graph could be time consuming. Our approach first derives the semantic relations only between adjacent nodes, and each semantic relation is represented as an edge in the spatial graph. Based on the derived semantic relations, we can limit the spatial parsing to objects that have semantic relations and thus reduce the search space to speed up the parsing process.

According to the HCI principle that related contents are placed in proximity [Shn09], we use the distance to derive the semantic relation between two objects. However, the shape of each information object is considered as a rectangle, the distance cannot be calculated between the two central points due to various sizes. We design a novel way to calculate the distance between objects a and b . We extend the size of object a to a certain degree. If at least two corners of object b fall in the extended rectangle of object a , objects a and b have a short distance that indicates a and b are semantically related. Accordingly, an edge is created in a spatial graph to connect objects a and b .

It is challenging to determine the threshold of extending an object. We have tried different means to determine the best value. First, we use an absolute value that produces a poor result since the size of an object does affect the calculation of distance. Then, we have designed a linear formula based on the size, which has an improved performance, but is still not optimal. Finally, we have designed a hash function between the size of an object and the extension value, as presented in Figure 2-5. Figure 2-6 presents the evaluation result on comparing those three approaches. The hash function based approach produces the best result in terms of both precision and recall².

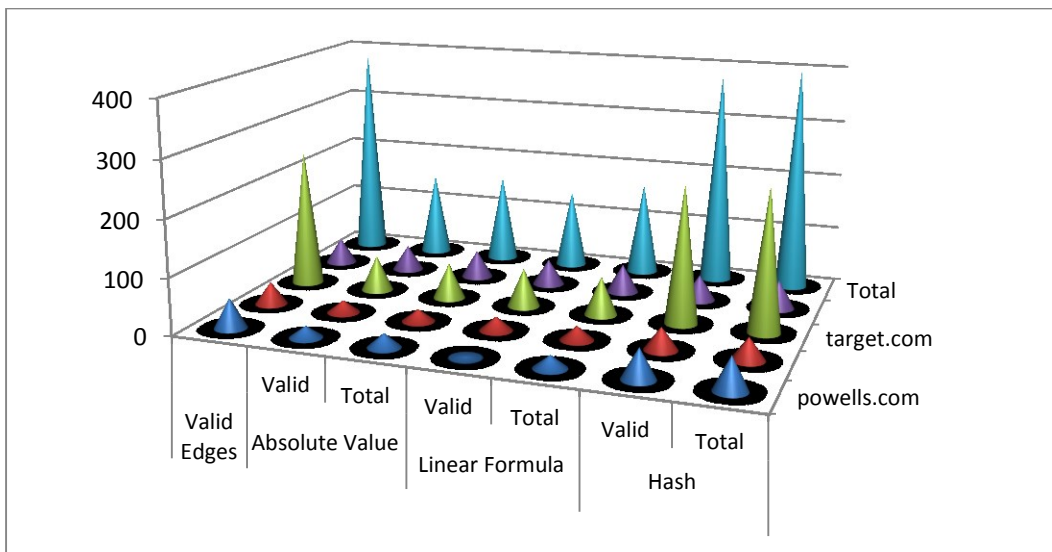


Figure 2-5. Hash function determining relationships between an object size & threshold value

² Definitions of Recall and Precision are provided at section 2.8.1.

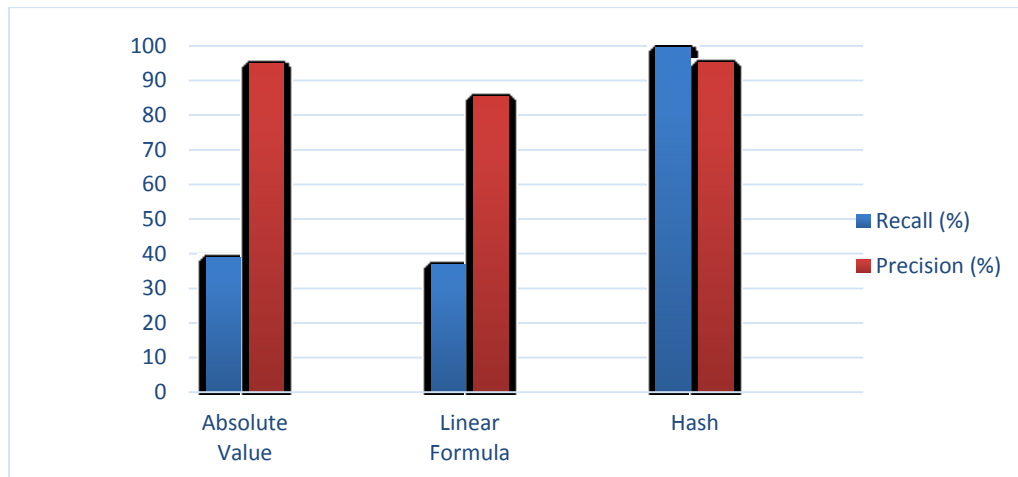


Figure 2-6. Recalls and precisions of distance calculation approaches

In addition to distance, the hierarchical DOM structure also provides hints to derive semantic relations. The HTML tags `<table>` and `<div>` are recognized as containers, and each container indicates a separate region. Since closely related objects are in general placed in the same region, semantic relations are limited to information objects which are in the same container or adjacent containers. According to the HTML DOM structure, we define a containment tree that specifies hierarchical relations among containers and information objects. For example, if an HTML element `<p>` (corresponding to a *text* object *t*) is enclosed in an HTML element `<table>` (corresponding to a container *c*), then the container *c* has a parent-child relation with the text object *t* in the containment tree. However, CCS style sheets or script languages could change the position of an information object and present it at a position which is completely different from the one defined by the DOM structure. Therefore, after rendering a Web page, we adjust the containment tree based on the real rendering. We traverse every object in the containment tree: if an object is completely contained in its parent container, we keep the current containment relation. Otherwise, the parent of this object becomes the smallest container that completely contains it.

Based on the containment tree, if an information object a has a semantic relation with an information object b , a must have one of the following containment relations with b , as presented in Figure 2-7, where a solid bold rectangle represents a container:

- As presented in Figure 2-7 (a), a and b are in the same container; or
- As presented in Figure 2-7 (b), the container of object a is the direct parent of the container of object b ; or
- As presented in Figure 2-7 (c), the container of object a is the sibling of the container of object b .

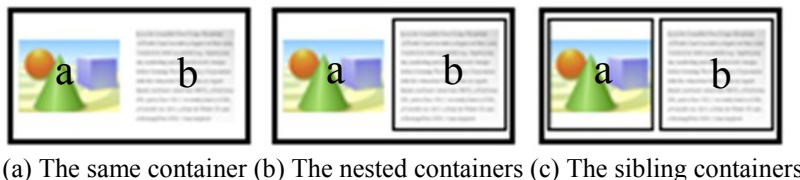


Figure 2-7. Different combinations of containers

2.4.3. Optimization

A Web page includes some information objects that are useless for data extraction, such as small icons. Those useless objects could add deviation to instances of Web patterns. For example, some Web designers place a small icon (such as the icon of a truck to indicate shipping) between a product image and product description. This small icon could establish two semantic relations, one between the product image and the icon, and the other one between the icon and the production description. Such connections make it slightly different from other instances of the same pattern that the product image is directly connected to the product description. We can eliminate the deviation by removing the small icon that does not really contribute to the real contents. We have manually evaluated 50 Web sites, and observed that small elements (such as icons) are in general

not important to real contents and commonly used for formatting purpose or for displaying extra information. Based on the observation, we have summarized two heuristic rules to remove small objects:

- Remove a *text* or *link* node whose size is smaller than 200 pixels;
- Remove an image node whose size is smaller than 1400 pixels.

A Web page is generally divided into 5 regions: top, bottom, left, right and center [Kov02]. Menus and advertisements are in general placed in the border areas. Removing menus and advertisements can reduce the size of a spatial graph without losing useful information, since they are irrelevant to data extraction. Instead of searching and removing individual information objects, our approach searches for a container that is at least 200 pixels in width and overlapping with at least 60 percent of a border area. If such a container is found, all the information objects contained by this container are removed. We are using containers to establish semantic relations among information objects, but containers themselves are not real data objects. Therefore, all containers are removed in the final spatial graph. Finally, any image that occurs more than twice is removed from the final spatial graph. These repetitive images are buttons, icons or styling images which are used for formatting and layout purpose without holding real contents. In summary, removing the above nodes can reduce the complexity of a generated spatial graph and eliminate noise in the data extraction process.

2.5. Pattern Specification and Validation

A concrete Web page is abstracted as a spatial graph, which not only simplifies the original page but also eliminates some variations. Based on spatial graphs, Web patterns are formally defined through graph grammars.

2.5.1. Spatial graph grammar

Graph grammars with their well-established theoretical background can be used as a natural and powerful syntax-definition formalism [Roz97] for visual languages, which model structures and concepts in a 2-dimensional fashion [Cox00]. The parsing algorithm based on a graph grammar can be used to check the syntactical correctness and to interpret the language's semantics.

Different from other graph grammar formalisms, the Spatial Graph Grammar (SGG) [Kon06] introduces spatial notions to the abstract syntax. In SGG, nodes and edges together with spatial relations construct the pre-condition of a production application. The direct representation of spatial information in the abstract syntax makes productions easy to understand since grammar designers often design rules with similar appearances as the represented graphs. In other words, using spatial information to directly model relationships in the abstract syntax is coherent with the concrete representation. Allowing designers to specify design knowledge in both structural and spatial properties simultaneously, the spatial graph grammar is ideal for specifying the extraction of structured records underlying a Web page. For example, the SGG production in Figure 2-8.a models the composition of an information object from both structural relations and spatial properties among relevant objects. In the above example, one structural relation (i.e. an edge between two nodes) indicates two objects are semantically closely related. According to the production in Figure 2.8a, a product information object is made of three information objects, i.e. link, image and text. The link has a close semantic relation with the image, and the image has one with the text. Furthermore, the link is placed above the image, and the image above the text. Such a display is defined through the spatial specification in Figure 2.8a. SGG also supports the syntax-

directed computation through action code. An action code is associated with a production, and is executed when the production is applied. Writing an action code is like writing a standard event handler in Java.

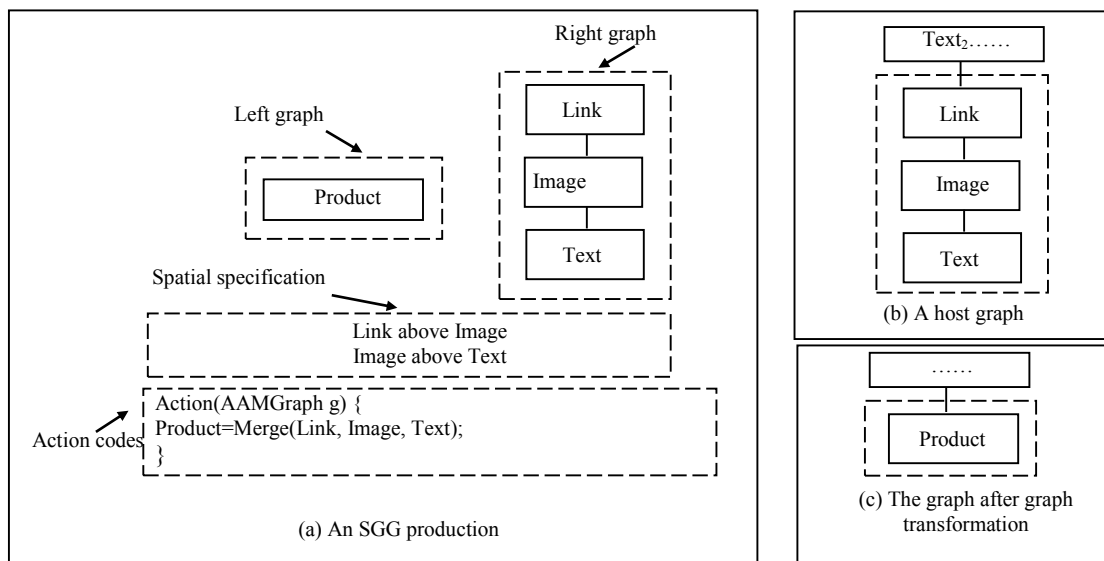


Figure 2-8. The spatial graph grammar formalism

Applying a production to a graph, usually called a *host graph*, is referred to as a graph transformation, which can be classified as an *L-application* (in a forward direction) or *R-application* (in a reverse direction). A *redex* is a sub-graph in the host graph which is isomorphic to the right graph in an R-application or to the left graph in an L-application. A production's L-application to a host graph is to find in the host graph a redex of the left graph of the production and replace the redex with the right graph. The language, defined as all possible graphs that have only terminal objects, can be derived through L-applications (i.e. a generating process) from an initial graph. On the other hand, an R-application is the reverse replacement (i.e. from the right graph to the left graph) used to parse a graph. In this paper, R-applications (i.e. a parsing process) are used to extract structured records from a Web page. For example, Figure 2.8.b shows a host graph, i.e. an abstraction of a Web page. The redex that matches the right graph in Figure 2.8.a is

highlighted in the dotted rectangle. After one R-application, we can extract a product record and the new host graph is updated as shown in Figure 2.8.c.

2.5.2. Validating web patterns

Based on the spatial graph, we formalize a Web pattern as a graph grammar. The left graph in a production includes one composite information object (other nodes in the left graph are context objects), and the right graph contains several atomic/composite information objects, which hold required spatial relations. In a spatial graph, the spatial relation of “short distance” is abstracted as an edge while other spatial relations are dynamically calculated during the parsing process. Based on the defined graph grammar, the SGG parser takes a spatial graph as input and recognizes in the spatial graph all substructures that are consistent with the defined the grammar. Figure 2-9 shows the graph grammar for Pattern 1 while Figure 2-10 presents the graph grammar for Pattern 2. In Figure 2-10, more specifically, Productions P1 to P3 specify the variations among instances of Pattern 2. P1 specifies that a product is made up of a title (i.e. a *link* object), a brief description (i.e. a *text* object), a production picture (i.e. an *image* object) and other descriptions (i.e. a *text* object). P2/P3 specifies that a product include a title, a product picture and other descriptions. All those objects are displayed vertically from top to bottom. In P2, a title is placed above an image and vice versa in P3. P4 is a recursive production that is used to recognize an arbitrary number of lines of product description.

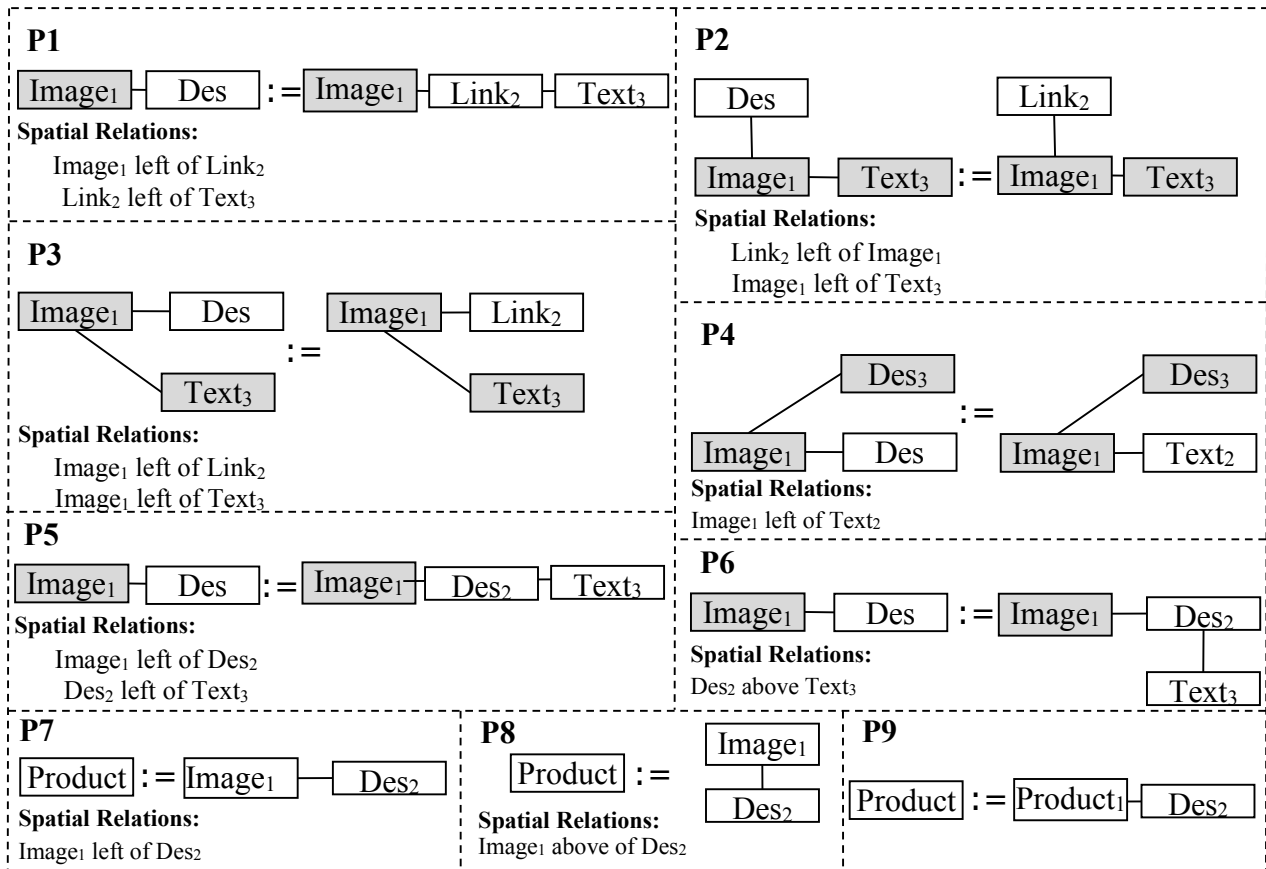


Figure 2-9. The graph grammar for pattern 1

Each graph transformation reveals a local composition. For example, the application of Production P1 in Figure 2-10 indicates that a composite object *product* consists of four information objects, i.e. *image*, *link* and two *text* objects. A sequence of graph transformations, i.e., the parsing process, assembles local compositions into a global hierarchical structure. For example, Figure 2-11.a presents a spatial graph. Three types of information objects, i.e. *image*, *link* and *text*, are recognized. An *image* object displays the picture of product; a *link* object represents a hyperlink that directs users to a separate page for more detailed information; and all other information objects are recognized as *text* objects, which provide textual description on a product. This spatial graph follows Pattern 2. Accordingly, the SGG parser applies the graph grammar in Figure 2-10 to the

spatial graph in Figure 2.11.a and the recognized instances of Pattern 2 are presented in Figure 2-11.b.

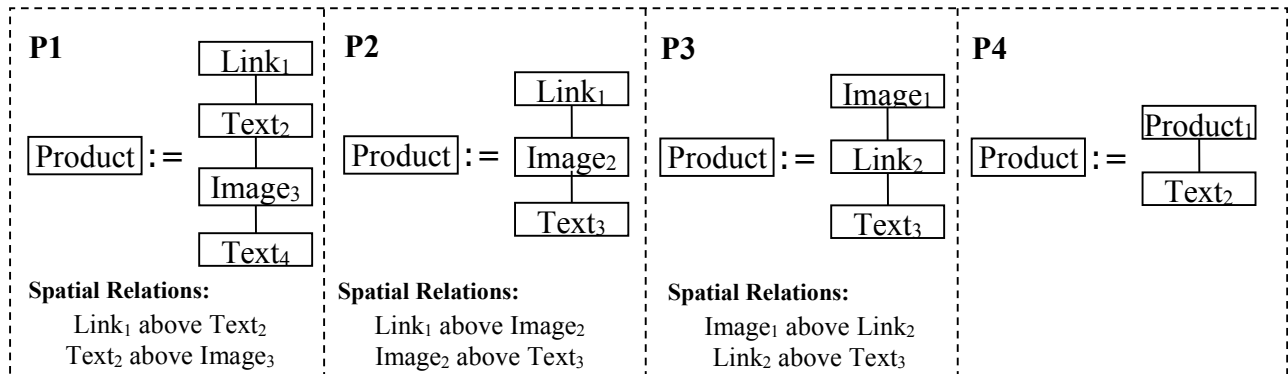


Figure 2-10. The graph grammar for pattern 2

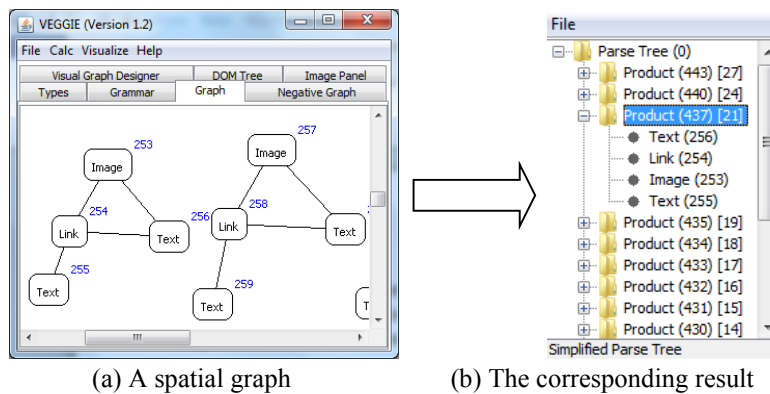


Figure 2-11. Extraction of product information from a spatial graph

2.6. Grammar Induction

Though a graph grammar and its associated parser provide a solid foundation for deriving the information organization underlying a Web page, it is time-consuming and challenging to design a graph grammar. Instead of writing a graph grammar from scratch, our framework includes a novel grammar induction engine for automating the grammar design and construction. Our grammar induction engine is featured by considering spatial properties in the induction process. In addition to a grammar induction engine, our toolset provides a sample-based grammar editor,

which allows grammar designers to intuitively design a graph grammar by directly manipulating information objects in a concrete Web page. In summary, the grammar induction engine, supplemented with a graphical grammar editor, improves the efficiency of grammar design.

2.6.1. A grammar induction engine

With the popularity of tree and graph structures in various applications, it is useful to automatically discover and formalize repetitive structures as a graph grammar, i.e., grammar induction. Most work focused on tree structures, such as [Liu03, Cre01, Ara03, Chu04, Rei04, Zha05], while only few approaches work on graphs. Node replacement is the basic operation in a grammar induction process, such as [Jon04, Ate06, Kuk07, Blo08]. Single node graphs are built up into more complex sub-graphs by adding edges and other nodes as they are present in the original graph. Each sub-graph, from simple to complex, is tested against the original graph for its compressive power—how well the sub-graph compresses the original graph. The best sub-graph is used to reduce the original graph by replacing all instances of the sub-graph with a single node. The compression consequently creates a graph grammar production for the sub-graph and replacement node [Ate06]. Instead of applying node replacements, Kukluk [kuk08] introduced an efficient algorithm to derive edge replacement graph grammars.

Our grammar induction algorithm has two distinct features from the previous approaches. First, to the best of our knowledge, none of existing grammar induction engines considers spatial properties of a graph in the induction process. However, spatial information may play an important role in real applications. The layout in which a line of texts is displayed above an image can have a different meaning from the layout which places an image above a line of texts. Therefore, our approach considers the spatial information in the induction process. In other words, our approach

considers two graphs identical only if they have the same structure (i.e., connections among nodes) and layout. Second, traditional node replacements in the induction process cannot keep the spatial relations among terminal nodes. For example, Figure 2-12 shows a spatial graph, in which each label near an edge indicates the spatial relation between the corresponding pair of nodes. If the *image* and *text* nodes are induced to a non-terminal node *P1*, the spatial relation between *P1* and the remaining node *Link* becomes undefined due to the irregular shape of *P1*. More specifically, previous approaches induce a graph grammar in a bottom to top fashion through node/edge replacements. Instead, our approach starts the induction from a single node and gradually extends this single node to more complex graphs. During the extension, we search for repetitive structures and induce them as productions. The extension based approach preserves spatial relations among terminal objects during the induction process.

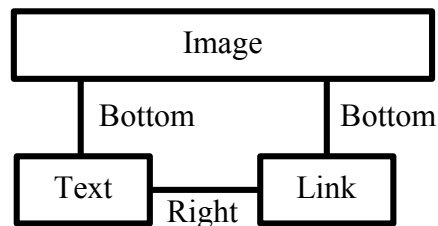


Figure 2-12. A spatial graph

Searching repetitive structures in a graph is an NP-Complete problem due to cycles. Our approach orders terminal objects based on left-right and top-bottom relations. Such an order allows us to search for structures in one direction to avoid cycles. More specifically, our induction approach proceeds in the following steps. First, each node in a graph is coded as a unique string, which specifies both the structural relations (i.e., the connection with its neighbors) and the spatial relations (i.e., right, bottom and containment) with those neighbors. Second, the string of each node is summarized as a java regular expression, which specifies a search pattern for strings.

Especially, nodes that have the same structural and spatial properties produce the same regular expression. In order to improve the search efficiency, the third step divides a large graph into multiple subgraphs. More specifically, given a node a , we gradually extend this node to the right and to bottom by following a directed path from node a . The extension continues until the longest path between any node in this subgraph and the original node (i.e., the node located in the top-left corner in the subgraph) reaches 10 or the extension reaches the node in the right-bottom corner. The extension is applied to every single node in the original graph. Consequently, given a graph with n nodes, the third step produces n subgraphs. Later, the search of repetitive structures is performed within each subgraph. Finally, the grammar induction (refer to lines 5-10 in Figure 2-13) is converted as a problem that searches for common java regular expressions (produced from the second step) in strings of subgraphs (produced from the third step). Figure 2-13 gives an overview of our approach. In forth Step, using the regular expressions which are created on second step, we count the number of subgraphs, which each regular expression can match. We select the top 20 most common regular expressions. These represent the most common search patterns in the graph. Afterward, in line 5-8 there is an iterative process, which basically start expanding those initial top 20 patterns by replacing the neighbors with other regular expressions. Then, recounting the number of subgraphs matches for each new pattern, and again selecting the top 20. This will expand the size of the patterns at each iteration, while always keeping the most common ones. Finally on Line 9 we use the last 20 regular expressions and calculate the area, which could be covered by each pattern on the actual Web Page. In this step we will select the pattern, which covers the most area on the Web page. Final step, creates a graph grammar based on selected regular expression on line 10.

1. A unique string is generated for each node in the graph G .
2. Summarize each string as a java regular expression
3. Extend each node in G to the right and to the bottom. Each produced subgraph is specified through a string.
4. Find the top 20 common java regular expressions.
5. for (i=1; i<MaxLevel;i++)
6. { extend the top 20 java regular expressions;
7. find the top 20 java regular expressions;
8. }
9. Among top 20 java regular expressions, search for the regular expression that covers the most area.
10. Convert the regular expression to a graph grammar

Figure 2-13. The grammar induction

2.6.1.1. Step 1. Create a graph string for each node

Our approach is distinct by converting a graph to a string, which consequently translates a graph search problem to a string search problem. Therefore, the first step is to convert each node in a graph to a unique string. In the conversion, we consider three types of spatial relations, i.e., *Bottom*, *Right* and *Containment*. In other words, given a node a , the corresponding string defines its adjacent nodes located to the right and to the bottom of node a . More specifically, each string includes two parts, separated by the symbol \sim . The first part presents the type and ID of node a , and the second part has three sections, separated by a semicolon. The first section presents the node types and IDs, which locate to the right node a , the second section contains nodes which are below, and the last section defines nodes that are contained by node a . The exclusion of relations left and top support an ordered search and extension in the following steps to avoid cycles. For example, Figure 2-14 shows a spatial graph. Figure 2-15 presents the generated strings for node $Image_1$ and $text_2$, respectively. In the above example, abbreviations of I, T, L represent Image, Text

and Link, prospectively; letter *N* means that there is no node in that section; and symbol # indicates the end of a string.

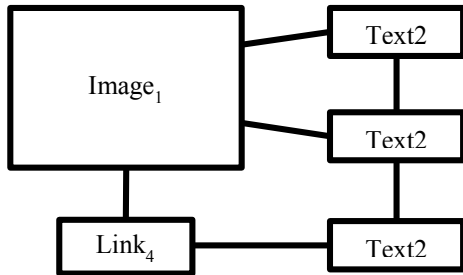


Figure 2-14. A sample spatial graph

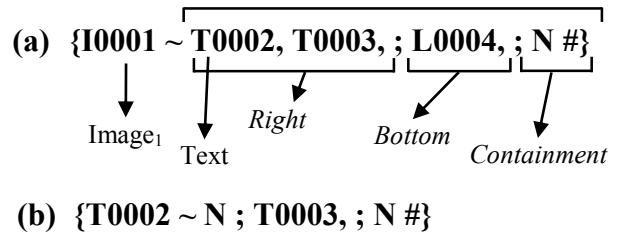


Figure 2-15. The text equivalent of Image1

2.6.1.2. Step 2. Summarize java regular expressions

In this step, we abstract the string of every node as a java regular expression and summarize a list of unique regular expressions. For example, the string of node *Image₁* in Figure 2-15.a is abstracted as the following regular expression.

$$\{ \setminus \{ I \setminus d \{ 4 \} \sim (T \setminus d \{ 4 \} \setminus ,) \{ 1 \} ? (T \setminus d \{ 4 \} \setminus ,) \{ 1 \} ? [\wedge ;] * ; (L \setminus d \{ 4 \} \setminus ,) \{ 1 \} ? [\wedge ;] * ; [\wedge ;] * \# \setminus \}$$

A regular expression keeps the node type, but it removes the detailed ID. Instead, each detailed ID is represented with a general format that includes 4 digit numbers. For example, node *Image₁* is abstracted as *I\d{4}* that indicates an image node with four digits as its ID. The above regular expression also defines required structural properties among *Image₁* and its adjacent nodes that have right, bottom or containment relations with *Image₁*. For example, *T\d{4}\,){1}* in the above regular expression indicates that there is one text node locating right to *Image₁*. *[^;]** is used as a wide card that allows variations when searching for an instance of this regular expression. In summary, a regular expression defines required structural and spatial relations among the node in the top-left corner in a sub-graph and its adjacent nodes. Graph strings that have the same spatial and structural properties are abstracted to the same regular expression. For example, any subgraph,

which has an *Image* node in the top-left corner with at least two adjacent *Text* nodes at the right side and at least one adjacent *Link* on the bottom, is considered as an instance of the above regular expression. Given the example in Figure 2-14, we can summarize the following regular expressions:

```
{\{I\d{4}~(T\d{4}\,){1}?(T\d{4}\,){1}?[^;]*;(L\d{4}\,){1}?(^;)*;[^;]*#\}
```

```
{\{T\d{4}~([^;]*;(T\d{4}\,){1}?(^;)*;[^;]*#\}
```

```
{\{L\d{4}~((T\d{4}\,){1}?(^;)*;[^;]*;[^;]*#\}
```

In the following description, set *Set_RegularExpression* is used to describe all regular expressions generated in Step 2.

2.6.1.3. Step 3. Expand graph strings

In order to have an efficient search, we divide a graph into a group of subgraphs. The division is implemented by extending the graph string generated in the first step. More specifically, a graph string in step 1 indicates a minimal subgraph, where the distance between the node in the top-left corner (i.e., Image₁ in the example in Figure 2-15.a) and any other node in the subgraph is one. In the following description, the node in the left-top corner is referred to as the origin of the subgraph. Then, we can extend this subgraph to the right and to the bottom by replacing each adjacent node of the origin in the graph string with the graph string of this adjacent node. For example, the graph string of node Image₁ in Figure 2-15.a can be extended to the following string:

```
{I0001~{T0002~N;T0003,;N#},{T0003~N;T0005,;N#},,{L0004~T0005,;N;N#};N#}
```

In the above extension, except the origin, each single node in the graph string is extended with a subgraph that takes this node as the origin. For example, node T0002 has been replaced by the subgraph of {T0002~N;T0003,;N#}. The first round of extension increases the maximum distance between the origin (i.e., Image₁ in the above example) and any other node in the extended string

to 2. The extension continues until the node in the bottom-right corner is reached or the maximum distance between the origin and any other node in the extended string reaches 10. This extension is applied to every graph string generated in Step 1. Therefore, given a graph with n nodes, the third step will generate n subgraphs. In other words, the original graph is divided into n subgraph and the search of a structured record is performed within each subgraph.

2.6.1.4. Step 4. Graph grammar induction

Based on the regular expressions produced in Step 2 and n subgraphs generated in Step 3, we search for the instances of each regular expression in those subgraphs, and select the top 20 regular expressions that have most instances among n subgraphs. Any of the top 20 regular expressions, which are recorded in set *Candidates_{GG}*, indicates a repetitive structure that occurs in a graph multiple times. However, each selected regular expression only specifies a minimal structure, in which the distance between the node in the top-left corner of the structure and any other node in the structure is only 1. We need to gradually extend the selected regular expression to identify a complete structure. Accordingly, we take those top 20 regular expressions as the initial set of repetitive structures, which are gradually expanded. In the expansion, we will traverse every node except the origin in the regular expression (i.e., the node in the top-left corner), and expand node a with every possible regular expression in *Set_{RegularExpression}*, where a regular expression has node a in the left-top corner. Since each expansion causes a new regular expression and several regular expressions may have the same node type in the top-left corner, expanding one regular expression will produce multiple new expanded regular expressions. For example, considering the example in Figure 2-14, expanding following pattern can produce a total of 7 new regular expressions.

$\{\backslash\{I\backslash d\{4}\sim(T\backslash d\{4}\backslash,)\{1\}?(T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;(L\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash\}$

Expanded regular expressions is presented as the following:

1. $\{\backslash\{I\backslash d\{4}\sim(10000\backslash\backslash\{T\backslash d\{4}\sim([^\wedge;]*;(T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash)10000\backslash,)\{1\}?(T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;(L\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash\}$
2. $\{\backslash\{I\backslash d\{4}\sim(T\backslash d\{4}\backslash,)\{1\}?(10000\backslash\backslash\{T\backslash d\{4}\sim([^\wedge;]*;(T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash)10000\backslash,)\{1\}?[^\wedge;]*;(L\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash\}$
3. $\{\backslash\{I\backslash d\{4}\sim(T\backslash d\{4}\backslash,)\{1\}?(T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;(10000\backslash\{L\backslash d\{4}\sim((T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash)10000\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash\}$
4. $\{\backslash\{I\backslash d\{4}\sim(10000\backslash\backslash\{T\backslash d\{4}\sim([^\wedge;]*;(T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash)10000\backslash,)\{1\}?(10000\backslash\backslash\{T\backslash d\{4}\sim([^\wedge;]*;(T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash)10000\backslash,)\{1\}?[^\wedge;]*;(L\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash\}$
5. $\{\backslash\{I\backslash d\{4}\sim(10000\backslash\backslash\{T\backslash d\{4}\sim([^\wedge;]*;(T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash)10000\backslash,)\{1\}?(T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;(10000\backslash\{L\backslash d\{4}\sim((T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash)10000\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash\}$
6. $\{\backslash\{I\backslash d\{4}\sim(T\backslash d\{4}\backslash,)\{1\}?(10000\backslash\backslash\{T\backslash d\{4}\sim([^\wedge;]*;(T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash)10000\backslash,)\{1\}?[^\wedge;]*;(10000\backslash\{L\backslash d\{4}\sim((T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash)10000\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash\}$
7. $\{\backslash\{I\backslash d\{4}\sim(10000\backslash\backslash\{T\backslash d\{4}\sim([^\wedge;]*;(T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash)10000\backslash,)\{1\}?(10000\backslash\backslash\{T\backslash d\{4}\sim([^\wedge;]*;(T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash)10000\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash\};(10000\backslash\{L\backslash d\{4}\sim((T\backslash d\{4}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash)10000\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash\}$

As shown on first expanded regular expression, node $T\backslash d\{4\}$ is expanded with the regular expression $\{T\backslash d\{3}\sim[^\wedge;]*(T\backslash d\{3}\backslash,)\{1\}?[^\wedge;]*;[^\wedge;]*#\backslash\}$, which has a text node as the origin. Each expanded regular expression is added to *Candidates_GG*. Number 10000 indicates the first level of expansion. On first 3 expanded regular expression we have just replaced one node, while on 4, 5, 6 we have replace two nodes, and on on 7 we have replaced all three nodes. In the provided example we only have one pattern which start with Text Node, but if we have more than one, then we need to also iterate over those and create more combinations based on those. After expansion, each regular expression in *Candidates_GG* is calculated with its instances in all subgraphs. We again select top 20 regular expressions based on number of instances, and each selected regular expression must have at least 4 instances in subgraphs. Only those selected regular expressions are kept in *Candidates_GG*. Then, we start the second round of expansion and verification. In order to match the maximum subgraph generated in Step 3, the expansion and verification are performed for 10 rounds. After all rounds of expansion and verification, we finally select the top 20 regular

expressions, and each indicates one graph grammar. Based on the observation that important information takes the main area in a Web page, we calculate the area that is covered by the instances of each regular expression, and choose the regular expression that covers the largest area as the final induced graph grammar.

A regular expression can consist of multiple levels. Starting for the deepest level (i.e., level 10), we extract productions in a bottom-up fashion. For example, given the following regular expression, we can summarize a graph grammar as shown in Figure 2-16.

```
{\{I\d{4}~(10000\{T\d{3}~[^;]*; (L\d{3}\,) {1}?[^;]*;[^;]*#\}10000\,) {1}?(T\d{4}
}\,) {1}?[^;]*; (L\d{4}\,) {1}?[^;]*;[^;]*#\}
```

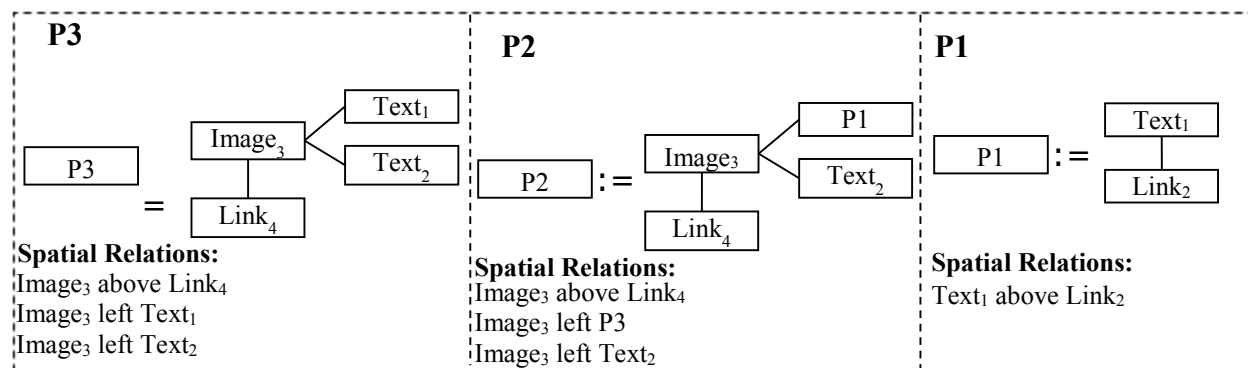


Figure 2-16. The graph grammar for text pattern

The first production P1 is extracted from the regular expression with the deepest level, i.e., `10000\{T\d{3} ~[^;]*; (L\d{3}\,) {1}?[^;]*;[^;]*#\}10000`. The second production (P2) is summarized based on P1. Since P1 is extended from a single text node, we also summarize P3 that includes a terminal node instead of substructure of P1.

2.6.2. Interactive grammar design

Due to the lack of domain knowledge, a grammar induction algorithm in general assigns recognized substructures with program-generated names, which make it hard for Web designers

to refine the induced grammar. In order to facilitate the automatic grammar induction, we develop a sample-based grammar editor, in which grammar designers directly manipulate the screenshot of a Web page to design a graph grammar.

A spatial graph that is abstracted from a concrete Web page simplifies the original DOM structure, and thus facilitates the spatial parsing to recognize instances of a Web pattern. However, such an abstraction introduces a gap between a spatial graph and its concrete representation in a Web page. A sample-based grammar editor bridges the gap by supporting grammar designers to design a graph grammar directly on a concrete Web page. We developed sample-based grammar editor that functions like a regular browser to display a Web page. In addition, the editor highlights each recognized information object in a rectangle, as shown in Figure 2-17. When a user clicks an information object, semantic relations related to this information object are displayed accordingly. In other words, the editor visualizes a spatial graph in a concrete Web page. Such a visualization instantiates the abstract organization of a Web pattern in a concrete context. A Web designer can construct a production by intuitively selecting several information objects in the visualization. According to user's selections, the selected information objects with semantic relations among them make up the right graph of a production.

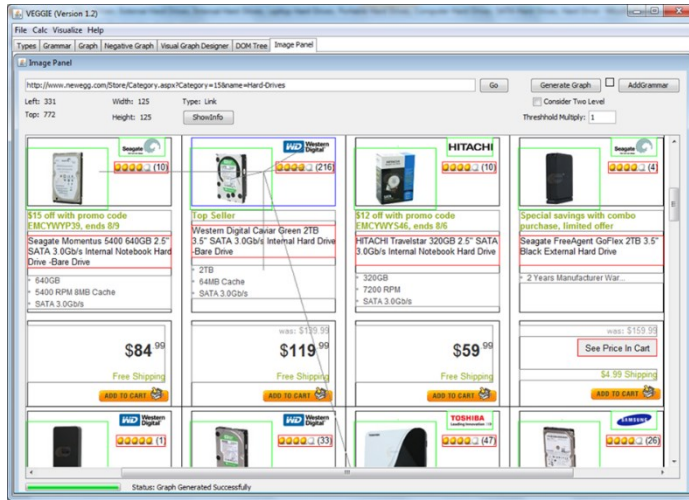


Figure 2-17. A sample-based grammar editor

2.7. Experiment

This section compares our approach with the state-of-the-art information extraction system, *Mining Data Records* (i.e. MDR) [Liu03]. MDR has been commonly used as a benchmark to compare with other approaches, such as ViNTs [Zha05b] and ViPER [Sim05], and is available to the public (<http://www.cs.uic.edu/~liub/WebDataExtraction/MDR-download.html>). We first discuss the design of the experiment, then present the results, and finally discuss the results.

2.7.1. Setup

The experiment is explained in the following sections.

2.7.1.1. Experiment web pages

Liu *et. al.* [Liu03] evaluated MDR on 46 Web sites. From those 46 Web sites, we eliminate the Web sites that are not accessible and not in the category of ecommerce, and finally select 21 Web sites as the test set.

2.7.1.2. Measurement

We measured the performance with the standard metrics:

$$recall = \frac{E_{correct}}{N_{total}}; \quad (\text{Equation 2-1})$$

$$precision = \frac{E_{correct}}{E_{total}}; \quad (\text{Equation 2-2})$$

Where N_{total} is the number of data records contained in a Web page; $E_{correct}$ indicates the total number of correctly extracted data records; and E_{total} denotes the total number of data records extracted from a Web page. We also calculated the F1-Score, which is the harmonic mean of $precision$ and $recall$ and is defined as $\frac{2 \times recall \times precision}{recall + precision}$. The F1-Score has been commonly used as a metric to evaluate the overall performance in many approaches [Che08, Lab09]. MDR is a general process that can recognize any repetitive structures within a Web page. Those recognized records may belong to different categories. Since those product-unrelated records may affect the recall and precision of MDR, we only calculate the number of product records recognized by MDR in order to compare two approaches fairly. In other words, structured records that are extracted by MDR but not related to products are not counted toward $E_{correct}$ and E_{total} . In addition, we evaluate the processing time and the complexity of the spatial graph in our approach. For our approach we had two options. First, we manually design a grammar similar to what discussed on section 2.5. Second using our automated grammar induction algorithm described at section 2.6.1. We have evaluated our approach using both methods and then compare all together.

2.7.1.3. Execution platform

We have evaluated the MDR and our approach on a desktop with a Core 2 Duo CPU 2.26 GHz and 4 GB RAM, running Windows 7 Professional.

2.7.2. Evaluation

2.7.2.1. Precision/Recall/F1-Score

The results are presented in Table 2-2. The recall of our approach is 99.5% for both manual and automated grammar, compared to 53.5% of MDR. The high recall rate in our approach indicates that visual analysis based on graph grammars is powerful to recognize structured records. The precision of our approach is 95.5% for manually designed grammar and 95.0% for automated grammar induction, which both are close to 97.9% of MDR. Our approach may falsely recognize some records, which are mainly caused by noise. For example, if an advertisement is placed in the central area and its overall layout is similar to pattern 2 (i.e. including a link, a picture and several lines of textual description that are displayed vertically); this advertisement may be recognized as a product record. Similar issue exists for automated grammar induction, besides since the grammar is more customized based on frequency of existing patterns in a webpage, it slightly increase the chance of falsely recognizing a product. In order to improve the precision, it is critical to improve the graph generation process by removing potential noise. F1-Score shows the overall performance. As shown in Table 2-2, our approach has a high F1-Score of 97.49% for manual grammar design and 97.16% for automated grammar induction, compared to 69.19% of MDR. In summary, the results indicate our approach has a good performance in terms of both precision and recall, and very high F1-Score.

Table 2-2. Experimental results

	Domain Name	# of Structured Records	MDR		VGE (Manual)		VGE (Auto)	
			Correct	Found	Correct	Found	Correct	Found
1	shopping.yahoo.com	15	0	0	14	14	14	14
2	scistore.cambridgesoft.com	13	13	13	13	14	13	15
3	shop.lycos.com	18	0	0	18	18	18	18
4	barnesandnoble.com	48	0	0	48	48	48	48
5	borders.com	27	27	32	28	29	28	30
6	circuitcity.com	5	5	5	5	7	5	7
7	compusa.com	18	0	0	18	21	18	21
8	drugstore.com	15	15	15	13	14	13	14
9	ebay.com	20	20	20	20	20	20	20
10	etoys.com	32	0	0	32	32	32	32
11	kidsfootlocker.com	29	29	29	29	29	29	29
12	kodak.com	20	0	0	20	20	20	20
13	newegg.com	20	0	0	20	26	20	26
14	nothingbutsoftware.com	24	24	24	24	24	24	24
15	overstock.com	18	18	18	18	18	18	18
16	powells.com	50	50	50	50	51	50	51
17	softwareoutlet.com	14	0	0	14	15	14	16
18	ubid.com	8	0	0	8	9	8	9
19	amazon.com	7	0	0	7	8	7	8
20	shopping.hp.com	5	5	5	5	5	5	5
21	qualityinks.com	24	24	24	24	26	24	26
	Total	430	230	235	423	443	423	451
	Recall/Precision		53.5%/97.9%		99.5%/95.5%		99.5%/95.0%	
	F1-Score		69.19%		97.49%		97.16%	

2.7.2.2. The complexity of spatial graphs

The graph generation process simplifies the original Web page by consolidating information pieces and removing noises. Figure 2-18 compares the size of a Web page to that of its corresponding spatial graph. The y axis indicates the size of a Web page/spatial graph that is measured in terms of the number of HTML elements or spatial graph nodes, and the x axis means the index of 21 Web pages. On average, the graph generation process produces the information that is of only 40% of the DOM's complexity. The detailed information about the sizes of each Web page and its corresponding spatial graph is presented in Figure 2-18.

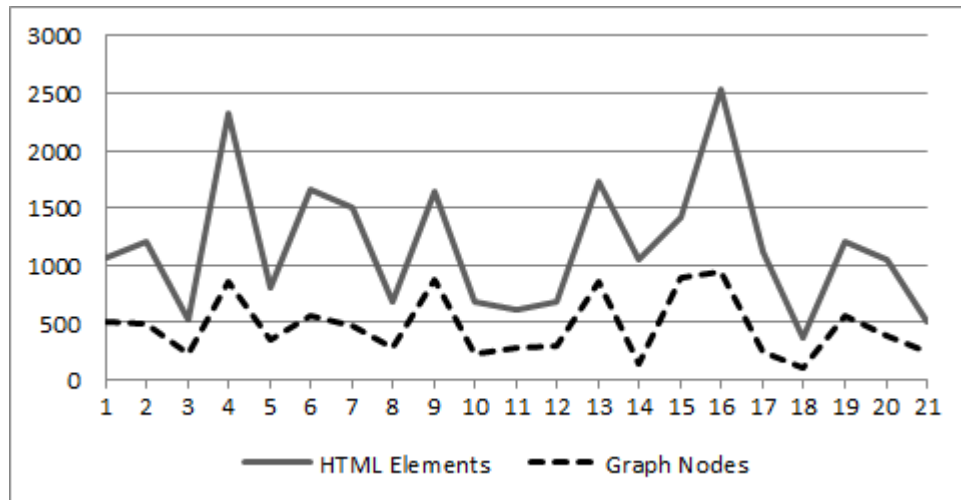


Figure 2-18. Comparison between HTML elements and spatial graph nodes

2.7.2.3. Processing time

Table 2-3 presents the processing time that is measured as the time of graph generation, data extraction time. The average graph generation time for an average Web page with 1160 HTML tags is less than 1 seconds. While data extraction will be less than 1.5 seconds on average. The processing time does not include the time used for downloading and rendering the page since it depends on the network speed. We were not able to record the MDR processing time since we can only access to the executable version of MDR. Due to the user interaction in MDR, we cannot precisely measure the processing time.

Table 2-3. The complexity of spatial graphs and processing time

	URL	Size			Processing Time (Milliseconds)	
		HTML Tags	Nodes	Reduction Percent	Graph Generation	Extraction
1	shopping.yahoo.com	1065	504	47.3%	1063	373
2	scistore.cambridgesoft.com	1213	488	40.2%	1301	368
3	shop.lycos.com	522	233	44.6%	210	139
4	barnesandnoble.com	2328	851	36.6%	2702	1899
5	borders.com	812	342	42.1%	384	297
6	circuitcity.com	1664	557	33.5%	1094	198
7	compusa.com	1509	473	31.3%	752	228
8	drugstore.com	690	279	40.4%	248	507
9	ebay.com	1647	875	53.1%	2714	2112
10	etoys.com	677	234	34.6%	176	140
11	kidsfootlocker.com	608	276	45.4%	174	68
12	kodak.com	679	301	44.3%	238	85
13	newegg.com	1724	861	49.9%	1200	6703
14	nothingbutsoftware.com	1048	148	14.1%	305	33
15	overstock.com	1416	889	62.8%	1841	2038
16	powells.com	2531	951	37.6%	1488	14860
17	softwareoutlet.com	1117	247	22.1%	260	137
18	ubid.com	360	111	30.8%	100	22
19	amazon.com	1208	552	45.7%	530	511
20	shopping.hp.com	1046	392	37.5%	257	239
21	qualityinks.com	513	237	46.2%	129	144
	Average	1160.8	466.7	40%	817.4	1481

2.7.3. Discussion

Graph Generation is one of the most important components in our approach. The performance of our approach is significantly affected by the quality of the spatial graph. As shown in Table 2-3, the generated spatial graphs reduce the complexity of the original Web page by 60%. The precision can be further improved by introducing machine vision techniques to the graph generation process, such as analyzing the background and foreground colors or styles. In the evaluation, some product records are not recognized by our approach. For example, in Figure 2-19, text objects 1 and 2 are very far and the distance exceeds the defined threshold. Therefore, text

objects 1 and 2 are not connected, and they are not recognized correctly by our approach. This problem could be solved by analyzing the background color of information objects. The same group of information objects often uses the same background color. It is highly possible that two objects that have the same background color, no other object between them and belong to the same container are semantically related. The complex usage of *div/table* may also reduce the recall. Some Web page designers use tables or divisions for the layout purpose. Therefore, information object of the same product may be organized at different containers that cause those objects are not semantically related in the graph generation process. If we also define some containment relations based on the background color of information objects, then they may be connected in the spatial graph. We also notice it is not sufficient to solely depend on the background color. For example, though text objects 1 and 3 have the same background color, but they should not be connected since there is a separation line between them. Therefore, it is necessary to combine background color, other machine vision techniques (such as line detection), distance analysis and DOM structure analysis together to derive the semantic relations.

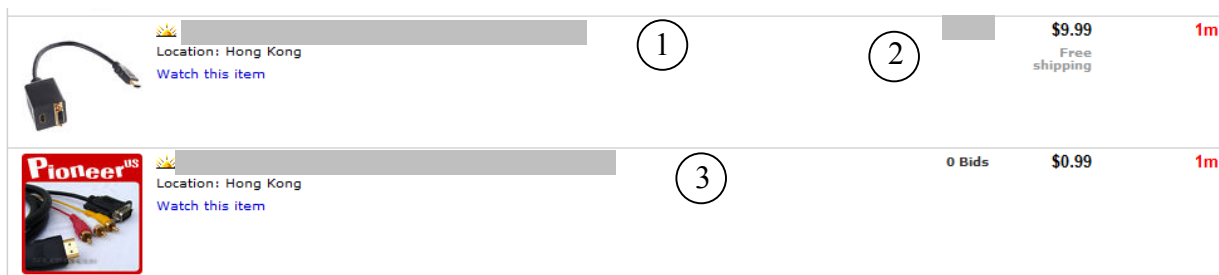


Figure 2-19. Far distance between text elements

The evaluation result shows that our approach is powerful to recognize structured records that are consistent with the specified graph grammar. However, it may falsely recognize some product-unrelated records, which is in general caused by some sorts of advertisements that are presented similarly to product records. Though the graph generation process has removed

advertisements placed in the border areas, some advertisements are placed in the central area which makes it hard to be removed. In the future, we will consider introducing content analysis techniques (such as the techniques proposed in [Kov02] or [Fer08]) to our approach. Based on the content analysis, we can classify main contents from other sections (e.g. advertisements or menu) in a Web page and limit the data extraction in the main area.

2.8. Related Work

The problem of extracting structured data from Web documents has been widely studied during the recent years [Lae02, Sar02]. Based on the theoretical foundation, we can classify data extraction techniques into wrapper-based approaches and statistical model based ones. The Wrapper-based approaches, such as [Fre00, Mus01], use a set of rules to define the knowledge of data extraction while the statistical model based approaches [Che08] use machine learning techniques or statistical models (such as Hidden Markov Models [Gen04, Sko03]) to discover related information. Wrappers and statistical models are used in different application domains. In general, wrappers extract structured records of a certain type and tag the semantic role of each information object. On the other hand, statistical model based approaches are useful to discover any repetitive structures without further analyzing semantics among information objects. Distinct from previous approaches, our approach applies the state-of-the-art grammar technology (i.e. the Spatial Graph Grammar) as the theoretical foundation. Based on the Spatial Graph Grammar, the knowledge of data extraction is visually defined through grammatical rules, which are decoupled from the data extraction process.

With a clear structure to specify the layout of a Web page, the HTML source codes have been commonly analyzed to extract structured data records [Kus97, Hsu98, Mus99, Cre01, Ara03,

Liu03, Chu04, Rei04, Zha05a, Zha07, Lab09]. Several approaches [Kus97, Hsu98, Mus99] use the machine learning technique to automatically derive a wrapper based on a set of manually labeled training data. In a training set, information pieces of interest are manually tagged with different labels, such as *title* or *price*. Based on the labels, those approaches [Kus97, Hsu98, Mus99] consider an HTML Web page as a sequence of characters and search for common delimiter strings between labeled information pieces. Those common delimiter strings are formalized as extraction rules in a wrapper for data extraction. Though the above approaches apply different technologies to derive a wrapper, they all require a set of training data, which are manually labeled by human experts. Several approaches [Cre01, Ara03] automatically derive a template from sample Web pages and use the extracted template to discover structured records. In general, Web pages that are generated from the same template have the same DOM structure but different actual contents. Crescenzi *et. al.* [Cre01] proposed an automatic approach to extracting structured records from a set of Web pages that follow the same template. This approach takes a set of sample Web pages as input and automatically generates a wrapper by recognizing identical DOM structures among those sample pages. Similarly, EXALG [Ara03] takes a set of template-generated pages as input, and automatically derives the template underlying those Web pages. The extracted template is used to generate extraction rules in a wrapper. TTAG [Chu04] takes several training Web pages as input. It traverses those DOM structures from the root to leaf nodes in a top down fashion, and uses the dynamic programming to compare and extract repetitive patterns. Reis *et. al.* [Rei04] propose a *tree edit distance* method. This approach takes clustered pages as input, and Web pages in each cluster use a common template. It calculates tree distance to derive the underlying template, and uses the template for data extraction. The above approaches [Cre01, Ara03, Chu04, Rei04] do

not require manually labeled data, which greatly reduces the manual effort in the data extraction process. However, they require that Web pages being analyzed must follow the same template as the sample Web pages. MDR [Liu03] generates an HTML tag tree based on table and form related tags, e.g., *table*, *form*, *tr*, *td*, and etc. This HTML tag tree significantly reduces the complexity of the original Web page. Based on the HTML tag tree, MDR uses the string comparison technique to divide a Web page into different regions. In each region, it identifies data records by calculating similarity between tag strings. Zhao *et. al.* [Zha05] has proposed a new system DEPTA that improves MDR. Especially, DEPTA supports partial tree alignment. Therefore, this approach accommodates variations among different Web pages and improves the applicability. Instead of automatically deriving a template, some approaches provide a graphical user interface for users to intuitively define data patterns. Zhai *et. al.* [Zha07] renders a Web page and allows users to select information objects in the screen shot to define a data pattern. Different from our approach, this data pattern is defined on the DOM structure, not on the visual layout. Laber *et. al.* [Lab09] proposed a heuristic algorithm for extracting news articles based on DOM tree structure. This approach used some statistical analysis to analyze the DOM tree elements and identify the relevant information objects. All of discussed methods use HTML DOM structure as the main source for data extraction. Instead, our approach, i.e. *Visual Grammar Based Extraction - VGE*, implements data extraction based on the information presentation. The visual analysis can address the issues of complexity and diverse usages of HTML DOM structures to a certain degree. By actually rendering a Web page, our approach supports dynamic information objects which are generated at run time.

Recently, the visual perception technique has been applied to extract structured data since it is independent from the detailed implementation underlying a Web page. These approaches [Zhe07, Che08, Sim05] basically calculate the visual similarity among different Web pages to group semantically related information. Zheng *et. al.* [Zhe07] introduced a *template independent* system to identify news articles based on visual consistency. This approach summarizes a set of visual features to present news stories and accordingly automatically generates a template-independent wrapper based on those visual features. Chen *et. al.* [Che08] proposed a system for extracting news stories based on visual perception. First, it identifies the areas that contain news stories based on content functionality, space continuity, and formatting continuity. After detecting the news areas, news stories are extracted based on the position, format and semantics. The above two approaches [Zhe07, Che08] are limited to extract news stories, and are not applicable to other domains. Distinct from those two approaches, our approach is general to different application domains. Furthermore, the usage of HTML structures in our approach makes it efficiently to recognize the boundary between different blocks. Some visual perception approach [Sim05] is implemented on statistical models that emphasize on extracting repetitive data records. For example, ViPER [Sim05] uses visual information to separate and weight different data regions.

The Hybrid method takes benefits from both DOM Structure and Visual Perception, and combines them together. ViNTs [Zha05b] automatically recognizes different content shapes based on the visual position of information objects. Afterward, a wrapper is generated based on an HTML structure which represents each shape. This approach still extracts information based on HTML DOM structures, though the wrapper is derived from visual analysis. Instead, our approach

specifies extraction rules from both the layout and the DOM structure. ViNTs is limited to search results, while our approach is general to different domains.

Our work is also related to the researches that divide a Web page into different regions. Kovacevic *et. al.* [Kov02] used visual information to build up an M-Tree, and defined heuristic rules to recognize common page areas, such as header, left and right menu, footer and etc. Some approaches analyze the importance of different blocks within a single web page, such as a learning based approach [Son04] or random walk model based approaches [Yin04, Yin05]. In these approaches, visual features, such as width, height, or position, are used to recognize and analyze information blocks. Joshi *et. al.* [Jos09] used natural language processing techniques and DOM tree structures for extracting the main content of a Web page. The TWWF approach [Zha09] used the DOM structure to divide a Web page. Ahmadi and Kong [Ahm08] used a hybrid approach for block recognition. The above technologies can be potentially integrated with our approach to improve the performance by recognizing information blocks to eliminate noises.

2.9. Conclusion

This paper presents a novel and general solution to extract data across different Web sites. Our method works based on graph grammars to extract the same type of information from different websites without the need of training and adjustment for different websites. Our approach utilizes both the visual content features of a rendered web page and the HTML DOM structure to extract structured records. First, a Web page is rendered; based on both the visual presentation and the HTML DOM structure, a spatial graph is be generated. Spatial graph nodes represent atomic information objects (such as images, texts and links) and edges show semantic relations between two connecting objects. Distinct from other approaches, the hybrid analysis can reduce the

complexity of HTML based analysis and address the issue of HTML diversity to a certain degree. We have chosen the electronic commerce domain as a case study to evaluate our method. We have summarized two common Web patterns for presenting product information on E-Commerce web sites and formally formalized those two patterns as graph grammars. We have implemented a prototype and tested the prototype on 21 Web sites. The evaluation shows promising results. Our approach has a high F1-Score as 97.49%, compared to 69.19% of the MDR approach. The evaluation results indicate our approach has a good performance in terms of both precision and recall. The main advantage of our approach lies in its ability to distinguish the most important content from less important and noisy information and to convert the complex HTML DOM structure to a simple spatial graph. On average, the generated spatial graphs reduce the complexity of the original Web page by 60%. Based on the simplified spatial graph, our approach is efficient to extract structured records. The average processing time on a Web page with about 1000 HTML elements is nearly 2 seconds. The visual analysis makes our approach a general solution which could be applied to different Web sites. This paper focused on the electronic commerce domain. By changing the graph grammar, our approach can be used in other domains. The implementation of an interactive grammar design tool reduces the manual efforts of designing a graph grammar. In the future work, we will add and identify more spatial relations between information objects, and optimize the graph generation algorithm. These optimizations increase the quality of generated spatial graphs, which can affect both the precision and recall. Especially, we plan to use some machine vision techniques to recognize boundary between different regions and to use content analysis techniques to analyze actual contents.

2.10. References

- [Ahm08] Ahmadi, H. and Kong, J. 2008. Efficient web browsing on small screens. In *Proceedings of the Working Conference on Advanced Visual interfaces* (Napoli, Italy, May 28 - 30, 2008). AVI '08. ACM, New York, NY, 23-30.
- [Ara03] Arasu, A. and Garcia-Molina, H. 2003. Extracting structured data from Web pages. In *Proceedings of the 2003 ACM SIGMOD international Conference on Management of Data* (San Diego, California, June 09 - 12, 2003). SIGMOD '03. ACM, New York, NY, 337-348.
- [Ate06] Ates, K., Kukluk, J., Holder, L., Cook, D., Zhang, K. 2006. Graph Grammar Induction on Structural Data for Visual Programming, In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, 2006. 232-242.
- [Ate07] Ates, K. and Zhang, K. 2007. Constructing VEGGIE: Machine Learning for Context-Sensitive Graph Grammars. In *Proceedings of the 19th IEEE international Conference on Tools with Artificial intelligence - Volume 02* (October 29 - 31, 2007). ICTAI. IEEE Computer Society, Washington, DC, 456-463.
- [Chu04] Chuang, S. and Hsu, J. Y. 2004. Tree-Structured Template Generation for Web Pages. In *Proceedings of the 2004 IEEE/WIC/ACM international Conference on Web intelligence* (September 20 - 24, 2004). Web Intelligence. IEEE Computer Society, Washington, DC, 327-333.
- [Che05] Chen, S., Hong, D., Shen, V.Y. 2005. An experimental study on validation problems with existing HTML webpages. In *Fourteenth International World Wide Web Conference* (May 10-14, 2005). WWW 2005

- [Che08] Chen, J. and Xiao, K. 2008. Perception-oriented online news extraction. In *Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries* (Pittsburgh PA, PA, USA, June 16 - 20, 2008). JCDL '08. ACM, New York, NY, 363-366.
- [Cox00] Cox, P. T., Smedley, T. 2000. Building Environments for Visual Programming of Robots by Demonstration, *Journal of Visual Languages and Computing*, Vol. 11(5), 2000. 549-571.
- [Cre01] Crescenzi, V., Mecca, G., and Merialdo, P. 2001. RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In *Proceedings of the 27th international Conference on Very Large Data Bases* (September 11 - 14, 2001). P. M. Apers, P. Atzeni, S. Ceri, S. Paraboschi, K. Ramamohanarao, and R. T. Snodgrass, Eds. Very Large Data Bases. Morgan Kaufmann Publishers, San Francisco, CA, 109-118.
- [Fre00] Freitag, D. and Kushmerick, N. 2000. Boosted Wrapper Induction. In *Proceedings of the Seventeenth National Conference on Artificial intelligence and Twelfth Conference on innovative Applications of Artificial intelligence* (July 30 - August 03, 2000). AAAI Press, 577-583.
- [Fer08] Fersini, E., Messina, E., and Archetti, F. 2008. Enhancing web page classification through image-block importance analysis. *Inf. Process. Manage.* 44, 4 (Jul. 2008), 1431-1447.
- [Gen04] Geng, J. and Yang, J. 2004. Automatic extraction and integration of bibliographic information on the Web. *IDEAS'04*, 193-04.
- [Hog05] Hogue, A. and Karger, D. 2005. Thresher: automating the unwrapping of semantic content from the World Wide Web. In *Proceedings of the 14th international Conference on World Wide Web* (Chiba, Japan, May 10 - 14, 2005). WWW '05. ACM, New York, NY, 86-95.
- [Hsu98] Hsu, C. and Dung, M. 1998. Generating finite-state transducers for semi-structured data extraction from the Web. *Inf. Syst.* 23, 9 (Dec. 1998), 521-538.

- [Hui06] Hui, G., Amanda, S. 2006. Taxonomy Based Data Extraction from Multi-item Web Pages. In *Proceedings of Workshop on Web Content Mining with Human Language Technologies at the International Semantic Web Conference, ISWC 2006*.
- [Ivo01] M. Y. Ivory and M. A. Hearst, "The State of the Art in Automating Usability Evaluation of User Interfaces," *ACM Computing Surveys*, 33(4), pp. 470-516, 2001.
- [Jos09] Joshi, P. M. and Liu, S. 2009. Web document text and images extraction using DOM analysis and natural language processing. In *Proceedings of the 9th ACM Symposium on Document Engineering* (Munich, Germany, September 16 - 18, 2009). DocEng '09. ACM, New York, NY, 218-221.
- [Kon06] Kong, J., Zhang, K., and Zeng, X. 2006. Spatial graph grammars for graphical user interfaces. *ACM Trans. Computer Human. Interact.* 13, 2 (Jun. 2006), 268-307.
- [Kon12] Kong, J., Barkol, O., Bergman, R., Pnueli, A., Schein, S., Zhao, C. Y., Zhang, K. 2012. Web Interface Interpretation Using Graph Grammars. *IEEE Transactions on SMC – Part C*, 42(4), 590-602.
- [Kov02] Kovacevic, M., Diligenti, M., Gori, M., and Milutinovic, V. 2002. Recognition of Common Areas in a Web Page Using Visual Information: a possible application in a page classification. In *Proceedings of the 2002 IEEE international Conference on Data Mining* (December 09 - 12, 2002). ICDM. IEEE Computer Society, Washington, DC, 250.
- [kuk08] J. Kukluk, L. Holder, and D. Cook. Inference of edge replacement graph grammars. *International Journal on Artificial Intelligence Tools*, 2008.

- [Kus97] Kushmerick, N., Weld, D., & Doorenbos, R. 1997. Wrapper induction for information extraction. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*. 729-737.
- [Kus00] Kushmerick, N. 2000. Wrapper induction: efficiency and expressiveness. *Artif. Intell.* 118, 1-2 (Apr. 2000), 15-68.
- [Lab09] Laber, E. S., de Souza, C. P., Jabour, I. V., de Amorim, E. C., Cardoso, E. T., Rentería, R. P., Tinoco, L. C., and Valentim, C. D. 2009. A fast and simple method for extracting relevant content from news webpages. In *Proceeding of the 18th ACM Conference on information and Knowledge Management* (Hong Kong, China, November 02 - 06, 2009). CIKM '09. ACM, New York, NY, 1685-1688.
- [Lae02] Laender, A. H., Ribeiro-Neto, B. A., da Silva, A. S., and Teixeira, J. S. 2002. A brief survey of web data extraction tools. *SIGMOD Rec.* 31, 2 (Jun. 2002), 84-93.
- [Liu03] Liu, B., Grossman, R., and Zhai, Y. 2003. Mining data records in Web pages. In *Proceedings of the Ninth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining* (Washington, D.C., August 24 - 27, 2003). KDD '03. ACM, New York, NY, 601-606.
- [Mus99] Muslea, I., Minton, S., and Knoblock, C. 1999. A hierarchical approach to wrapper induction. In *Proceedings of the Third Annual Conference on Autonomous Agents* (Seattle, Washington, United States). O. Etzioni, J. P. Müller, and J. M. Bradshaw, Eds. AGENTS '99. ACM, New York, NY, 190-197.

- [Mus01] Muslea, I., Minton, S., and Knoblock, C. A. 2001. Hierarchical Wrapper Induction for Semistructured Information Sources. *Autonomous Agents and Multi-Agent Systems* 4, 1-2 (Mar. 2001), 93-114.
- [Pas09] Pasternack, J. and Roth, D. 2009. Extracting article text from the web with maximum subsequence segmentation. In *Proceedings of the 18th international Conference on World Wide Web* (Madrid, Spain, April 20 - 24, 2009). WWW '09. ACM, New York, NY, 971-980.
- [Rei04] Reis, D. C., Golgher, P. B., Silva, A. S., and Laender, A. F. 2004. Automatic web news extraction using tree edit distance. In *Proceedings of the 13th international Conference on World Wide Web* (New York, NY, USA, May 17 - 20, 2004). WWW '04. ACM, New York, NY, 502-511.
- [Roz97] Rozenberg, G. (Ed.) 1997. Handbook on Graph Grammars and Computing by Graph Transformation: Foundations, Vol.1, *World Scientific*, 1997.
- [Sar02] Sarawagi, S. 2002. Automation in information extraction and data integration (tutorial). In *Proceedings of VLDB 2002*.
- [Shn09] Shneiderman, B. 2009 Designing the User Interface: Strategies for Effective Human-Computer Interaction. *Addison-Wesley Longman Publishing Co., Inc.*
- [Sim05] Simon, K. and Lausen, G. 2005. ViPER: augmenting automatic information extraction with visual perceptions. In *Proceedings of the 14th ACM international Conference on information and Knowledge Management* (Bremen, Germany, October 31 - November 05, 2005). CIKM '05. ACM, New York, NY, 381-388.
- [Sko03] Skounakis, M., Craven, M., and Ray, S. 2003. Hierarchical hidden Markov models for information extraction. In *Proceedings of the 18th international Joint Conference on Artificial*

intelligence (Acapulco, Mexico, August 09 - 15, 2003). Morgan Kaufmann Publishers, San Francisco, CA, 427-433.

[Son04] Song, R., Liu, H., Wen, J., and Ma, W. 2004. Learning block importance models for web pages. In *Proceedings of the 13th international Conference on World Wide Web* (New York, NY, USA, May 17 - 20, 2004). WWW '04. ACM, New York, NY, 203-211.

[Wik10] Wikipedia. 2010. F1 Score. http://en.wikipedia.org/wiki/F1_score. Visited on 7/21/2010

[Wu06] Wu, C., Zeng, G., Xu, G. 2006. A Web Page Segmentation Algorithm for Extracting Product Information. In *Proceeding of the IEEE International Conference on Information Acquisition*, ICIA 2006. 1374-1379.

[Yin04] Yin, X. and Lee, W. S. 2004. Using link analysis to improve layout on mobile devices. In *Proceedings of the 13th international Conference on World Wide Web* (New York, NY, USA, May 17 - 20, 2004). WWW '04. ACM, New York, NY, 338-344.

[Yin05] Yin, X. and Lee, W. S. 2005. Understanding the function of web elements for mobile content delivery using random walk models. In *Special interest Tracks and Posters of the 14th international Conference on World Wide Web* (Chiba, Japan, May 10 - 14, 2005). WWW '05. ACM, New York, NY, 1150-1151

[Zha04] Zhang, Z., He, B., and Chang, K. C.-C. 2004. Understanding Web Query Interfaces: Best-Effort Parsing with Hidden Syntax. In *Proc. 2004 ACM SIGMOD International Conference on Management of Data*, 107-118.

[Zha05a] Zhao, H., Meng, W., Wu, Z., Raghavan, V., and Yu, C. 2005. Fully automatic wrapper generation for search engines. In *Proceedings of the 14th international Conference on World Wide Web* (Chiba, Japan, May 10 - 14, 2005). WWW '05. ACM, New York, NY, 66-75.

- [Zha05b] Zhai, Y. and Liu, B. 2005. Web data extraction based on partial tree alignment. In *Proceedings of the 14th international Conference on World Wide Web* (Chiba, Japan, May 10 - 14, 2005). WWW '05. ACM, New York, NY, 76-85.
- [Zha07] Zhai, Y. and Liu, B. 2007. Extracting Web Data Using Instance-Based Learning. *World Wide Web* 10, 2 (Jun. 2007), 113-132.
- [Zha09] Zhang, Q., Shi, Y., Huang, X., and Wu, L. 2009. Template-independent wrapper for web forums. In *Proceedings of the 32nd international ACM SIGIR Conference on Research and Development in information Retrieval* (Boston, MA, USA, July 19 - 23, 2009). SIGIR '09. ACM, New York, NY, 794-795.
- [Zhe07] Zheng, S., Song, R., and Wen, J. 2007. Template-independent news extraction based on visual consistency. In *Proceedings of the 22nd National Conference on Artificial intelligence - Volume 2 (Vancouver, British Columbia, Canada, July 22 - 26, 2007)*. A. Cohn, Ed. Aaai Conference On Artificial Intelligence. AAAI Press, 1507-1512.

3. A LOW-COST SPATIALLY-AWARE MOBILE INTERACTION DEVICE

Large paper sheets are still the most preferred medium used by engineers for inspection in remote sites. However, those paper documents are hard to modify and retrieve. This paper presents a novel spatially-aware mobile system (called PhoneLens), which combines the merits of paper documents and mobile devices. It augments paper documents with digital information. Different from previous approaches, PhoneLens is inexpensive. It includes two infrared LEDs, one Wiimote and one Android device. Based on the hardware setting, we developed an efficient spatial tracking algorithm to track the movement of a mobile device within a large workspace. Our approach is robust and is applicable in various scenarios that use multivalent maps, diagrams or floor plans. PhoneLens provides different functions to link a multivalent document with digital information, including browsing different layers, annotation, zooming and so forth. We conducted a controlled study that compares the participant's performance of the PhoneLens against the traditional paper-based method on a multivalent paper document. The results supported the hypothesis that participants using the PhoneLens method were more efficient.

3.1. Introduction

Though the concept of "paperless office" was proposed long time ago, physical paper documents are still resilient even in a digital age since they have many advantages, such as easy to use, superior readability and availability. However, the information printed on paper is static, which is hard to modify and index. On the other hand, mobile devices provide great flexibility and convenience to access and modify digital information anywhere and anytime. However, the small screen makes it frustrating to browse a large amount of information. By augmenting paper documents with digital information, spatially-aware mobile interaction can overcome the

shortcomings of paper documents and mobile devices. A user can move on top of a paper document a spatially-aware mobile device, which dynamically visualizes user-focused information within a large workspace that provides sufficient context.

Various approaches have been proposed to support spatially-aware mobile interaction, such as marker-based methods [Rei06, Roh04, Sch07] or content-recognition based solutions [Ero08, Har08, Lia10a, Liu08]. Instead of using markers or complex image processing techniques, this paper proposes a Wiimote based system to provide a digital overlay on paper documents (see Figure 3-1 for a system setup). Two infrared LEDs are mounted to a mobile device. A Wiimote tracks those LEDs and transmits their coordinates to a mobile device through Bluetooth. The mobile device calculates its relative position within a large workspace and displays relevant digital information. The advantages of our system (called *PhoneLens*) are summarized as follows:

1. **Low cost.** Wiimote and smartphone have been commonly used in daily life with a low cost. Furthermore, our approach works on traditional paper documents without altering the original contents.
2. **Low computing complexity.** Our approach avoids complex image processing. The smartphone serves as an input device (i.e., movement-based interaction), an output device (i.e., overlaying digital information), and a computing device (i.e., tracking hand movements).
3. **Robust.** A smartphone connects with a Wiimote through Bluetooth without requiring a wireless or 3G network. By using an infrared camera, the infrared tracking provides a reliable sensing in different indoor environments. The simple hardware implementation also makes our approach applicable in a public environment.

4. **Portable.** Our approach only needs a smart phone with a Wiimote, which makes it very portable. Besides, PhoneLens works with traditional unmodified paper documents, which users can easily roll and carry.

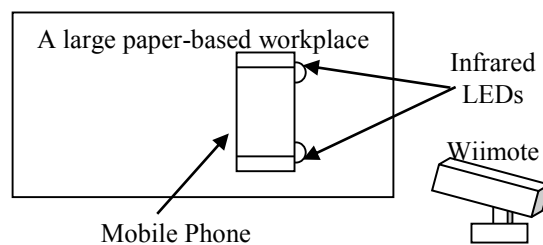


Figure 3-1. PhoneLens interacting with a paper document

In summary, our key contribution is to propose an efficient low-cost spatial tracking algorithm in a 3D environment based on a cheap infrared camera. According to the spatial tracking algorithm, we developed a framework that supports a bimanual interaction to augment paper documents with digital information. A user uses his/her non-dominant hand to move a mobile device while using the dominant hand to make a more precise operation. Movement based interaction allows users to view and edit virtual objects in a physical environment [Yee03] that is much larger than the size of a mobile screen.

To evaluate the Phone-Lens method, we conducted a controlled empirical study that required the participants to use the Phone-Lens for browsing the digital information contained in a multivalent document. A multivalent document [Phe97], such as an architectural plan, consists of multiple layers of distinct, but closely coupled contents. The results from this study show significant improvements in the efficiency and usefulness of the proposed system over the traditional paper-based approach.

The remaining of the paper is organized as follows. Section 3.2 discusses related work. Section 3.3 describes our design goals. Section 3.4 overviews the system architecture. Section 3.5 illustrates the spatial tracking algorithm. Section 3.6 presents the accuracy evaluation on the spatial tracking algorithm. Section 3.7 shows a case study. Section 3.8 describes a user evaluation. Section 3.9 presents the evaluation results. Section 3.10 discusses those results, followed by conclusion and future work in Section 3.11.

3.2. Related Work

Markers have been used in augmented reality to identify virtual and real objects [Sch07]. Rohs [Roh04] proposed to print visual codes on a paper document. Digital information is then retrieved by recognizing a code tag through a camera phone. Marked-up maps [Rei06] use RFID to link paper maps with digital information. An RFID reader is attached to a mobile device, and RFID tags are placed beneath items of interest on a paper map. Chao and Chen [Cha09] used line numbers as tags to specify locations of sentences in a textbook. Those line numbers serve as anchors that link to related digital information, and thus bridge the communication between a mobile device and a paper document. These marker-based approaches are reliable, but they require altering original paper documents by inserting markers [Lia10a]. Chan *et al.* [Cha10] proposed invisible infrared markers that do not interfere with original contents. However, the proposed device is not portable since it includes an infrared projector and one standard projector. *SideBySide* [Wil11] is a novel approach that supports multi-user interaction with handheld projectors. This approach uses invisible fiducial markers to track projected images from multiple devices.

Camera-based document recognition and detection have been used in interactive paper that overlays digital information to paper documents. One seminal prototype is DigitalDesk that used

a camera and a projector to provide a computer augmented environment for paper documents [Wei93]. PaperLink [Ara97] used a highlighter pen and a camera to link dynamic digital information with paper documents. In content-based analysis, some approaches focus on text-based documents, such as Mobile Retriever [Liu08], a clickable printed URL [Hul07], and Hotpaper to augment paper with multimedia annotations (such as video or audio) [Ero08], while several approaches have been proposed to augment picture-based documents, such as augmented maps [Mor09, Roh07a]. The MapSnapper project proposed a robust algorithm that maps a low-quality image to a high-quality digital image of the same content [Har08]. PACER [Lia10a] used a camera phone to capture the image of a patch in a paper document. Based on the image recognition, PACER supported gesture-based commands on paper documents. FACT [Lia10b] is also aligned with the research of content-based interaction that is built on precise image recognition. This approach used a camera-projector unit, in which the camera captures video frames and the mobile projector provides direct visual feedbacks on a paper document. The above approaches depend on markerless optical tracking techniques. Recently, the natural feature tracking method [Mor09, Wag10] is proposed and optimized for real-time tracking on mobile devices with limited processing power.

Different from these above content-based methods, our work uses an infrared camera (i.e., Wiimote), instead of a regular camera, to track the position of a mobile device within a large workspace, with the following considerations. First, our approach tracks movements in a 3D space rather than a 2D space. Second, in some applications (such as an architectural floor plan), it is hard to use image processing to detect locations, because several sections may look very similar or exactly the same. Third, image recognition is computationally expensive. Fourth, normal cameras

may be affected by environmental light (too dark or too bright). Finally, the camera base approach needs a certain distance between the camera and the paper document while our approach is working under situations with or without a distance.

Our approach is related to spatially-aware displays [Fit93a, Fit93b], in which a positioned screen is moving around to see different parts of a large virtual workspace. Chameleon [Tsa02] was a novel input/output device that mounts a touch screen on a tracked mechanical boom. The movement of the touch screen in a physical environment is mapped to a 3D virtual environment, and virtual objects presented from the corresponding viewpoint are displayed on the touch screen. The peephole display [Yee03], which combines pen inputs with spatially aware displays, was built based on Chameleon [Fit93a] and Toolglass [Bed94]. The tracker in the peephole is either associated with a table or is tethered to a power cord, which limits its portability. A-book used a WACOM 4d mouse to track the position and orientation of a handheld PDA [Mac02]. The PDA acts as a transparent window that provides a digital overlay on the patch of the paper document beneath the PDA. Distinct from A-book, our approach does not need a remote server, and provides gesture based interaction, in addition to point and click interaction.

Built-in cameras on cell phones or accelerometers have been used to implement motion based interaction, which responds to user's motion. Camera based optical tracking [Har05, Wan06, Han07] analyzes image sequences captured by a built-in camera to estimate the device motion, which is translated to interaction commands. Cho *et al.* [Cho07] used an accelerometer to control a photo browser based on a tilt dynamics model, which provides a proportional mapping from the sensor data to the cursor movements in the image space. Motion sensing interfaces provide natural and intuitive interaction through an enlarged workspace. The motion based interaction has been

applied into different applications, such as image browsing [Cho07] or mobile gesture [Wan06]. Recently, different studies have been conducted to evaluate the usability of motion based interaction in different applications, such as image browsing [Yim11] and map navigation [Roh07a, Roh07b, Mor09], under different settings, such as field trials [Mor09], physical movement with and without visual context [Roh07a]. Furthermore, different motion sensing techniques have been compared and evaluated, such as accelerometer [Roh07b], optical tracking [Roh07a, Mor09], and combined accelerometer and camera [Yim11]. Those studies provide valuable clues about design guidelines for motion sensing interfaces.

The Anoto technology [Ano02] enables tracing handwriting on physical paper. Based on Anoto digital pens, researchers have implemented pen-based interaction to augment paper documents with digital information, such as integrating paper notes with digital photos in biology [Yeh06] and a paper-based proof reading application [Wei08]. The PADD system (i.e., Paper Augmented Digital Documents) seamlessly integrated paper and digital documents together, and supported editing annotations in both digital and paper forms [Gui03]. PapierCraft [Lia08] extended the PADD system by providing gesture based commands on a paper document. Users can draw gesture commands through an Anoto digital pen on paper printouts. Then, those commands are executed, and results are displayed in a digital document viewer. Recently, projectors are combined with digital pens to provide a digital overlay on top of a paper document [Cao06, Son09, Son10, Lia10b]. Cao and Balakrishnan provided a systematic exploration of the design space of handheld projector interaction [Cao06]. PenLight [Son09] provided mobile, spatially-aware projector-based interaction. This approach integrated pen input with projector output to visualize and modify virtual information on a paper document. Due to the limited

technology of miniature projectors, the prototype of PenLight simulates a mobile projection mounted on a digital pen through a standard video projector. Our approach is fully implemented through existing techniques (i.e., smartphone and Wiimote). The MouseLight system [Son10] decoupled pen input with projector output, so that users can interact with the projector and digital pens bimanually. In MouseLight, a spatially aware projector was made of a mobile laser projector, two mirrors and two digital pens. The NiCE discussion room [Hal10] facilitates content creation and sharing during group discussion meetings by integrating Anoto digital pens, wall displays and personal computers. The Anoto digital pen based tracking avoids calibration. However, it can only be implemented on digital paper, on which a digital pattern of dots is printing. In contrast, our approach is applicable to traditional paper. Furthermore, the simple hardware implementation makes our approach robust in public environments with a high portability and a low cost.

LightSense [Olw06] used a camera to capture the LED light of a mobile phone to track the position. This approach needs to place the camera behind a diffused surface (such as a mirror) in order to eliminate indirect or ambient light, which may limit its portability. On the other hand, our system is implemented based on the infrared LED, which does not interfere with ambient light. Therefore, our approach is more robust. The Visual Interaction Platform [Ali06], which combines an electronic sketching board with optical tracking of tangible bits, uses a digital pen with a drawing tablet for precise positioning (such as sketching), and supports tracking tangible objects based on infrared tags for selecting and positioning virtual objects in a workspace. This approach is not portable and only tracks objects in a 2D space. Johnny Lee has applied the Wiimote-based infrared camera to track object movements in various applications [Lee08, Lee11]. Our approach is different in the following ways. First, the head tracking algorithm from Johnny Lee tracks a

person's head in a 3D space, where the distance is calculated between a Wiimote and a pair of LEDs. In order to support a distance based zooming and the location locking functionality, our approach calculates the distance between the workspace and a pair of LEDs of a mobile device (not between the Wiimote and the mobile device). This is more challenging, since it calculates the location of the workspace in a 3D space. Second, our approach eliminates the hand shaking. In Johnny Lee's applications, it is not necessary to consider hand shaking. But in our case, a user needs to hold and move a phone above a paper document. Therefore, it is inevitable to have small hand movements or hand shaking, which is considered as noise.

Recently, depth cameras have been applied to provide spatial tracking in a 3D space, such as multitouch interaction on everyday surfaces [Har11] or Microsoft Kinect. Compared with Wiimote, the depth information from a depth camera is especially useful to detect the movements of an object in a 3D space. However, a depth camera needs a server for image processing, which can reduce the portability. In the application of interactive paper, we do not need to detect the precise depth information. Therefore, our approach adopts Wiimote with the emphasis on portability, and proposes an efficient approximation algorithm (Refer to Section 3.5) to estimate the depth information.

3.3. Approach Overview

This section gives an overview of PhoneLens that has three important features, i.e., movement-based interaction, portable and low cost.

3.3.1. Movement-based interaction

A natural user interface makes the interface itself invisible so that users can focus on the task. PhoneLens provides a natural interaction to manipulate digital information on a paper

document. More specially, PhoneLens combines the easiness of using a paper document with the dynamics of digital information. For example, a user can type in a digital annotation in PhoneLens, similar to writing a note on a paper document.

It is challenging to browse a large amount of information on small screen devices [Bur08]. PhoneLens provides a spatially aware display to facilitate navigation of a large information space. A user can move a mobile device on top of a large paper document. PhoneLens can identify the position of the mobile device within the workspace and display the corresponding information in the area beneath the mobile device. Hand movement replaces the traditional panning interaction on a mobile device, and supports browsing information in a “context+focus” fashion.

Humans do many tasks using both hands. As stated in [Bux86], two handed interaction increases performance. In PhoneLens, a user uses the non-dominant hand to move a mobile device to change his/her focus, while the dominant hand issues a precise operation based on a natural interaction, such as a pinch gesture for zooming. Furthermore, the two handed interaction provides the option of using the index finger to operate PhoneLens while the index finger in general performs better than the thumb on the front of mobile devices [Wob08].

3.3.2. Portability

Portability is an important goal of our system. PhoneLens is applicable in different indoor environments, such as dark or bright environments, with or without a network connection. On the other hand, the portability should not compromise the usability and performance. We must address two challenging issues (i.e., a small screen size and a limited computing capacity) which exist for any small portable device. We addressed the screen size issue by providing a spatially-aware display. A user uses a large paper-based workspace as a reference to access detailed information

on a small mobile screen. In order to support smooth interaction with a limited computing capacity, the spatial tracking is implemented through an infrared camera to avoid complex computation. Furthermore, the Scalable Vector Graphics (SVG) format is used to provide high-quality graphical outputs on mobile devices. Different from traditional bitmap graphics, the SVG format does not have the zooming limitation, since it can be drawn with different scales. Besides, it uses processor and memory efficiently, because the system only needs to draw the portion of a picture, which is currently in the viewport, instead of the whole picture. In contrast with bitmap pictures, SVG files are very small in size, so they are more applicable in limited resources devices and we do not need to render the picture pixel by pixel. Instead, the picture is drawn by commands such as line, circle, arc, paint and etc.

3.3.3. Low cost

We intend to implement our system based on existing techniques with a low cost to increase the applicability in reality. Our system includes two infrared LEDs (less than \$1), a Wiimote Controller (around \$20), and a smart phone. PhoneLens can be implemented on any Android 2.2 device with at least 500 Mhz processor and 256M memory, which is currently the minimum configuration for smart phones.

3.4. System Architecture

Figure 3-2 shows the software architecture of PhoneLens. The PhoneLens mobile application mainly includes four layers, i.e., the *spatial data layer*, the *data layer*, the *logical layer* and the *presentation layer*. In addition, the PhoneLens system includes a stand-alone desktop application that allows developers to edit SVG files and the annotation database.

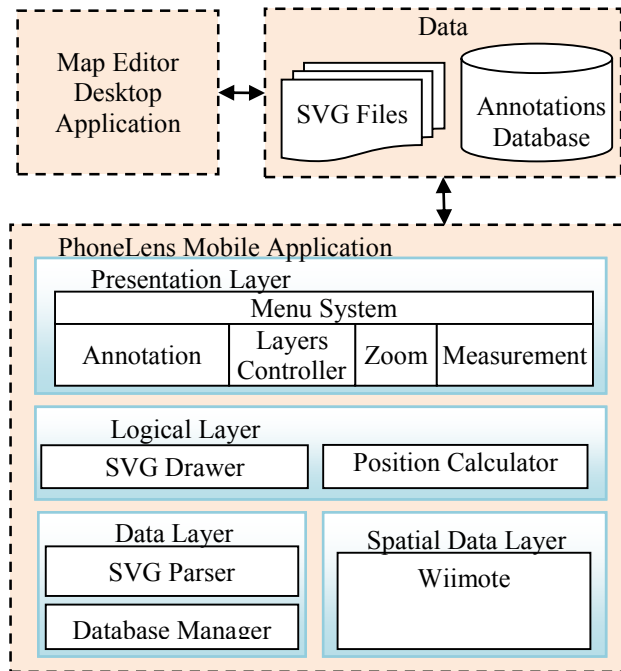


Figure 3-2. The system architecture of PhoneLens

3.4.1. Hardware design

We have chosen Wiimote as the tracking device, since it is portable with a low cost. As shown in Figure 3-3, we stick a paper board to the back of a smart phone, and attach two infrared LEDs to the paper board at one side. Two button batteries, which are stuck to the paper board, provide power to the LEDs. The button batteries and LEDs are light, and do not interfere interacting with a mobile device.

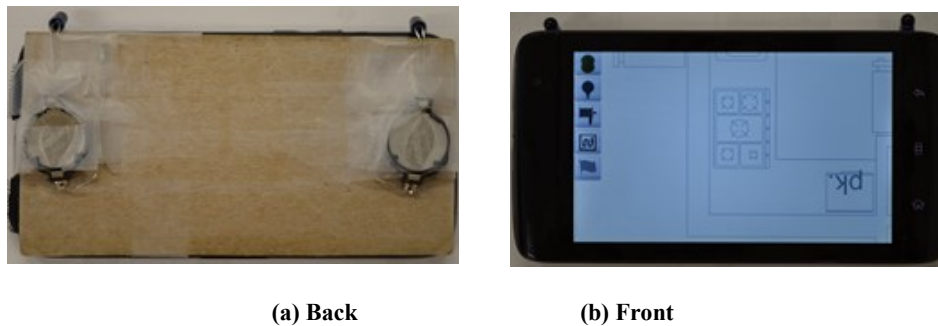


Figure 3-3. The mobile phone holder with infrared LEDs

The infrared LEDs in conjunction with a Wiimote efficiently track the location of a mobile device. As shown in Figure 3-1, a large paper document is placed on a wall. A user holds the mobile phone in front of the paper. The Wiimote is located at the right side of the workspace in a way that it covers the whole workspace. The phone connects to the Wiimote through a Bluetooth connection. Then, the camera continuously sends the raw position data of two visible infrared LEDs to the phone, and the phone uses an approximation algorithm (See section 3.5) to transform the raw position data to the position coordinates, relative to the workspace.

3.4.2. Software design

The PhoneLens mobile application is running on an Android 2.2 mobile device, and includes four major layers, as shown in Figure 3-2. The Spatial Data layer includes a Wiimote controller that receives the raw spatial data from the Wiimote. The data layer is responsible to parse SVG files and retrieve customizable digital information (such as annotations) from a database. The logical layer calculates the relative position of a mobile device within a workspace and visualizes corresponding digital information. An efficient position calculation algorithm is described in details in the next section. In order to calculate the position, a user needs to calibrate the mobile device with five points in a workspace (i.e., four corners and the central point). The presentation layer provides a GUI for users to manipulate digital information on top of a paper document.

3.4.2.1. Spatial data layer

This component directly communicates with the Wiimote through a Bluetooth connection, and contiguously receives the raw spatial data of two infrared LEDs from the view point of the Wiimote.

3.4.2.2. Data layer

There are two major components in this layer, i.e., SVG Parser and database manager. The SVG parser reads SVG files. We have chosen the SVG format due to the following reasons. First, most construction and design plans are recorded in a vector format, and almost all vector-based files (e.g., AutoCAD) can be easily converted to SVG. Second, SVG provides a scalable zooming without losing quality. Third, SVG is especially useful on mobile devices that have a smaller screen and a smaller storage capacity [Sch04], since it is light-weight and easy to parse. We have developed an SVG Parser component in Java for Android. The SVG parser is able to read XML documents and accordingly create graphical objects. The database manager communicates with a database that records customizable digital information.

3.4.2.3. Logical layer

This layer, which includes the Position Calculator and SVG Drawer, essentially calculates the relative position of a mobile device within the workspace and draws the corresponding portion of an SVG file on the screen. The Position Calculator is the kernel in PhoneLens that converts raw spatial data to coordinates in a 3D space. Based on the relative position of a mobile device, the SVG Drawer component renders the relevant digital information on the screen.

3.4.2.4. Presentation layer

The Presentation Layer provides a user interface for PhoneLens. This layer provides the following commands:

1. Different types of zooming approaches;
2. Select and show different layers of a multivalent document;
3. Add, search and view annotations; and

4. Measure distance and area.

Since different multivalent documents have different layers, the interface in PhoneLens is customizable to different applications. More specifically, the PhoneLens mobile application first reads the layer information from SVG files (i.e., the number of layers and the name of each layer) and customizes the interface based on the number and names of different layers. In the search interface, a user can search for annotations by selecting a specific annotation type from a list. The annotation types are specified in a database, and the list of annotation types in the search interface is customized according to the database.

In addition to the PhoneLens mobile application, our prototype also includes a stand-alone desktop editor, as shown in Figure 3-2. This editor provides a graphical user interface to design the structure of annotations (i.e., annotation types) in a database in PhoneLens. Different multivalent documents need to organize annotations in different ways. For example, in a floor plan design, we may classify annotations into several categories (such as ceiling, wall and floor) while another application can have a completely different classification. This editor allows designers to efficiently specify the annotation types in a database. Furthermore, designers can use this editor to add annotations to some default objects. Those default annotations with their corresponding positions are recorded in a database. Through the PhoneLens mobile application, users can edit those default annotations and add/edit new ones.

3.5. Position Calculator

Since Wiimote only captures objects in a 2D space, we propose a novel approximation algorithm to detect the vertical gap (i.e., the z-distance) between a mobile device and the workspace, and the moving direction in the z-axis (i.e., moving toward or away from the

workspace), which is used to support distance based zooming. The Position Calculator calculates logic positions within a workspace with the following steps:

1. Transform the location of a mobile device from the point of view of a camera (i.e., a trapezium) to that of the workspace (i.e., a rectangle).
2. Retrieve the distance between the Wiimote and the mobile device.
3. Calculate the z-distance between the mobile device and workspace.
4. Lock the position when a gap between the mobile device and the workspace is detected.
5. Eliminate hand shaking or small hand movements.

3.5.1. Step 1. Point-of-view conversion

The Wiimote views the workspace from the side. Consequently, a Wiimote camera views the workspace as a trapezium, similar to the illustration in the left in Figure 3-4, while the actual workspace is of a rectangular shape. We need to map a trapezium to a rectangular shape. We have used the method proposed by [Lie98] to transform each point from a trapezium to its corresponding point in a rectangular shape based on four calibration points at corners.

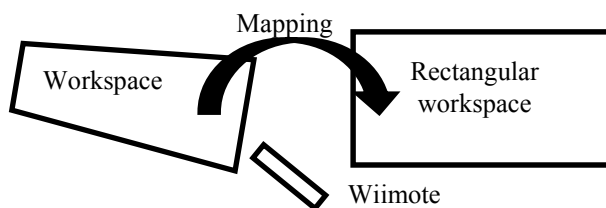


Figure 3-4. View conversion

3.5.2. Step 2. Approximating the distance between a Wiimote and a mobile device

Two infrared LEDs are mounted to a mobile device. According to perspective rules, when a mobile phone is moving away from a Wiimote, the length between two LEDs decreases from the point of view of an infrared camera, and vice versa. In other words, the perceived length

between two LEDs implies the distance between a mobile device and a Wiimote. Accordingly, we use Equation 3-1 [Wii11] to calculate the distance. In Equation 3-1, z is the vertical distance between the center of two LEDs and the Wiimote; (x_1, y_1) is the position of the first infrared LED, and (x_2, y_2) is the second one; R_d indicates the actual length between two LED points; the infrared camera in a Wiimote can capture 1024 pixels in the horizontal direction (HFOV=41°) and 768 pixels in the vertical direction (VFOV = 31°).

$$z = \frac{R_d}{2 \tan\left(\frac{\left(\frac{HFOV + VFOV}{1024 + 768}\right) \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{4}\right)} \quad \text{(Equation 3-1)}$$

Equation 3-1 requires that LEDs are positioned vertically in front of the Wiimote (i.e., 90 degrees). However, our application places the Wiimote at a side position, which makes it infeasible to directly apply Equation 3-1. In our approach, the distance between a mobile device and Wiimote is used to derive the z -distance between the mobile device and the paper workspace (refer to Step 3). More specifically, we need to detect the direction of hand movements in the z -axis, rather than the actual gap between a mobile device and a workspace. Therefore, we approximate the distance calculation according to Figure 3-5, in which the two solid circles indicate the physical positions of two LEDs, d_p represents the perceived length between two LEDs, and d_1 is the actual distance between the mobile device and the Wiimote. Our approach uses Equation 3-1 to calculate the distance d_2 (See figure 3-5) and uses it as an approximation for d_1 . This approximation reduces the accuracy, but it still satisfies our goal to detect the direction of hand movements in the z -axis, as validated by the performance evaluation in Section 3.6.

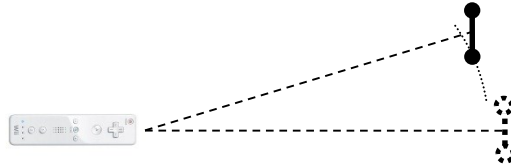


Figure 3-5. Distance approximation

3.5.3. Step 3. Calculating the z-distance between the mobile device and the paper workspace

A movement toward or away from a paper document can significantly change the coordinates detected by a Wiimote. In order to address this issue, we need to derive the z-distance between the mobile device and the workspace. The z-distance calculation also supports the distance based zooming.

First, we calculate the angle (i.e., a_c in Figure 3-6) of the viewing area relative to the Wiimote. The center calibration point is indicated by a solid circle. Equation 3-2 calculates the angle of a_0 (d_c is the approximation distance based on Equation 3-1). d_w , a predefined parameter, indicates the vertical distance between the Wiimote and the Workspace (See Figure 3-6). Equation 3-3 calculates the angle of a_1 based on the angle-per-pixel factor (x_c is x coordinate of the center point). Then, Equation 3-4 calculates the angle of a_c . The relations of those angles are illustrated in Figure 3-6.

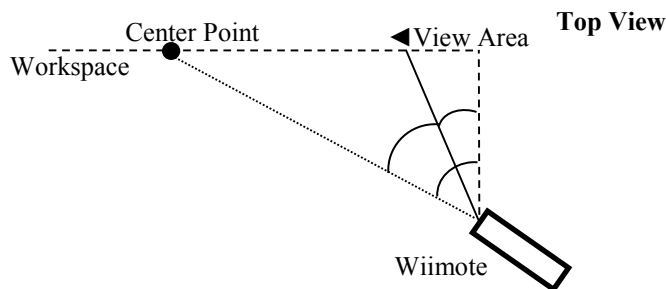


Figure 3-6. The calculation of the wiimote angle based on the center point

$$\mathbf{a}_0 = \cos^{-1} \frac{d_w}{d_c} \quad (\text{Equation 3-2})$$

$$\mathbf{a}_1 = x_c \times \frac{HFOV}{1024} \quad (\text{Equation 3-3})$$

$$\mathbf{a}_c = \mathbf{a}_0 - \mathbf{a}_1 \quad (\text{Equation 3-4})$$

At real time, we use the angle a_c to calculate the z-distance between the mobile phone and the workspace. Given a mobile phone that has the coordinates (x_m, y_m) , as shown in Figure 3-7, we can calculate the angle a_m based on Equation 3-3. According to a_m and a_c , we can calculate the vertical distance V_m based on Equation 3-5, in which d_m is the approximation distance, calculated through Equation 3-1.

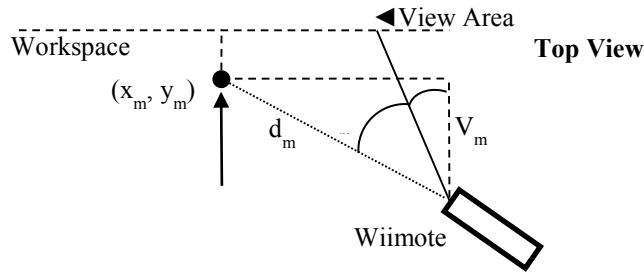


Figure 3-7. The calculation of the gap

$$V_m = \cos(a_m + a_c) \times d_m \quad (\text{Equation 3-5})$$

Ideally, we can compare V_m (See Figure 3-7) with d_w (See Figure 3-6) to derive the z-distance between a mobile device and the workspace. However, due to the approximation in the calculation, when a mobile device touches the workspace and moves around, we will get different values of V_m though they should be equal to d_w in an ideal situation. In order to handle the deviation, we introduce the concept of a *reference vertical distance* for each point. Given a point p in workspace, its reference vertical distance indicates an approximation of the vertical distance between the Wiimote and the mobile phone when the mobile phone is placed on top of the

workspace (i.e., the z-distance between the phone and the workspace is 0). For each point, its corresponding reference vertical distance is calculated as the following:

- a) **A calibration point p.** Its reference vertical distance is calculated based on Equation 3-5 in the calibration process.
- b) **Other points.** We divide the whole workspace into four regions, as shown in Figure 3-8. Each region includes two calibration points, i.e., the center point and the corresponding corner point. Consider the two calibration points p_0 (with the coordinate (x_{p0}, y_{p0})) and the reference vertical distance V_{e-p0} and p_1 (with the coordinate (x_{p1}, y_{p1})) and the reference vertical distance V_{e-p1} in the region 1 in Figure 3-8, the difference of the two reference vertical references is $|V_{e-p0} - V_{e-p1}|$. We evenly distribute the difference to each point in this region. For example, given a point p_i (with the coordinate $((x_{pi}, y_{pi}))$) in region 1, its reference vertical distance is calculated based on Equation 3-6.

$$V_{e-p_i} = \min(V_{e-p_0}, V_{e-p_1}) + |V_{e-p_0} - V_{e-p_1}| / (y_{pi} - y_{p0}) \quad \text{(Equation 3-6)}$$

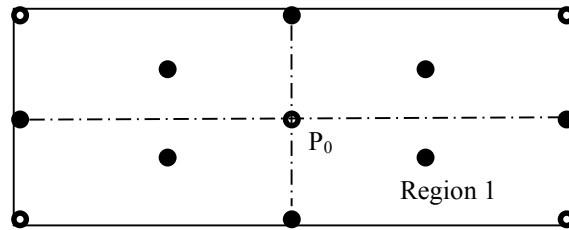


Figure 3-8. The eight points for evaluation

At run time, given a point p , we compare the value of V_m calculated through Equation 3-5 with its reference vertical distance (i.e., V_{e-p}) to calculate the z-distance between the phone and the workspace, i.e. Equation 3-7

$$z - \text{distance} = V_{e-p} - V_m \quad \text{(Equation 3-7)}$$

3.5.4. Step 4. Lock the location when a mobile device does not touch the workspace

When there is a gap between a mobile device and the workspace (i.e., $z\text{-distance} > 0$), the coordinates detected by Wiimote cannot correctly reflect the relative position within a workspace. Therefore, we lock the information displayed in a viewport when a gap is detected. However, except the calibration points, the reference vertical distance may not be equal to the calculated distance (i.e., V_m in Figure 3-7) when a mobile device touches the workspace, due to the approximation in Equation 3-6. We have chosen eight points that have the longest distance to calibration points, as shown in Figure 3-8. Then, we calculate the deviation between the reference vertical distance and the calculated distance, when the mobile device touches the workspace on those points. The deviation is ranging from 0.84 to 3.5 among those eight points. Therefore, we set a threshold as 5. At run time, when the difference between the reference vertical distance and the calculated distance is less than 5, we consider no gap between the phone and the workspace. Otherwise, a gap is detected.

3.5.5. Step 5. Noise elimination

It is normal that user's hands are shaking, while he/she holds the mobile phone. These pulses cause unstable position information. Besides, phone rotations can change the detected distance between two infra LEDs, which consequently results in an unstable viewport. Therefore, PhoneLens must eliminate those noises.

In order to prevent a jittering picture caused by the hand shaking, PhoneLens is equipped with a noise filter. We record the coordinates of hands movements during the last 1 second, and calculate the medians on both x and y axes as the target location. This will stabilize the output and prevent drawing a shaking picture on the screen.

In the case of a device rotation, the length between two LEDs detected by a Wiimote will change, which causes our algorithm to detect a gap between the mobile device and the board. Consequently, the information in the viewport is locked, but the zooming factor can be changed continuously due to the rotation, which may cause an unstable zooming. If the rotation is larger than a threshold (i.e., 10 degrees) detected by using an accelerometer, we lock the viewport and show an icon on the screen that alerts the user that he/she should hold the device in parallel to the workspace.

3.6. Accuracy Evaluation

Accuracy is an important factor, which affects user experience. We measured the accuracy of the spatial tracking algorithm by evaluating (a) the direction of hand movements in z-axis and (b) the sensitivity to hand movements.

The evaluation is performed on a paper document with the size of 18x25 inch. In order to evaluate the worst cases, we have selected 8 points, which have the largest distance with the calibration points, as shown in Figure 3-8 (white circles indicate the calibration points and solid black circles indicate evaluation points).

3.6.1. Moving direction in z-axis

We first evaluate the correctness of predicting the moving direction in z-axis. For each evaluation point (See Figure 3-8), we first recorded its reference vertical distance, and then calculated the vertical distance (i.e., V_m in Figure 3-7) in the cases of 1-inch, 2-inch and 4-inch distances between the mobile device and the workspace. As shown in Figure 3-9, when the phone is moving away from the workspace, the calculated vertical distance is decreasing. Therefore,

according to Equation 3-7, it implies a moving-away action that validates the correctness of our algorithm. The X-axis in Figure 3-9 indicates the index of each evaluation point.

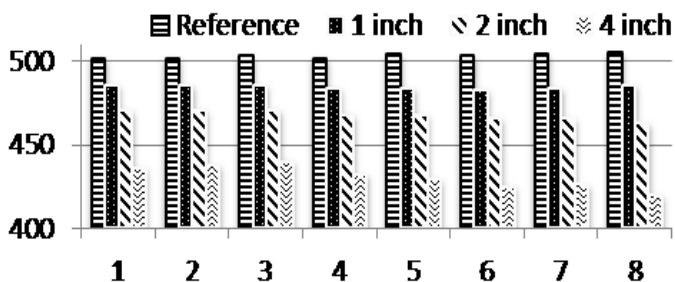


Figure 3-9. Evaluating the moving direction in z-axis

3.6.2. Sensitivity to the phone movements

We then evaluate the sensitivity of the movements. In the evaluation, we place the phone on one of the evaluation points (See Figure 3-8), and move the phone slowly to the right until the information being viewed is changed. The movement distance that triggers a coordinate change is recorded in Table 3-1. In summary, the average movement sensitivity is 2.6 mm.

Table 3-1. The movement sensitivity result

Point	1	2	3	4	5	6	7	8
Sensitivity	4mm	3mm	3mm	2mm	3mm	2mm	2mm	2mm

3.7. An Application – An Architectural Plan

The PhoneLens system provides a systematic way to overlay digital information on multivalent paper documents. A stand-alone desktop editor provides a GUI to customize the structure of digital information. The PhoneLens mobile application eliminates the need of carrying various multivalent paper documents to the field. This gives the convenience of both a paper document and having access to all other layers and related digital information efficiently.

PhoneLens is applicable in different contexts, such as construction building plan, electronic circuit plan, oil refinery plan and etc. This paper chooses an architectural plan as an example domain to evaluate the usability of PhoneLens. The architecture plan contains information that is classified into five different layers, i.e., the floor plan, sockets plan, lighting plan, fire & water plan, and waste plan. The PhoneLens allows users to view information at different layers simultaneously and provides them with functions (e.g., Layers, Annotation, Measurement, zooming) to help improve their browsing experience as discussed follows.

3.7.1. Layers

One of the most important features in the PhoneLens mobile application is to read information in various layers on top of a paper document. In a construction plan, the reference design is the floor plan, which mainly shows the placement of walls, stairs, and doors. The floor plan is a reference to related information in other layers. PhoneLens allows users to only carry the floor plan and to use their mobile phone on top of the floor plan to view other layers. This function is not limited to construction plans, and it is also useful in various fields, such as electronic circuits or oil refinery plans. Users can easily switch between different layers based on the layer specification in SVG files. A user can select one layer or even multiple layers simultaneously. For example, Figure 3-10.a shows a paper floor plan, and Figure 3-10.b presents a portion of the floor plan with corresponding lighting information.

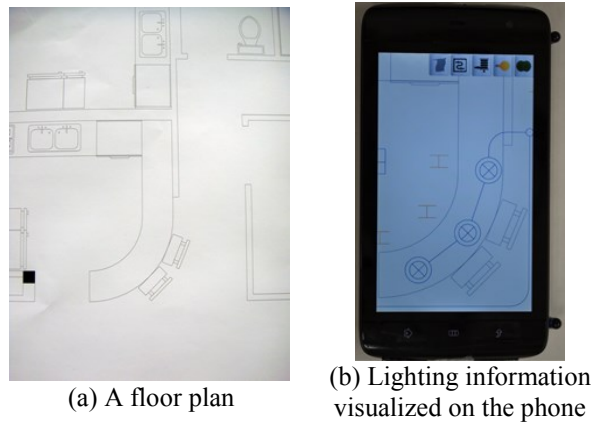


Figure 3-10. The lighting layer

3.7.2. Annotation

This feature allows users to annotate an object in a construction plan with digital multimedia information. For example, if an engineer inspects a building and finds that an electricity socket is not installed correctly according to the wiring plan, he can take a picture and use the picture as an annotation.

PhoneLens also allows users to search for an annotation. A user can either input a keyword or select an annotation type to trigger a search. After a user determines an annotation, an arrow pointing to the annotated object is displayed on the screen, as shown in Figure 3-11. When the user moves the mobile device toward or away from the annotated object, a number that indicates the distance between the current spot and annotated spot, is displayed besides the arrow. This feature helps users to locate the annotated spot quickly and easily.

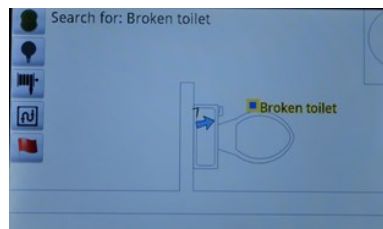


Figure 3-11. An arrow points to the annotation

3.7.3. Measurement

One of the most common tasks during the inspection of a design plan is to measure the distance and the area. In order to measure the distance, a user first moves the phone to the location where he/she wants to start the measurement, and touches the start point. A cross marker highlights the start point, as shown in Figure 3-12.a. The user then moves the phone to the target location and clicks on it. PhoneLens calculates the distance between two points and shows the distance in a message box (See 3-Figure 3-12.b). The same technique has been used to measure the area in terms of square inches. This feature allows users to perform a measurement without going to the physical location.



(a) Start Point

(b) Showing a measurement result

Figure 3-12. Measuring distance

3.7.4. Zooming

Users can issue a zooming command through three types of interaction.

- **Touch based zooming.** A user can trigger the *zoom in/out* menu items.
- **Gesture based zooming.** A pinch gesture allows users to zoom in by moving both fingers toward each other and zoom out by moving away from each other.
- **Distance based zooming.** When this type of zooming is activated, the mobile device acts like a camera viewfinder. When it gets closer to the paper document, it will zoom in.

Otherwise, it will zoom out. When the user reaches the desirable level of zooming, he/she just touches the screen to fix the zooming factor.

3.8. Empirical Study

The major goal of this study is to evaluate the usability of the PhoneLens for browsing the digital information contained in a multivalent document (an architectural plan). Paper documents are still one of the most popular and preferred media to carry information even in a digital age. For example, paper based maps are still very popular [Nor05], and traditional paper and pen in early architectural design are preferred by many professionals over computer-based tools [Ali06]. Our approach intends to improve the usability and accessibility of paper documents with newest technologies while still keeping the advantages of traditional paper documents. Therefore, this study compares the performance of the PhoneLens against the traditional paper-based browsing method, i.e., the state-of-the-art method used by engineers to inspect architectural plans.

This study is a pre-posttest repeated measures experiment (with the complete counterbalancing of the treatments) in which the participants browsed the multivalent document (an example architectural plan) using both the PhoneLens and the traditional paper-based method. The subjects were briefed on the multivalent document and trained on how to use each method to browse through the document. Next, the subjects were asked to complete different types of tasks and answer the questions based on the information contained in the document using each method. These answers were then evaluated by researchers. The researchers recorded the time it took the subjects to answer each question. The browsing efficiency of the browsing methods was then evaluated by comparing the total time spent on browsing the information. The details of the study are provided in the following subsections:

3.8.1. Research question and hypotheses

Using the Goal Question Metric (GQM) approach [Bas94] to define the goals for this study, we obtained the following goals and related hypotheses.

Goal 1: *Analyze* PhoneLens and Paper-based browsing methods *for the purpose of* their evaluation *with respect to* the browsing efficiency of multivalent documents.

Hypothesis 1: Participants using the PhoneLens spent significantly less time than the participants using the Paper-based browsing method.

Goal 2: *Analyze* PhoneLens and Paper-based browsing methods *for the purpose of* their evaluation *with respect to* their usability to browse the multivalent documents.

Hypothesis 2: Participants rated the PhoneLens significantly better than the paper-based method on its ease of use, comfort level and the practical usefulness for browsing through the multivalent documents.

Goal 3: *Analyze* the effect of independent variables *for the purpose of* improving the performance of PhoneLens method *with respect to* the browsing usability of the multivalent documents.

Hypothesis 3: Other independent variables (e.g., comprehension skills, prior domain knowledge, and usefulness of training) can affect the individual performance when using the PhoneLens method.

3.8.2. Independent and dependent variables

As a part of GQM approach, a set of metrics devised for the study served as independent and dependent variables for the study. The experiment manipulated the independent variables as listed and defined in Table 3-2. Table 3-2 also defines the dependent variables that were measured.

Hypothesis 3 investigates the effect that the independent variables defined in Table 3-2 have on the dependent variables.

Table 3-2. Study variables

Independent Variable	Definition
Browsing Method	The treatment method (i.e., PhoneLens or Paper-based) used to browse the document.
Training Usefulness	Measures the perceived usefulness of training procedure for each subject and each treatment method combination
Reading Comprehension	Measures the reading comprehension skills of each subject prior to the study
Domain Knowledge	Measures the prior knowledge of the domain of the document used in the study
Comfort Level	Measures the degree of comfort level for browsing information using the hand-held devices for each subject prior to the study
Dependent Variable	Definition
Browsing Efficiency	The time spent by each subject while browsing through the information contained in the document for each task and browsing method combination
Correctness	Measures the correctness of the information provided by subjects for each task and browsing method combination

The browsing method (i.e., PhoneLens or Paper-based) is the treatment method of this study. Details of the assignment of the subjects to different treatment methods are described in Section 3.8.3. As mentioned earlier, each subject was asked to complete different types of tasks using each treatment method. Detailed description of these tasks is also provided in Section 3.8.3.

3.8.3. Study details

Because the researchers wanted to test every subject for each treatment method, this study utilized a complete counterbalanced repeated measures design. That is, each subject was tested with both the PhoneLens and the Paper-based browsing method and was asked to perform two different types of tasks using each treatment method. The details of the study are provided in the following subsections.

3.8.3.1. Participating subjects

Twenty eight undergraduate/graduate students enrolled in the Computer Science and Information Science programs at a middle-west university participated in this study. The subjects were randomly selected from different courses and are not specifically targeted to benefit the study results. The subjects were trained by one of the researchers and worked individually to browse the information contained in the multivalent document (described in the next subsection) using both the treatment methods.

3.8.3.2. Artifact / document

The multivalent document used in this study was an architectural plan of an apartment building that had five different layers namely the: a) *Floor Plan*, b) *Waste Plan*, c) *Sockets Plan*, d) *Fire & Water Plan*, and the (e) *Lighting Plan*. The floor plan is a scaled diagram which shows a top view of walls, doors, spaces (rooms) and other physical features and relationships between them at one level of a structure. With the reference to a floor plan, the waste plan shows the locations and directions of waste pipes and waste outlets; the sockets plan shows locations of electric sockets and cabling between them; the fire & water plan shows locations of water and fire outlets and water piping; and the lighting plan shows locations of light switches and cabling between them. Each of these layers was drawn on a separate paper-document. The participating subjects using the *paper-based method* used all the five different paper-documents (corresponding to each layer) to complete the tasks that required them to browse the information contained in different layers. The participating subjects using the *PhoneLens* used the PhoneLens mobile application on top of the floor plan document to access all the other layers digitally to complete the tasks.

While the PhoneLens can be used on any multivalent document, the architectural plan used in this study was motivated by that fact that it is a standard apartment building plan with different layers that everyone is familiar with. Furthermore, the plan was kept simple in order to avoid any “noise” due to the subject’s lack of understanding of the architectural plan. The impact of having a more complex architectural document is discussed in the threats to validity section later in the paper.

3.8.3.3. Study tasks

As mentioned earlier, each subject was asked to browse through the architectural plan to complete two different sets of tasks using the PhoneLens and the paper-based method. A brief description of these tasks follows. Based on the *IEE Wiring Regulation 17th Edition (BS7671) - Section 528*, when a wiring system is installed in proximity to any other services, it shall be arranged so that any foreseeable operation carried out on either service will not cause damages to another service. Consequently, an inspector must be able to check different layers (i.e. piping & wiring layers) simultaneously based on the above regulation. Therefore, we design a navigation task that requires participants to browse several layers.

- a) *Navigate* an architectural plan and search for specific information at different layers (e.g., looking for an electricity plug behind a dishwasher in the sockets plan or a sewage outlet behind a dish washer in the waste plan). For each method, the subjects were provided three different statements that pertain to the information contained in the different layers of the document. The subjects were then asked to browse the document and answer if the statements were true or false.

- a. Browse the *sockets plan* and check if there is at least one electricity plug on the wall behind the dish washer 1/1’;
 - b. Browse the *waste plan* and check if there is at least one sewage outlet on the wall behind the dish washer 1/1’; and
 - c. Browse the *fire & water plan*, and check if there is at least one hot and one cold water outlets on the wall behind the dish washer 1/1’.
- b) Search for an annotation and Measure the area of the corresponding room that includes that annotation. As opposed to the *navigation* tasks that had true or false answers, the *search* and *measurement* tasks required the subjects to provide the actual numbers as output. This task required the subjects to search for all broken ceilings (or broken tiles), count them, and measure the area of the room which contains one of them. The exact questions provided to them are listed below:
- a. Count the total number of “Broken Ceilings”/“Broken Tiles”; and
 - b. Measure the area of a room (in square inch) which includes a “Broken Ceiling” / “Broken Tile” annotation.

The “Search and Measurement” subtasks were designed to simulate the actual working conditions of an engineer that would need to search for a piece of information, and then perform computation using that information. Even if a participant could not correctly count the number of broken ceilings or tiles, he/she still could measure the area of a room independent of the success on the pervious task.

Since each subject performed these tasks using both treatment methods, in order to avoid the learning effect, the exact questions were altered slightly in order to ensure that they were not

looking for the same information using both the treatment methods. More details of the order of tasks and treatment method in discussed in the following subsection.

3.8.3.4. Study procedure

The experiment operation included three different steps and one training session. Figure 3-13 provides an overview of the experiment steps. The details of each step are provided in this section.

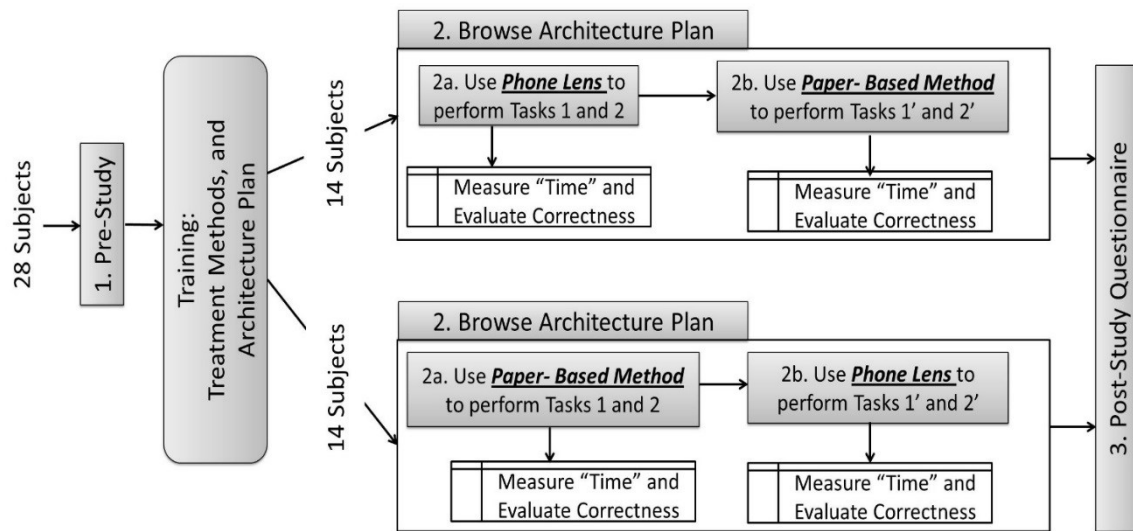


Figure 3-13. Study design

3.8.3.4.1. Step 1. Pre-study survey

The first step was to collect the background information from the participating subjects regarding their reading comprehension skills, their prior knowledge of the architectural plans, and their experience and comfort level with browsing on hand-held devices. The information during the pre-study was used to gain additional insights into the individual performance of subjects during the experiment.

3.8.3.4.2. Training session

Following the pre-study survey, the subjects were trained by the same researcher on each treatment method (i.e., PhoneLens and Paper-based) prior to the start of study. During the ten to fifteen minutes training session, the subjects were briefed on the architectural plan and the different layers of the architectural plan. Next, subjects were taught how to use the paper-based method and the PhoneLens to browse through the architectural plan. The subjects were then asked to try both the methods to get familiar with the architectural plan and how to use the PhoneLens to browse the architectural plan.

3.8.3.4.3. Step 2. Browsing architectural plan

The 28 participating subjects were randomly divided into two groups of 14 subjects each as shown in Figure 3-13. One group of subjects browsed the architectural plan using the PhoneLens followed by the paper-based architectural plan, whereas the other group of subjects was tested with paper-based architecture plan followed by the PhoneLens method. Therefore, we tested each subject for both treatment methods.

During this step, each subject was asked to complete two types of tasks (i.e., *Navigation* task and the *Search and Measurement* task) using each treatment method. The details of these tasks are provided in Section 3.8.3.3. Since subjects used one method followed by the other method (i.e., either PhoneLens followed by the paper-based or vice-versa) to complete these tasks, the questions were altered that required the subjects to look at different portions of the architectural plan layers to answer the questions related to each task. For example, if a subject used the PhoneLens to check if there is at least one electricity plug on the wall behind the dish washer 1, then he/she used the

Paper-based method to check if there is at least one electricity plug on the wall behind the dish washer 1’.

Therefore, each subject in group 1 completed four different tasks, two using the PhoneLens (i.e., task 1 related to dish washer 1 and task 2 related to broken ceilings), and then completed two additional tasks using the paper-based (i.e., task 1’ related to dish washer 1’ and task 2’ related to broken tiles) as shown in Figure 3-13 and listed in Section 3.8.3.3. Conversely, each subject in group 2 uses the treatment methods in a reverse order as compared to subjects in group 1 to complete four different tasks. That is, each subject in group 2 used the paper-based method to complete task 1 (related to dish washer 1) and task 2 (related to broken ceilings) followed by the PhoneLens method to complete task 1’ (related to dish washer 1’) and task 2’ (related to broken tiles). This allowed the researchers to minimize potential learning effects, and to compare the performance of subjects using the PhoneLens and paper-documents across four different tasks (i.e., tasks 1, 2, 1’ and 2’).

The researcher who had trained the subjects also kept a log of the time it took them to complete each task using both the treatment methods as well as the total time spent to complete all the tasks. The same researcher also recorded the answers to the questions related to each task and then evaluated the students responses at the end of the study.

3.8.3.4.4. Step 3: Post-study questionnaire

In the end, participants were asked to provide feedbacks on their use of PhoneLens and paper-based browsing methods using a 5-point Likert-type scale (ranging from “1- very low” to “5- very high”).

3.8.4. Data collection

This section provides a brief description of qualitative and quantitative data collected during the study. The quantitative data include the time (in seconds) spent by subjects to complete each of the four tasks (two tasks per each treatment method). The researcher recorded the timing information including the start and end times for each task, the time they found the desired information in the architectural plan, and any breaks they took. The researchers also evaluated the correctness of the responses provided by the subjects for each task. For the *Navigation* task, the correctness was evaluated by comparing the subject's responses of "True/False" statements against the correct responses. For the *Search* and *Measurement* task, the count of the broken tiles/ceilings and the area recorded by the students were evaluated against the actual values.

For qualitative data, we gathered the subjective self-reported data during the pre-study and at the end of the study run. Using a five point scale (ranging from "1-very low" to "5-very high"), the participating subjects rated their overall experience in terms of their comfort level in browsing through the architectural plan using both treatment methods. Furthermore, each subject was asked to rate both the treatments methods on four different characteristics using the 5-point scale: 1) accessing the desired information, 2) during the searching for the annotations, 3) performing the measurement tasks, and 4) practical usefulness of the browsing method. Also, we asked subjects to rate the usefulness of training provided to them for each treatment method prior to the beginning of the study.

3.9. Data Analysis and Results

This section provides an analysis of the data collected during the study. The results are organized around the three research goals (and related hypotheses) presented in Section 3.8.1. An alpha value of 0.05 was selected for judging the significance of the results.

3.9.1. H1: Comparison of browsing efficiency of Phone-Lens and paper-based methods

We analyzed the browsing efficiency by comparing the average time spent (in seconds) by subjects in each group using the PhoneLens and paper-based method. The comparison of both methods was performed separately for the “navigation” tasks and for the “search and measurement” tasks. Therefore, we compared the average time spent by subjects for all the four tasks (i.e., tasks 1, 1’, 2, and 2’) as shown in Figure 3-14. Due to the nature of repeated measures tasks, a Wilcoxon test (instead of a t-test) was used to judge the significance of the results and is discussed below.

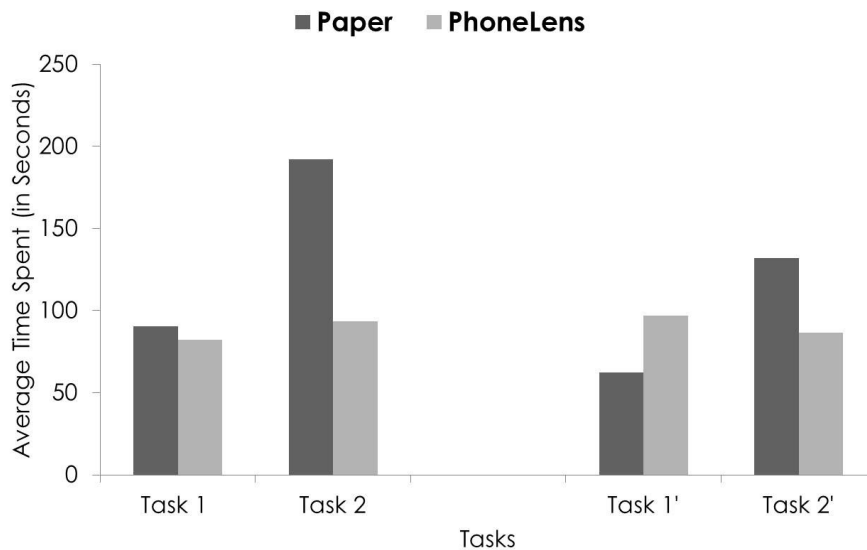


Figure 3-14. Comparison of average time spent on tasks 1, 1', 2 and 2'

During the *navigation* tasks (Tasks 1 and 1'), the PhoneLens is not always more efficient than paper documents. During task 1, the PhoneLens performed marginally better over paper-based approach (though non-significant); whereas during the Task 1', the paper-based approach showed improvement over the PhoneLens (again non-significant). Since the trends are in opposite directions, there is an *order effect* in place. Furthermore, since the navigation tasks were relatively simple and required subjects to confirm or refute the statements, this result is not surprising.

During the *Search and Measurement* tasks, the PhoneLens was more efficient than the paper-based method. During the task 2, the Phone-Lens method was significantly more efficient than the paper-based method spending an average of 93 seconds compared to the 192 seconds for the paper-based method ($p = 0.00096$). Similarly, during the task 2', the Phone-Lens method was again significantly more efficient than the paper-based method spending an average of 86 seconds compared to the 132 seconds for the paper-based method ($p = 0.001$). These results are shown in Figure 3-14. Analysis of these results shows that irrespective of the order of the treatment methods, the PhoneLens is a significant improvement over the traditional method during the *search* and *measurement* tasks. However, no significant difference was observed for the navigation tasks between the treatment methods. Analysis of the time spent by individual subjects showed that, the timing values in data sets for paper-based method are far apart from the average value as compared to the PhoneLens method for both the tasks. Combining the results for all the tasks for PhoneLens and paper-based method, the time spent by subjects also showed a much higher upper-bound (i.e., 75th percentile) when using the Paper-based method (a value of 3 minutes) as compared to the PhoneLens (1.5 minutes). Based on these results, the timing values for PhoneLens were closer to the average values and showed a more consistent distribution.

Furthermore, we analyzed the correctness of the responses provided by subjects using the PhoneLens and the paper-based method. The results showed that the subject's responses on the *navigation* tasks (i.e., true or false statements) were correct on using both the methods. However, the subject's responses on the *measurement* tasks (i.e., area size) using the paper-based method were incorrect for majority of the subjects (i.e., 23 out of 28 subjects) as compared to the 6 out of 28 subjects using the Phone-Lens that reported the incorrect output. In the paper-based method, the measurement errors are mainly caused by forgetting the scale in the calculation. On the other hand, PhoneLens has a much higher accuracy rate in the measurement task. In the evaluation, we found that the "fat finger" error that is common in touch-based screens can affect the accuracy in the measurement. A user may not precisely touch the point he/she needs to touch on a touch screen. This issue can be addressed by providing an option that snaps a user's touch point to the nearest intersection or corner.

Therefore, based on these results, PhoneLens is significantly more efficient and can help users achieve more accurate information as compared to the paper-documents on the *annotation and measurement* tasks. The overall evaluation results justify the usefulness of the PhoneLens. Since this was an initial investigation, some advanced features were not explored. For example, subjects were not asked to take the advantage of assessing the digital information at different layers simultaneously or using the zooming features using the PhoneLens. We will investigate the usefulness of these features in future experiments. In future evaluation on more complex tasks with PhoneLens, we expect the efficiency results to show a similar significant improvement.

3.9.2. H2: Comparison of subjective feedbacks on PhoneLens and paper-based methods

Using a five-point scale (ranging from “1- very low” to “5- very high”), the participants evaluated their overall experience using PhoneLens and the Paper-based treatment on six relevant characteristics (C1 through C6)_listed in Figure 15. We categorized the rating results in to two main categories, “Disagree” and “Agree”³. “Disagree” category included all rating values lower or equal to 3 and “Agree” category included rating values 4 and 5.

Figure 3-15 compares the percentage of “Disagree” vs. “Agree” ratings on each characteristic for both methods. For each characteristic in PhonLens method, we conducted a non-parametric one-sample Wilcoxon Signed Rank test to determine whether the mean response was significantly greater than the midpoint of the scale. In the PhoneLens approach, the results indicate a significantly positive feedback (i.e., $p < 0.05$) on all characteristics except the “Comfort level in browsing through the plan” (C1). Overall, the results show a larger percentage of agreements for the PhoneLens method as compared to the paper-based method except C6, which means that the subjects rated the training on both methods equally well prior to performing the browsing tasks but rated PhoneLens more favorably during the browsing of architectural plan.

³ Based on the analysis method suggested in www.likert.org

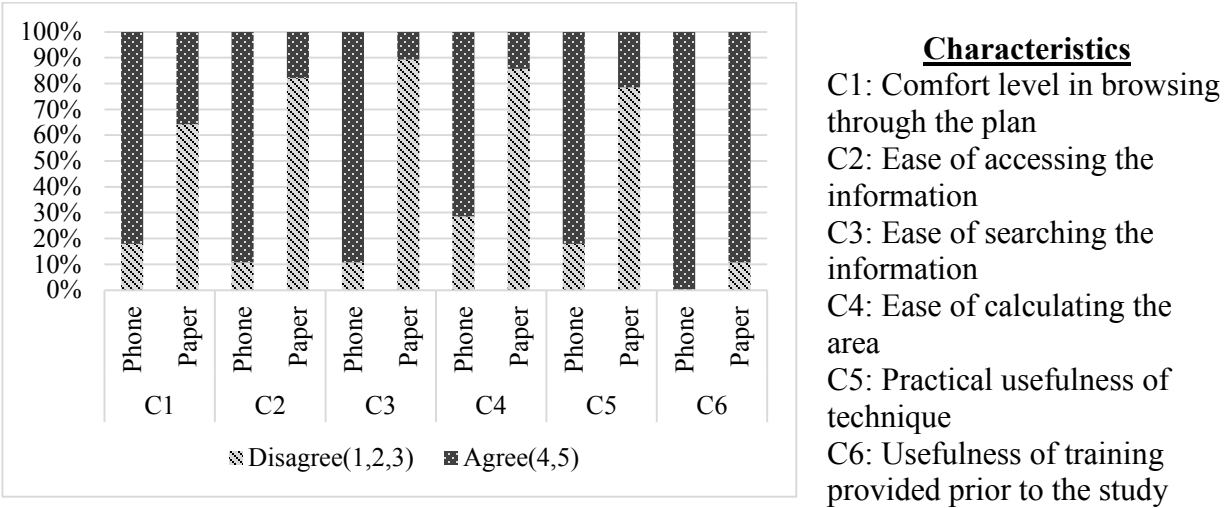


Figure 3-15. Percentage of the rating on each characteristic for each method (Categorized)

In addition, the feedback from the subject’s comments identified some issues that will help improve the design of the PhoneLens. Some participants commented that control icons are displayed too close, which accidentally triggered the function of an adjacent control. In the function of searching an annotation, it was also suggested to provide a search history in future implementations.

3.9.3. H3: Effect of independent variables on individual performance of subjects

We analyzed the effects of other independent variables on the dependent variables (Table 3-2) using multiple regression [Fie07]. The goal was to find any significant correlations between those variables and participants browsing efficiency during the experiment run. The training usefulness, comfort level with hand-held devices, the reading comprehension skills, and the prior experience with the architectural plans were measured on a 5-point scale. The results show that only the *Usefulness of training* on the PhoneLens was significantly correlated with the browsing efficiency of subjects when considering the total spent by subjects on all the tasks using the PhoneLens. That is, the participants who found the PhoneLens training useful were more effective

when using the PhoneLens during the study run. The multiple regression analysis did not show any significant positive correlations among the independent variables.

3.9.4. Threats to validity

We faced the following threats to validity during the study that could not be completely addressed:

3.9.4.1. Conclusion validity

The threat due to the heterogeneity of participants was not controlled since all participants were not drawn from the same course and were a mix of bachelor and master students (i.e., had different levels of education). While we used a 5-point likert scale, there remains a threat that we treated these scales as an interval scale rather than ordinal scale, following the standard practice in social sciences. This practice means that the p-value needs to be treated with care when interpreting the results.

3.9.4.2. External validity

There remains a threat because the participants were all undergraduate and graduate students in an educational setting and likely not represent typical users of the PhoneLens when used in reality. Furthermore, the students are less experienced in working with architectural plans as compared to the intended users.

3.9.4.3. Internal validity

To increase internal validity, we did not inform the participants of the study goals. Therefore, they should not have been biased in the data provided. Participants volunteered to take

part in this study and were not graded on their performance. The most important internal validity threat in the study was the lack of a classic control group. We plan to address this threat in future.

3.10. Discussion of Results

The result in Section 3.9.1 showed that the PhoneLens was significantly more efficient than the paper-based architectural plan on more complex “*search and measurement*” tasks as compared to the “*navigation*” tasks. The architectural plan used in the evaluation was kept simple to isolate the effect of the treatment method. While the study subjects (i.e., students recruited from CS and MIS programs) are more inclined to try a new technology as compared to the intended users (i.e., engineers), they may not well-versed with the architecture models as compared to the engineers. Therefore, our future studies will investigate the generalizability of our study results by testing the Phone-Lens with actual intended users in their work-setting.

Furthermore, we expect the efficiency of the PhoneLens to show further improvement over the paper-based method on more complex examples. A complex multivalent document has several layers with a large size. PhoneLens overlaps information of different layers on the same reference design simultaneously, which avoids frequent flipping between different layers. With a natural interaction, PhoneLens facilitates searching, editing and annotating information efficiently. Therefore, we expect that the *search* and *navigation* efficiency using the PhoneLens can be significantly improved on complex cases over paper documents.

Also, the result in Section 3.9.1 showed that the responses provided by the subjects were more accurate and precise when using the PhoneLens method. Because the subjects were Computer and Information science students, and are not the actual users of architectural plans, we do not claim the accuracy of the results (e.g., area size) to be generalizable. However, the results

in Section 3.9.3 show that the subject's computer science background (e.g., user interface development) and their browsing experience on hand-held devices are not correlated with their performance during the study. Therefore, we anticipate the efficiency and other results to be valid with more generic non-computer/information science participants in our future studies. As additional evidence, the result in Section 3.9.3 showed that the subjects, who rated the training on PhoneLens highly, were significantly more efficient using the PhoneLens than the subjects who did not rate the PhoneLens training as favorably. This result provides evidence that the usefulness of the training is more relevant than the background experience of the participating subjects.

Additionally, the results in Section 3.9.2 showed that subjects rated the PhoneLens method more favorably than the traditional paper-based method on the tasks the subjects were asked to perform during the study run. According to the feedback from users at the end of the study, we were able to identify some areas to help improve the usability of the PhoneLens, such as increasing the space between two adjacent icons.

The intended use of the PhoneLens is believed to be in domains that include coupled information in different layers. PhoneLens provides a "context+focus" interaction, which is useful to browse a large amount of information, such as maps or graphs, on mobile devices. In this study an apartment architectural plan (a common type of multivalent documents) was used to avoid losing generality. The positive results have motivated us to investigate the use of PhoneLens in other domains as well. Last, while only a subset of PhoneLens features were investigated during the study, we anticipate that some of the more advanced features (e.g., zooming or browsing multiple layers simultaneously) would further improve the browsing efficiency of the PhoneLens

method when used on multivalent documents. We plan to investigate the use of PhoneLens method on other types of documents and with more complex study tasks.

3.11. Conclusion

In this paper, we have presented a spatially aware mobile system, called PhoneLens. This system augments paper documents with digital information. The main strengths of this system are being low cost and robust in different indoor environments, and portable with low complexity and ease of set-up. This system is developed to support various multi-layer vector graphical files, such as construction plans, electronic circuits or oil refinery plans. The PhoneLens system enables a spatially-aware display. When a user places a mobile device on any part of a printed paper document, it will show the corresponding digital information related to that area. PhoneLens is developed with different functions, including zooming, exploring various layers, measurement and annotation. A preliminary user study justifies the usefulness and high usability of the proposed prototype. More comprehensive studies will be conducted in the future, such as comparing PhoneLens with mobile based applications or evaluating PhoneLens in a real setting.

3.12. References

- [Ali06] Aliakseyeu. D., Martens. J. B., and Rauterberg. M., “A computer support tool for the early stages of architectural design”, *Interacting with Computers*, Vol. 18, 528-555, 2006.
- [Ano02] Anoto, Development Guide for Service Enabled by Anoto Functionality, 2002.
- [Ara97] Arai. T., Aust. D., and Hudson. S. E., “PaperLink: a Technique for Hyperlinking from Real paper to Electronic Content”, *Proc. CHI'97*, pp.327-334, 1997.

- [Bas94] Basili. V. R., Caldiera. G., Rombach. H. D., “The Goal Question Metric Approach,” *Technical Report, Department of Computer Science, University of Maryland, 1994*, <ftp://ftp.cs.umd.edu/pub/sel/papers/gqm.pdf>.
- [Bed94] Bederson. B. and Hollan. J. D., “Pad++: A Zooming Graphical Interface for Exploring Alternative Interface Physics”, *Proc. UIST’ 94*, pp.17-26, 1994.
- [Bur08] Burigat. S., Chittaro. L. and Gabrielli. S., “Navigation Techniques for Small-Screen Devices: An Evaluation on Maps and Web Pages”, *International Journal of Human-Computer Studies*, 66, pp.78-97, 2008.
- [Bux86] Buxton. W. and Myers. B., “A Study in Two-handed Input”, *Proc. CHI’86*, pp.321-326, 1986.
- [Cao06] Cao. X. and Balakrishnan. R., “Interacting with Dynamically Defined Information Spaces using a Handheld Projector and a Pen”, *Proc. UIST’ 06*, pp.225-234, 2006.
- [Cho07] Cho. S., Murray-Smith. R., and Kim. Y., “Multi-context photo browsing on mobile devices based on tilt dynamics”, *Proc. Mobile HCI*, pp. 190–197, 2007.
- [Cha09] Chao. P. Y. and Chen. G. D., “Augmenting paper-based learning with mobile phones”, *Interacting with Computers*, Vol. 21, 173-185, 2009.
- [Cha10] Chan. L. W., Wu. H. T., Kao. H. S., Ko. J. C., Lin. H. R., Chen. M. Y., Hsu. J., and Hung. Y. P., “Enabling beyond-surface Interactions for Interactive Surface with an Invisible Projection”, *Proc. UIST’20*, pp. 263-272, 2010.
- [Ero08] Erol. B., Antunez. E., Hull. J. J., “Hotpaper: Multimedia Interaction with Paper Using Mobile Phones”, *Proc. Multimedia ’08*, pp. 399-408, 2008.

- [Fie07] Field A., *Discovering Statistics Using SPSS*. 2nd ed. 2007, SAGE Publications Ltd, London.
- [Fit93a] Fitzmaurice. G. W., “Situated Information Spaces and Spatially Aware Palmtop Computers”, *Communications of the ACM*, Vol.36(7), pp.38-49, 1993.
- [Fit93b] Fitzmaurice. G. W., Zhai. S., and Chignell. M., “Virtual Reality for Palmtop Computers”, *ACM TOIS*, Vol.11(3), 1993.
- [Gui03] Guimbretière. F., “Paper Augmented Digital Documents”, *Proc. UIST’03*, pp. 51-60, 2003.
- [Han07] Hannuksela. J., Sangi. P., and Heikkilä. J., “Vision-based motion estimation for interaction with mobile devices”, *Computer Vision and Image Understanding*, 108 (1–2), pp.188–195, 2007.
- [Hal10] Haller. M., Leitner. J., Seifried. T., Wallace. J. R., Scott. S. D., Richter. C., Brandl. P., Gokcezade. A., and Hunter. S., “The NICE Discussion Room: Integrating Paper and Digital Media to Support Co-Located Group Meetings”, *Proc. CHI 2010*, 2010.
- [Har05] Haro. A., Mori. K., Setlur. V., and Capin. T., “Mobile camera-based adaptive viewing”, *Proc. International conference on Mobile and ubiquitous multimedia*, pp 78-83, 2005.
- [Har08] Hare. J. S., Lewis. P. H., Gordon. L., and Hart. G., “MapSnapper: Engineering an Efficient Algorithm for Matching Images of Maps from Mobile Phones”, *Proc. Multimedia Content Access: Algorithms and Systems II*, 2008.
- [Har11] Harrison. C., Benko. H., and Wilson. A. D., “OmniTouch: Wearable Multitouch Interaction Everywhere”, *Proc. UIST’11*, pp.441-450, 2011.

- [Hul07] Hull. J. J., Erol. B., Graham. J., Ke. Q., Kishi. H., Moraleda. J., and Olst. D. G. V., “Paper-based Augmented Reality”, *Proc. International Conference on Artificial Reality and Telexistence*, pp.205-209, 2007.
- [Lee08] Lee. J. C., 2008. Hacking the Nintendo Wii Remote. *IEEE Pervasive Computing* 7, 3, 2008, 39-45.
- [Lee11] Lee. J. C., Wiimote Whiteboard, Finger Tracking Projects, 2011. <http://johnnylee.net/projects/wii/>.
- [Lia08] Liao. C. Y., Guimbretière. F., Hinckley. K., and Hollan. J., “PapierCraft: A Gesture-Based Command System for Interactive Paper”, *ACM ToCHI*, Vol.14(4), pp.1-27, 2008.
- [Lia10a] Liao. C. Y., Liu. Q., Liew. B., and Wilcox. L., “PACER: Fine-grained Interactive Paper via Camera-touch Hybrid Gesture on a Cell Phone”, *Proc. CHI'2010*, pp.2441-2450, 2010.
- [Lia10b] Liao. C. Y., Tang. H., Liu. Q., Chiu. P., and Chen. F., “FACT: Fine-grained Cross-media Interaction with Documents via a Portable Hybrid Paper-laptop Interface”, *Proc. the International Conference on Multimedia*, pp.361-370, 2010.
- [Lie98] Liebowitz. D. and Zisserman. A., “Metric Rectification for Perspective Images of Planes”, *Proc. CVPR '98*, pp. 482-488, 1998.
- [Liu08] Liu. X. and Doermann. D., “Mobile Retriever: Access to Digital Documents from Their Physical Source”, *International Journal on Document Analysis and Recognition*, Vol.11(1), pp.19-27, 2008.
- [Mac02] Mackay. W. E., Pothier. G., Letondal. C., Boegh. K., and Sorensen. H. E., “The Missing Link: Augmenting Biology Laboratory Notebooks”, *Proc. UIST' 02*, pp.41-50, 2002.

- [Mor09] Morrison. A., Oulasvirta. A., Peltonen. P., Lemmela. S., Jacuci. G., Reitmayr. G., Nasanen. J., Juustila. A., "Like Bees Around the Hive: A Comparative Study of a Mobile Augmented Reality Map", *Proc. CHI'09*, 2009.
- [Nor05] Norrie. M. and Signer. B., "Overlaying Paper Maps with Digital Information Services for Tourists," in *Information and Communication Technologies in Tourism*, Austria, 2005, pp. 23-33
- [Olw06] Olwal. A., "LightSense: Enabling Spatially Aware Handheld Interaction Devices", *Proc. IEEE/ACM ISMAR*, pp. 119-122, 2006.
- [Phe97] Phelps. T. A. and Wilensky. R., "Multivalent Annotations", *Proc. ECDL '97*, pp.287-303, 1997.
- [Rei06] Reilly. D. R., Rodgers. M., Argue. R., Nunes. M., and Inkpen. K., "Marked-up maps: Combining Paper maps and Electronic Information Resources", *Personal and Ubiquitous Computing*, Vol.10(4), pp.215-226, 2006.
- [Roh04] Rohs. M., "Real-world Interaction with Camera-phones", *Proc. 2nd International Symposium on Ubiquitous Computing Systems*, pp.74-89, 2004.
- [Roh07a] Rohs. M., Schöning. J., Raubal. M., Essl. G., Krüger. A., "Map Navigation with Mobile Devices: Virtual versus Physical Movement with and without Visual Context", *Proc. ICMI'07*, 2007.
- [Roh07b] Rohs. M., and Essl. G., "Sensing-based interaction for information navigation on handheld displays", *Proc. Mobile HCI*, pp. 387-394, 2007.
- [Sch04] Schrott. G. and Gluckler. J., "What Makes Mobile Computer Supported Cooperative Work Mobile? Towards a Better Understanding of Cooperative Mobile Interactions", *International Journal of Human-Computer Studies*, 60, pp.737-752, 2004.

- [Sch07] Schmalstieg. D. and Wagner. D., “Experiences with Handheld Augmented Reality”, *Proc. IEEE/ACM ISMAR*, pp.3-15, 2007.
- [Son09] Song. H.Y., Grossman. T., Fitzmaurice1. G., Guimbretière. F., Khan. A., Attar. R., and Kurtenbach. G., "PenLight: Combining a Mobile Projector and a Digital Pen for Dynamic Visual Overlay", *Proc. CHI'09*, pp.143-152, 2009.
- [Son10] Song. H. Y., Guimbretière. F., Grossman. T., Fitzmaurice. G., “MouseLight: Bimanual Interactions on Digital Paper Using a Pen and a Spatially-Aware mobile Projector”, *Proc. CHI'10*, pp.2451-2460, 2010.
- [Tsa02] Tsang. M., Fitzmaurice. G., Kurtenbach. G., Khan. A. and Buxton. B., “Boom Chameleon: Simultaneous Capture of 3D Viewpoint, Voice and Gesture annotations on a Spatially-aware Display”, *Proc. UIST' 02*, pp.111-120, 2002.
- [Wag10] Wagner. D., Reitmayr. G., Mulloni. A., Drummond. T. and Schmalstieg. D., “Real-Time Detection and Tracking for Augmented Reality on Mobile Phones”, *IEEE Transactions on Visualization and Computer Graphics*, Vol.16(3), pp.355-368, 2010.
- [Wan06] Wang. J., Zhai. S., Canny. J.,”Camera phone based motion sensing: interaction techniques, applications and performance study”, *Proc. the ACM Symposium on User Interface Software and Technology*, pp.101–110, 2006
- [Wei08] Weibel. W., Ispas. A., Signer. B., and Norrie. M.C., “Paperproof: a Paper-digital Proof-editing System”, *Proc. CHI '08*, pp.2349-2354, 2008.
- [Wel93] Wellner. P., “Interacting with Paper on the DigitalDesk”, *Communications of the ACM*, Vol.36(7), pp.87-96, 1993.

- [Wii11] Distance Measurements with the WiiMote, 2011, <http://wiiphysics.site88.net/physics.html>
- [Wil11] Willis. K. D.D., Poupyrev. I., Hudson. S. E., Mahler. M., “SideBySide: Ad-hod Multi-user Interaction with Handheld Projectors”, *Proc. UIST'11*, pp.431-440, 2011.
- [Wob08] Wobbrock. J. O., Myers. B. A., and Aung. H. H., “The Performance of Hand Postures in Front- and Back-of-Device Interaction for Mobile Computing”, *International Journal of Human-Computer Studies*, 66, pp.857-875, 2008.
- [Yeh06] Yeh. R. B., Liao. C., Klemmer. S.R., Guimbretière. F., Lee. B., Kakaradov. B., Stamberger. J., and Paepcke. A., “ButterflyNet: A Mobile Capture and Access System for Field Biology Research”, *Proc. CHI'06*, pp.571-580, 2006.
- [Yee03] Yee. K. P., “Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers”, *Proc. CHI'03*, pp.571-580, 2003.
- [Yim11] Yim. S., Lee. S., and Choi. S., “Evaluation of motion based interaction for mobile devices: A case study on image browsing”, *Interacting with Computers*, 23(3), 268-278, 2009.

4. A LOW-COST AND INTELLIGENT CAMERA MANAGEMENT SYSTEM

Automatic camera management systems have been developed to automatically record videoconferencing. These systems provide many benefits, such as reducing production costs and conveniently documenting events for future viewing. However, automatically recorded videos generally lack visual interest. This paper presents a novel approach that intelligently manages camera shots and angles to improve the visual interest. We use 3D infrared images captured by a Kinect sensor to recognize active speakers and their positions in a meeting. A movable camera, which is constructed by placing a wireless PTZ camera on top of a motorized rail, can automatically move its position to frame the active speaker in the center of the screen. Without interfering with a meeting, a speaker can seamlessly switch video sources (such as a PPT or images) through gesture commands. We have summarized and implemented a set of heuristic rules to simulate a virtual director. Those rules can be visually customized through a graphical user interface. The customization of a virtual director makes our system applicable in various scenarios. We have conducted a user study, and the evaluation results are promising.

4.1. Introduction

Videoconferencing systems automatically record a meeting and transmit the video to remote participants. However, an automatically captured video is often not engaging, and lacks visual variety such as switching from an overview shot of the whole group to a close-up shot of the active speaker. On the other hand, a professional director produces engaging videos by switching between multiple cameras to provide a variety of interesting views [Ran10], however, human-operated video recording is labor intensive and costly.

Various automatic camera management systems [Yu10] have been developed to produce an engaging video without human involvement. In an automatic video production, it is critical to recognize a dominant speaker and accordingly adjust the camera's shot and angle. Various approaches have been proposed to control a camera for framing active speaker, such as an omnidirectional camera [Rui01, Foo00, Cut12] with a 360-degree view or an array of microphones and PTZ cameras [Ran10].

Previous approaches generally place a camera at a fixed position. Since each camera can only cover a portion of the whole scene, it is necessary to calibrate multiple cameras. This paper presents a novel system, i.e., *SmartCamera*, which only uses one movable camera to frame an active speaker based on his/her position and pose. More specifically, *SmartCamera* is built on a Kinect sensor that provides an overview shot and captures 3D images to track speakers and their gestures. After detecting an active speaker, a movable camera, which is constructed by mounting a PTZ camera on a motorized rail, moves to an appropriate position with a suitable angle based on the location of the dominant speaker and his/her pose. In order to provide an engaging video with various camera shots, we implement a virtual director to control cameras through a set of heuristic rules, which can be visually edited and customized by a video production professional.

Our work focuses on recording a discussion or meeting in which all speakers' activities are covered within the range of an overview camera, i.e., a Kinect sensor. The meetings where a speaker has his/her back towards the overview camera are out of scope of this paper.

The major contributions of this paper are summarized as follows:

- **A movable camera.** We designed a movable camera which is able to capture every speaker, while avoiding the calibration of multiple cameras. Furthermore, a movable camera assures that the camera can always directly face an active speaker.
- **Seamlessly switching video sources.** It is useful for speakers to access multimedia content (such as PPTs) without interrupting the meeting. Our approach supports gesture-based and voice-based commands to naturally interact with multimedia contents and seamlessly switch the video source between multimedia contents and speaker.
- **A customizable virtual director.** A virtual director is essential to produce an engaging video. It includes a set of director rules that intelligently control a movable camera according to different events, such as the change of a speaker. Each director rule is defined through a workflow that specifies a series of camera activities to respond to an event. A workflow can be modified visually through a graphical user interface. This customization makes our approach applicable in various scenarios.
- **Efficiently detecting speakers.** To the best of our knowledge, we are the first to utilize the Kinect sensor to automatically record videos. 3D infrared images captured through Kinect are applicable in different lighting conditions to detect the skeleton of each speaker, and the depth information makes the body recognition more efficient and accurate. Furthermore, the integration of sound and vision tracking in the Kinect sensor simplifies the hardware setup.

4.2. Related Work

An automatic meeting capturing system should meet three main criteria:

1. *Track speakers, their movements, voices and gestures.* It is challenging to track and capture speakers in a meeting because speakers have different behaviors over time (such

as changing his/her body pose, start or stop speaking) [Pol97, Ran08]. Furthermore, speakers may have varying requests.

2. *Provide an engaging and attractive video output.* An automatically produced video is often not engaging and attractive. Consequently, people get bored when watching the video [Rub02, Ran08, Ran10]. This issue is mainly caused by the lack of various shots from different angles [Ino95], which forces people to watch the video from a fixed shot. In order to solve this issue, it is important to apply TV production rules in the video production process [Ino95, Bia98, Liu01, Ran08].
3. *No human effort.* In a meeting, speakers need to focus on the discussion. They should not be disrupted to guide an automatic system to record the meeting [Ran06, Rui03]. Therefore, a meeting capturing system should be intelligent with only minimal manual inputs from speakers.

Researchers have proposed different solutions to meet the above three criteria. The following subsections review previous solutions.

4.2.1. Tracking technologies

Tracking speakers' activities in a meeting provides the necessary information for a virtual director to decide what shots should be taken. Therefore, tracking speakers is one important component in an automatic video production system. Its goal is to constantly recognize the position of each speaker and identify the dominant speaker at any time.

Various sensing techniques have been proposed to track speakers. Microphone array [Bra01, Lee02, Liu01] is developed to recognize the location of sound in a physical environment. Rui *et al.* [Rui01] combined a microphone array with an omni-directional camera, which can

capture a 360-degree view. This approach is suitable for recording meetings, in which people are seated in a circle. Lee *et al.* [Lee02] used an omni-directional camera together with four microphones. This approach uses motion analysis and skin detection to recognize speakers. In order to reduce the cost, Cutler *et al.* [Cut12] used multiple inexpensive cameras to replace an omni-directional camera. Those cameras are deployed as a ring, and have a similar function to an omni-directional camera. Similarly, the FLYCAM system [Foo00] used an array of inexpensive cameras, organized in a circular manner, to capture every direction. This approach also supported the motion detection technique (such as the movement of a body or hands) to identify active speakers. Nickel [Nik05] used two microphones to recognize the sound source, which is combined with complex image processing techniques to track speakers in a meeting. Ranjan *et al.* [Ran08] used infrared cameras, combined with passive reflective markers, to identify all speakers in a meeting. Though this approach can efficiently recognize speakers, it requires sticking markers on speakers in advance. Later, Ranjan *et al.* [Ran10] used a high resolution camera to recognize speakers' faces and their positions. In addition, they have placed a microphone in the front of each speaker to identify the sound source. Rinzhin *et al.* [And10] used multiple omni-directional and PTZ cameras on the wall and ceiling to detect all speakers using the face detection technique. Based on Radial Basis Function Networks, Howell and Buxton [How02] recognized speakers' gestures, and accordingly adjusted the attention of a camera.

Identifying speakers based on image processing and face detection techniques can be erroneous, as mentioned by [Ran10]. Rinzhin *et al.* [And10] also reported that the level of illumination can affect the face detection accuracy. In addition, Yu *et al.* [Yu10] reviewed various smart meeting recording systems, and identified an inefficient face recognition algorithm as one

major limitation. Diverging from previous work, we have used a Kinect sensor to track speakers. The Kinect sensor provides a 3D infrared image. Compared with 2D color images, the depth information in 3D images makes it easy and robust to track speakers and their gestures. Our approach detects the body and produces a skeleton for each speaker. The skeleton includes two points of a person's shoulder, which imply the body pose. In addition, the Kinect sensor includes a microphone array, which is able to track multiple sound sources simultaneously. The combination of audio and vision tracking enables us to identify a dominant speaker in a meeting.

4.2.2. Camera management techniques

Recording a meeting involving several speakers requires multiple camera shots from various angles. Some systems [Rui01, Cut12, Foo00, Lee02] used an omni-directional camera to capture a 360-degree view. A variation of an omni-directional camera is to organize an array of inexpensive cameras as a ring. Those approaches can provide a close-up shot of any speaker sitting around the camera. Other approaches [Ran10, Liu02] used pan-tilt-zoom (PTZ) cameras to provide multiple shots.

Previous approaches are limited by the variety of shots they can provide. Specifically, they cannot adjust the angle of a camera based on the body pose of a speaker in a way that the camera can directly face the eye of a person. Jones *et al.* [Jon09] discussed the importance of eye contact when capturing videos. However, it is challenging to recognize the pose based on the image processing techniques [Ran10]. In order to improve the eye contact in the video recording, our approach combines a 3D infrared camera, which is embedded in a Kinect sensor, with a movable camera. The 3D camera is used to derive the body pose of a person. According to the body pose, the movable camera moves to an appropriate position so that the camera directly faces the front

part of a speaker's body. Compared with omni-directional cameras or PTZ cameras which have a fixed location, a movable camera is flexible enough to adjust a camera's location and angle to fit the position and the pose of a speaker.

4.2.3. Video directing techniques

Some approaches [Kun90, Nag05] use post-processing techniques to improve the quality of an automatically recorded video. However, some scenarios need real-time recording. An experienced director, who carefully selects an appropriate shot at any moment and changes the shot based on the discussion among meeting attendees, is critical to produce an engaging and attractive video in real-time. Researchers have summarized a list of heuristic rules [Liu01, Ran0, Ran10] to simulate an experienced director. Based on previous experiences and consulting with professional directors, we have developed a set of heuristic rules that specifically fit our hardware design. Those rules synchronize various hardware components (e.g., the movable camera and the Kinect sensor) and choose appropriate shots based on the discussion among meeting attendees.

One of the important information sources in a meeting is media files (i.e., PPT, movies or pictures). Gestures have been used to support the camera management in previous approaches (such as grabbing a camera's attention [How02] or changing a camera's shot [Ran10]). Our approach supports speakers to fully control media files through voice commands and gestures.

4.3. Approach Overview

There are two major challenges in SmartCamera. First, it is challenging to sense and track the behavior of speakers under different situations, with the occurrence of various noises. Second, it is critical to intelligently guide a video production process, which involves various actions, such as selecting an appropriate video source and controlling a movable camera.

SmartCamera takes the advanced 3D sensing technology through Kinect for efficient meeting participant recognition and tracking. A movable camera, controlled by a customizable virtual director, provides the flexibility to shoot speakers at various angles and positions. In summary, SmartCamera has the following five design goals.

4.3.1. A natural user interface

The SmartCamera system not only automatically controls a movable camera to take the best shot in a meeting, but also supports meeting attendees to naturally interact with multimedia content (such as PowerPoint). In SmartCamera, a speaker controls multimedia content through predefined gesture commands without interrupting the meeting. Such a natural user interface makes an interface invisible and allows users to focus on their tasks.

4.3.2. Robustness

One important feature in SmartCamera is being robust. Tracking a speaker can be affected by many environmental factors, e.g., noises, echoes, light conditions, part of a body being behind an object, and body movements. SmartCamera can mitigate different types of noises. Especially, the system can still correctly recognize the dominant speaker even with the occurrence of ambient noises and sound echoes. Our approach uses the Kinect microphone array to calculate the sound source. When there are multiple sound sources (e.g., one is coming from the dominant speaker and the other is the ambient sound, such as air conditioning), SmartCamera first uses the Kinect SDK to eliminate the noises and sound echoes, and then uses some heuristics (refer to Section 4.5.1) to identify the dominant speaker.

It is important to recognize all meeting attendees and track their movements and gestures in a video production scene. The speaker detection and tracking should not be affected by ambient

lights (e.g., too dark or too light). The 3D infrared camera in Kinect is applicable in various conditions, and the depth information in 3D images reduces the computational complexity, compared with face recognition based on 2D images.

4.3.3. Low cost

Though different smart camera systems have been proposed [Foo00, Liu02, Ran10, And10], they include an expensive panoramic camera or multiple cameras to cover various angles. The hardware components in SmartCamera essentially include one Kinect Sensor (i.e., sensing various events and providing an overview shot), one PTZ camera (i.e., providing a close-up shot of the dominant speaker), a motorized rail (i.e., moves a camera to an appropriate position), and a laptop (i.e., running a virtual director that controls a movable camera and multimedia content).

4.3.4. Easy to setup and operate

One objective of the SmartCamera system is to simplify the video production procedure. We can easily set up and operate SmartCamera in various environments, such as a conference room. All hardware components (e.g., Kinect sensor, camera, laptop, and rail) are portable and can be flexibly connected through WiFi. The movable camera replaces an automated movable arm, which is expensive and has been commonly used in a TV production studio, and assures the best angle of framing a speaker based on his/her pose.

SmartCamera does not need any calibration. We simply place the Kinect sensor in the front to cover all speakers. After the set-up, a virtual director automatically controls the movable camera without the need of any human operators. The easy setup and operation improves the applicability.

4.3.5. Customizable virtual director

Automatically directing a video requires intelligent decision-making. In practice, a human director provides various shots based on different events that happen in the scene. In order to simulate a human director in SmartCamera, we investigated previous experiences [Liu01, Ran10] and consulted with several professional directors to summarize a set of heuristic rules. In responding to a specific event, a heuristic rule is triggered to control the movable camera. Each heuristic rule is specified through a workflow, which can be visually modified and customized through a graphical user interface. The customization of the virtual director makes SmartCamera applicable in different scenarios.

4.4. System Architecture

In practice, a professional TV director first places cameras in appropriate locations to cover the whole scene. Then, the director chooses one camera as the output video feed, and cuts between different cameras upon certain events, such as the change of dominant speakers, gestures of a speaker, verbal requests, etc. During the video production process, common video production rules must be observed to make an engaging video, such as the maximum time to show a person, the maximum time of an overview shot, etc. In summary, a professional director observes various events, and uses his/her video production knowledge to control cameras based on observed events.

SmartCamera simulates a professional director through four essential hardware components: a Kinect sensor, a PTZ camera, a motorized rail, and a laptop. The Kinect sensor has three important functions: tracking speakers through an infrared camera, identifying the dominant speaker through a microphone array, and providing an overview shot. A PTZ camera that mounts on top of a motorized rail realizes a movable camera, which takes a close-up shot for an active

speaker. The motorized rail is implemented through an *EasyDriver Stepper Motor Driver* and a stepper motor. The motor driver receives commands from a virtual director and accordingly controls the stepper motor to move a camera on the rail. Finally, SmartCamera requires a laptop that connects the above hardware component together. The laptop collects environment sensing information from the Kinect sensor and then uses a virtual director to control the movable camera.

Base on the hardware platform, the SmartCamera software consists of four main components as shown in Figure 4-1, which are: Environment Sensing, Camera Manager, Media Manager, and Virtual Director.

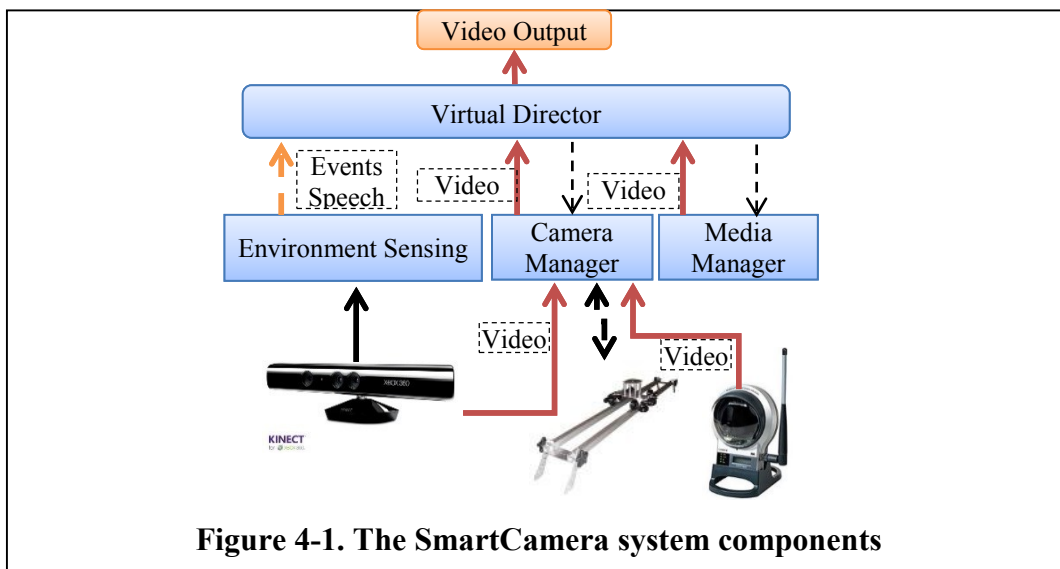


Figure 4-1. The SmartCamera system components

The Environment Sensing component is responsible to analyze and convert environmental sensory data (such as 3D images or sound) to video production events (such as gestures or a switch of active speakers). The Camera Manager component controls the movable camera to a designated location. The Media Manager component accesses to multimedia content, such as pictures and slides. The Virtual Director component directly interacts with the above three components. In response to events received from the Environment Sensing component, the virtual director either

commands the camera manager to take a shot at an appropriate position and angle or switches the video source to a multimedia document through the media manager.

4.5. Environment Sensing

The Microsoft Kinect sensor captures a wide range of sensory information, which includes RGB videos, 3D infrared images and sounds. The sensory data are collected by the environment sensing component, which converts those data to important video production events. We classify those events into three categories: 1) user recognition events that are implemented by the user tracking module; 2) speech commands that are recognized by the speech analyzer; and 3) gesture commands that are identified by the gesture analyzer.

4.5.1. User tracking

In practice, a professional director continuously tracks the person who is speaking in a meeting, and then determines how to control the camera(s). The user tracking module implements the function of user tracking. This module first identifies all meeting attendees and then recognizes a dominant sound source. Finally, we identify the dominant speaker by mapping the dominant sound source to a person.

Distinct from previous approaches [Nik05, Ran10] that use the face recognition to identify meeting attendees through 2D images, this paper uses 3D infrared images for a robust body and face recognition, which is supported by the Microsoft Kinect SDK. Our approach works even when some parts of the body aren't visible, such as legs under a table. In addition, the depth information is useful to efficiently detect the pose of a speaker, which enables the virtual director to move the camera in a way that matches the speaker's pose.

In the user tracking module, the sound analyzer is an essential function that identifies the dominant sound and its position. More specifically, the sound analyzer continuously tracks sound beams and their angles. Each detected sound beam has a dynamic power value, which implies the strength of the detected sound. A higher power value means a more reliable and continuous sound. The power value of a newly detected sound is set as zero in the beginning. As the sound continues being detected, its power value increments gradually every 100 milliseconds, until it reaches a predefined maximum value. Similarly, when an existing sound is not detected any more, its power value gradually decrements every 100 milliseconds and finally reaches zero. Among all detected sound beams, their values are compared every 100 milliseconds and the sound with the largest power value is considered as the dominant sound while other sound beams are treated as noises. After detecting the dominant sound, the analyzer will match its angle to a specific speaker, as shown in Figure 4-2.



The introduction of a power value avoids falsely recognizing a sound burst as a new speaker. Since the duration of a sound burst is very short, its power value remains at a low level. Furthermore, the power value is useful to continuously track a speaker. During a meeting, a speaker

may pause for a while. During the pause, though the power value of the dominant speaker gradually decrements, it still remains at a relatively high level that avoids losing the dominant speaker by recognizing a noise (e.g., a sound burst) as the new dominant sound.

Since the power value of each sound beam is updated and compared every 100 milliseconds, the odd that two sources have the same value is very small. In the case that two sources have the same power, we randomly choose one as the dominant speaker.

We have evaluated the time taken to recognize a new speaker. In the test, we switch speakers for 20 times and measure the time spent on identifying the new speaker. The result shows that SmartCamera takes on average 1.06 seconds (standard deviation = 0.19) to report a new speaker.

In summary, the integration of 3D infrared images and a Microphone Array provides several benefits. First, it avoids the placement of multiple microphones around the scene, which simplifies the set up process. Second, the 3D infrared images provide a robust user tracking. Finally, the depth information in 3D images can efficiently detect the pose of a speaker so that a camera can be controlled to directly face the speaker all the time.

4.5.2. Speech analyzer

Speech is a natural communication means in our daily life, and has an important role in the video production. It is especially useful when a speaker asks the director to switch the video source to a media file, such as a slide or a video. Besides, a speaker may require a specific shot in some scenarios. The Speech Analyzer component allows a speaker to control a camera or a media file through voice commands.

SmartCamera supports seven voice commands. In order to issue a voice command, a speaker must first speak the keyword “CAMERA” to activate the command recognition process, and then speak one of the seven commands. The two-stage command recognition process reduces the probability of recognizing a false command from a conversation in a meeting. The seven commands are as the following: 1. Overview shot 2. Close-up shot 3. Show this 4. Show movie 5. Show picture 6. Show next 7. Show previous

The first three commands are used to control a camera shot. The “show this” command is used to shoot the object in the dominant speaker’s hand. These commands allow a speaker to select a specific shot when needed. The fourth and fifth commands switch the video output to a media file. A speaker uses the last two commands to navigate multiple pages in a media file, such as a PowerPoint.

4.5.3. Gesture analyzer

Gestures enable people to exchange information without relying on sound. It is especially useful in a video production process since it eliminates the sound distraction in a meeting. In order to differentiate gesture commands from normal hand movements in a meeting, the gesture analyzer is triggered only after a multimedia document is opened through a voice command. We have defined four gestures to control presentation media:

1. Slide a hand to the left. Navigate to the next page in a multimedia document.
2. Slide a hand to right. Navigate to the previous page.
3. Slide a hand top-down. Stop playing the media.
4. Slide a hand bottom-up. Start playing media.

A recognized gesture is sent to the virtual director to trigger the corresponding command.

4.6. Video Sources

The output of a video production can be either images captured from a camera or a multimedia file. This section discusses the camera manager and the media manager.

4.6.1. Camera manager

In SmartCamera, it is critical to intelligently manage various cameras to take the best shot at any time. SmartCamera includes one fixed camera and /one movable camera. The fixed camera, which is integrated within a Kinect sensor, provides an overview shot, while the movable camera captures close-up shots of speakers from different angles.

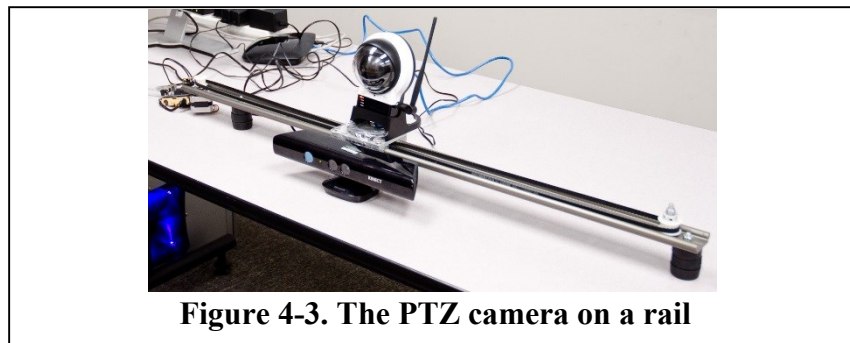
4.6.1.1. Fixed camera

A fixed camera provides an overview shot of the whole scene. Instead of having a separate fixed camera, the RGB camera embedded in the Kinect sensor is used as an overview camera. Therefore, the Kinect sensor should be placed in the front of all meeting attendees to cover the whole scene.

4.6.1.2. Movable camera

PTZ cameras have been commonly used in smart camera management systems [Liu02, Ran10]. In the previous approaches, a PTZ camera in general has a fixed position. Therefore, it is necessary to include multiple PTZ cameras, and each camera covers a portion of the whole scene. In addition to an increasing cost, the calibration of multiple cameras complicates the hardware setup. Furthermore, a camera with a fixed position cannot adjust its angle based on the position and pose of a speaker. Instead, a movable camera, as presented in Figure 4-3, provides the flexibility to exercise a wide range of possible views. Especially, the capability of changing the

position of the camera assures that a speaker directly looks at the camera, which makes the video more engaging. In summary, the advantage of using a track-based camera versus a series of PTZ cameras is the ability to place the camera on any position along the track. This gives the system greater potential to create a well-composed shot, especially if the subject moves at all during the video shoot. There is also the potential to capture video while the camera moving on track, which could add significant aesthetic interest.



In an initial design, we constructed a movable camera by mounting a regular camera on top of an iRobot. However, we found several issues with the original design. First, an iRobot makes loud noises when it is moving; second, the iRobot is often moving off a straight-line track, which requires an additional camera to monitor and adjust the movement of the iRobot; third, the movement of the iRobot is slow. Therefore, we designed a motorized rail system, which provides a rapid and accurate movement. We have tested the camera moving from one end to the other end for 20 times, and the average time is 4.61 seconds with the stand deviation 0.34. The rail system does not require any special calibrations.

The movable camera system is powered by a stepper motor through a belt and two pulleys attached to the rail. The camera itself is fixed to a small cart fit to the rail with a low friction material. To control the motor, the camera manager communicates with an Arduino

microcontroller through a USB port, and through WiFi to exchange information with the PTZ camera. The movable camera has two states, i.e., *Ready* and *On-Move*. By default, the movable camera is in the state of *Ready*. When the movable camera receives a moving command, its state is changed to *On-Move*. When the camera arrives at the designated position, the state is changed back to *Ready*.

Based on the position and the pose of a speaker, it is critical to calculate the position of the PTZ camera on the rail and the panning/tilting angle so that the camera straightly points at the speaker. The position of the PTZ is specified as the distance relative to the center of the rail, i.e., d_m in Figure 4-4. In order to calculate d_m , we need to first determine the panning angle a_1 (refer to Figure 4-4). Ideally, the Kinect sensor can calculate the panning angle based on the positions of face. In the case that a person's face is partially visible to the Kinect, we use the shoulder angle to indicate the pose of the speaker. Equation 4-1 shows the calculation of the panning angle a_1 based on the positions of the right shoulder (SR_x, SR_y, SR_z) and the left shoulder (SL_x, SL_y, SL_z). Using the panning angle a_1 , d_m is calculated according to Equation 4-2, in which (d_{px}, d_{py}, d_{pz}) is the center of the head of the dominant speaker detected by the Kinect sensor (refer to Section 4.5.1). In the case that a speaker turns his head/body extremely to one side, the calculated d_m may be larger than the half size of the rail, i.e., d_{max} , then we move the camera to the end of the rail (i.e., d_{max}) and calculate camera panning angle a_1' based on Equation 4-3. The tilting angle a_2 is calculated according to Equation 4-4, where d_{py} is the height of the speaker's head center relative to the Kinect sensor. The calculation must consider the height of the PTZ camera relative to the Kinect sensor, i.e., *CameraHeight* in Equation 4-4, which is 15 cm in our implementation. Tilting the camera according to the head position of the speaker assures to frame the head in the center.

$$a_1 = \frac{\pi}{2} - \tan^{-1} \left(\frac{SL_z - SR_z}{SL_x - SR_x} \right) \quad (\text{Equation 4-1})$$

$$d_m = \frac{d_{pz}}{\tan(a_1)} - d_{px} \quad (\text{Equation 4-2})$$

$$a_1' = \tan^{-1} \left(\frac{d_{pz}}{d_{px} + d_{max}} \right) \quad (\text{Equation 4-3})$$

$$a_2 = \tan^{-1} \left(\frac{d_{py} - \text{CameraHeight}}{d_{pz}} \right) \quad (\text{Equation 4-4})$$

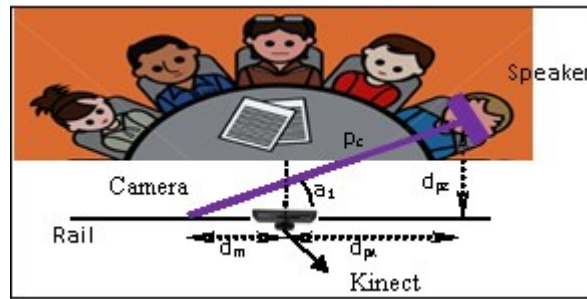


Figure 4-4. Adjusting a camera based on the speaker

4.6.2. Media manager

During a meeting, it is useful to show some pictures or videos related to the talk at certain points. The presentation media can increase the engagement of audiences and make the topic more understandable and interesting. The Media Manager Component controls and produces video outputs based on media files in a predefined media folder. Meeting attendees can activate the media output by using speech commands, and then control the media file by speeches or gestures (e.g., sliding a hand). Gesture commands enable a speaker to control the presentation media without stopping talking.

4.7. Virtual Director

It is complicated to direct a scene and produce a video. The director should carefully follow the subjects, and identify the dominant speaker at any time. Besides, he/she should continuously

listen to the requests (e.g., voice or gesture commands) from speakers. Based on all signals and information received from various sources in a meeting, the director decides a specific action (e.g., moving cameras to an appropriate location, selecting the output feed, and changing the shot) to make the video engaging.

The virtual director component simulates a professional human director. It receives environmental events (such as speaker change) and decides corresponding actions based on a set of heuristic rules. Action commands are sent to various components (e.g., the movable camera or the media manager) to actually perform those actions. Some actions may be performed immediately (such as voice commands) while some may have a delayed effect (e.g., it takes a while to move a movable camera to a designated position). According to previous successful work [Bas94, Ino95, Liu01, Ran08] and by consulting with professionals, we have summarized a set of heuristic rules, which are implemented through a workflow engine.

The virtual director workflow visually represents heuristic rules through a graphical user interface, which makes it easy to modify those rules. The workflow is embedded in the system through an open interface. In other words, we can update a heuristic rule through the workflow without changing the source code. This versatile design allows the user to adapt the virtual director to their specific needs. For example, a user may want to switch cameras between speakers rapidly in a fast-paced debate, or a user may want to remain on a single camera angle longer in a slow-paced documentary style presentation. The mapping between an action and a gesture/speech command can also be customized to fit a specific scenario. In addition, the separation of the virtual director and other components supports reusing those heuristic rules. For example, if our system

is extended with multiple Kinect sensors or cameras, a portion of the virtual director can re-used in the new system.

The virtual director workflow, as presented in Figure 4-5, reacts to events that come from various hardware/software comments (such as the Kinect sensor or a timer) and accordingly controls the video output. The heuristic rules in the virtual director workflow are grouped into three categories, which are discussed in detail in the following.

4.7.1. Speaker shot

When the user tracking component (Refer to Section 4.5.1) recognizes a new dominant sound source and accordingly identifies the pose of the body of the new dominant speaker, a *Speaker Detected* event is triggered. At this point, if the following conditions are all satisfied, the workflow first changes the video output to an overview shot, and then repositions the camera based on the detected speaker for a close-up shot.

1. The Output is not set to Media Shot (Slide show).
2. At least 7 seconds have passed since the last switch between cameras. This condition prevents an unexpectedly timed switch between different shots, which could be disorienting to the viewer and offset the visual pace of the video. If the user desires, he/she may shorten or lengthen this delay to match their specific scenario.
3. If the current speaker is the same as the newly detected speaker, the pose of this speaker must have been changed by more than 15 cm or 20 degrees.
4. Camera is not moving. The dominant speaker may change, while the camera is moving to a designated position $p1$. In this case, we do not move the camera to a new position until it reaches $p1$ first.

5. The total number of speaker transitions within the last minute is less than 5 times. This condition prevents a camera from changing speakers with a close-up shot very frequently during a heated discussion. Once the limitation is reached, SmartCamera will provide an overview shot.

4.7.2. Command

Speech and gesture commands are triggered through *Speech Detected* and *Gesture Detected* events. For example, a speaker may say *Show Picture*. In this case, the virtual director will set the output as a slide show. Then, the speaker can go through the slides by gestures or speech commands. All commands, except the *close-up shot* command, are performed immediately once they are recognized by the speech analyzer or the gesture analyzer. The close-up shot command switches a camera from a presentation media file to a speaker. The virtual director first displays an overview shot, until the camera moves to a designated location, and then changes to a close-up shot.

4.7.3. Timing

In order to make the video more engaging, the director needs to change the shots over time. Those timing heuristic rules, defined as the following, specify the cutting between different shots.

4.7.3.1. Close-up shot timer

When the virtual director starts a close-up shot, this close-up timer is activated (i.e., *30 Seconds No Event* in Figure 4-5). This timer will be expired after 30 seconds, and the expiration switches a close-up shot to an overview shot, which makes the video more engaging by avoiding a continuous fixed view.

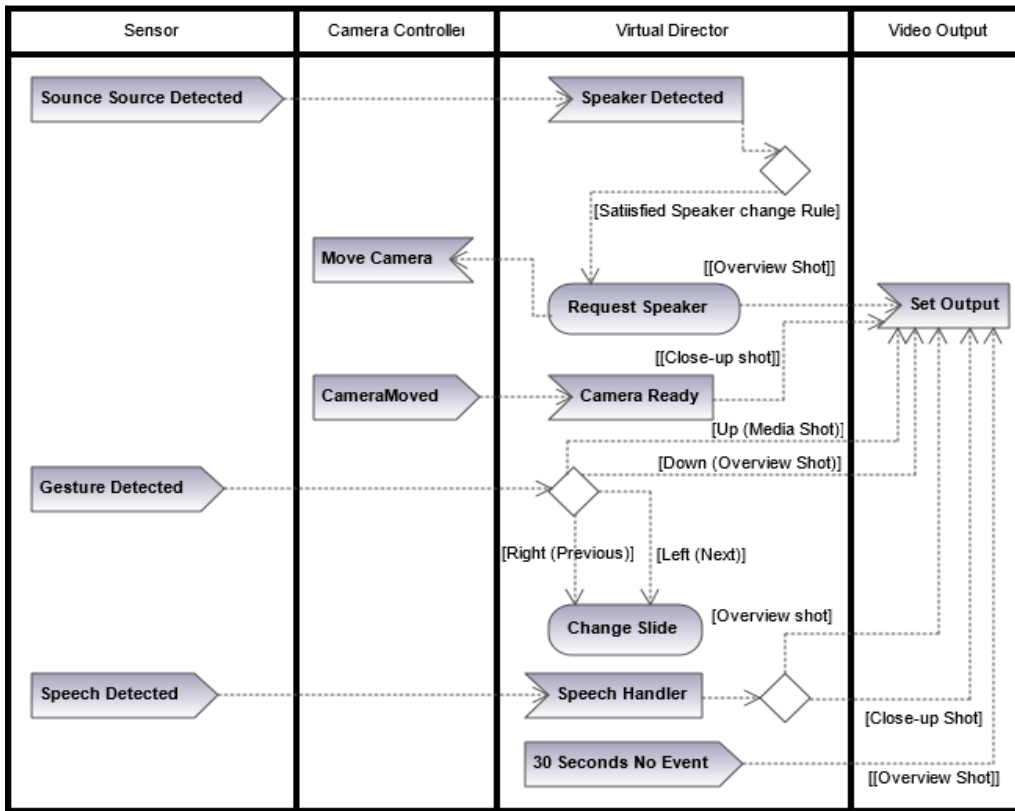


Figure 4-5. The virtual director activity diagram

4.7.3.1.1. Change shot timer

A change of shots is disallowed, if the last change happened in less than 7 seconds. This timer prevents a fast switch of different shots, which may cause unpleasant distraction.

4.7.3.1.2. Frequent speaker transition timer

By allowing at most five switches of close-up shots, this timer prevents a frequent switch among speakers.

4.8. An Empirical Study

The major goal of this study is to evaluate the video production quality and engagement of the SmartCamera system. In this study, we have invited three people from an *Improv Comedy Club* at a midwest university to discuss their club. The SmartCamera system automatically films the

discussion. At the same time, a professional camera-man is recruited to record the discussion. The SmartCamera system produces an output video in real-time, while the professional needs a post-processing step to mix and select shots to produce the final output.

We apply a between-group evaluation to compare the quality of the video produced by SmartCamera with that produced by the professional. More specifically, participants are randomly divided into two groups. The first group (referred as **G1** in the following) watches the video produced by SmartCamera, and the second group (referred as **G2** in the following) watches the other video. Each group has 27 subjects. In order to avoid biased opinions, participants are not informed on how a video is produced. After watching the video, each participant was asked to complete a questionnaire that uses a 5-point Likert-type scale (i.e., ranging from “1- Strongly disagree” to “5- Strongly agree”). The details of the study are provided in the following.

4.8.1. Research question and hypotheses

Based on the Goal Question Metric (GQM) approach [Bas94], we define the following goals and hypotheses.

Goal 1: *Compare* the engagement and the overall quality of the video produced by SmartCamera with the video by a human professional.

Hypothesis 1: Participants rated the SmartCamera video close to the human directed video in terms of engagement and overall production quality.

Goal 2: *Compare* SmartCamera and human directed videos in terms of overview and close-up shots.

Hypothesis 2: Participants rated the SmartCamera video close to the human directed video in terms of overview and close-up shots.

4.8.2. Participating subjects

Fifty four undergraduate students enrolled in the Computer Science and Information Science programs at a middle-west university participated in this study. The subjects were randomly selected and are not specifically targeted to benefit the study results.

4.8.3. Artifact/videos

Two videos was produced; i.e., one by SmartCamera and the other by a professional human director. Both videos have the same resolution, frame rate, length and contents. The video mainly consists of two sections. In the first section, speakers introduce their club through PowerPoint slides, which are controlled by speech and gesture commands. SmartCamera automatically provides the PowerPoint slides as the designated output video. In the human-directed video, the professional needs to manually choose the slides as the output in the post-processing step. In the second section, speakers give some detailed examples about the work they had performed and their experiences. This section involves speaker recognition and shot transitions among speakers. In summary, this video covers all important features in SmartCamera, such as speech and gesture commands, speaker recognition, transition between overview and close-up shots and etc.

4.9. Data Analysis

This section provides an analysis of the data collected during the study. The results are organized based on the two research goals and related hypotheses. An alpha value of 0.05 was selected for judging the significance of the results.

4.9.1. H1: Comparison of subjective feedbacks on engagement and video production

quality

Participants are asked to evaluate the engagement and overall video production quality based on the five questions listed in Table 4-1, in which G1 represents the first group that watched the SmartCamera video while G2 indicates the second group that watched the human-directed video. Since, we have two independent unpaired group, we have performed a two-tail Mann-Whitney's U test. Shaded cells in Table 4-1 indicate a significant difference (i.e., $p < 0.05$). We found that the rates about the engagement are close without significant difference between two videos. In other words, we have achieved the major design goal of making an automatic video interesting and engaging, which is one of most challenging issues in the automatic video production [Ran08, Ran10, Rub02, Yu10]. The human-directed video has a higher mean ranks on the overall quality (i.e., the second question in Table 4-1) than the SmartCamera video. However, no significant difference has been observed ($p=0.067$). Similarly, both videos have close rates on the third question on number of times that we have switched the shot.

Table 4-1. Overall production quality result

	Median		STD		Mean Ranks		P
	G1	G2	G1	G2	G1	G2	
1. You find the video interesting or engaging.	3	3	0.92	1.06	28.02	26.98	0.82
2. Rate the quality of the video production as a whole.	3	3	0.68	0.89	23.55	31.44	0.067
3. The camera views switched just the right amount of times.	3	3	1.13	0.94	25.3	29.7	0.31
4. Overall, the selected shots and timing of camera were switched appropriate.	3	3	0.85	0.79	20.15	34.85	0.0006
5. The director framed the speaker well.	2	4	1.05	0.9	19.26	35.74	0.0001

The human-directed video has a higher rate on both the camera switch time (i.e., question 4) and framing (question 5). This result meets our expectation since a human director can

understand the content and the context of a discussion and accordingly control cameras faster and more precisely. In the evaluation, we observed that SmartCamera has some delays to move a camera to the designated position when speakers switch frequently and fast in a discussion. The delay is caused by the following three reasons. First, it takes a while to move a camera to a designated position. Second, the power value in the sound analyzer (refer to Section 4.5.1) avoids falsely recognizing a sound burst while it also increases the time to detect a new sound source. However, the delay caused by the sound analyzer is limited within 1.06 seconds (refer to Section 4.5.1.). Third, the 7 seconds rule in the virtual director (refer to Section 4.7) which avoids a disrupting and frequent switch between different shots, while it also may delay pointing the camera at a new speaker. More specifically, if the first speaker speaks for less than 7 seconds, the movable camera will not point at the second speaker until the 7 seconds timer is expired. In order to minimize the delay of shot switches, we can consider updating the hardware with a faster and more powerful motor. More important, the camera movement should be optimized. During the duration of an overview shot, we can move the camera from its current position to the center of the rail, which may potentially reduce the average moving distance. The 7 seconds rule can delay pointing a camera at a new speaker when the previous speaker spoke less than 7 seconds. The virtual director should be updated to balance the frequent switch of various shots and the delay of pointing the movable camera at a speaker with a close-up shot.

The quality of framing in SmartCamera is reduced due to the following two reasons. First, if a speaker turns his head/body to one side, the Kinect sensor cannot capture the speaker's face/body clearly, which causes inaccurate body recognition. Second, when a speaker is speaking, his/her head/body movement requires adjusting the position of the camera. However,

SmartCamera currently only considers large movements (more than 15 cm or 20 degrees). In summary, the close-up framing issue is caused by body recognition, especially when a speaker does not face the Kinect sensor well. One possible improvement is to employ multiple sensors in various positions to capture different angles of a speaker and accordingly construct a full 3D view [Wil12]. In addition, a longer rail for a PTZ camera could provide more space to move the camera to a position with a better angle.

In summary, the engagement and the overall quality of the SmartCamera video are close that of a human directed video. However, the quality of camera switch and framing in SmartCamera still needs to be improved.

4.9.2. H2: Comparison of subjective feedbacks on timing /length of overview and close-up shots

In addition to engagement and overall quality, we also performed a two-tail Mann-Whitney's U test on the quality of overview and close-up shots from the perspectives of timing and length, as presented in table 4-2. A significant difference is observed on questions 2 and 3 in Table 4-2. In SmartCamera, the length of an overview shot and the timing of switching to a close-up shot are co-related. More specifically, in order to switch to a close-up shot, SmartCamera first needs to move the camera to a designated position. While the camera is moving, SmartCamera continues providing an overview shot as the output. Consequently, the duration of an overview shot is lengthened, while the duration of a close-up shot is shortened. In order to address the above issue, it is critical to reduce the transition time from an overview shot to a close-up shot by moving the camera faster to its designated position. On the other hand, SmartCamera can immediately switch

from a close-up shot to an overview shot. Therefore, participants are satisfactory with the timing of switching to an overview shot (Question 1 in Table 4-2).

Table 4-2. Overview/close-up shots

	Median		STD		Mean Ranks		P
	G1	G2	G1	G2	G1	G2	
1. The director switched to the overview shot at the right time.	3	4	0.97	0.79	23.59	31.40	0.069
2. The length of the overview shot was appropriate.	3	4	1.12	0.83	21.59	33.40	0.006
3. The director switched to the close-up shot at the right time.	2	3	1.02	0.83	19.69	35.31	0.0002
4. The length of the close-up shot was appropriate.	3	4	0.98	0.89	23.31	31.68	0.052

4.10. Conclusion and Future Work

In this paper, we have presented a low-cost and intelligent camera management system, called *SmartCamera*, which has several advantages, such as being robust and easy to setup. The usage of a Kinect sensor integrates different functions in one single hardware device, which simplifies the hardware assembly. Different from the face recognition based on 2D images, our approach uses 3D images captured by a Kinect sensor to efficiently recognize the skeleton of each meeting attendee. We have designed a movable camera so that a close-up shot is captured based on the pose of a speaker. Besides, *SmartCamera* supports voice and gesture commands to control media files (i.e., pictures and movies) during a meeting. This feature allows speakers to naturally interact with multimedia documents without interrupting the discussion. Based on the hardware design, we summarized a set of video production rules that intelligently synchronize different hardware devices. An empirical user study justifies the usefulness and the video quality of the proposed prototype.

The SmartCamera system is not limited to recording meetings. For example, by changing the virtual director, we can adapt SmartCamera to record a lecture in a classroom. The flexibility of an open workflow-based system opens the possibility to easily extend the system. The future work includes applying and testing SmartCamera in different scenarios.

The number of speakers which can be covered in SmartCamera is limited by the Kinect sensor's field of view. If one Kinect sensor cannot cover all speakers in a meeting, we can divide the whole space into several non-overlapping regions, each of which is covered by one Kinect sensor. Therefore, SmartCamera is flexible to cover a small scene, while it is also scalable to support a large number of speakers. In the future, we will extend SmartCamera with several Kinect sensors.

4.11. References

- [And10] Andrey L. Ronzhin, Maria Prischepa, and Alexey Karpov. A video monitoring model with a distributed camera system for the smart space. *Proc. ruSMART/NEW2AN'10*, Springer-Verlag, (2010), 102-110.
- [Bas94] Basili, V.R., Caldiera, G., Rombach, H. D. , The Goal Question Metric Approach, *Technical Report, Department of Computer Science, University of Maryland*, (1994), <ftp://ftp.cs.umd.edu/pub/sel/papers/gqm.pdf>
- [Bia98] Bianchi, M. AutoAuditorium: A Fully Automatic, Multi-Camera System to Televisе Auditorium Presentation, In *Proc. Joint DARPA/NIST Smart Spaces Technology Workshop*, (1998)
- [Bra01] Brandstein, M. and Ward, D. Microphone Arrays: Signal Processing Techniques and Applications. *Springer Verlag*, (2001).

- [Cut12] Cutler, R., Rui, Y., Gupta, A., Cadiz, J., Tashev, I., He, I., Colburn, A., Zhang, Z., Liu, Z., and Silverberg, S. Distributed meetings: a meeting capture and broadcasting system. *Proc. MULTIMEDIA*, ACM (2012) 503-512.
- [Foo00] Foote, J. and Kimber, D. FlyCam: practical panoramic video. *Proc. MULTIMEDIA*. ACM, (2000) 487-488.
- [How02] Howell, A. J. and Buxton, H. Visually mediated interaction using learnt gestures and camera control. *HCI 2002*. Springer-Verlag. (2002) 272-284.
- [Ino95] Inoue, T., Okada, K. and Matsushita, Y. Learning from TV programs: application of TV presentation to a videoconferencing system. *Proc. UIST 1995*, ACM Press, (1995) 147-154.
- [Jon09] Jones, A., Lang, A., Fyffe, G., Yu, X., Busch, J., McDowall, I., Bolas, M., and Debevec, P. Achieving eye contact in a one-to-many 3D video teleconferencing system. *ACM Trans. Graph July* (2009). 28, 3, Article 64.
- [Kun90] Kuney, J. Take One: Television Directors on Directing. *Praeger Publishers*, (1990).
- [Lee02] Lee, D., Erol, B., Graham, J., Hull, J. J., and Murata, N., Portable meeting recorder. *Proc. MULTIMEDIA*, ACM (2002), 493-502.
- [Liu01] Liu, Q., Rui, Y., Gupta, A., and Cadiz, J. J. Automating camera management for lecture room environments. In *Proc. CHI 2001*. ACM (2001), 442-449.
- [Liu02] Liu, Q., Kimber, D., Foote, J., Wilcox, L., and Boreczky, J. FlySPEC: a multi-user video camera system with hybrid human and automatic control. *Proc. Multimedia 2002*. ACM (2002), 484-492.
- [Nag09] Nagai, T. Automated lecture recording system with AVCHD camcorder and microserver, *Proc. SIGUCCS* (2009), 47-54.

- [Nic05] Nickel, K., Gehrig, T., Stiefelhagen, R., and McDonough, R. A joint particle filter for audio-visual speaker tracking. *Proc. ICMI 2005*. ACM (2005), 61-68.
- [Pol97] Poltrock, S.E. and Engelbeck, G. Requirements for a virtual collocation environment. In *ACM GROUP (1997)*, 61-70.
- [Rab06] Ranjan, A., Birnholtz, J.P. and Balakrishnan, R., An exploratory analysis of partner action and camera control in a video-mediated collaborative task. *Proc. ACM CSCW*, (2006) 403-412.
- [Ran08] Ranjan, A., Birnholtz, J.P. and Balakrishnan, R., Improving meeting capture by applying television production principles with audio and motion detection, *Proc. CHI 2008*, ACM (2008) 227-236.
- [Ran10] Ranjan, A., Henrikson, R., Birnholtz, J., Balakrishnan, R., and Lee, D., Automatic camera control using unobtrusive vision and audio tracking. *Proc. Graphics Interface 2010*. ACM (2010) ,47-54.
- [Rub02] Rubin, A.M., The uses-and-gratifications perspective of media effects. *Media Effects: Advances in theory and persuasion (2002)*, 525-548.
- [Rui01] Rui, Y., Gupta, A., and Cadiz, J. J. Viewing Meeting Captured by an Omni-Directional Camera, *Proc. CHI 2001*, ACM (2001), 450-457.
- [Rui03] Rui, Y., Gupta, A. and Grudin, J. Videography for telepresentations. *Proc. CHI 2003*, ACM, (2003) 457-464.
- [Yu10] Yu, Z. and Nakamura, Y. Smart Meeting Systems: A survey of state-of-the-art and open issues, *ACM Computing Surveys*, (2010), *Vol. 42, No. 2*, Article 8.

[Wil12] Williamson, B., LaViola, J., Roberts, T., and Garrity, P. "Multi-Kinect Tracking for Dismounted Soldier Training, *Proc. Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*, (2012) 1727-1735.

APPENDIX. IRB APPROVAL

Institutional Review Board

...for the protection of human participants in research

North Dakota State University
Sponsored Programs Administration
1735 NDSU Research Park Drive
NDSU Dept #4000
PO Box 6050
Fargo, ND 58108-6050 231-8995(ph) 231-8098(fax)



Protocol Amendment Request Form

Changes to approved research may not be initiated without prior IRB review and approval, except where necessary to eliminate apparent immediate hazards to participants. Reference: SOP 7.5 Protocol Amendments.

Examples of changes requiring IRB review include, but are not limited to changes in: investigators or research team members, purpose/scope of research, recruitment procedures, compensation scheme, participant population, research setting, interventions involving participants, data collection procedures, or surveys, measures or other data forms.

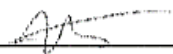
Protocol Information:

Protocol #: **sm12066** Title: **An automatic camera mangement system**

Review category: Exempt Expedited Full board

Principal investigator: **Jun Kong** Email address: **jun.kong@ndsu.edu**
Dept: **Computer Science**

Co-investigator: **Amin Roudaki** Email address: **amin.roudaki@ndsu.edu**
Dept: **Computer Science**

Principal investigator signature, Date:  8/17/2012

In lieu of a written signature, submission via the Principal Investigator's NDSU email constitutes an acceptable electronic signature.

Description of proposed changes:

1. Date of proposed implementation of change(s)*: **09/15/2012**
** Cannot be implemented prior to IRB approval unless the IRB Chair has determined that the change is necessary to eliminate apparent immediate hazards to participants.*
2. Describe proposed change(s), including justification:
 1. **Amin Roudaki has been added as a co-investigator** *... 8/21/12*
 2. **The questionair has been slightly changed.**