

**MOVISHARE: BUILDING LOCATION-AWARE MOBILE
SOCIAL NETWORKS FOR VIDEO SHARING**

by

Zonggao (Michael) Jia

B.Sc., Shanghai JiaoTong University, 1995

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF COMPUTING SCIENCE
in the School
of
Computing Science

© Zonggao (Michael) Jia 2009
SIMON FRASER UNIVERSITY
Summer 2009

All rights reserved. This work may not be
reproduced in whole or in part, by photocopy
or other means, without the permission of the author.

APPROVAL

Name: Zonggao (Michael) Jia
Degree: Master of Computing Science
Title of Project: MOVISHARE: BUILDING LOCATION-AWARE MOBILE
SOCIAL NETWORKS FOR VIDEO SHARING

Examining Committee: Dr. Steven Pearce
Chair

Dr. Jiangchuan Liu, Senior Supervisor

Dr. Ke Wang, Supervisor

Dr. Qianping Gu, SFU Examiner

Date Approved:

July 9th 2009



SIMON FRASER UNIVERSITY
LIBRARY

Declaration of Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the "Institutional Repository" link of the SFU Library website <www.lib.sfu.ca> at: <<http://ir.lib.sfu.ca/handle/1892/112>>) and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author's written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

While licensing SFU to permit the above uses, the author retains copyright in the thesis, project or extended essays, including the right to change the work for subsequent purposes, including editing and publishing the work in whole or in part, and licensing other parties, as the author may desire.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, BC, Canada

Abstract

Social networking and content sharing are the two major reasons for the explosion of web 2.0 applications in recent years. Although more people enjoy Internet and multimedia via their cell phones, most of the services for online video sharing are still restricted to traditional PC users. The existing systems of mobile video sharing do not fully utilize the mobility benefits or decently address the performance issues caused by the limitations of network and energy in mobile devices.

In this report, we propose MoViShare (Mobile Video Share), a universal video sharing platform that will provide anytime anywhere video browsing and publishing services for mobile devices. MoViShare targets to create and maintain location-aware mobile social networks, and to apply video abstraction technique for saving bandwidth and energy. We detail the MoViShare design in this report, and demonstrate a prototype design based on Nokia Symbian development kit and N96 smartphones.

Keywords

Mobile video-sharing; Video abstraction; Social network; Location-aware; Mobility

Subject Terms

Mobile communication systems; Wireless communication systems; Multimedia systems; Video-sharing systems; Social networks

Acknowledgments

I want to thank my Senior Supervisor, Dr. Jiangchuan Liu, for the guidance and encouragement throughout this project. I would not be able to complete this project without his stimulating suggestions and keen insights.

Thank my supervisor Dr. Ke Wang and my thesis examiner Dr. Qianpin GU, for reviewing this report and providing suggestions to improve the quality of the report. Also, thank Dr. Steven Pearce for taking the time to chair my project defence. I would like to extend my appreciation to the faculty and staff in the School of Computing Science of Simon Fraser University.

Thank my colleagues and friends at Simon Fraser University for their help. Especially thank my teammate Li Ma, for such a nice working experience of the whole project. Also, thank my colleague Jeff Jiang and Xu Cheng, for their priceless suggestions and professional technical supports on my project.

Thank my family for their selfless love and unlimited support. Thank my parents and in-laws for taking care of our living all the time and looking after our infant girl. Greatly thank my wife for giving me an adorable angel and making her always happy, active and healthy. Also thank my little daughter, Larisa, for all the fun and enjoyments she bring to us.

Finally I want to say that I love you all and wish all the best to you.

Contents

Approval	ii
Abstract	iii
Acknowledgments	iv
Contents	v
List of Tables	viii
List of Figures	ix
1 Introduction	1
2 Background and Related Works	4
2.1 Internet Video Sharing and Social Networking	4
2.2 Mobile Video Sharing	5
2.3 Mobile Social Networking	6
2.4 Multimedia Encoding/Decoding Standards	8
2.5 Video Abstraction Algorithms	10
3 Problem Statements	12
3.1 Mobility-Aware Social Network	12
3.2 Bandwidth and Energy Limitation	13
4 Proposed Solutions	14
4.1 Location-Aware Social Networking	14

4.2	Video Abstraction	15
5	Design Architecture	19
5.1	Project Proposal	19
5.2	User Scenarios	19
5.3	Modules	20
5.3.1	User Interface and Social Network Module	20
5.3.2	Location Management Module	20
5.3.3	Video Abstraction and Streaming Module	21
5.4	System Architecture	21
5.4.1	Design and Dataflow	21
5.4.2	Components	23
5.5	Application Interface	26
5.5.1	Client Side Main Interface	26
5.5.2	Uploading Video Interface	27
5.5.3	Video Browsing Interface	27
5.5.4	Location-Wise Video Display	27
5.5.5	Video Searching Interface	32
6	Implementation Details	34
6.1	Programming Modules and Tools	34
6.2	Database Structure	35
6.3	Challenges and Solutions	35
6.3.1	Display Videos More Intuitively	37
6.3.2	Extracting Key Frames	37
6.3.3	Decoding, Converting and Creating	38
6.4	Outstanding Issues	38
6.4.1	Display Multiple Videos at Same Position	38
6.4.2	Different Video Formats	39
6.4.3	Video Abstraction Delay	39
7	Conclusions and Future Works	41
7.1	Conclusions	41
7.2	Future Works	41

7.2.1	New Search and Social Features	42
7.2.2	Video Skimming Instead of Video Summary	43
7.2.3	Proxy Caching	44
7.2.4	Understanding Mobile Social Networks	45
	Bibliography	46

List of Tables

2.1	The top 10 video competitors (ranked by US-based visits in December 2007)	5
2.2	Top 10 mobile sites and top 10 PC sites	7
2.3	Four media formats in Nokia N96 mobile phone	8
6.1	The structure and sample tuples of 'Medias' table	36

List of Figures

1.1	The increasing number of videos added to YouTube	2
1.2	Worldwide mobile social network users, 2007-2012	3
2.1	Structure of a video clip in Nokia N96 mp4 format	9
4.1	An example of video abstraction	16
5.1	MoViShare video uploading data flow	22
5.2	MoViShare video retrieving data flow	22
5.3	MoViShare client-server structure	23
5.4	MoViShare client side main interface	28
5.5	MoViShare interface of uploading video	29
5.6	MoViShare video browsing interface	30
5.7	MoViShare location-wise video display interface	31
5.8	MoViShare interface of video search	33

Chapter 1

Introduction

Since its first appearance five years ago, Web2.0 [47] has become one of the favorite topics among people who are related to the Internet in any way. Among all the characteristics of Web2.0, social networking and content sharing, especially the sharing of video resources, are proved to be the most two important features. YouTube, as one of the most successful sites that bring them together, now features over 100 million videos and enjoys 20 million visitors each month. A recent research about YouTube produced Figure 1.1 that shows the increasing number of videos added to YouTube as time goes by [27]. We can see it has a 20% growth rate per month. YouTube's great achievement lies in the combination of the content-rich videos and, equally or even more importantly, the establishment of a social network. YouTube-like sites have created a video village on the web, where anyone can be a star, from lip-synching teenage girls to skateboarding dogs. With no doubt, they are changing the popular culture.

At the same time, the cyber-world also witnesses the tremendous fast growing of the mobile industry in recent years. By the end of 2008, there were 4.1 billion cell phone subscriptions globally, compared with about 1 billion in 2002 [33][29]. The Internet use by cell phone was more than doubled. An estimated 23% of mobile users on the planet used the Internet in 2008, up from 11% in 2002. Following the trend of Web2.0, in last two or three years, mobile-specific social networks like airG [1] and MocoSpace [13] have started attracting clients worldwide. The market leader, Nokia, has built its own mobile social network, Ovi [15], which has successfully pitched to European operators. These mobile social networking services emphasize the activities people tend to do on their phones, such as chatting, instant messaging, friends locating, and data sharing. Those moves create vibrant

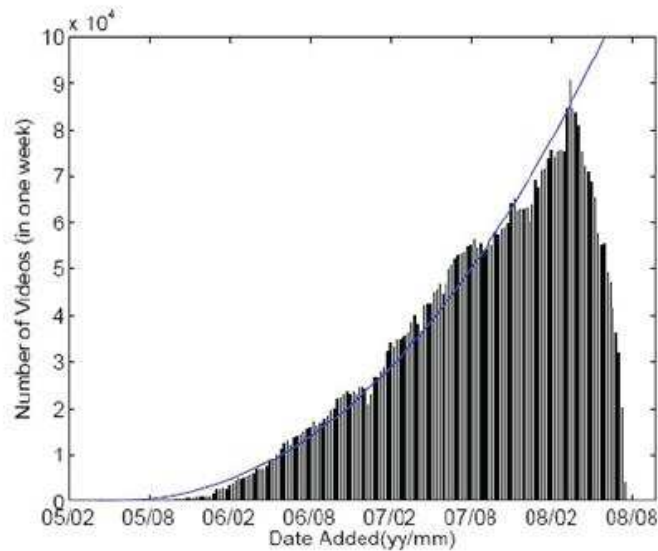


Figure 1.1: The increasing number of videos added to YouTube

groups for consumers and a new profit center for carriers. Figure 1.2 shows a forecasting from eMarketer [2] that over 800 million people worldwide will be participating in a social network via their mobile phones by 2012, up from 82 million in 2007.

It is interesting to point out that the clients of video sharing sites and that of advanced mobile devices are largely overlapping, with a majority of them belonging to the young generation. Unfortunately, while more and more mobile devices are multimedia-ready (e.g., Nokia N-series, iPhone, etc.), the available video resources for them are quite limited and often bounded to specific network service providers. According to a recent report from Veeker Ltd., only about 50% of U.S. mobile phone subscribers with built-in cameras actually snap photos, not to mention videos [26]. On the other hand, most of the video sharing services are still restricted to traditional Internet users, and only recently have we seen preliminary attempts that explore mobile video sharing, e.g., by YouTube and Veeker. Yet, for video sharing in this context, enormous issues still have to be addressed.

In this project, we design and develop MoViShare (Mobile Video Share), a universal video sharing platform that accommodates mobile accesses. MoViShare will realize anytime anywhere video browsing and publishing services for diverse multimedia-ready mobile devices, e.g., 2G/3G camera phones. It goes beyond to be a customized website for mobile

Worldwide Mobile Social Network Users, 2007-2012 (millions)						
	2007	2008	2009	2010	2011	2012
Mobile phone subscribers*	3,078	3,417	3,697	3,894	4,150	4,275
Mobile Internet users	406	490	596	757	982	1,228
Mobile social network users**	82	147	243	369	554	803
Mobile social network users % of mobile phone subscribers	2.7%	4.3%	6.6%	9.5%	13.3%	18.8%

*Note: *data for 2007-2010 from European Information Technology Observatory (EITO), March 2007; **registered users (identified by their mobile number) who create, edit and view personal content using their phone*
Source: eMarketer, April 2008

094335 www.eMarketer.com

Figure 1.2: Worldwide mobile social network users, 2007-2012

subscribers within a service provider only, but provides a ubiquitous interface that bridges mobile clients and existing Internet video sharing sites. As such, the abundant video resources available in these sites can be fully utilized, making the system instantly content-rich and up-to-date with a well-built social network. The incorporation of mobile clients also enriches these sites, given that the video contents can now be captured and published anywhere anytime. More importantly, MoViShare will create and maintain location-aware social networks among the clients. It will explore their mobility by effectively utilizing the location information and activating context-sensitive location-aware video browsing and sharing. In addition, to address the issue of bandwidth and energy limitation in mobile industry, every video clip will be abstracted to a small GIF (Graphics Interchange Format) file, which only contains a few key frames and can be displayed on normal web pages. The abstractions are usually representative while their lengths are only 1% to 5% of original video clips. In mobile environment, these unique features will greatly benefit clients when using MoViShare system.

The rest of the paper is organized as follows. Chapter 2 presents background and related work. Chapter 3 states the problems and challenges of such mobile video sharing system. The proposed solutions by MoViShare are described in Chapter 4, following by the design architecture (Chapter 5) and implementation details (Chapter 6). Finally, Chapter 7 concludes the paper and claims the potential future works.

Chapter 2

Background and Related Works

2.1 Internet Video Sharing and Social Networking

Internet services for video sharing have existed long before YouTube-like sites entered the scene. However in recent four or five years, YouTube and its competitors have been growing extremely fast along with the Web 2.0 [47] trend. According to a review from compete.com in 2008, the overall online video market which consists of over 50 sites and sub-domains serving video to US-based visitors, commands 426 million visits on 145 million unique visitors in December 2007 [28]. Table 2.1 represents the top 10 video competitors ranked by US-based visits in December 2007. YouTube, as the most successful case, dominated the field of video contenders with 52.3% market share. The most important factor toward their success is the establishment of social networks among massive video sources and their owners. The videos are no longer stand-alone units or categorized contents as they were in traditional streaming media servers or peer-to-peer downloading systems. Instead, they are hyper-linked to each other based on their owners' social relationships. This feature highly motivated individuals' participants. The statistical measurements against social network of YouTube site in [33] further proved the strong correlations between the videos and the users.

Realizing the importance of social networking in video sharing business, we are naturally thinking of the possibility of establishing social networks in mobile mode. Actually, due to the incredibly fast renovation pace in mobile service industry, more and more mobile devices are acting as a "moving PC". They keep getting better performance and richer functionalities nowadays. In other words, we can retrieve equally much, or even more information of social communication from one's mobile device than his/her personal computer, such as

December, 2007	People	Visits	Market Share
YouTube	58,549,750	237,648,150	52.3%
Yahoo	22,329,644	42,412,499	9.3%
AOL	17,940,686	38,148,895	8.4%
MSN	16,776,045	33,995,012	7.5%
MySpace	14,398,198	32,351,147	7.1%
Veoh.com	6,106,389	15,694,048	3.5%
ManiaTV.com	3,079,817	8,026,196	1.8%
Heavy.com	3,594,414	6,720,730	1.5%
Break.com	2,726,584	6,375,410	1.4%
Metacafe.com	3,365,198	5,104,333	1.1%

Table 2.1: The top 10 video competitors (ranked by US-based visits in December 2007)

address book, email list, etc. Moreover, most devices of recent model have been integrated with GPS (Global Positioning System) service [5]. Therefore, a more attractive feature, location-aware social networking, becomes available in mobile area.

2.2 Mobile Video Sharing

Mobile video sharing systems are not new either. As early as May 2006, YouTube started a service that allows users to upload homemade clips via their mobile phones or PDAs [25]. Later in same year YouTube Mobile site (<http://m.youtube.com>) has been launched to open its video repository to cell phone users. In the early stage there are only very small amount of video clips with specific format can be watched via cell phone. In recent two years, YouTube collaborated with giants of the industry such as Nokia, Verizon, Apple, Vodafone, and extended the mobile video service to 16 other countries outside USA and 10 other languages than English [24]. For example, during the corporation with Apple Inc., they offered a feature in Apple's products, iPod and iPhone, that allows user to browse YouTube website through a single click on one icon.

Does it mean that we have already had an ideal mobile video sharing system? Unfortunately there are still several fundamental issues to be addressed. The performance can be compromised due to the restricted hardware conditions of mobile devices. Comparing to PCs, mobile devices have weaker processors and smaller screen size. Most cell phones are

also suffered from power limitations when watching videos. Comparing to wired network, mobile phone network has averagely narrower bandwidth in turns of Internet data transferring. Therefore, simply launching a mirror of the original video sharing site and opening all sources to mobile users may not give satisfying services as expected. Mobile users may face difficulties to find a wanted video, or experience long buffering time and lagging performance when playing back. We believe that more efforts need to be made to overcome the above obstacles in mobile video sharing business.

2.3 Mobile Social Networking

Also, as the result of the tremendous fast growing of the mobile industry, mobile-specific social networks like airG, MocoSpace and Nokia's Ovi have started attracting clients worldwide in recent two years. airG's mobile community [1] has more than 20 million unique users worldwide and is interconnected to more than 100 mobile operators in over 40 countries. airG's mobile community solutions are proven to increase customer affinity and brand recognition for leading mobile operators and media companies globally. MocoSpace is a social network specifically designed for use on a mobile device [13]. The features of the site are similar to other social networking sites including mobile chatting, instant messaging, photo and video sharing, as well as forums. MocoSpace is the largest mobile social network in the U.S. with over six million registered users and over one billion monthly page-views. Nokia's Ovi is marketing as "personal dashboard" where users can share photos with friends, buy music and access third-party services [15]. It has some significance in that Nokia is moving deeper into the world of Internet services. So far, Ovi has successfully pitched to European operators.

To catch up the trend, traditional Internet social network providers poured money into mobile services as well. Facebook Mobile [3], Yahoo! Mobile [23] and MySpace Mobile [14] are the most successful cases in this category. Early this year, Google launched a new service called 'Latitude' which allowed mobile users to see where friends were and what they were up to [6]. Taking advantages of their existing gigantic social networks, various web2.0 services and rich contents, they got double results with half efforts. Table 2.2 shows the top 10 popular mobile sites and top 10 PC web sites recorded in February 2009 [19]. We can find those names in both lists. According to IT Facts [19], Facebook even received more attractions from mobile users than PC users. Mobile users accessing Facebook spend an

average of 24 minutes per day on the site, similar to the 27.5 minutes spent by PC users, while mobile users averaged 3.3 visits per day versus 2.3 visits by PC users.

Top Mobile sites	Top PC sites
Mobile Operator Sites	Google Sites
Google Sites	Microsoft Sites
Facebook.com	Yahoo! Sites
Yahoo! Sites	Facebook.com
BBC Sites	EBay
Apple Inc. Sites	BBC Sites
Microsoft Sites	AOL (inc. Bebo)
Sony Online (inc. Sony Ericsson)	Amazon Sites
Nokia	Ask Network
AOL (inc. Bebo)	Wikimedia Foundation

Table 2.2: Top 10 mobile sites and top 10 PC sites

Moreover, a bunch of new smart phone applications took off in recent years to fulfil the fast growing demands of social activities on mobile devices. Limbo (joined forces with Brightkite recently), Whrrl, Twitter, Loopt are the most popular social networking applications now [2]. Relatively, those services are designed for modern mobile devices with GPS, multimedia, and the ability to add third-party applications, such as iPhone (3G), Nokia N-series, etc.

These mobile social networking services emphasize the activities people tend to do on their phones, such as chatting, instant messaging, friends locating, and multimedia source sharing. Those features create groups for consumers, and a new profit center for carriers. eMarketer forecasts that over 800 million people worldwide will be participating in a social network via their mobile phones by 2012, up from 82 million in 2007 [2].

But on the other hand, these systems are more focused on mobile-based communications or simple user-based social networks. Most of them have very limited location-wise considerations when building the mobile social networks. For those who make great efforts on the utilities of positioning components of mobile devices, they are not designed specifically for video sharing. Our project, MoViShare, is targeting a seamless combination of location-based mobile social networking and mobile multimedia source sharing.

2.4 Multimedia Encoding/Decoding Standards

The video format we use is MPEG-4 which is a standard video format adopted by Nokia N-series cell phones and many other mobile devices such as PSP, iPhone, Nintendo DS, etc. The MP4 file format [22] [10] defines the storage of MPEG-4 audio, scenes and multimedia content using the ISO Base Media File Format. The device we used in our project, Nokia N96 cell phone, provides four different recording quality levels of MP4 video. They are "TV high", "TV normal", "Email high", and "Email normal", listed by the order of video quality, from high to low. Table 2.3 represents detail technical specs of each format. "TV high" has the best quality with 640 X 480 screen size, 29 fps frame rate, and averagely 4000kbps video bit-rate. On the other hand, "Email normal" format gives a much lower video quality with 320 X 240 resolution and only 15 fps frame rate, and the video bit-rate drops to around 400Kbps. Although "Email normal" setting yields the lowest performance in turn of video quality, it is still a preferred format in our project because of the following concerns.

- The video files generated in this setting is much smaller so that the uploading time will be much shorter;
- The screen size of mobile devices is normally less than 320 X 240, which means the video quality is acceptable under this resolution;
- The streaming performance is suffered from high bit-rate video sequences, thus lower bit-rate is preferred.

Format	Resolution	Frame Rate	Video Bit-rate	Audio Bit-rate
TV High	640 X 480	30 fps	4000 Kbps	96 Kbps
TV Normal	640 X 480	15 fps	1500 Kbps	96 Kbps
Email High	320 X 240	30 fps	768 Kbps	48 Kbps
Email Normal	320 X 240	15 fps	384 Kbps	48 Kbps

Table 2.3: Four media formats in Nokia N96 mobile phone

According to the international standard of coding of audio and visual objects, ISO/IEC 14496-12 (the part 12: ISO base media file format [9]), the data of information is stored in a nested box structure. Figure 2.1 shows an example of file structure of a video clip taken

by Nokia N96 cell phone. The Media Data Box ("mdat") holds the actual media data for the presentation, while the Movie Box ("moov") and its sub-boxes define the metadata for the video. In short video clips taken by Nokia cell phones, there are usually two Track Boxes ("trak") inside the Movie Box; one contains video metadata and the other has audio metadata. Other important boxes, that help the video abstracting operation in MoViShare, include Time-To-Sample-Box ("stts"), Sample-To-Chunk Box ("stsc") and Chunk-Offset Box ("stco"). Using such nested-box file structure, we manage to get key frames of a video clip and convert them into images. We will describe the details of video abstraction process in Section 4.2.

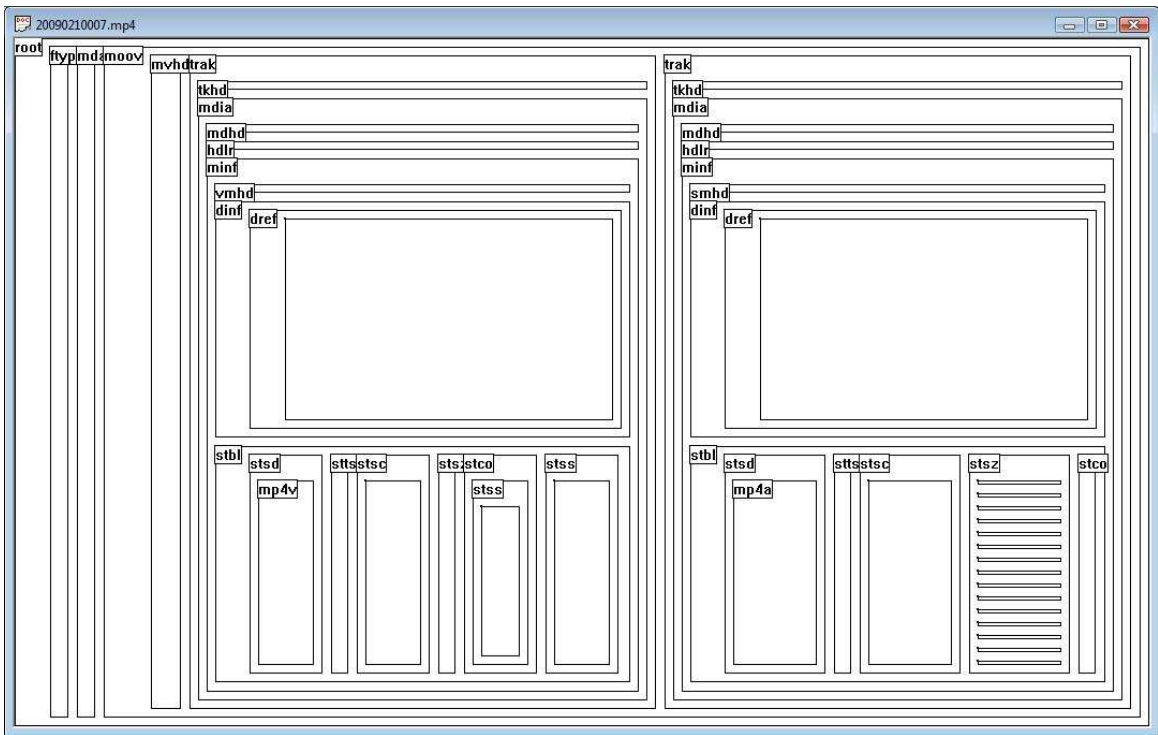


Figure 2.1: Structure of a video clip in Nokia N96 mp4 format

2.5 Video Abstraction Algorithms

Video abstraction is a technique to generate a short summary of the content of a longer video document. A video abstract is a sequence of still or moving images representing the content of a video in such a way that the target party is rapidly provided with concise information about the content while the essential message of the original video is well preserved [48]. Based on the way the salient images are extracted, the related techniques can be categorized into the following classes: sampling-based, shot-based, cluster-based, feature-based and others [42] [45].

Most of earlier work in video summarization chose to be sampling based, i.e., select key frames by randomly or uniformly sampling the video frames from the original sequence at certain time intervals. This is as well one of the most intuitive ways for video abstracting, however there is no guarantees for the qualities of the video abstracts generated by this method.

More sophisticated work tends to use the first frame of each shot as its key frames to be extracted and form the video abstraction. This solution is so called shot-based video summary. To define a shot is easy (i.e., a video segment within a continuous capture period), however, to locate the shots from a video remains to be a very difficult challenge. The algorithms in most of existing work tend to interpret the content by employing some low-level visual features such as color and motion.

The complexity of each shot is hardly reflected by one frame. Several cluster-based techniques have been proposed that select a proper number of key frames from a video. Authors of [56] use the color histogram of a frame as the feature and select the frame that is closest to the centroid of a cluster as a key frame; they also balance the cluster size by merging small size clusters into large ones. The solution proposed in [50] performs an iterative partitional-clustering procedure for key frame selection. Another algorithm [38] extracts multiple key-frames to represent a cluster based on the frame's similarity to the centroid of the cluster and a pre-defined similarity threshold.

The feature-based key frame selection considers that the frames in a video sequence are characterized by one or multiple features. Each frame can be represented by one visually perceptual feature, such as color, motion, edge, shape and spatial relationship. If the difference between the current frame and the most recent key frame exceeds a pre-defined threshold, the current frame is selected as a key frame. The efforts on this category include

color-based selection in [54], the motion-based selection in [52] and the object-based selection in [41]. A video sequence can also be represented by a vector of multiple features [35] [55]. And the entire feature vectors of the frames in a video sequence can form a curve in the feature space. Key frames are selected based on the property of the curve such as sharp corners or transformations.

Other key frame extraction work integrates some different technologies into their summarization framework, such as wavelet transform, face and skin-color detection-based [36], statistic-based [53], and time-constrained based [37] methods.

In order to keep it simple and give relatively intuitive visualization to users, we choose to apply sample-based technique by simply extracting key frames of the video (i.e. the I-frames) and merge them together as a gif image. It is a simple but also reasonable solution because most of videos in such sharing systems are short and single-shot only. Therefore, it is not necessary to apply shot-based, cluster-based or feature-based algorithms. We describe the detail design in Section 4.2. Also in Section 7.2.2 we discuss the potential improvements for video abstraction by using video-skimming techniques.

Chapter 3

Problem Statements

Through the discussions in the previous sections, we realized that although a large population of video sharing clients and that of advanced mobile devices are overlapping, unfortunately, most of the video sharing services are still restricted to traditional Internet users, even though more and more mobile devices are multimedia-ready. There are tremendous business opportunities and growing spaces in this area. So far, mobile operators have had a hard time offering inexpensive, easy-to-use and fast services that consumers can use to share images and videos captured by their camera phones. Two major problems regarding mobile video system can be identified as the following.

3.1 Mobility-Aware Social Network

The current existing mobile video sharing systems have not fully utilized the biggest advantage of the mobile devices - the mobility. A multimedia-ready cell phone can take pictures or capture videos anytime anywhere. So ideally clients expect to share their works with others anytime anywhere as well. And the whole service should be affordable, efficient and enjoyable. Instead of simply gathering video clips and streaming them to cell phones, a nice service provider should target on building and maintaining a mobility aware social network among massive end users by effectively utilizing the location information. As such, clients can instantly retrieve nearby photos and videos, make new friends during trips, and share live experiences with old friends. This kind of services is still lack at present.

3.2 Bandwidth and Energy Limitation

Another problem is that the cell phones have limited battery power and restricted network bandwidth. These problems must be addressed because they will affect the overall usability of the whole system. To search for a video, no one is willing to spend a lot of time on waiting for loading a page with searching results. And among the list of results, user should be able to easily find the wanted ones or at least filter out most of the unwanted videos quickly. It is not reasonable to ask user to watch all the candidates in order to find the right one, because it is quite time and energy consuming to streaming a video clip by cell phone. Therefore we come up with the solution of giving video abstraction of the available video clips. So that users can browse quickly and have a look at the abstraction first, then they can decide whether to watch the whole video or not. The details about the solutions MoViShare gives will be discussed in the following section.

Chapter 4

Proposed Solutions

The problems in existing mobile video sharing systems became our motivations of building MoViShare system. In MoViShare, we proposed the following solutions towards the issues stated in Chapter 3.

4.1 Location-Aware Social Networking

Facing the challenge of mobility issue discussed in Section 3.1, MoViShare will create and maintain location-aware social networks among the clients. It will explore users' mobility by effectively utilizing the location information. Specifically, MoViShare will enable context-sensitive location-aware video browsing and sharing. User can hunt for videos based on their current positions or expected locations. Enhanced social networking features can also be applied easily based on the prototype we built, such as locating a friend, communicating with strangers in same event or party, sharing visualized location info to others, etc. This kind of service is unique in mobile industry, and will conveniently facilitate clients to interact with their environment.

However, the way to obtain users' location information is not as obvious as we imagine. We need to consider the cases where the GPS data on cell phones are not available or not feasible. For example, in many indoor zones the GPS devices will be masked; and sometimes user takes shots in one place and uploads them in another place. In MoViShare, the media's location information is always submitted to the server when the content is uploaded. In the stage of this prototype, we will try to get the location information by three ways in the order of priority. If GPS information was recorded when taking the shot, then this is definitely the

first choice because it is the most accurate data representing the video's location. However currently most of the video formats do not contain location data, then we will try to retrieve them from cell phone's GPS device when user is uploading the video. In worst cases, the GPS device is down when user uploads the content, then we have to use the location data retrieved at the last time user enters the system. Indeed, there may be significant position difference when we use the latter two alternatives to collect location data. How to address this problem remains to be an open issue at the next stage of development.

4.2 Video Abstraction

To address the issue of bandwidth and energy limitation, MoViShare adopts an important feature, video abstraction. Every video clip will be abstracted to a small GIF (Graphics Interchange Format) file, which contains only a few key frames and can be displayed on normal web pages. The length of abstraction files are normally only 1% to 5% of original video clips. Therefore, fast browsing and searching features on mobile devices can be activated. And since the abstractions consist of key frames from the original videos, they are decently representative in most of the cases.

By seeing the abstraction image, user can immediately get a global impression of that video. For example, the following questions can be answered after the first glance: what type of story was it (music, sports or nature scene)? Where was it taken (indoor, beach or a snow mountain)? Who was the protagonist of the story (a baby, a dog or a famous movie star)? Figure 4.1 displays a sample video abstraction by listing the key frames extracted from the source video clip. We can easily figure out that this video clip is talking about some highlights of a soccer game. With additional information provided by the system, such as the length, age, view times, rating and comments, user can make a much more persuasive decision of whether to watch the whole video. And further more, the video abstractions can be used by the notification system to promote video clips to selected user groups via emails or short messages.

Knowing the benefits of video abstractions, people may ask a practical question, that is, how to define key frames of a video clip and how to retrieve them. As we discussed in Section 2.5, there are existing researches and sophistic proposals on this problem in recent years. Among bunches of video summary techniques, we choose to apply sample-based technique by simply extracting key frames of the video (i.e. the I-frames) and merge them together



Figure 4.1: An example of video abstraction

as a .gif image. The reasons are as the following. First, we want to keep the process to be simple and fast when running on the server side. Since we adopt the centralizing mode for running the video abstraction task (see next paragraph for detail discussions), we want to avoid the congestions caused by massive uploaded videos during peak times. So a quick running algorithm is preferred in this case. Secondly, it is a simple but also reasonable solution because most of videos in such sharing systems are short and with single shot only. It is not necessary to apply shot-based, cluster-based or feature-based algorithms that cost much more processing time and development efforts while only work better for longer videos. However, we have to realize that this is not a perfect solution because the representative can be compromised in some cases, i.e., the abstraction generated by this way may not nicely represent the real story. In Section 7.2.2, we will discuss another option for video abstraction by using video-skimming techniques.

There are two optional running modes regarding the video abstraction process, at the phone and at the server. The task can be done by mobile devices themselves before uploading the videos. In such way there are no extra server workloads required. And it allows users to flexibly control the ways to manipulate or utilize the video summaries. Users may select from different parameters for different videos to generate best results. After the summary is created, there will be a copy stored on the device. So users can apply further utilities such as short messaging, emailing, or self reviewing. However, there are still some problems we cannot afford to overlook in implementing this mode. First of all, mobile devices are greatly various nowadays in turns of both hardware configurations (i.e., processor, memory, storage, and screen size, etc.) and software deployments (i.e., operating system, development kit, and supported video format, etc.). It is almost impossible to implement a universal application which works for all modern mobile devices. And further more, cell phones usually have no processors powerful enough to run even the simplest video abstraction algorithms. Due to the above reasons, we choose to implement a centralized service for making video summaries. Such service is running at the back-end on our server all the time. At the time a new video is uploaded, the process is triggered to extract salient images from the source. Servers, as known as high-end computers, have the ability to execute complex algorithms for abstracting videos. Thus the workload on mobile devices is minimized. However the flexibility of making and using abstracts on client side will be sacrificed. And we will face another issue regarding the scalability.

When server is busy for handling large amount of requests or long videos, there will

be a delay between video upload time and abstract available time. How to minimize such delay becomes an interesting problem to solve. On one hand, we can upgrade the hardware, e.g., apply multiple processors and more memories, and configure the server to assure the video abstracting service always has sufficient system supports. On the other hand, we can implement a scheduling scheme to apply more flexible queuing algorithms than First-Come-First-Serve behavior. For example, the videos with more streaming or viewing requests should have higher priority to be abstracted. By such way the most urgent cases can be served in first time.

Chapter 5

Design Architecture

5.1 Project Proposal

The ultimate goal of MoViShare system is to provide a universal video sharing platform that accommodates mobile accesses. As the first stage of the whole system, we target a basic prototype system demonstrating the MoViShare fundamental concepts and ideas based on Nokia's hardware and software development environments. Via Wi-Fi access available in most new Nokia mobile device models, the prototype allows users to build connections with other mobile users and share videos. Basic functions of location-aware sharing and video abstracting will be demonstrated as well.

5.2 User Scenarios

First scenario we have addressed is the mobility. Using Nokia's multimedia-enabled mobile devices, users can capture video clips anytime and anywhere. Then via the MoViShare prototype, they upload the media files to the system to share to other users. At the mean time, working with the GPS hardware integrated in cell phone, users' location information (e.g., latitude and longitude value) will be uploaded as well. Therefore, location-aware social network can be built with individual's presence, location and contextual information. Via location-wise social network, people can see where other users are, how far are they away from, or even a visualized location information by pictures or videos. In addition, they can get other contextual information such as what are other users' interests. Thus, they can further reveal nearby friends, places of interests, and potential friends with similar tastes.

Secondly, MoViShare users may discover and then enjoy a new type of social networking experience, i.e. the real-time social activities via cell phone. In traditional social networking, time is an un-honoured matter. For example, user A writes some comments on B's page to ask about a place B has been to or a movie B has watched, but B might not be able to reply to A instantly. Even if A is quite eager to know the answer, it is not expectable to get instant response via traditional web2.0 tools. By contraries, a location based mobile social network will let users communicate with others simultaneously during some events. For instance, users can invite somebody for a drink or a party, or chat up with someone in a bar or an event, while in a traditional way they might be shy or unconfident to talk to others.

5.3 Modules

In view of the scenarios about mobility, location-based video sharing and social networking, we outline the major modules of the full-fledged MoViShare. In the timeframe of this proposed project, we will identify the key design issues of MoViShare, and implement a preliminary workable prototype.

There will be three major modules for the whole MoViShare System.

5.3.1 User Interface and Social Network Module

This module will serve as a transparent middleware to intercept the requests, translate them, and then redirect users to the video sharing sites, with streaming being realized by a dedicated module to be described later. This module also plays an important role in creating and maintaining the mobile social network among the users and their friends. We will carry out an in depth study on the structure of the mobile social networks as well as their impacts and potentials toward video sharing services.

5.3.2 Location Management Module

The location management module keeps track of the client location (through GPS or Wi-Fi access point's information), and works together with video sharing sites to provide context- and location-aware video content. The most distinct feature (and flexibility) of mobile users, as compared to regular fixed-location Internet users, is that they can move over time.

The location management module can redirect a client to a site dedicated for the current location, or, with support from the video site, provide a list of related video clips sorted by a location-aware score. More importantly, it will work together with the social network module to promote location-aware medias to selected groups of users.

5.3.3 Video Abstraction and Streaming Module

The streaming module bridges mobile users and remote sites for high-quality real-time video delivery. While streaming has been realized in current mobile platforms, MoViShare will distinguish itself from existing solutions by exploring video popularity and correlations derived from social networks. Our experience with YouTube videos has shown that there are strong clustering behaviors among videos and users, thanks to social networks [33]. This offers great potentials for effective caching and pre-fetching of videos of interest, which will be explored in later stages (see Section 7.2.3).

In addition, there should be a module to abstract the video clips, i.e., to generate the abstraction from large video files for users to glimpse. Therefore, users might be able to get a clear idea about the main idea of the video within first sight. The details about this module have been presented in Section 4.2.

5.4 System Architecture

5.4.1 Design and Dataflow

Figure 5.1 displays the video uploading dataflow. All videos are uploaded with associated Meta data such as owners' identifications, video titles, keywords, creating time, and most importantly, location information. All Meta information is saved in database for effective searching and management, while the physical media files are stored in a file server for fast streaming purpose.

Figure 5.2 shows the dataflow of retrieving video. User submits a request to WAP server. The system will perform a query through database to retrieve the target Meta information that includes the physical location of the video file in the file server. Then the Meta data is sent back via WAP service while the content of video is played back through streaming service.

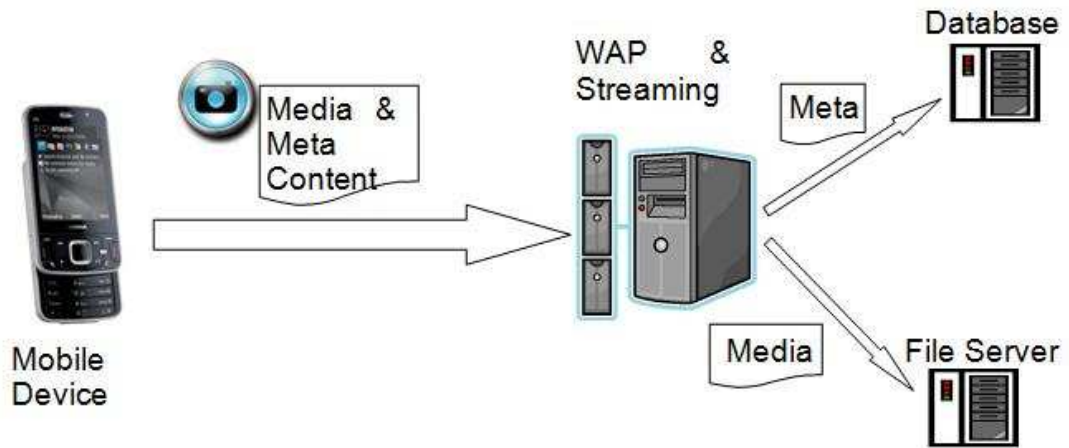


Figure 5.1: MoViShare video uploading data flow

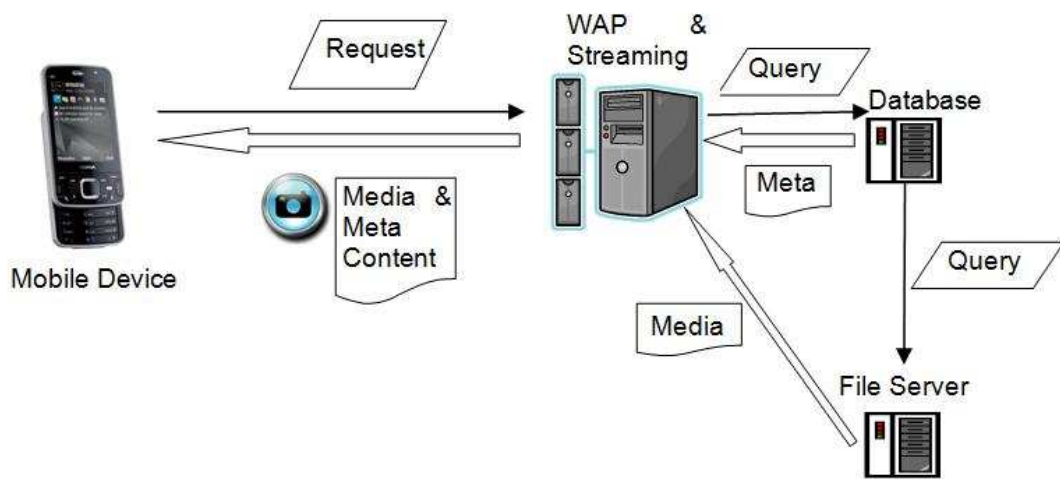


Figure 5.2: MoViShare video retrieving data flow

5.4.2 Components

MoViShare system consists of two fundamental components, Server and Client. Figure 5.3 shows this client-server structure in MoViShare system, as well as the major services and functions in each component.

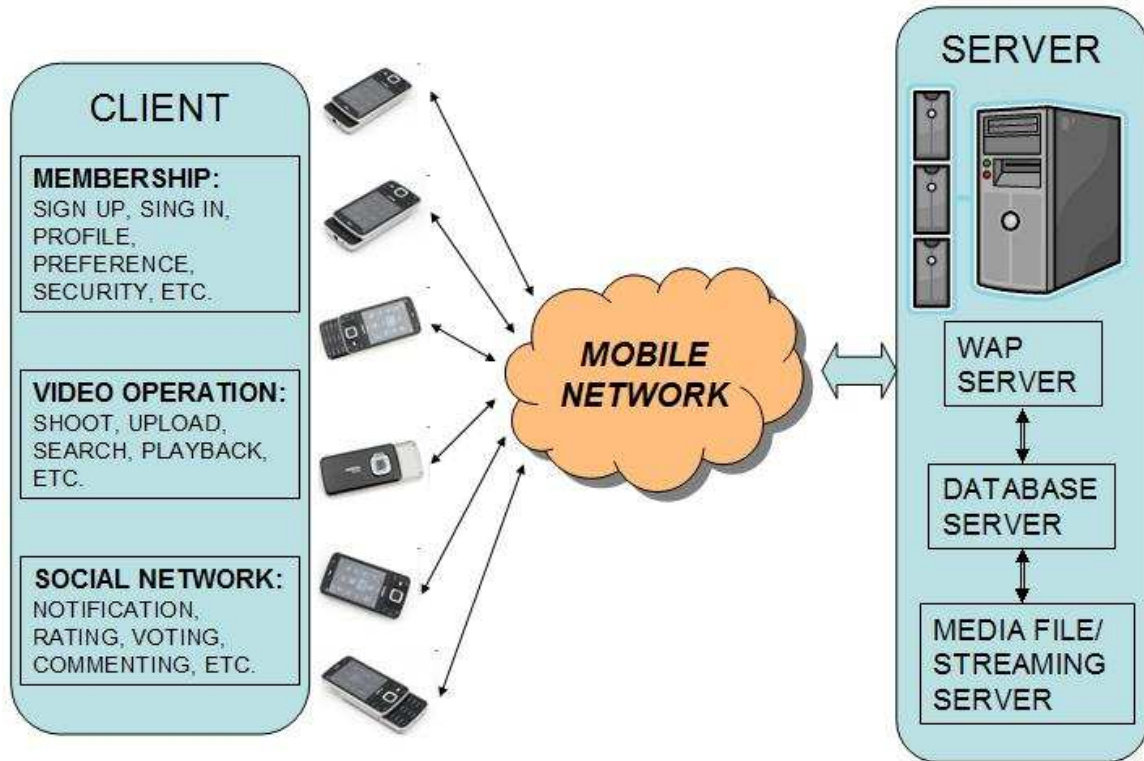


Figure 5.3: MoViShare client-server structure

MoViShare Server

MoViShare Servers provide WAP (Wireless Application Protocol), database and file services for the whole system.

A WAP site enables access to the Mobile Web from a mobile phone or PDA [21]. MoViShare WAP site will provide services like video searching and retrieving, message notification, membership registration, security setting, etc. Since WAP specification defines a powerful and functional user interface model that is appropriate for hand-held devices [20],

when enjoying these services, users will have very similar experiences as they are using PC web browsers. Users navigate with up and down scroll keys instead of a mouse. Soft keys allow the user to perform specific operations appropriate to the application context. Users can use a traditional 12-key phone keypad to enter alphabet and numeric characters, as well as a full set of standard symbols. Navigation functions such as Back, Home, and Bookmark are also provided. The micro-browser allows devices with larger screens to automatically display more content. These design standards provide familiar functionality for those accustomed with the Web. They also offer user interfaces that are easy to learn and highly discoverable for first time users.

Besides friendly user interface, WAP specification can address some other important design objectives of the MoViShare system. First, WAP Gateway leverages standard web proxy technology to connect the wireless domains, and decreases the response time to the mobile device by aggregating data and caching frequently requested video contents (see Section 7.2.3). In addition, the protocol stack defined in WAP optimizes standard Web protocols for use under the low bandwidth, high latency conditions often found in wireless networks. Take HTTP for an instance, a number of enhancements to the session, transaction, security and transport layers provide HTTP functionality better suited to the wireless network environment. And last but not least, the WAP specification ensures that a secure protocol is available when in some cases clients require more privacy protections when using MoViShare.

A database is set up to store both users' profiles and videos' Meta data. To build location aware social networks, we must maintain not only the individuals' data like membership info, profiles, relationships, preferences, security settings, etc, but the location-wise media information as well, such as where and when the video was taken, how long it was and what kind of content it was about. Theoretically any mature relational database systems on the market can be a qualified candidate for MoViShare. In this prototype of MoViShare, we choose Microsoft Access to be the database system due to its simplicity and flexibility of transferring.

However in real-live video sharing systems, a more comprehensive database product should be inducted. We suggest when choosing a proper database product from the market, the following features need to be considered in first place. First of all, the performance will be the major concern in database with massive amount of data. And in MoViShare many database query requests involve the computations of location coordinators. That will be an

extra burden for the database server. The database products with multi-threading feature and SQL preprocessor can be more helpful in this issue. Secondly, we need to take the scalability and availability into account. The potential users of MoViShare will scatter all over the country, the continent, or the world, and there are usually small amount of videos that attract large portion of views. A recent analyzing shows that 10% of the top popular videos in YouTube account for nearly 80% of views, while the rest 90% of the videos account for very few requests [31]. Therefore, we prefer the database to have high availability for most popular videos, i.e., to maintain the availability of applications so that the perceived downtime for users can be minimized. And moreover, the security and safeness of the database should never be overlooked. The candidate database system must have a powerful security schema and a reliable backup/recover system.

All the media files can be stored and managed by a separate server or a stand-alone service. For the sake of simplicity, we use only one server to provide WAP, database and file services in this prototype. However, in real system, a dedicated file server or a server group is highly recommended. Video sharing servers are definitely one of the most heavy-loaded server groups over the Internet. Therefore, sophisticate storage management is quite demanding. The candidate techniques include load-balancing [11], failover-recovering [4], RAID [16], etc. These techniques are usually working together to retrieve maximum gain. Load-balancing is often used to implement failover - the capability to switch over automatically to a redundant or standby component upon the failure. When one component becomes unavailable, the load balancer is informed and no longer sends traffic to it. When a component comes back on line, the load balancer begins to route traffic to it again. In a RAID disk controller, using RAID1 (mirroring) is analogous to the "live/backup" approach to failover, where RAID5 analogous to load balancing failover.

MoViShare Client

MoViShare Client component integrates a series of functions for the end users to provide an intuitive application on mobile devices.

The membership-related functions offer full set of membership services including signing in, signing up, subscribing, personal profile managing, location information updating, preference choosing, and security setting, etc. Unlike the traditional membership registration in Internet, the process in mobile environment should be and can be much more simplified.

Noticing that the input systems on most mobile devices are not as convenient as PCs' keyboards and mice, no one is willing to experience a tedious, multi-paged registration process on cell phone. To address this concern, MoViShare uses the cell phone number as the default identification field and makes the whole process neat and simple. Unless user volunteers to provide more personal information, he/she can sign up to MoViShare by simply submitting their cell phone number to the system. Similarly, user can log in MoViShare by pressing on an icon to load the client application, which will automatically retrieve the cell phone number and submit to the server, and afterwards user can upload or watch videos. For sure, users always have the option to choose a password or a different identification to make the process more secure if they feel necessary.

Bunches of important functions are related to the videos processing procedure in MoViShare system. This procedure involves straightforward operations such as video capturing, video sharing (i.e. uploading), video searching and playing back. In addition, it also includes some assistant functions for notification system and rating/voting/commenting system. After taking a shot by cell phone, user can easily upload the video to MoViShare system, and send notification short messages to specific groups of other users, such as friends, nearby active users, content subscribers, or random users. The multimedia message will include the video's title, description, location info, and most importantly, the abstraction (see Section 4.2). Being notified, users can sneak peek the video by its abstraction, then decide whether to play back the entire video or not. They can also rate, vote or comment on this video afterwards.

5.5 Application Interface

5.5.1 Client Side Main Interface

The main interface after launching MoViShare in mobile devices is showed in Figure 5.4. This is the first interface users will see when entering the MoViShare system by their cell phone. Two big icons lead users to two main functions in this prototype, "upload videos" (blue icon) and "search videos" (red icon). Note there are some hidden fields with default values appear in the bottom part of the screen. They are important for building location-aware social networks and are transparent to clients in real-live system. The latitude and the longitude values are retrieved from the built-in GPS device of the cell phone. The username (or userid) is the major identification for each individual and is associated to each mobile

device automatically. The location information and the client identification will be passed to server as soon as the user enters MoViShare system.

5.5.2 Uploading Video Interface

The video uploading interface is showed in Figure 5.5. The media-uploading interface is quite straight forward and friendly to users. After selecting a media file which is stored in the device, user can input description, keywords and regional information, and then press "Upload Media" button. During this step, the media's location information will be uploaded as well. We will get the location information by three alternatives in the following order of priority. First, if the GPS information was recorded when taking the shot, then this is definitely the most accurate data representing the video's location. In such case, system will extract the location information from the media file and store into database. However currently most of the video formats do not contain location data, then we will retrieve them from cell phone's GPS device when user is uploading the video and submit them together to the server. In worst cases, the GPS device is not available at that time either, and then we have to use the location data retrieved at the last time user enters the system.

5.5.3 Video Browsing Interface

The video clips listing page is showed in Figure 5.6. On the video clips listing page, we list current available videos by the order of the distance to the mobile device by default. The closest videos are showed in first place. User can see the abstracted GIF file and other text context such as title, timestamp, view times, and how far the video's location is. User can further click the GIF file to start watching the video. Other browsing options include listing videos by the popularity, showing friends' latest videos, and reviewing all self-uploaded videos.

5.5.4 Location-Wise Video Display

As showed in Figure 5.7, videos can also be displayed directly on a map. This is a much more intuitive way to show the videos. On a Google map, the small red icons indicate the exact locations of the videos. When clicking the icon, user will see the video abstract and introduction in a pop-up box. User can further click the 'watch this' link to start streaming the video.



Figure 5.4: MoViShare client side main interface

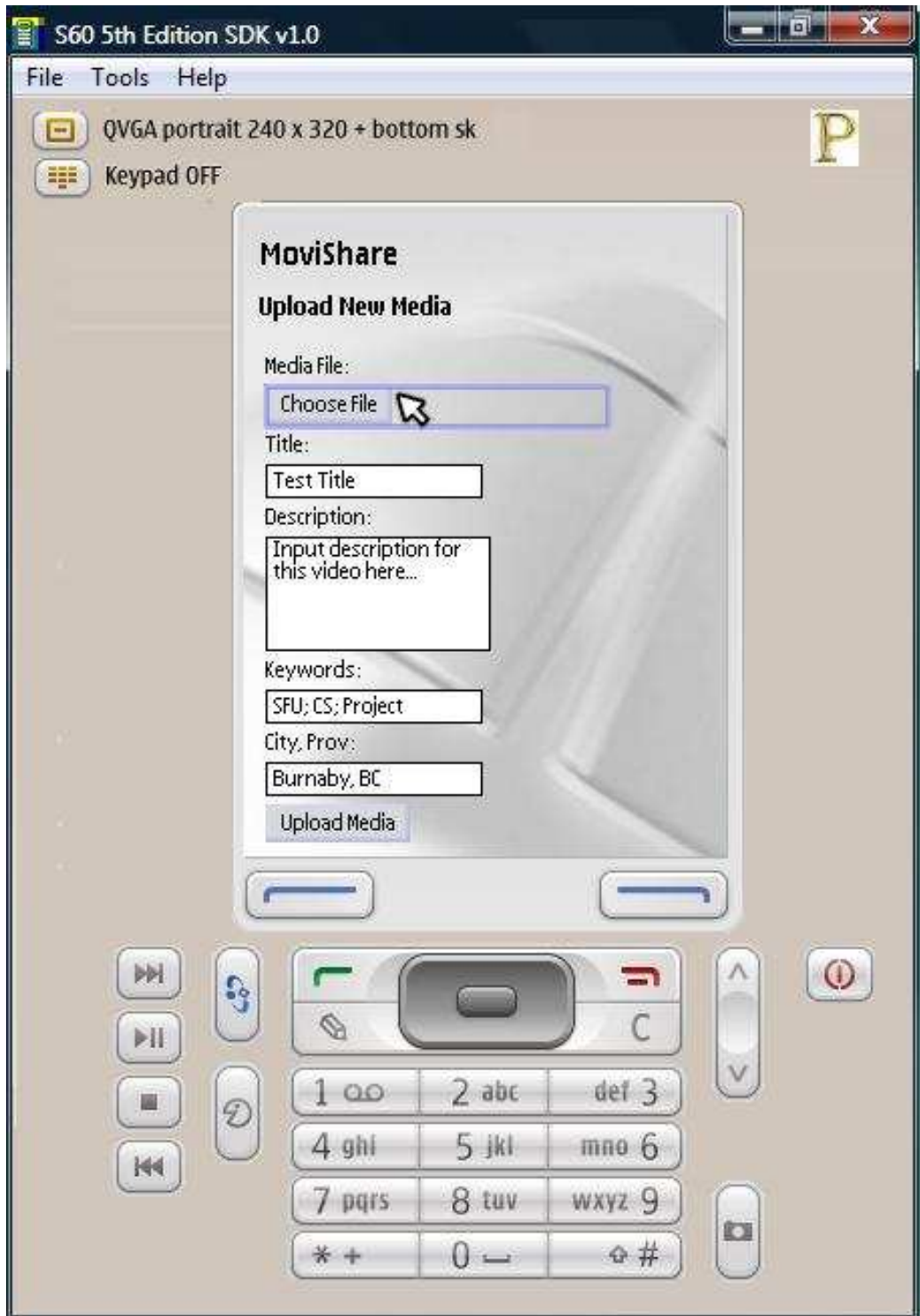


Figure 5.5: MoViShare interface of uploading video

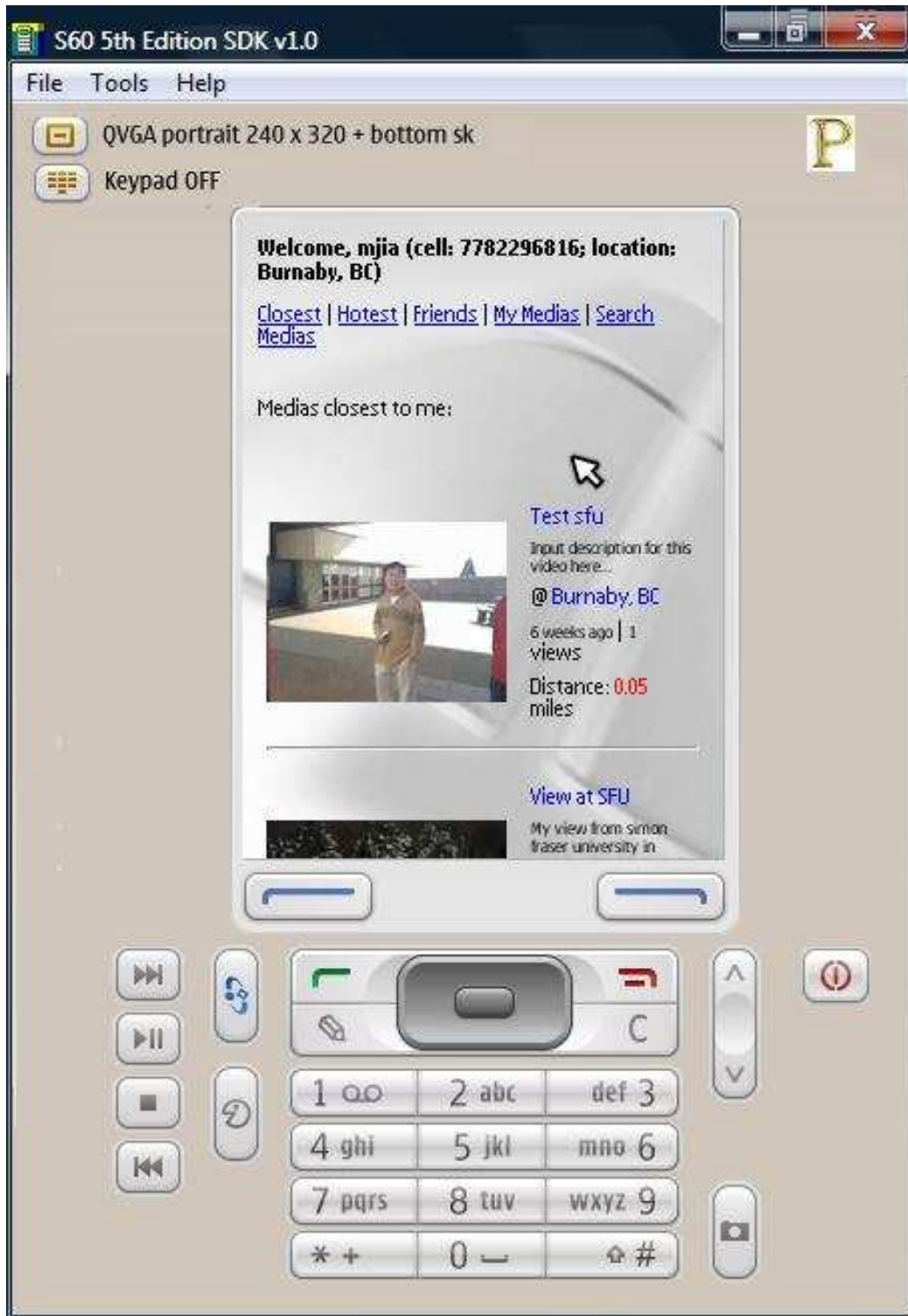


Figure 5.6: MoViShare video browsing interface

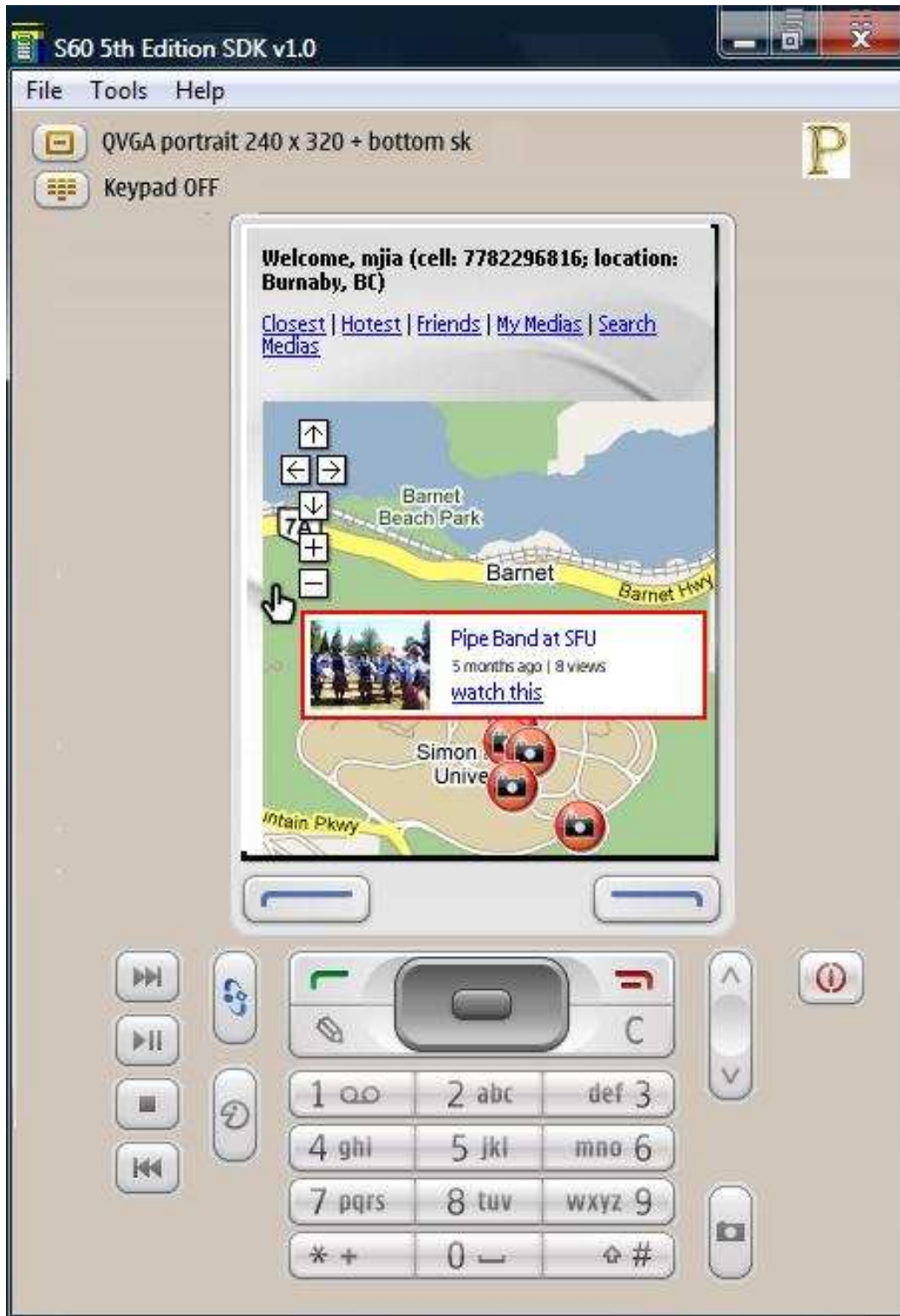


Figure 5.7: MoViShare location-wise video display interface

5.5.5 Video Searching Interface

Clients can search videos in different ways, as showed in Figure 5.8. MoViShare provides advanced search options for users to locate an expected video source quickly and accurately. Figure 5.8 shows that clients can look for medias by keyword, postal code, or regional information. The searching functions can always be extended based on the prototype we built.

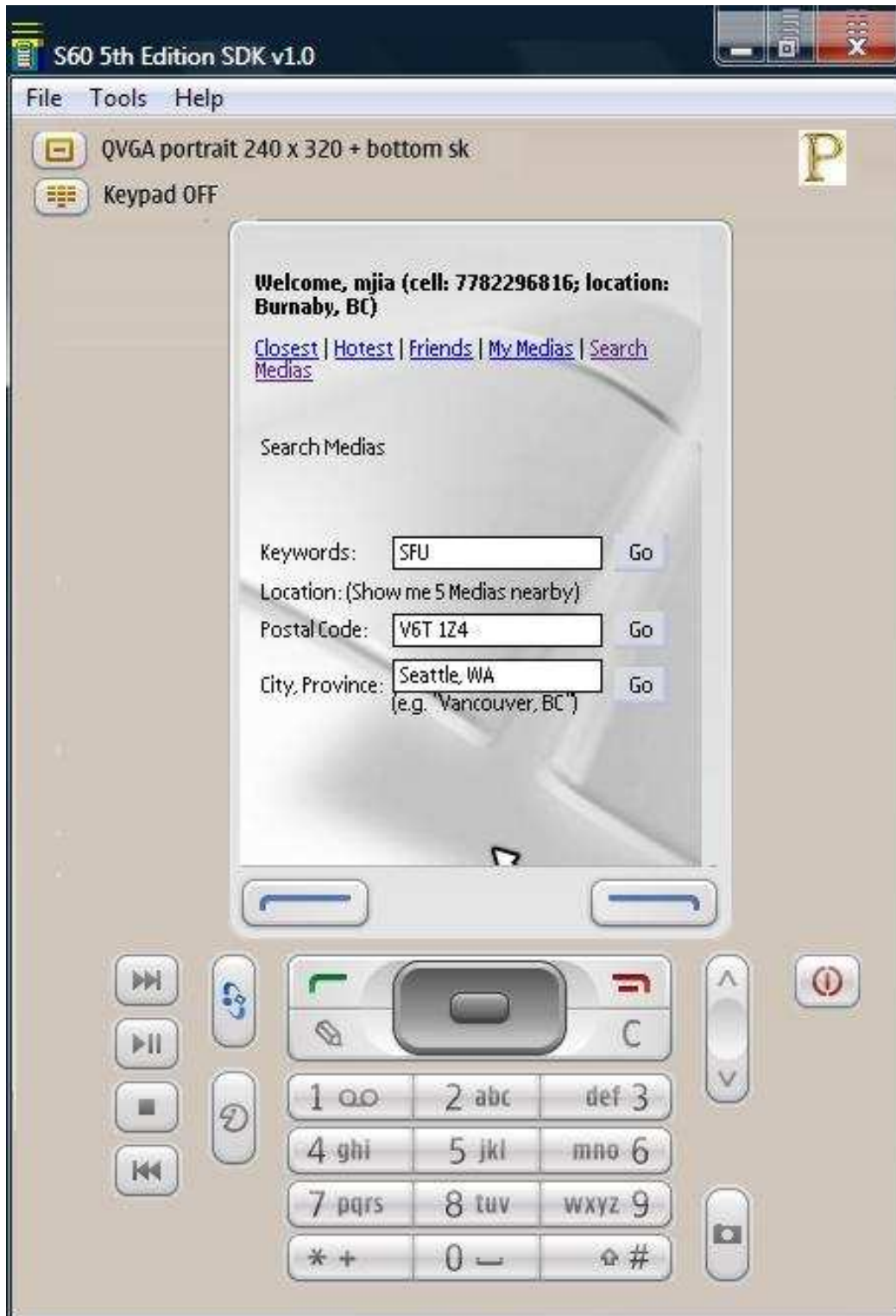


Figure 5.8: MoViShare interface of video search

Chapter 6

Implementation Details

Our MoViShare prototype was built on the N96 smartphone with the Nokia Symbian S60 development platform. The entire period of building MoViShare prototype was close to 4 months, including planning, designing, environment setting, coding and testing. I served as the principle developer for the system architecture and the database structure design. I was also responsible for the installation and configuration of the WAP site, the Database server, and the streaming server. Further more, I developed all web pages for video sharing and retrieving functions, as well as the whole video abstraction service running on the server.

In this section, I will discuss some key technical challenges and solutions for the implementation.

6.1 Programming Modules and Tools

Mobile video sharing system is a very complex system because it involves various IT technologies. During the period of developing this prototype, we have experienced different operating systems, development tools, and coding languages. There are three major programming modules within the MoViShare prototype, Client side application, WAP Site, and video abstraction service. The cell phone we use in this project is Nokia N96 whose operating system is Symbian OS 9.3. For the client side application, we use Symbian S60 3rd edition as the software platform [18]. The main function of client side application is to provide user an entrance to MoViShare system. After the application is launched on user's cell phone, it will automatically connect to the server and pass user's identity and location information to the database. Then user can enjoy movie sharing and other social networking

services. We code the client application by using the S60 Web Runtime technique [17]. Web Runtime (WRT) extends the Web Browser for S60 to enable widgets and offers an optimized web experience that a user can access with a single click. And on the server side, we adopt Microsoft .Net2.0 as the framework. The development tool is Microsoft Visual Studio 2005 Web Developer. The WAP site pages offer video uploading, searching and other functions to end users. The database system we choose is Microsoft Access 2003 (Section 5.4.2.1). We also integrate other web developing technologies such as Javascript, Google Map API to make the interface more friendly and intuitive; we will discuss in more detail in Section 6.3.1. Also on the server side, we use Microsoft Visual Studio 2005 to develop a windows application for the video abstraction service. In turns of programming language, most of the code is programmed by *c#*, while a small portion is written in *c++* language. We will introduce more techniques and concerns regarding video abstraction module in Section 6.3.2, 6.3.3 and 6.4.3. Beside the above programming modules, at the server side we also installed and configured Helix server from RealNetworks Inc. to provide video streaming service.

6.2 Database Structure

The database in current prototype consists of four tables. 'Users' table is storing every user's profile, preference, location and security information. 'Friends' table draws a map of relationships among different users. 'Postalcode' table saves all zip codes and postal codes in North America and their geographical positions. And the most important table, 'Medias', keeps every video's meta information. Table 6.1 shows the structure and several sample tuples of 'Medias' table where we can retrieve the video's title, description, keywords, physical file location, owner, feedbacks, and most importantly, the location information (longitude and latitude).

6.3 Challenges and Solutions

During the development of MoViShare project, we had faced couple practical issues that well challenged our knowledge and skills. We listed three representative problems when developing the functions of web searching and video abstraction. We managed to solve them efficiently by studying many related works and making considerable attempts and

Media ID	3	8	16
User ID	2	4	1
Title	SFU Clan vs. Seattle U.	Why UBC?	Vancouver Gastown steamclock
Description	SFU Clan vs. Seattle University Redhawks - 2008 Game Recap	Video I made for an international student competition at UBC on why we chose UBC	Vancouver Gastown steamclock
Keywords	SFU; Soccer; Clan	UBC; Intro	Vancouver; Gas Town; Steamclock
File Path	sfu1_soccer.rmvb	ubc1_intro.rmvb	van4_gastown.rmvb
Abstract File Path	sfu1_soccer.gif	ubc1_intro.gif	van4_gastown.gif
Time generated	9/27/2008	12/24/2007	1/5/2008
Time uploaded	9/27/2008	12/25/2007	1/6/2008
View Number	7	5	8
Download Number	2	1	4
Latitude	49.278333	49.266667	49.284444
Longitude	-122.921944	-123.247222	-123.109444
Region	Burnaby, BC	Vancouver, BC	Vancouver, BC

Table 6.1: The structure and sample tuples of 'Medias' table

tests. At the meantime we improved our skills of programming and problem analyzing, and gained priceless experiences of various technologies.

6.3.1 Display Videos More Intuitively

In MoViShare, in order to emphasize the location feature of videos, we decided to display videos directly on a map with small icons to indicate their exact positions (see Figure 5.7). It is a challenge to dynamically draw icons on a map based on users' search results. We found a common tool, Google Map API [7], to implement this feature. The Google Maps API allows user to embed Google Maps in web pages with JavaScript. The API provides a number of utilities for manipulating maps and adding content to the map through a variety of services. We made efforts to apply this API into .net2.0 web pages, and to adjust the output to be fit in cell phone browsers. However, one remaining practical issue regarding this feature is how to draw multiple videos in same location? We will discuss it in Section 6.4.1.

6.3.2 Extracting Key Frames

In Section 2.4 we discussed the nested box structure of MPEG-4 video files. Theoretically we can extract any frames from the video file based on the defined structure. We can simply build different objects for all types of boxes, and then read data from file into these objects. However it is quite expensive to create, maintain and operate on such a nested box structure in memory. And it is definitely not scalable for long videos or more videos. We notice in video abstraction service, our purpose is not to decode whole video sequences; all we need is to extract several key frames. Therefore our code is simplified to only work on necessary boxes and only extract enough key frames. The Sample Table Box ("stbl") contains all the time and data indexing of the media samples (frames) in a track (see Figure 2.1). One of its child boxes, the Sync Sample Box ("stss"), provides the set of the sequence numbers of all the random access points (i.e. I-Frames) within the stream. After getting the sequence numbers of I-Frames, we can retrieve their physical positions in the stream from the Chunk Offset Box ("stco") and their sizes from the Sample Size Box ("stsz"). Therefore we can extract the key frames from the file by dealing with these three boxes only; in such way the consumed memory and the processing time are both minimized.

6.3.3 Decoding, Converting and Creating

In video abstraction process, there are still some tedious works to do after extracting key frames from video sequences. The frames extracted from the video file are encoded by MPEG-4 standard. We need to decode each frame and convert to normal image file format such as BMP. And then we can generate GIF file based on these BMP files. These processes involve complex algorithms and massive coding efforts. Fortunately they are also common tasks and there are many open source code and public free software that solve them well. We found proper existing solutions for these tasks and integrated them in MoViShare. We used the code developed by Andrea Graziani in project Mayo [12] for decoding the MP4 frames to YUV format stream. In same project we found a function called 'YUV2BMP' that could convert YUV stream to BMP file. Since the code was written in c++, we made necessary efforts to convert them to c#. For generating GIF file, we found a free software called ImageMagick [8] which did a fantastic job on creating, editing, and composing bitmap images. By simply calling ImageMagick commands in c#, we can get expected GIF file composed by the BMP files.

6.4 Outstanding Issues

Due to time constraint and resource limitations, there are still some technical issues remaining outstanding in this prototype. The major flaws can be found in video searching function and video abstraction process. Fixing these defections will be the serious task in next stage of development.

6.4.1 Display Multiple Videos at Same Position

We need to design a more intuitive interface to display more than one videos in same position. In the map that shows videos' exact locations, if two or more videos happen to be taken in exactly same position, they will be overlapping each other in current design. Thus user can only see one video's information when clicking on the icon (see Figure 5.7). We need to figure out a proper solution to handle this issue. One option is to display all videos in slide-show manner, i.e. show first video when the icon is clicked, then show next video after 2-3 seconds, and so on. This may not satisfy impatient users because it forces user to wait several seconds for the expected video. It also requires users to make decision

in a short time because if they miss the video, they have to wait for a round for its coming back. Another option is to pile up all videos in a large box so user can see everything immediately. It solves the problems of the slide-show manner; however it is not scalable because the cell phone screen is not large enough to display many videos together. A hybrid way that combines slide-show and pile-up manners may be a good solution, i.e. in each slide we display 2-3 videos together in a larger box whose size is still good to show in cell phone screen, and after 2-3 seconds we show the next slide. We may also add in other features for quickly locating any videos, such as allowing user to jump among slides by clicking the slide number, or to select video directly from a dropdown list of all video titles.

6.4.2 Different Video Formats

So far our video abstraction service can only deal with one video format, MPEG-4. Although it is a popular video format that can be played back by many mobile devices such as Nokia N-series, PSP, iPhone, and Nintendo DS, it is not the default video file format when taking shots by all other cell phones than Nokia. For example, iPhone 3GS is using MOV format when shooting videos, and Motorola is using 3GP format in some models. It is necessary to enhance the current video abstraction service to support other popular video formats such as MOV, 3GP, RM, AVI, etc. Discussed in Section 6.3.2, the design principle of the video abstraction process is to extract key frames rather than to decode whole video sequences. Thus when we extend the code to handle other video formats, we must follow the principle as well to minimize the cost of memory and processing time.

6.4.3 Video Abstraction Delay

In Section 4.2 we mention the scalability issue, i.e., when server is busy for handling large amount of requests or long videos, there will be a delay between video upload time and abstract available time. We also list possible methods to minimize this delay. But the first thing we need to do is to see whether the current code for video abstracting can be improved. The whole service is programmed by *c#*, and we adopt some existing open source code written in *c++* for decoding frames, and integrate a public software for generating GIF file. This remains a deficient design since it creates more delay when jumping among different running environments. The proper way is to implement own *c#* code for MP4 decoding and GIF making functions so that the whole process keeps in same running environment

and optimizations can be applied when compiling the code. However it requires much more time and efforts for digging deeply into the MP4, GIF standards and studying existing code samples. So this yields to be a task in next stage of development.

Chapter 7

Conclusions and Future Works

7.1 Conclusions

The ultimate goal of MoViShare project is to build a universal video sharing platform that accommodates mobile accesses. We have represented the massive growth in both Web2.0 and mobile industry with extreme rapidity in recent years. We also showed the vacancy in mobile multimedia source sharing area. Two major problems regarding mobile video system are identified. Firstly, the biggest advantage of the mobile devices, the mobility, has not been fully utilized. And second, the cell phones have limited battery power and restricted network bandwidth. To address the first problem, MoViShare effectively utilizes the location information and enables context-sensitive location-aware video browsing and sharing. So that location-aware social networks can be built by our system to benefit mobile users. MoViShare also adopts video abstraction technique to summarize each video into a tiny animation GIF image file which consists of key frames from the source video clip. This enables fast browsing and searching feature in mobile devices and thus saves energy and bandwidth. MoViShare prototype applies client/server architecture with three major built-in modules, named social network, location management and streaming module. We have demonstrated the detail design aspects, data flow charts and user interfaces of the system.

7.2 Future Works

The mobile social networks are becoming the industry trend and attracting tons of both marketing revenues and technique innovation efforts. A mature mobile video sharing system

is a very complex system integrated with various technologies such as multimedia representing, network modeling, video streaming, and social networking, etc. We can only make some basic attempts in this prototype. Many areas of our work can still be improved upon. We discuss below possible future works and the difficulties in solving them.

7.2.1 New Search and Social Features

First, we will enhance the system by appending bunches of new search and social features on the next stage. As we are on the first stage of the whole system, we build a basic prototype that allows users to connect with other mobile users and share videos. The location-aware searching and social networking functions are still moderate at this time. We will improve them by making the following efforts.

Searching function is extremely important for clients when using services that connect to massive amount of online resources. It is even more critical in systems like MoViShare because users are being a mobile environment and using mobile devices when doing searches. We will improve the MoViShare searching tool by adding more options and simplifying the process. Although location is the key searching criteria in MoViShare, feedbacks from social networks are determinative in many cases. So we will offer more searching options such as lists of top rating videos, most recommended videos, and videos with most positive comments. And clients can always use these options combined with locality during the video searching. Due to the restricted input capabilities in mobile devices, we will simplify or avoid keyword input to the utmost degree by offering various alternative means as the following.

- User can choose from the list of most frequently used keywords in this area;
- User can choose from the list of pre-selected location related keywords;
- User can walk through an established taxonomy structure;
- User can get auto-completing service when typing the first several characters of the keyword;
- User can always see and select from the keywords he/she used before.

On next stage, we also consider to consummate the membership/friendship system in MoViShare. Social networking is the soul concept of MoViShare. Functions that help to

build a favored and effective social network are never adequate. In future MoViShare system, user can label his/her friends by multiple tags, so people with same tags among each other can form groups. Every active user will be labeled by the current location information as soon as he/she logs into the system, so people in same location can form a special group automatically. We will also enhance the notification system so that fresh videos can be notified quickly within groups and/or among groups. In addition to commenting and voting systems that most video sharing sites already have, we will develop the new recommending system which allows user to advertise his/her favorite videos to different types of groups. As we mentioned before, combined with the location data, videos with most recommendations, votes or positive comments will appear most frequently on users' screens. Moreover, based on video sharing service, we will develop more real-time social networking tools such as chatting, poking, twittering, etc. In short, MoViShare will provide users anytime anywhere social experiences via cell phones.

7.2.2 Video Skimming Instead of Video Summary

Video skimming is another research direction in video abstraction which is focusing on the moving-image abstract. This type of abstract is also known as moving storyboard, or multimedia summary. As opposed to the method we used, i.e., to collect key frames (still images) from the source video, the skimmed file consists of a collection of image sequences, as well as the corresponding audio abstract extracted from the original sequence and is thus itself a video clip but of considerably shorter length.

In mobile video system, there are also advantages using video skimming. Compared to a still-image abstract, it makes much more sense to use the original audio information since sometimes the audio track contains important information such as those in training courses, touring guidance, or visible voice messages. Besides, the possibly higher computational effort during the abstracting process pays off during the playback time: it's usually more natural and more interesting for users to watch a trailer than watching a slide show, and in many cases, the motion is also information-bearing [42].

There are quite a few efforts on video skimming in recent years, however, not all the proposed solutions fit in MoViShare. Some straightforward approaches would be to compress the original video by speeding up the playback, such as what Microsoft Research [46] and IBM [30] did. These techniques are barely suitable for MoViShare because the maximum time compression ratio is only 1.5-2.5 depending on the speech speed [40], beyond which the

speech becomes incomprehensible. Some algorithms depend heavily on the text information in the video such as closed captioning, manual transcript, etc., such as [34] [51]. Obviously they cannot be applied by MoViShare since most user generated contents don't have any text information. Solutions like [44] totally ignore the accompanying audio track and thus have no significant advantages over the current method of still-images abstraction.

Among the existing video skimming techniques, the cluster-validity approach proposed by [39] may be a qualified candidate. All video frames are grouped into n clusters, while the optimal value of n is determined by a cluster-validity analysis. One representative keyframe is then chosen from each of these clusters and finally, the skimming is generated by concatenating all video shots which contain at least one keyframe. This approach exhibits a nice performance on specific events with a well-defined and reasonably structured content. However further modifications and enhancements are mandatory before it can be plugged into MoViShare system.

7.2.3 Proxy Caching

Scalability is one major issue before MoViShare can be widely promoted to the public. MoViShare not only provides the video clips taken by mobile devices of registered users, it offers a bridge to traditional video sharing resources such as YouTube. Scalability is one of the biggest challenges that the existing video sharing providers currently face, and this problem gets even worse in mobile environment due to the hardware and bandwidth restrictions. Caching frequently used data at proxy servers close to clients will be an effective way to save backbone bandwidth and prevent users from excessive access delays. The WAP specifications we discussed in Section 5.4.2.1 guarantee the feasibility of proxy caching in MoViShare.

Numbers of algorithms have been developed for caching web objects or streaming videos [43]. Different from traditional web video sharing, two unique characteristics of mobile video source have to be addressed when we design the proxy caching schema. First of all, the size of video files is usually small and the length is short. Secondly, the locality may be the deciding factor to persuade people to watch instead of popularity, i.e., users prefer to watch videos uploaded by others in same or close location, while in traditional sites the videos with most views always attract more eyes. Considering these factors, full-object caching for web or segment caching for streaming video are not practical solutions for MoViShare, whereas prefix caching [49] is probably the best choice, i.e., for each video, the proxy will

cache a five second initial clip (about 200 KB) of the video. The cache efficiency can be further improved by exploring the importance of locality characteristic. A proxy server should cache the videos closest to the region it serves with first priority.

7.2.4 Understanding Mobile Social Networks

MoViShare is also a potential practical platform to learn the characteristics of mobile social networks. Understanding the nature and the impact of social networks is crucial to network traffic engineering and to sustainable development of such services. There has been a significant research effort into studying the characteristics of traditional Internet media sharing services and social networks [32]. However, the differences of scale, dynamics, and user behaviors in mobile environment make traditional predictions unsuitable. Studies on mobile video sharing social network are still moderate due to the lack of a universal mobile video sharing platform, which is also the goal MoViShare is designed for. Although the prototype we made is still far to be live, we successfully illustrated the fundamental concepts and architectures that make MoViShare a potential promising solution in near future. We will develop an integrated data collecting schemes so that once MoViShare is going live and attracting clients, useful data can be collected in first place. The following data is of the most interest to us: videos' meta data (length, time, location, etc.), users' anonymous profile information, users' interests and behaviors. However we must assure that users' privacy would not be compromised during the process. Obtaining these important data, further analysis and researches can be applied to learn the characteristics of mobile social networks.

Bibliography

- [1] airG's Mobile Community. <http://www.airg.com>.
- [2] Everyone is Talking About Mobile Social Networking. http://corp.airg.com/news/inNews/eMarketer_05222008.htm.
- [3] Facebook Mobile. <http://m.facebook.com/>.
- [4] Failover wiki. <http://en.wikipedia.org/wiki/Failover>.
- [5] Global Positioning System. <http://en.wikipedia.org/wiki/Gps>.
- [6] Google Latitude. <http://www.google.com/latitude/intro.html>.
- [7] Google Map API. <http://code.google.com/apis/maps/>.
- [8] ImageMagick. <http://www.imagemagick.org/script/index.php>.
- [9] ISO/IEC 14496-12, ISO Base Media File Format; technically identical to ISO/IEC 15444-12.
- [10] ISO/IEC 14496-14, MP4 File Format.
- [11] Load Balancing wiki. [http://en.wikipedia.org/wiki/Load_balancing_\(computing\)](http://en.wikipedia.org/wiki/Load_balancing_(computing)).
- [12] Mayo. <http://www.projectmayo.com/>.
- [13] MocoSpace wiki. <http://en.wikipedia.org/wiki/Mocospace>.
- [14] MySpace Mobile. <http://www.myspace.com/index.cfm?fuseaction=mobile>.
- [15] Ovi wiki. [http://en.wikipedia.org/wiki/Ovi_\(Nokia\)](http://en.wikipedia.org/wiki/Ovi_(Nokia)).
- [16] RAID wiki. <http://en.wikipedia.org/wiki/RAID>.
- [17] S60 Web Runtime (WRT). [http://wiki.forum.nokia.com/index.php/Category:Web_Runtime_\(WRT\)](http://wiki.forum.nokia.com/index.php/Category:Web_Runtime_(WRT)).
- [18] Symbian S60 SDK 3rd Edition. <http://www.forum.nokia.com/info/sw.nokia.com/id/4a7149a5-95a5-4726-913a-3c6f21eb65a5/S60-SDK-0616-3.0-mr.html>.

- [19] Top Mobile Sites and Top PC Sites. <http://www.itfacts.biz/top-mobile-sites-and-top-pc-sites/12684>.
- [20] WAP White Pages. http://www.wapforum.org/what/WAP_white_pages.pdf.
- [21] WAP wiki. http://en.wikipedia.org/wiki/Wireless_Application_Protocol.
- [22] What is MP4 Format Video? <http://www.tansee.com/what-is-mp4-video.html>.
- [23] Yahoo! Mobile. <http://mobile.yahoo.com/>.
- [24] YouTube Extends Mobile Video Sharing Service. <http://www.it.com.cn/f/news/081/24/541937.htm>.
- [25] YouTube Offers Mobile Upload Service. http://news.cnet.com/2100-1025_3-6070527.html.
- [26] YouTube Goes 'Moblogging'. <http://money.cnn.com/2006/11/02/magazines/business2/moblogging.biz2/index.htm>, 2006.
- [27] YouTube Serves up 100 Million Videos a Day Online. http://www.usatoday.com/tech/news/2006-07-16-youtube-views_x.htm, 2006.
- [28] Online Video: 2007 Year In Review. <http://blog.compete.com/2008/01/16/online-video-share-december-2007-year-in-review/>, 2007.
- [29] World's Poor Drive Growth in Global Cell Phone Use. <http://apnews1.iwon.com/article/20090302/D96LSAE80.html>, 2009.
- [30] A. Amir, D. Ponceleon, B. Blanchard, D. Petkovic, S. Srinivasan, and G. Cohen. Using audio time scale modification for video browsing. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, volume 1, pages 3046–3055, 2000.
- [31] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. 2007.
- [32] Xu Cheng. *On the Characteristics and Enhancement of Internet Short Video Sharing*. PhD thesis, Simon Fraser University, 2008.
- [33] Xu Cheng, Cameron Dale, and Jiangchuan Liu. Statistics and Social Network of YouTube Video. In *Proceedings of IEEE IWQoS*, 2008.
- [34] Michael G. Christel, Michael A. Smith, C. Roy Taylor, and David B. Winkler. Evolving video skins into useful multimedia abstractions. In *Proceedings of Conference on Human Factors in Computing Systems (CHI98)*, pages 171–178, 1998.
- [35] Daniel DeMenthon, Vikrant Kobra, and David Doermann. Video Summarization by Curve Simplification. In *Proceedings of ACM Multimedia*, pages 211–218, 1998.

- [36] Frederic Dufaux. Key Frame Selection to Represent a Video. In *Proceedings of IEEE 2000 International Conference on Image Processing*, pages 275–278, 2000.
- [37] Andreas Girgensohn and John Boreczky. Time-constrained Key Frame Selection Technique. In *IEEE International Conference on Multimedia Tools and Applications*, pages 347–358, 2000.
- [38] Riad Hammoud and Roger Mohr. A Probabilistic Framework of Selecting Effective Key Frames from Video Browsing and Indexing. In *Proceedings of International Workshop on Real-Time Image Sequence Analysis*, pages 79–88, 2000.
- [39] Alan Hanjalic and HongJiang Zhang. An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 9, pages 1280–1289, 1999.
- [40] G. W. Heiman, R. J. Leo, G. Leighbody, and K. Bowler. Word intelligibility decrements and the comprehension of time-compressed speech. 40:407–411, 1986.
- [41] Changick Kim and Jenq-Neng Hwang. An Integrated Scheme for Object-based Video Abstraction. In *Proceedings of ACM Multimedia*, pages 303–309, 2001.
- [42] Ying Li, Tong Zhang, and Daniel Tretter. An overview of video abstraction techniques. 2001.
- [43] Jiangchuan Liu and Jianliang Xu. Proxy caching for media streaming over the internet. 2004.
- [44] Jeho Nam and Ahmed H. Tewfik. Video abstract of video. pages 117–122, 1999.
- [45] Jung Hwan Oh, Quan Wen, Sae Hwang, and Jeongkyu Lee. Video abstraction. In Sagarmay Deb, editor, *Video Data Management and Information Retrieval*, pages 321–346. Idea Group Inc. and IRM Press., 2004.
- [46] Nosa Omoigui, Liwei He, Anoop Gupta, Jonathan Grudin, , and Elizabeth Sanocki. Time-compression: System concerns, usage, and benefits. In *Proceedings of ACM Conference on Computer Human Interaction*, pages 136–143, 1999.
- [47] Tim O’Reilly. What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. <http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, 2005.
- [48] Silvia Pfeifer, Rainer Lienhart, Stephan Fischer, and Wolfgang Effelsberg. Abstracting digital movies automatically. *Visual Communication and Image Representation*, 7:345–353, 1996.
- [49] Subhabrata Sent, Jennifer Rexford, and Don Towsley. Proxy prefix caching for multimedia streams. 1999.

- [50] Xinding Sun, M.S. Kankanhalli, Yongwei Zhu, and Jiankang Wu. Content-based Representative Frame Extraction for Digital Video. In *Proceedings of IEEE Multimedia Computing and Systems*, pages 190–193, 1998.
- [51] Candemir Toklu, Shih-Ping Liou, and Madirakshi Das. Videoabstract: A hybrid approach to generate semantically meaningful video summaries. pages 1333–1336, 2000.
- [52] W. Wolf. Key Frame Selection by Motion Analysis. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1228–1231, 1996.
- [53] Evangelos A. Yfantis. An Algorithm for Key-frame Determination in Digital Video. In *Proceedings of 16th ACM Symposium on Applied Computing*, pages 312–314, 2001.
- [54] Hongjiang Zhang, Jianhua Wu, Di Zhong, and Stephen W. Smoliar. An Integrated System For Content-based Video Retrieval and Browsing. *Pattern Recognition*, 30:643–658, 1997.
- [55] Li Zhao, Wei Qi, Stan Z. Li, Shi-Qiang Yang, and HongJiang Zhang. Key-frame Extraction and Shot Retrieval using Nearest Feature Line (NFL). In *Proceedings of ACM Multimedia Workshop*, pages 217–220, 2000.
- [56] Yueting Zhuang, Yong Rui, Thomas S. Huang, and Sharad Mehrotra. Adaptive Key Frame Extraction using Unsupervised Clustering. In *Proceedings of International Conference on Image Processing*, volume 1, pages 866–870, 1998.