LAZYCAP - TEMPLATE-BASED MEDIA CAPTURING

Xian-Sheng Hua,
Microsoft Research Asia
xshua@microsoft.com

ABSTRACT

The rapid adoption of digital cameras and camcorders leads to an explosive growth of personal photos and home video in digital form. This leads to a huge demand for new tools and systems that enables average users to more efficiently and more effectively process, manage, author and share these digital media contents. However, these tasks are always tedious, as well as extremely time consuming, requiring necessary professional skills. To tackle this issue, content analysis based methods has become an interesting and promising research topic. However, due to the difficulties in content analysis after the media data is captured, these methods have insurmountable limitations. In this paper, we propose a novel scheme, Template-Based Media Capturing, which provides a uniform scheme to collect extra information during the process of media capturing in an easy and convenient manner. The extra information captured using LazyCap remarkably facilitates further personal media applications such as management, authoring and sharing. The main idea to accomplish this objective is using uniformly defined *capturing templates*, which are installed on the capturing devices and instantly guide users to capture necessary raw data. LazyCap not only improves the users' experiences on media management, authoring and sharing after capturing, but also assists users to capture high-quality raw media data.

1. INTRODUCTION

The recent rapid adoption of consumer digital photo cameras and video camcorders has redefined the landscape for personal media management, authoring and sharing mechanisms. However, to manage personal media data (photos and videos), in particular video editing, is always a tedious task, as well as extremely time consuming, requiring necessary professional skills. Though there are a couple of commercial media authoring tools available, the situations have not been changed. Many researchers believe that intelligent content analysis based systems are promising to solve these difficulties. However, these systems actually highly depend on the effectiveness and efficiency of the automatic media content analysis algorithms, which still have insurmountable limitations due to the difficulties in bridging the gap between high-level semantics with low-lever features [1].

A promising idea to tackle this difficulty is try to collect more information during the process of media capturing [2]. This information may include GPS data, speech, text, etc. However, few scheme for this purpose is available either in academic or in industry. This paper addresses this issue by introducing an efficient and unified extra-information capturing scheme.

As we know, there are many templates-based text composing tools available such as Microsoft Office (Word, Excel, PowerPoint, Project, Visio, InfoPath, etc.). These editing templates have significantly improved the efficiency of office workers. Inspired by this, in this paper, we propose a content-aware template based media capturing scheme, named *LazyCap*, which provides an uniform scheme to collect extra information during the process of media capturing in an easy and convenient manner. The main idea to accomplish this goal is to use predefined *capturing templates*, which are installed on the capturing devices and instantly guide users to capture necessary raw data. LazyCap can not only remarkably facilitates further applications on personal media data such as management, authoring and sharing, but also assists users to capture higher-quality raw media data.

In fact a few "templates"-like methods for video capturing have already appeared in some portable devices with embedded cameras [6]. However, these "templates" are only for some special video effects, such as making sepia tone and old movie effects, adding frames/borders for photos or videos, or drawing graphics or animations over photos or videos. They really have not tackled the most difficult issues in video or photo capturing and management such as indexing, editing and sharing.

In the proposed LazyCap system, the template scheme for capturing is a uniform approach that helps collect necessary content related information, as well as guide users to obtain relatively higher-quality raw content in an easy and convenient manner. The extra information collected using LazyCap will significantly improve the performance of media indexing, browsing, authoring and sharing.

The reminder of this paper is organized as follows. Section 2 briefly introduces the architecture and work flow of LazyCap. LazyCap capturing template scheme is detailed in Section 3, followed by introducing the prototype system and results based on Pocket PC in Section 4. Finally we conclude the paper in Section 5 with remarks on future works.

2. LAZYCAP OVERVIEW

A typical lifecycle of personal media data may be divided into the following three major steps:

1. **Raw Media Data Acquisition**: Such as capturing videos or photos by camcorders, digital cameras or mobile phones, and then importing them into a computer system. For automatic or semi-automatic media management systems, media content analyses may also be embedded in this step. Actually, LazyCap templates are especially designed for this step, though the results are applied in the later two steps.
2. **Indexing, Browsing and Searching**: In this step, users browse the imported media data based on automatic, semi-automatic or manual indexing results, enjoy the recorded experiences with family members or friends, or search for a particular photo or video clip. With the extra information captured using LazyCap, better experiences are enabled in this step.
3. **Authoring and Sharing**: In this step, the users may select appropriate video clips and/or photos from the imported media library, and then put them onto a timeline to generate an edited storyline. Captions, credits, transitions and video effects may also be added in this step. This is the most time-consuming and tedious step. While with the information collected using LazyCap, as well as multimedia content analysis technologies [1][3], users are able to efficiently accomplish this goal without much effort.

In accordance with this typical lifecycle, we design a personal media capturing, managing, authoring and sharing system with LazyCap support, as illustrated by Figure 1.

Firstly, a user may download LazyCap capturing templates from the Internet or other compatible devices to their capturing device. Before doing capturing with the device, the user may select a desired template, which will be rendered on the monitor of the device, and guide the user to capture necessary data, as well as how to capture good quality materials. Thereafter, the captured media data with the selected template is imported to a personal media library on PC and then the content is analyzed with the help of the information embedded in the template. With these, an integrated system that supports browsing, authoring and sharing that sufficiently uses the template information is provided. If the device has sufficient computation power, certain indexing, browsing, authoring and sharing operations may also applied directly on the device.

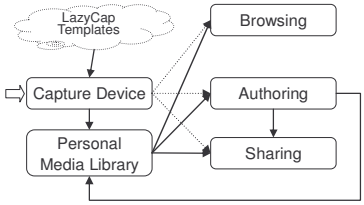

Figure 1. A Personal Media System with LazyCap.

It needs to be mentioned here that, though currently typical capturing devices do not support installing software programs such as LazyCap, we believe more and more smart capturing devices enabling programming, such as Pocket PC and Smart Phone, will be available. As to be shown in Section 4, we will demonstrate a prototype of LazyCap implemented in a Pocket PC environment. Another thing need to be mentioned is that metadata such as GPS data and camera parameters are not the focus of this paper, as this information can be directly embedded into the media stream if they are supported by the corresponding capturing devices.

3. LAZYCAP CAPTURING TEMPLATE

As aforementioned, there are a number of template-based text-editing systems available. LazyCap templates play a similar role while it is for video and photo capturing. Our goal is to provide a uniform template schema for non-professional media capturing, so LazyCap users are able to efficiently and easily use and modify existing templates, design new templates, as well as share their templates with others. In LazyCap, two types of capturing templates are designed, including *Spatial LazyCap Template* (SCT) and *Temporal LazyCap Template* (TCT). As to be detailed, SCT is applied for capturing still images, and also can be regarded as a temporal unit contained within a TCT.

3.1 Spatial LazyCap Template

Spatial template mainly consists of a sketch or sample photograph with text descriptions that demonstrates and specifies how to frame the scenes, people or objects, or how the people or objects pose in front of the camera. Figure 2 shows several sample sketches for spatial templates. The XML description for a multiple-element SCT is shown in Figure 3. The sketch pictures are embedded in the XML file as binary data element.

Figure 2. Sketch/Photograph Samples of Spatial Templates.

When doing capturing, the sketch will be displayed semi-transparently on the monitor, thus the users are able to match the real scene with the sketch to obtain a better-quality photo or video. A good example for the application of SCT is to photograph wedding pictures. These kinds of photographs typically have general poses and positions/layout. Another example is to capture souvenir pictures when visiting well-known scenery or key points of interest. In this case, SCT helps users find the best picture spots and photograph from the best angle of view.

```
<?xml version="1.0" ?>
<LazyCap type="Photo">
<Info>
   <Title> Wedding - Romantic Style 1 </Title>  <Author>John </Author>
   <Email> john@abc.com </Email>  <URL> http://www.abc.com/john</URL>
   <Description> This LazyCap capturing template is for … </Description>
</Info>
<Photo>
   <Title>Wedding - On the Beach</Title>
   <Sketch> <![CDATA[ ..... ]]> </Sketch>
   <Description> The bride stands closely with the groom, and the groom … </Description>
</Photo>
<Photo>
   <Title>Wedding - Kiss Close-Up</Title>
   <Sketch> <![CDATA[ ..... ]]> </Sketch>
   <Text></Text>
   <Speech><![CDATA[ ..... ]]></Speech>
   <Description> The bride groom … </Description>
</Photo>
</LazyCap>
```

Figure 3. XML Description for Multiple-Element SCT.

3.2 Temporal LazyCap Template

TCT defines the temporal structure of the to-be-captured media data, also described by XML. Similar to authoring template [1], the basic temporal unit of TCT is "MSeg" (also called "Slot" when it is rendered – to be detailed later), which stands for "Media Segment". MSeg could be a chapter, a scene or a shot, or whatever temporal segment of a media data. For a specific template, MSegs may be arranged hierarchically or "flatly". All MSegs are sharing the same definition and structure. The default settings for a child MSeg are the settings of its parent MSeg, while child MSeg can has its own settings which have higher priority. A typical hierarchical structure could be "Chapter – Scene", which is similar to a general DVD content menu. In this paper, we will use this structure to describe our idea.

A template should at least contain one chapter (MSeg). A Chapter may contain several Scenes (also MSegs), while a Scene can contain one or more smaller scenes, and so on. There are three types of MSegs, including Video, Photo and Phodeo (stands for combination of photos and videos). Video MSegs will guide users to capture one or more video clips, Photo MSegs will guide users to photograph one or a series of pictures using Spatial LazyCap Template, while Phodeo MSegs mean it contain both Video and Photo sub-MSegs.

3.2.1 TCT Samples

A good example of a real TCT template is similar to the comprehensive shot list for a typical birthday party proposed by Jan Ozer [4], which includes *establishing*, *pre-party*, *guests arriving*, *meeting and greeting*, *environment*, *lighting candles*, *singing happy birthday*, *eating cake*, *giving and opening gifts*, etc. Figure 4 shows the structure of a sample TCT template based on Ozer's shot list (some shots are removed or merged into one scene or chapter). It contains six chapters, including one leader chapter (*Location and Preparation*), four body chapters (*Guests arriving and greeting*, *The party*, *Guests leaving and giving favors*, and *Final words of the birthday child*), and one tail chapter. And the second body chapter contains three scenes.

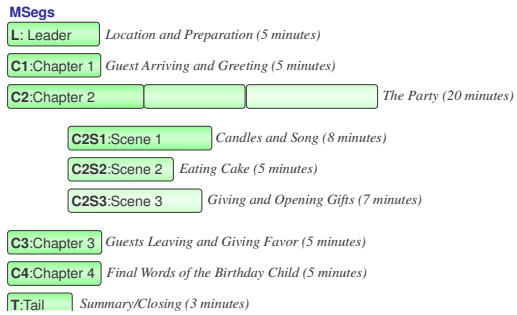

Figure 4. Temporal Structure of a TCT Sample.

Figure 5 shows the XML description of MSeg "Chapter 2" who contains three child MSegs. XML syntax of TCT templates will be introduced in next sub-section.

```
<MSeg level="1" mtype="Video">
  <Title>The Party</Title>
  <MSeg level="2" mtype="Video">
    <Title>Candles and Song</Title>
    <Duration fixed="false">480</Duration>
    <Sketch> <![CDATA[ ..... ]]> </Sketch>
    <Description> This slot captures ... </Description>
    <Text></Text>    <Speech><![CDATA[ ..... ]]></Speech>
  </MSeg>
  <MSeg level="2" mtype="Video">
    <Title>Eating Cake</Title>
    <Duration fixed="true">300</Duration>
    <Sketch> <![CDATA[ ..... ]]> </Sketch>
    <Description> In this slot, you ... </Description>
    <Text></Text> <Speech><![CDATA[ ..... ]]></Speech>
  </MSeg>
  <MSeg level="2" mtype="Video">
    <Title>Giving and Opening Gifts</Title>
    <Duration fixed="true">420</Duration>
    <Sketch> <![CDATA[ ..... ]]> </Sketch>
    <Description> In this slot, capture ... </Description>
    <Text></Text>    <Speech><![CDATA[ ..... ]]></Speech>
  </MSeg>
</MSeg>
```

Figure 5. XML Description of an MSeg with 3 Sub-MSegs.

3.2.2 TCT XML Syntax

In this section, we describe the primary elements and the corresponding syntax of TCT templates. Typically a TCT file contains one root element which includes a sub-element called "TCTInfo", as well as a series of "flat" or hierarchical MSegs. TCTInfo provides the basic information of the TCT, including five basic sub-elements, as listed in Table 1.

Table 1. Sub-Elements of CDTInfo.

Name	Description
Title	The title/name of the TCT template.
Author	The author of this template.
Email	The email of the template author.
URL	The URL of the relevant website.
Description	Description of the template.
Icon	Icon of the TCT (binary data element).

MSeg has two primary attributes and four sub-elements, as listed in Table 2 and Table 3, respectively.

Table 2. Attributes of Element "MSeg".

Name	Description
level	The structure level. The first level is "1". MSeg may contain multiple child MSegs, the level of a child MSeg is the level of its parent MSeg plus 1.
mtype	Specify media type of the source data. May be "Video", "Photo" or "Phodeo" (stands for Photo and Video).

Table 3. Sub-Elements of "MSeg".

Name	Description
Title	The title of the MSeg, e.g., the caption of a chapter or a scene.
Duration	The suggested duration of the raw content to-be-captured in the MSeg. It has only one attribute called "fixed", which specifies whether the duration is fixed, or can be altered.
Sketch	A static picture (graphical or photographic) or animation to show how to capture video/photo for this slot.
Description	Text description for how/what to capture for this slot.
Text	User's text description for the captured data in this slot.
Speech	User's speech description for the captured data.

An extension for the above definition of LazyCap template is to integrate "editing method" as a sub-element, similar to the authoring template in [1]. In the authoring procedure, tThise element will be applied when do authoring after capturing. Below table shows the attributes of "Method", which is the editing method that will be applied on the raw media content that is fed into this slot in the authoring procedure.

3.3 Template UI Rendering

In a capturing device supporting LazyCap, a UI engine will parse the user selected TCT (XML file), and then construct an interface to get users' inputs (to be exact, to capture photos or video clips). As a TST will be regarded as an MSeg of a TCT, so mainly we will present TCT. Since TCT is defined by well-structured XML, a uniform UI (user interface) engine is designed. To be clear, we use the TCT for birthday video introduced in Section 3.2.1 as an example to illustrate how the UI engine works. It should be mentioned that, the UI engine is device-depend, and can also be personalized. That is, we may have different versions of UI engines that provide different experiences for capturing.

Firstly, the UI engine parses the hierarchical structure of the TCT, and draws a series of corresponding "lattices" representing the MSegs in the TCT as a timeline (a sketch map is illustrated in Figure 6 and a real UI is shown in Figure 7). The slot titles will be displayed in the corresponding lattices, and the details (title, duration, description, etc.) of a certain slot (i.e., MSeg, such as L, C1, C2, C2S1, etc.) will be displayed in a window when the corresponding slot is clicked or got focused. The static or animation sketch will be played if users press the button "Show". The users are able to adjust some of the parameters (say, the duration and caption) for a certain MSeg in the detail window. Slot adding, deleting, copying, pasting and moving are also supported in a manner similar to typical editing software. These functionalities may also be dependent on the capability of the capturing devices, such as the size of the monitor and the power of the CPU. During the process of capturing, users are abled to find a specific lattice and then begin to do capturing.

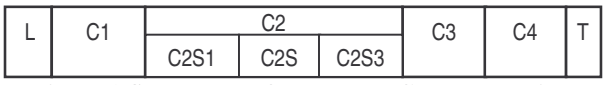

Figure 6. Sketch Map of Rendered TCT by UI Engine.

After capturing raw media content with a certain TCT template, the result file, named Captured Content File (CCF), is in the same form as TCT file, except the "Text" and "Speech" elements may filled with data, an element called "Content" into element "MSeg" (specify the links to the raw media data). CCF files provide all the information for further applications, such as browsing, authoring and sharing.

4. RESULTS AND FURTHER APPLICATIONS

As aforementioned, though currently it is difficult to install software into typical mainstream capturing devices, we believe more and more smart capturing devices enabling programming like Pocket PC and Smart Phone will be available. In this section, we will show a prototype of LazyCap based on Pocket PC in Windows Mobile environment, and demonstrate the LazyCap framework and the concepts of template based media capturing, as well as introduce further applications with the support of LazyCap.

4.1 Typical Capturing Process with LazyCap

To demonstrate how LazyCap works, we will capture a birthday party video using a birthday LazyCap template. The following features will be demonstrated:

1. Download and install TCT package.
2. Modify selected TCT on the rendered timeline.
3. Capture raw media data into TCT slots.
4. Download CCF and media data into PC.

Firstly we check whether there is any suitable TCT template in the TCT template library in the device – here we take Pocket PC as an example device as currently only PDA-like capturing devices can have software installed. There are several built-in TCTs, but we try to search for a new one over the Internet (simulated by an internal website currently). While a desired TCT titled *Elegant Birthday Party* is found on the web, we download and imported it into LazyCap. We choose the TCT in LazyCap then it is rendered on the device screen. The temporal structure of this TCT is the same as the example we showed in Section 3.2.1, which contains one leader chapter, three body chapters and one tail.

Next, after right clicking on the lattice representing the tail chapter and on the popup window, we change the duration of tail chapter from 3 minute into 5 minutes. The modified TCT can be saved (or saved as a separate TCT) in the TCT library for later use.

Then we begin to capturing a real birthday party under the guidance of the selected templates. Figure 7(a) shows the interface of LazyCap, and the first slot is selected and then begins to capture. Figure 7(b) shows the TCT with captured content.

(a) (b)

Figure 7. LazyCap on Pocket PC.

Finally, we download the captured content with the CCF file into desktop PC, and further applications with this content will be introduced in next sub-section.

4.2 Further Applications

As the raw content is well organized under the capturing templates, we are able to do rich and convenient content-based browsing and authoring without much extra effort, similar to the video management and browsing system proposed in [5].

As the MSeg titles the temporal structure can be rendered on the desktop applications, the users are able to find an appropriate clip conveniently and efficiently. And, with these metadata, automatic content analysis algorithms may be more efficient (which will be our future work).

Furthermore, the temporal structure, MSeg titles and descriptions in the CCF file facilitate us to generate the edited results rapidly, which are more compelling as the storylines are well-preserved, compared with previous automatic video editing results [1][3].

5. CONCLUSION AND DISCUSSION

We have proposed a novel scheme for personal photo and video capturing, named LazyCap, which enables rich content capturing with the help of predefined capturing templates. It is observed that higher-quality raw content will be obtained when using LazyCap compared with traditional capturing scheme, because the users know what to capture and how to capture with the help of LazyCap. That is, LazyCap not only improves the users' experiences on media management, authoring and sharing after capturing, but also assists users to capture higher-quality raw media data.

An interesting extension to LazyCap is, tour guide and capturing guide at the same time. As LazyCap instantly guides the capturing process, actually, it is also can be used as a tour guide. Tour information can be embedded in the description element, or we may also add more tags to embed richer information, such as photographs, links, music, audio introduction, and so on. For example, a LazyCap template for Beijing City Tour may contain the tour information of the main attractions of the city, including transportation, shopping, schedule, ticket, main photo spots, and so on. And the capturing functions of course are also embedded. Tour organizations or average users may design these kinds of templates and shared through the Internet. With this support, both touring and media capturing will be more efficient.

Another functionality that LazyCap would have is data syncing. As currently the quality of the photos and videos captured by these devices are not good enough, people may carry their high quality capturing devices such as DCs, camcorders and so on. While the authors will use the LazyCap system at the same time, we can sync the timestamps on these devices, so the high quality media data can also be put into corresponding slots in the template, thus extra metadata are also valid for these media data. Future work will be to implement the above extensions, as well as to improve the performance of automatic content analysis algorithms with the information captured under LazyCap. A comprehensive user study is also desirable to prove that LazyCap help users capture higher-quality media content.

6. REFERENCES

[1] Hua, X.-S., et al. LazyCut - Content-Aware Template-Based Media Authoring. ACM Multimedia 2005..

[2] Boll, S., et al. Between Context-Aware Media Capture and Multimedia Content Analysis. *Panel of ACM Multimedia 2004*.

[3] Hua, X.-S., Lu, L., and Zhang, H.-J. AVE – Automated Home Video Editing. *ACM Multimedia 2003*. Berkeley, USA, Nov 2003.

[4] Ozer, Jan. Scripting the Birthday Party Video. *Doceo Publishing*. http://www.doceo.com/bday script.htm.

[5] Wang, Y, et al. MyVideos - A system for home video management. In *Proceeding of ACM Multimedia 2002*.

[6] Xda II - The mobile entertainment system. http://www.my-xda.com/.