Document Finder

Qiong Liu, Patrick Chiu, Lynn Wilcox
FX Palo Alto Laboratory
3400 Hillview Ave. Bldg. 4
Palo Alto, CA 94304, USA
1-650-813-6957
liu@fxpal.com

ABSTRACT

This demo introduces a tool for accessing an e-document by capturing one or more images of a real object or document hardcopy. This tool is useful when a file name or location of the file is unknown or unclear. It can save field workers and office workers from remembering/exploring numerous directories and file names. Frequently, it can convert tedious keyboard typing in a search box to a simple camera click. Additionally, when a remote collaborator cannot clearly see an object or a document hardcopy through remote collaboration cameras, this tool can be used to automatically retrieve and send the original e-document to a remote screen or printer.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Interaction Styles.

General Terms

Algorithms, Performance, Design, Experimentation.

Keywords

Camera enabled user interface, document retrieval, image local features.

1. INTRODUCTION

Even though paperless office was proposed a long time ago [6], paper is still a preferred media in our daily lives. From year 1983 to 2003, the world production and consumption of printing and writing paper were increased from 45,224,300 tons/year to 97,199,494 tons/year [4]. With piles of paper documents in our office, finding original or related files of these paper documents bothers many office workers. On the other hand, field workers have to handle thousands of real objects, such as machine parts, and finding documents related to a real object bothers many field workers.

With current technology, many office workers and field workers use interfaces shown on the left of Figure 1 to find the original or related documents of an object or document hardcopy. With this interface, a person needs to explore various directories and files to find the target documents. When a user can clearly remember various directory names, file names, and the associations between objects and file names, this interface is very useful. However, as the number of directories, files, and physical objects grows too big for a person to clearly remember them, using this interface for document access will be a tedious and time consuming task.

To reduce people's document access effort, many workers turn to interface shown on the right of Figure 1. This kind of interface may save people's efforts by restricting the search within files related to user typed keywords. However, users of this interface still need to do keywords' selections and typing for the file search. This is not very convenient for new workers who do not know the name of a machine part but need document help the most. Text-based document search may also become extremely difficult when it is difficult to find keywords in a document hardcopy. For example, a field worker does not know the keywords in a foreign language when he/she wants to see figure illustrations in a foreign document. Even if keywords are selected correctly, simple one-word or two-word input may frequently lead to a large number of results that a person cannot easily handle.

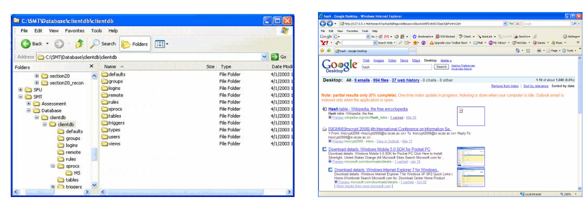

Figure 1. Two typical interfaces for file retrieval.

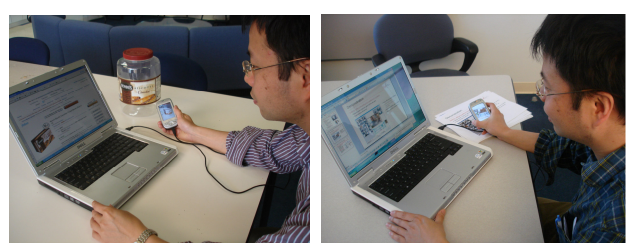

Figure 2. Some usage examples of the proposed system.
Left: get more information of a 3D object (e.g. cookie jar).
Right: get an original file by capturing a hardcopy.

In this demo, we will show a tool that can facilitate e-document access with a photo of a document hardcopy or a real object. With this tool, a user can access a file by clicking a camera or cell phone connected to a computer. Ideally, this type of interface can save people from boring directory exploration. It can also save people's typing effort and quickly feed a computer with a large amount of document related key features. Figure 2 shows two

MM'08, October 26–31, 2008, Vancouver, British Columbia, Canada.
ACM 978-1-60558-303-7/08/10.

usage scenarios where a user accesses an e-document by capturing an image of a 3D object (cookie jar in the example) or a document hardcopy. Beyond usage for regular desktop systems, this interface is also useful for cell phones whose keyboards are too tiny for comfortable typing. It is also useful for tele-collaboration, when collaboration cameras cannot capture a hardcopy clearly.

To make the proposed system working, we have to face several challenges. First, we need to find robust features to represent captured object or document under various camera setups and capture conditions. Second, the feature search in this algorithm should be fast enough to handle a large amount of files. Third, it is better that the algorithm can be language independent so that the system can support collaborations among different language groups. With these challenges in mind, we use low level image features directly to overcome the language problem. Additionally, we co-designed the feature extraction and search algorithm for improving feature robustness and search speed at the same time.

2. SYSTEM OVERVIEW

To make the system work for both programmable camera (e.g. camera-equipped cell phone) and non-programmable camera (e.g. camera directly connected to a PC), we separate the software into three modules: mobile-client module, service-proxy module, and file-manipulation module.

The mobile-client module is a client application that is used to collect input images for a system. If the system camera is installed on a cell phone, this application will be deployed on the cell phone to respond to user's image capture request. If the system camera is connected to a tele-collaboration PC or a server PC, the software module will be deployed to the PC connected to a camera. By separating this module out of the main system, it will be easy to scale up the system with multiple cameras connected to multiple computers or cell phones.

The service-proxy module is a web application that can talk to the document-manipulation module. It provides a web service for various mobile client modules to talk to the document-manipulation module. More specifically, it analyses one or more captured images, extracts low level image features, and search the original e-document based on these extracted features. By modifying proxy services for different document systems, our tool can be used for different document systems with the same hardware.

The file manipulation module is a service application module that has the authority to access those original e-documents and related applications. It normally resides on a user's document access machine. To assist the deployment of this service, this module can be started directly from a web site.

3. DOCUMENT FINDING

To reduce feature distortions caused by various image transformations and object occlusions, our system computes image local invariant features, and retrieves documents according to the number of local feature matches. Since object recognition accuracy and recognition time cost are two most important factors in our system, and both of them are heavily influenced by the feature extraction and search algorithm, improving feature extraction and search is our main focus for building the system.

SIFT feature [1] is well recognized as a robust feature set for general images. Many people have tried this feature set for natural images. To test the effectiveness of this feature on document recognition, we converted 400 pages (100 papers) of the ICME 2006 proceeding into images and saved them in our search repository. By randomly scale (0.5~2) and rotate (0º~360º) these images, we generated 1200 test images. Through matching these 1200 images against the search repository using SIFT feature, we can find original images of these 1200 input images 100% correctly.

Even though the SIFT feature is good for object and document representation, an effective search algorithm is needed for using this high dimensional feature on a large data set. kd-tree [2] is a popular data structure for fast data point search. However, directly feed high dimensional data to a kd-tree construction algorithm may lead to a very unbalanced tree and dramatically increase the search time. To overcome this problem, we break each SIFT descriptor into multiple sub-descriptors and build a kd tree for each low dimensional sub-descriptor set. In this way, we can get a kd-tree forest of sub-descriptors. Besides these approaches, we also used Approximate Nearest Neighbour approach [3] to further speedup each kd-tree search. To fuse the result generated by each kd-tree, each kd-tree is assigned one vote for each input, and the document which gets the maximum number of votes is selected as the output document. By using sub-descriptors and kd-tree forest, we can overcome the problems caused by dimensionality increase. Moreover, because those sub-descriptors are more localized than a full descriptor and the decision is made through voting, they are more immune to occlusion, background interferences near the boundary of an object, and other localized noise interference.

We tried the modified algorithm with 1000 small objects in the ALOI (Amsterdam Library of Object Images) library [5]. Compared with a typical SIFT & kd-tree based feature search, the proposed approach only uses about $1/260^{th}$ time for a similar accuracy. We believe that the time saving will become even bigger when the dataset is larger. This computation-time speedup strongly supports the effectiveness of our proposed approach. It also makes it possible for us to show a complicated demo on a laptop.

4. REFERENCES

[1] D.G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal on Computer Vision, vol. 60, pp. 91-110, 2004.

[2] A. Kaufmann et al., "kd-tree", http link: http://en.wikipedia.org/wiki/Kd-tree

[3] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman and A. Wu, An optimal algorithm for approximate nearest neighbor searching, *Journal of the ACM*, 45(6):891-923, 1998. (Also, in SODA 1994.)

[4] G. Sciadas, "Our Lives in Digital Times", *Connectedness Series*, catalogue no. 56F000MIE – no. 14, Statistics Canada.

[5] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, The Amsterdam library of object images, *Int. J. Comput. Vision*, 61(1), 103-112, January, 2005.

[6] Business Week (1975), "The Office of the Future", No. 2387, June 30, pp 48-70.