Automatic Management of Large Images Databases

Martin Klinkigt[*,**], Heiko Maus[**], Koichi Kise[*], Andreas Dengel[**]

[*] *Graduate School of Engineering, Osaka Prefecture University, Japan*
[**] *German Research Center for Artificial Intelligence DFKI GmbH, Kaiserslautern, Germany*
E-mail: klinkigt@m.cs.osakafu-u.ac.jp, Heiko.Maus@dfki.de, kise@cs.osakafu-u.ac.jp, Andreas.Dengel@dfki.de

Abstract

With an increasing amount of images a user would like to keep and share the desire for an adequate management for these images raises. The normal folder structure which is available on nowadays computers can fulfill this task only non satisfying. Often an image contains more individual information than can be expressed by a hierarchical tree. We introduce annotation techniques to support the user in this organization. Furthermore, we analyze how a computer system could do this organization automatically to keep the amount of work for the user as low as possible.

1. Introduction

With computers the user has the possibility to organize his images. He can choose a proper file name to describe the content of an image and can gather images into one folder to create collections. By using subfolders he can create a hierarchical tree. Such trees are mainly organized by time and events.

Problems mainly arise if the user would like to place an image in more than one folder. This is often the case for images from a social event. Here, a multidimensional categorization as a collection concerning the event is needed, as well as a collection of the images showing one specific person.

A multidimensional categorization would be better for organizing images. Here the user can browse his image collection by different motivations. If he searches for images from a specific event, his motivation would be the date, should interest be in images of a person he can abstract from the specific date by choosing this person as criteria for browsing or searching. The upcoming semantic desktop systems provide such a solution [1]. Here, all these additional semantic relations could be expressed and could be used at the time of browsing or searching. The drawback of such systems is, that it requires a certain amount of time to provide this information. Therefore, we analyzed how the results of computer vision in object detection could be used, to get this information automatically. The resulting system can support the user in his image management, while also reducing the work for him.

2. Semantic Desktop

In the Semantic Desktop the organization of the information objects is abstracted from the folder structure. The main idea is, to use statements about files to provide additional information. These statements can be compared to simple sentences in natural language.

2.1 Idea of Semantic Desktop for Images

For images the content could be described for example as follows:

- Image *A* shows *Paul*
- Image *A* was taken on *2009-04-20*
- Image *A* was taken in *Tokyo*
- Image *B* shows *Paul*
- *Paul* is a *Person*
- *Tokyo* is a *City*

If the user browses or searches for *Paul*, he will see image *A* and *B* and for images taken in *Tokyo* he will see image *A*.

2.2 Integration

To supply all these statements costs a lot of time and would be necessary for every image to be fruitful. Leaving such a task to the user would not be rational, since an image can contain a lot of different information like objects or persons and it can also happen that the user does not know which objects will be interesting for a later search.

Therefore, we provide a system which automatically detects some content of an image. We separate between explicit and implicit statements. Explicit ones are provided by the user and implicit are generated by the system, which can be used as recommendations. The user can accept these statements, rejects them as completely wrong or ignore them as “do not show again”. Beside the explicit statements these implicit statements are then

ISSN 1975-4736 ©MITA 2009

additionally used to browse or search in the image collection.

3. Object Detection

In computer vision the field of object detection is divided into two main parts, specific and generic object detection. While specific objects detection concern the problem to detect one specific entity like *this red sport* car, the generic object detection wants to identify the class of the object for example *a* car. We mainly focus on specific object detection since good and fast algorithms are available.

3.1 Learning Phase

To describe the content of an image, interest points are detected. These points are then characterized with the help of a so-called feature descriptor. We will use PCA-SIFT [2] which has a good performance. In a last step these features are stored in a database.

At the time of learning it is important to take care, that all images are provided with some information about their content. To automate this process we make use of Wikipedia and DBpedia. Wikipedia is a free encyclopedia which provides for nearly every interesting object some images. The DBpedia project reprocesses the information from Wikipedia and makes them available in a well defined structure. The database of DBpedia can be easily accessed and processed by a computer. Figure 1 shows these steps. We have some objects of interest and get the corresponding identifiers from DBpedia. From DBpedia we now get the information about this object and its Wikipedia page. Now we analyze the images and store them in our database.

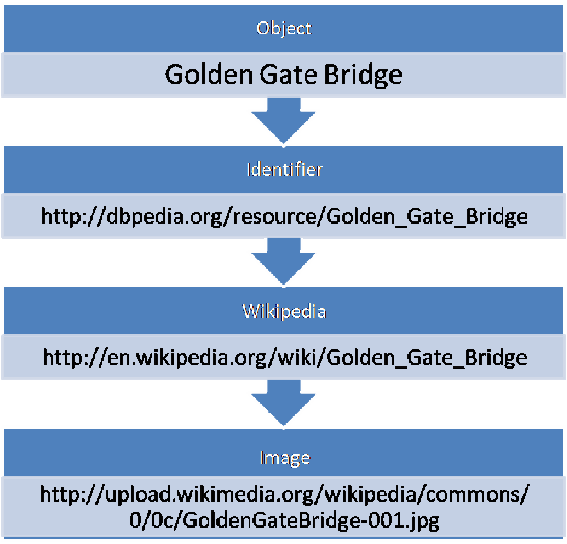
Object
Golden Gate Bridge
Identifier
http://dbpedia.org/resource/Golden_Gate_Bridge
Wikipedia
http://en.wikipedia.org/wiki/Golden_Gate_Bridge
Image
http://upload.wikimedia.org/wikipedia/commons/
0/0c/GoldenGateBridge-001.jpg

Figure 1: Steps of database creation.

3.2 Recognition Phase

For every image in the collection, we perform a search in the database. Therefore, each image is also characterized by a feature descriptor and compared with the features in the database. This will return a collection of objects for which we then use the provided information from the learning phase.

In this recognition phase performance becomes important. If we assume working on a several hundreds or even thousands of images, the computation for one image should not take too long (< 10sec), since we would then need several hours to analyze all images. The critical part of performance is the search for the best result in the database. We use the approach Kise et al. explained in [3]. The idea is to use a hashing technique to find a nearest neighbor.

3.3 Results

The images from Wikipedia often have a good quality which makes the system reliable for this automatic process. However, sometimes the images are of low quality or do not show the correct object. In such cases the system will fail to detect this object in an image of the user. This is no serious problem, since the user can always find his image again with the help of the remaining categorizations.

4. Conclusions

We have proposed a more suited approach to manage an image collection. We used the possibilities of the Semantic Desktop for multiple categorizations and showed how this organization could be done automatically by a computer. Furthermore, we connect the system with Wikipedia to learn interesting objects.

The overall system will provide a personal image collection with a support of easy categorization of images with the help of object detection.

（MITA 論文の最終版には下記の謝辞を入れて下さい）

5. References

[1] Leo Sauermann, Ansgar Bernardi, Andreas Dengel. Overview and Outlook on the Semantic Desktop, Proceedings of the 1st Workshop on The Semantic Desktop at the ISWC 2005 Conference, 2005

[2] Yan Ke, Rahul Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors, Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2004

[3] K. Kise and K. Noguchi and M. Iwamura. Simple Representation and Approximate Search of Feature Vectors for Large-Scale Object Recognition, BMVC, Vol. 1 pp. 182-191, 2007