Region-based Annotation Tool using Partition Trees

Xavier Giró, Neus Camps and Ferran Marqués
Technical University of Catalonia (UPC)
Email: {xgiro, alneus, ferran}@gps.tsc.upc.edu

***Abstract*—This paper presents an annotation tool for the manual and region-based annotation of still images. The selection of regions is achieved by navigating through a Partition Tree, a data structure that offers a multiscale representation of the image. The user interface provides a framework for the annotation of both atomic and composite semantic classes and generates an MPEG-7 XML compliant file.**

I. Introduction

The annotation of visual content has become a topic of interest in scientific community due to the recent advances in pattern recognition applied on computer vision systems. Most of automatic detection solutions rely on a training phase during which classifiers are trained with a corpus of manual annotated data. While traditionally visual content has been annotated at a large scale (whole image or video files), more recent region-based indexing techniques require a finer resolution to train those detectors that are able to localize objects in a specific part of the image or video. To do so, new initiatives have been developed, such as the project LabelMe [1] that provides a web interface to annotate polygonal regions by the definition of their vertices. A previous analysis of the visual data can help such a region-based annotation process, as the tablet PC techniques proposed in the Videotater project [2], where users refine and tag video segments that have been automatically delimited. In the case of still images, a hierarchical organization of partitions generated from the image can facilitate a multiresolution navigation in the image, as presented in [3]. The present paper proposes a solution to help human users to semantically annotate regions from a still image by combining the pixel-based visualization with an underlying region-based navigation system. The main contribution of this paper is the application of Partition Trees as a navigation framework to select and annotate regions as instances of atomic or composite semantic classes. The resulting tool is implemented in Java and uses MPEG-7 XML compliant files written by Java binding classes automatically generated by Apache XMLBeans and the MPEG-7 schemas.

II. Input and Output Data

The presented tool uses two main sources of information: visual data associated to the still image, and a semantic ontology from where objects or their parts are selected. The annotation tool allows the user to generate an output file describing which regions of the input image represent instances of semantic classes defined in the ontology. Moreover, the same interface includes tools for the creation and edition of the semantic ontology.

A. Visual Input

Three different types of data related to the visual content are necessary for the annotation tool to work. Firstly, the still image itself; that is, the actual pixels in any standard format (JPG, PNG...). Secondly, an initial partition of the still image previously generated through a segmentation process. Such data is represented by another image whose pixel values correspond to region labels. Finally, the annotation tool requires an additional set of regions defined as combinations of the regions in the initial partition and structured as a Partition Tree (PT). The PT leaves correspond to the regions in the initial partition, while the root represents the whole image. The tree structure guarantees that there exists a single path between any leaf and the root. The access to the three sources of data is achieved through a single MPEG-7 XML file that contains the parent-child relations among the nodes in the PT as well as references to the files with the input image and initial partition.

B. Semantic Input

The annotated concepts are uniquely identified and structured in a library of semantic classes. There exist two types of classes: atomic and composite. While the visual representation of *atomic* classes cannot be further splitted from a semantic point of view, *composite* classes are defined as a combination of parts visually distinguishable and semantically meaningful, that is, as a combination of other classes. The decomposition process can be iterated to define semantic hierarchies under the form of ontologies. The current tool version reads the semantic data from an MPEG-7 XML compliant file and allows its edition through a graphical user interface.

C. Annotation Output

The output data is a description of which regions or combinations of regions from the initial partition represent instances of those classes included in the semantic library. In the case of atomic classes, each annotated instance includes the list of associated regions from the initial partition. On the other hand, annotated instances of composite classes refer to the annotated instances of their parts. Such information is written in an MPEG-7 XML compliant file which also includes references to the input image and initial partition files.

III. User Interface

The graphical user interface is divided in three parts: on the left, the ontology panel for the selection and management of classes and, below it, the instances panel for the review of annotated data. Occupying most of the window, the image panel for the selection and visual display of regions.

A. *Ontology Panel*

The ontology panel is located shows the semantic classes defined in the library. When the user focus the interest in one of them, the image panel highlights the regions associated to those instances that have been previously annotated, if any. The ontology panel includes an edition tool that allows the definition of new classes as well as the edition of existing ones. Users can add and delete classes, but also add or remove parts from composite classes. The addition of parts is always based on those classes already defined, so when a brand new part is to be added to a composite class, it must be firstly defined as an atomic class and later added to the composite class. The editor allows multiple instances of the same part in the definition of a composite class, but prevents the creation of semantic cycles, that is, an ancestor class cannot be a child of its descendants.

B. *Image Panel*

The image panel is used to select those regions containing instances of the class selected on the ontology panel. After reading from disk the input visual data described in Section II-A, the application shows the image on screen and activates a region-based navigation system through the PT. Users can choose any pixel from the image as an anchor point by left-clicking on it, an action that toggles the state of the associated PT leaf between selected and unselected. The new state can be propagated to upper PT nodes by using the mouse wheel. The selected nodes are visualized by making their associated pixels transparent on an overlay mask of selectable colour and transparency.

C. *Instances Window*

The instances panel shows in a tree the annotated instances of the class selected in the ontology panel. By selecting any of them, their associated regions are shown in the image panel.

IV. Annotation Cycle

The process of annotation follows a cycle that can be repeated as many times as semantic instances in the image. Firstly, the user must select a class from the ontology panel by left-clicking on it or using the mouse wheel. In case of selecting an atomic class, the next step is to choose which regions instantiate it, while in the case of composite classes, a new annotation cycle is initiated for each of its parts.

The selection of regions starts with the first left-click on the annotation panel and it is achieved with the tools described in Subsection III-B. As a result of the process, a set of regions from the initial partition is selected and must be validated with a right-click on the mouse. Afterwards, a new class can be selected on the ontology panel, or another instance of the same class annotated by selecting another set of regions. In the case of composite classes, the annotation can also be performed by selecting the object parts from the previously annotated instances selected on the instances panel.

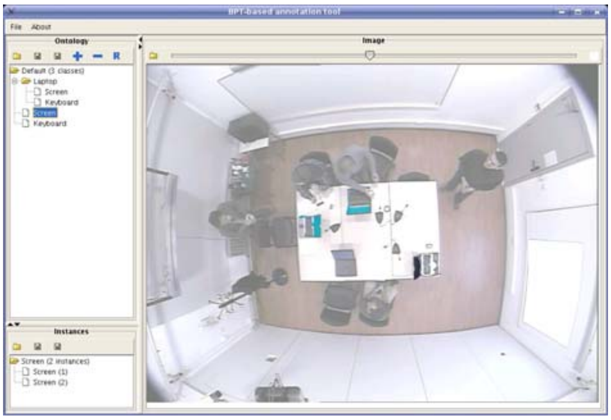

Fig. 1. Snapshot of the user interface

V. Example

In our case, the Binary Partition Tree (BPT) [4] image representation was selected to generate the PT. In the BPT, intermediate nodes are iteratively defined by merging the two adjacent regions which minimize a certain fusion criterion (eg. based on colour). Fig.1 shows a snapshot of the tool where the image contains four instances of the semantic class *Laptop*. The instances panel shows that two screens have already been annotated while a third is being selected on the image panel.

VI. Conclusions

The presented technology provides a user interface for the region-based annotation of still images. Its main advantages and limitations come from the underlying navigation framework provided by the PT. If the PT creation criterion adjusts to the semantic interpretation of the image, the selection of semantic relevant regions is fast and straightforward. If this is not the case, the PT will not provide much help, but the user is still able to annotate semantic instances by left-clicking on regions from the initial partition that represent them. Further steps will focus in the automatic selection of a semantically relevant PT node after a left-click on the annotation panel, to further help users reducing the number of required actions to select the semantic instance.

Acknowledgment

This work was partially founded by the European Commission NoE FP6-507752 MUSCLE and the Spanish projects CENIT-2007-1012 i3media and TEC2007-66858/TCM PROVEC.

References

[1] B.C. Russell et al, “Labelme: a database and web-tool for image annotation,” *MIT Artificial Intelligence Memo*, vol. 25, September, 2005.

[2] D. Diakopoulous and I. Essa, “Videotater: an approach for pen-based digital video segmentation and tagging,” in *ACM Symp. on User Interface Software and Technology*, pp. 221–224, Montreux, October, 2006.

[3] F. Marques et al, “Partition-based image representation as basis for user-assisted segmentation,” in *Proc. of the Int. Conference on Image Processing (ICIP)*, vol. 1, , pp. 312–315, Vancouver, October, 2000.

[4] P. Salembier and L. Garrido, “Binary partition tree as an efficient representation for image processing, segmentation and information retrieval,” *IEEE Trans. on Image Processing*, vol. 9, no. 4, pp. 561–576, April, 2000.