FI MU

Faculty of Informatics
Masaryk University Brno

Detection and Annotation of Graphical Objects in Raster Images within the GATE Project

by

Ivan Kopeček
Radek Ošlejšek
Jaromír Plhák
Fedor Tiršel

FI MU Report Series FIMU-RS-2008-10

December 2008

Publications in the FI MU Report Series are in general accessible via WWW:

http://www.fi.muni.cz/reports/

Further information can be obtained by contacting:

Faculty of Informatics
Masaryk University
Botanická 68a
602 00 Brno
Czech Republic

Detection and Annotation of Graphical Objects in Raster Images within the GATE Project[*]

Ivan Kopeček Radek Ošlejšek Jaromír Plhák
Fedor Tiršel
Faculty of Informatics, Masaryk University,
Botanická 68a, 602 00 Brno,
Czech Republic
{kopecek, oslejsek, xplhak, xtirsel}@fi.muni.cz

December 16, 2008

Abstract

This report presents a brief outline of the GATE (= Graphics Accessible to Everyone) project architecture and an analysis of some problems and approaches connected to the detection and annotation of graphical objects in raster images. It also mentions some experiments concerning the object detection and annotation. Some examples and illustrations are presented as well.

1 Introduction

Computer graphics accessibility is an important issue of the current assistive technologies. In this area, some papers have been published concerning tactile devices and their applications for the blind people, see e.g. [1, 2, 3, 4]. Some other works analyze to what extent the sound information can be used to investigate graphical objects, see e.g. [5, 6]. SVG graphical format has been recently often exploited to permit the access to object-oriented graphical information for vector graphics [7, 8, 9].

[*] Supported by the Grant Agency of Czech Republic under the Grant 201/07/0881.

An approach, that utilizes SVG format also for raster graphics and enables the user to obtain the required information in both verbal and non-verbal form, having simultaneously full control in the way in which they investigate the picture, is presented in [10].

The analysis of the related problems shows that it is advantageous to support picture annotation by a suitable ontology. It enables the system to help the user to annotate the picture as easily and efficiently as possible and simultaneously it prevents chaos in terminology and keeps the semantic hierarchy consistent. The ontology is coupling the information about the actual picture with the information about the objects that appear in the picture with their description in the real world, making the annotators' work easier by automatically supplying them with the attributes of the described objects.

2 GATE Dialogue System

The *GATE* (= Graphics Accessible To Everyone, see [10]) project aims to achieve the following goals. First, to develop utilities deployed for easy picture annotation. Second, to provide the blind users with support for exploring ("viewing") pictures. And finally, to develop a system for generating images by means of dialogue conversation. Let us briefly describe its basic modules.

The *ANNOTATOR* module supports image navigation and is closely connected to the graphical ontology. Its basic task is to inform the user about the graphical content in a non-visual way. The *GATE* system provides two basic ways doing so - verbally and by means of sound. There are two basic tools supporting this communication: *What-Where Language* and *Recursive Navigation Grid*. *What-Where Language*, *WWL*, is a simple fragment of English. Each sentence of this language has the form of "WHAT is WHERE" or "WHERE is WHAT". It enables the user to ask simple questions about the objects in the scene and their position (e.g. "Where is the tower?", "What is in the middle?", "What is in the background?").

Recursive Navigation Grid [11, 12], *RNG*, is the navigation backbone of the system, dividing the picture space into nine identical rectangular sectors. Each sector is subdivided in the same way recursively. This enables the user to investigate points in the region with chosen precision.

Verbal Information Module, VIM, controls the dialogue conversation including the *WWL* communication. Possible misunderstandings in the communication are solved by *VIM* by invoking dialogue repairing strategies.

GUIDE module provides verbal information, exploiting both the pieces of information obtained by tagging the picture as well as the pieces of information gained directly from the picture. *GUIDE* also provides *EXPLORER* with relevant information and cooperates with *VIM* and *RNG*.

The communication of *EXPLORER* is not primarily verbal, but analogue. It is controlled by mouse, digitizer, or numerical keyboard. The output sound information is also primarily non-verbal. The *RNG* module is exploited for navigation. The pieces of information that are related to the place or object are both verbal and non-verbal, allowing the user to perform a quick dynamic exploration of the non-annotated details of the picture. The information about colors is provided by a procedure which is based on the sound representation of colors. The basic idea of this representation of color assumes the sound information to be a combination of special sounds assigned to the primary colors of a suitable color model.

3 Dialogue Communication via DialogueStep Procedure

The dialogue communication within the *GATE* project is supported by the DialogueStep procedure. This routine enables an easy programming and managing the dialogue strategies. It is implemented in Java and based on some principles derived from the VoiceXML standard [13]. This procedure process the user's entries using predefined grammars and responds to the most common events that appear during the dialogue.

DialogueStep Java class analyzes the user's reply and parses the relevant values into separate variables. Correct initiation of the DialogueStep class assumes prompt and grammar files to be specified. These files have a standardized structure and determine the behaviour of the entire communication in the dialogue step until the final output is obtained. During the dialogue with the user, the system is capable to react to the events like "**no match**" (the system is not capable to analyze the input, the user is asked to repeat the input), "**help**" (more detailed information for the user is prompted), "**repeat**" (the last information is prompted once again), "**empty input**" (if the user is allowed to left the variable empty, then a variable filled with empty string is returned - otherwise the user is asked to repeat the input), "**no input**" (the user is not responding for a specified time period) and others.

We identify various types of grammars in the DialogueStep class. Some of them are common for all dialogues - they store all phrases that automatically generate the event. The other grammars are unique to each communication and they are stored in different data format according to the grammar rules structure.

The grammars contain two different regular expressions for each defined variable. These regular expressions are applied to the type verification that represents one of the implemented supply functions. The system inspects whether the given output, processed from the users input, matches the first regular expression. This expression covers all common entries for the current variable (e.g. nine numbers for cell phone number variable). If the output is successfully tested in the regular expression test, it is returned to the system as a correct result. Otherwise, we inspect the output against the second regular expression that covers all other, possibly correct, inputs (e.g. thirteen numbers beginning with ′00′ or ′+′ and twelve numbers for the cell phone number variable). If the output is correct, the user is asked to confirm the output in the yes-no dialogue. Otherwise, the user is asked to repeat the input.

Correction of the typing errors is another implemented function. The system is able to process the user′s input correctly even if one letter is missing or replaced by another one or when two adjacent letters are swapped.

4 Approaches to Objects Detection in Raster Images

4.1 Segmentation

Segmentation is a process of partitioning a digital raster image into a set of non-overlapping regions, corresponding to the image scene composition [14]. We assume that the result of the segmentation is presented as a vector graphics structure. Typically, the image preprocessing consist of the following phases:

- **Noise reduction** - suppression of different types of the noise presented in the image.

- **Adaptive filtering** - using filters that self-adjusts its transfer function according to an optimizing algorithm.

- **Anisotropic filtering** - enhancing the image quality of the textures on surfaces that are far away and steeply angled with respect to the point of view.
- **Impact of reflection** - corrections such as brightness and contrast adjustments.

The commonly used segmentation algorithms are described in what follows.

4.1.1 Edge Detection Methods

The edge detection methods are oriented to the detection of the significant edges in the image. Local edges are detected by edge detectors (e.g. Canny edge detection, see [15]) based on the difference of the pixels' color. Hough transform [16] finds imperfect instances of objects within a certain class of shapes by a voting procedure. The voting procedure uses object candidates that are obtained as local maxima in the so-called accumulator space. This space is explicitly constructed by the algorithm for computing the Hough transform.

4.1.2 Region Growing Methods

In comparison to the edge detection methods, these methods can better handle the noise in the analyzed image. Homogeneity of the regions is the main segmentation criterion for the region detection (gray levels, color, texture, shape, etc.). Split and merge technique [17] uses graph structures to represent the regions or boundaries.

4.1.3 Statistical Methods

The segmentation process starts with a statistical analysis of the image data. The structure of the corresponding information is usually discarded. These methods use thresholding, adaptive thresholding, component labeling, amplitude projection and clustering. Kohonen maps [18], also known as self-organizing maps (SOM), operate in two modes: training and mapping. Training builds the map using input examples, also called vector quantization. Mapping automatically classifies a new input vector.

4.1.4 Knowledge-based Methods

The knowledge related to the segmentation objects properties (shape, color, structure, etc.) is exploited in the detection process. These methods often use templates database.

This database is automatically generated from the training data. Alternatively, the related information is inserted manually on the basis of human experience. During the segmentation, the algorithm is trying to transform the well-known objects or templates stored in the database to the objects belonging to the input image. This process is called atlas-warping. The object variability is the most degrading factor of the knowledge-based methods. However, if the objects in the structure are similar, these methods are mostly very efficient. A representative method is Active Appearance Models (AAM, see [19]).

4.1.5 Hybrid Methods

This group of methods combines some of the above discussed ideas with other characteristics of the image obtained by means of Watershed transformation [20] or neural networks [21]. The method called fuzzy min-max neural network for image segmentation (FMMIS) grows boxes from a set of pixels called seeds, to find the minimum bounded rectangle.

4.2 Machine Learning Methods for Visual Object Detection

The methods belonging to this group represent statistical approach to the objects detection and classification based on pattern recognition. The main approaches can be divided into three categories: statistical (feature-based) pattern recognition, structural pattern recognition and artificial neural networks [22, 23, 24].

In general, machine-learning methods are very robust. The fact that they are also typically very time-consuming does not play a role for our application. These methods are not very useful when we want to locate some object in the picture. On the other hand, they are powerful when we need to detect whether the picture contains required type of object. Pattern recognition methods are therefore often used for the selection of relevant pictures from a larger database of images.

5 Detection of the Objects and Annotation of the Graphics in the SVG format

The SVG format enables us to handle a bitmap structure either by referring an external raster image or directly by encapsulating the bitmap data. Because referring external bitmap data is risky, as the referred data can be lost - especially in the web environment, *GATE* uses the second possibility and involves the routine encapsulating the bitmap data directly into the SVG format.

5.1 Strategies for Efficient Object Detection and Annotation

In order to create a sufficiently large database of annotated pictures, it is necessary to make the annotation process as simple and fast as possible. During the annotation, the user marks out borders of specified objects in the picture. This can be supported by combining automatic image segmentation and edge detection together with subsequent hand-refinement using fuzzy tools like magic wand, lasso and other well-known image tools.

Second step in the annotation process is the definition of the semantics of the marked out objects. The use of ontologies can significantly improve both the image recognition as well as the semantics definition steps.

5.1.1 Ontology-driven Annotation

To every semantic category in the ontology a set of specific properties that characterize the category is assigned. For example, the "car" semantic category can have "color" and "type" properties. The annotation supported by the ontology requires to choose an object in the picture, select its semantic category from the ontology and then assign values to the prescribed properties. Therefore, once this kind of information is presented in the ontology, it can be reused easily and the user is not forced to think up the best characteristics for the object in the picture. This significantly accelerates the annotation process.

The annotation data are structured into the levels of detail, LOD, where the top-level describes the semantics of the significant objects, while the bottom levels contain their details. LOD is necessary for the navigation in the picture as well as for information filtering. Graphical ontology stores this type of structural information and helps to improve and accelerate the information structuring. Once the user selects an object in the picture, the ontology offers valid sub-objects and supports the user during the creation of the lower levels.

The ontology supports automatic object detection as well. It stores the information in the form of semantic categories and relationships between them. It also stores the information about specific annotated objects. Based on this knowledge, the ontology formalism enables us to deduce relevant consequences and constraints automatically. For instance, we can deduce that "cat is never green" or "swan appears usually together with a watter in a picture". This knowledge can enhance the efficiency of the automatic recognition of objects.

5.1.2 Coupling Ontology-based Annotation with SVG

The SVG format is primarily used to store and organize graphical primitives. The idea behind the annotating raster images is to mark out interesting areas of the image, to annotate them and store them in a SVG file together with the original raster image.

The annotated areas are marked using invisible SVG geometries with unique identifiers, e.g. transparent polygons, points, rectangles, etc. The annotation data, e.g. the semantics of the marked regions, are stored using a <metadata> tag. The primary goal of this annotation is to link the regions with specific semantic categories and their properties in the ontology. Besides, the annotation can define additional annotation information prescribed by the ontology, e.g. relationships between the categories. The following fragment of SVG file shows an example of an annotation section. rdf:ID elements refer to the identifiers of the marked regions. Head, Eye and Pupil represent semantic categories defined in the ontology. The ontology definition is not included directly, but linked as an external OWL file:

```
<metadata id="ANNOTATION_METADATA">
<rdf:RDF>
    <owl:Ontology rdf:about="">
        <owl:imports rdf:resource="http://owl.com/ontology.owl"/>
    </owl:Ontology>

    <Head rdf:ID="head"/>
    <Eye rdf:ID="lefteye"/>
    <Pupil rdf:ID="leftpupil"/>
    ... classification continues here ...
</rdf:RDF>
</metadata>
... SVG content including invisible marked regions ...
```

6 Detection of Objects in Raster Images: An Example

To illustrate possible techniques in the object detection for raster images, we present an analysis of the objects in a photography (Genoa lake, see Figure 1), containing many graphical objects to be annotated (fire place, mountains, sky, water, beach, trees, birds, etc.). In the first step, we divide the image into five basic regions (see Figure 2). Corel TRACE [25], a powerful and accurate raster-to-vector tracing application, has been utilized for semi-automated image segmentation and objects detection. The Advanced Outline Trace method enables us to set the following parameters:

- **Noise Filter** - depending on the amount of noise in the original image, None, Low or High filtering can be selected.

- **Complexity** - allows us to adjust the level of complexity (the number of vector elements) in the trace result.

- **Max Colors** - enables us to choose the number of the colors that are used in the trace results.

- **Node Reduction** - allows us to adjust the level node reduction that will be used in the trace results.

- **Node Type** - depending on the type of nodes to be used in trace results, Cusp or Smooth type can be selected.
- **Minimum Object Size** - specifies the minimum object size to be used in the trace results.

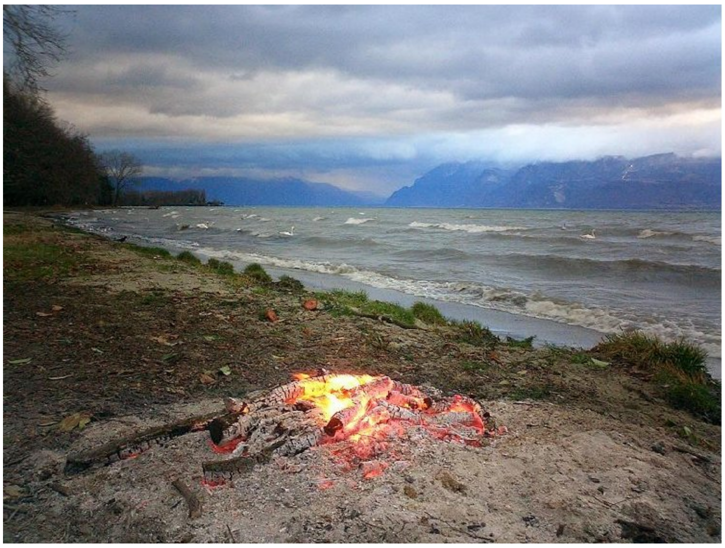


Figure 1: Original image.

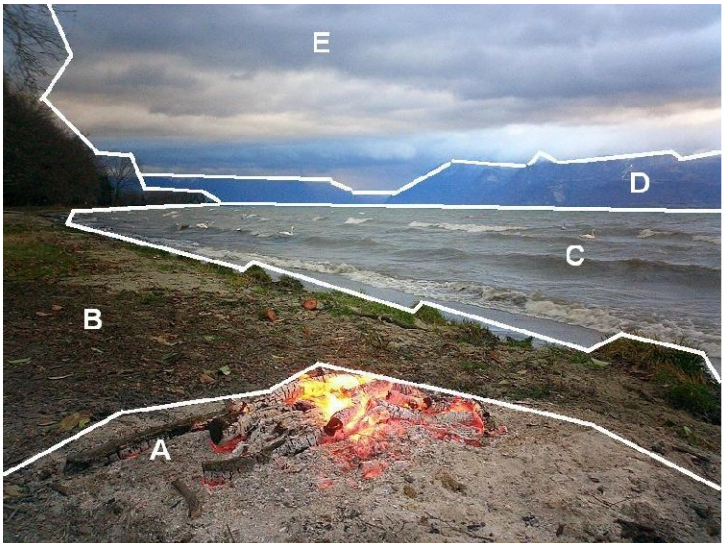


Figure 2: First iteration of the manual image segmentation.

In what follows, we present some results showing the dependence of the trace result on the selected complexity parameters:

- Noise Filter = low.
- Max Colors = 185.
- Node Reduction = 100.
- Node Type = Cusp.
- Minimum Object Size = 300.

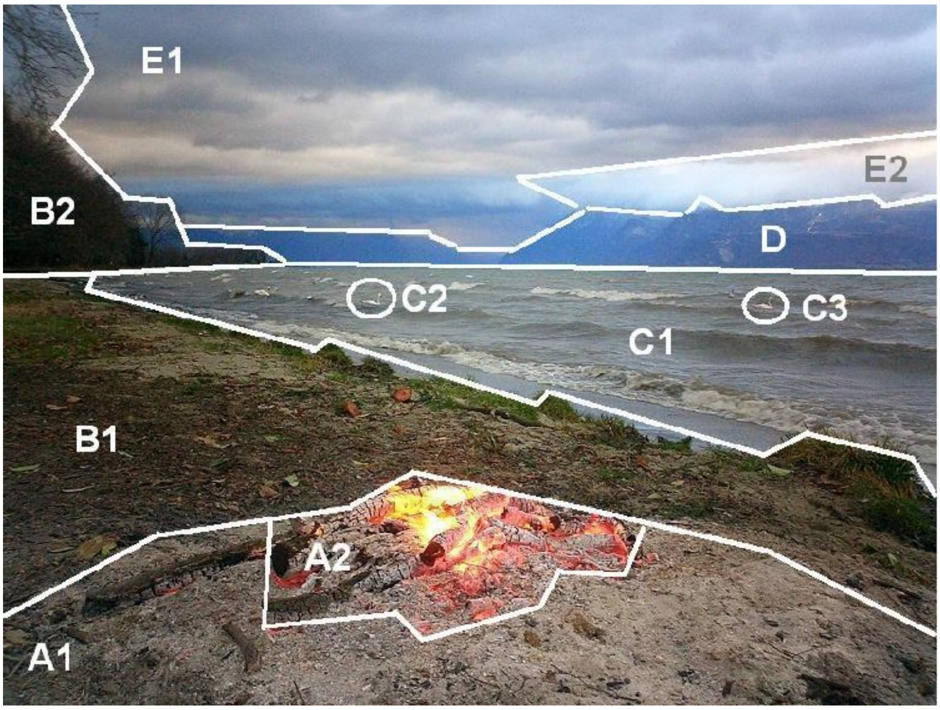
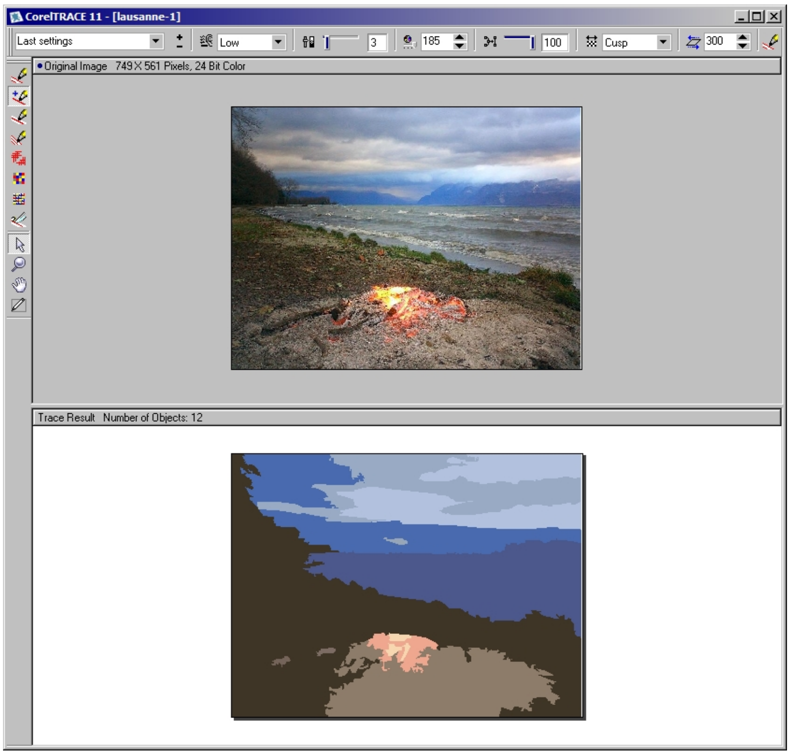


Figure 3: Second iteration of the manual image segmentation.

The Figure 4 shows the result for Complexity set to 3. In this case, the segmentation quality is not sufficient. Only 12 objects were discovered, and some significant objects were merged together. In the next step, we set Complexity to 5. The trace result and the selection of one recognized object are shown in Figure 5. The number of the matched objects increases to 35. The accuracy of the segmentation is much better than in the first attempt. Finally, if the Complexity parameter is set to 10, we get 99 various objects recognized, see Figure 6.



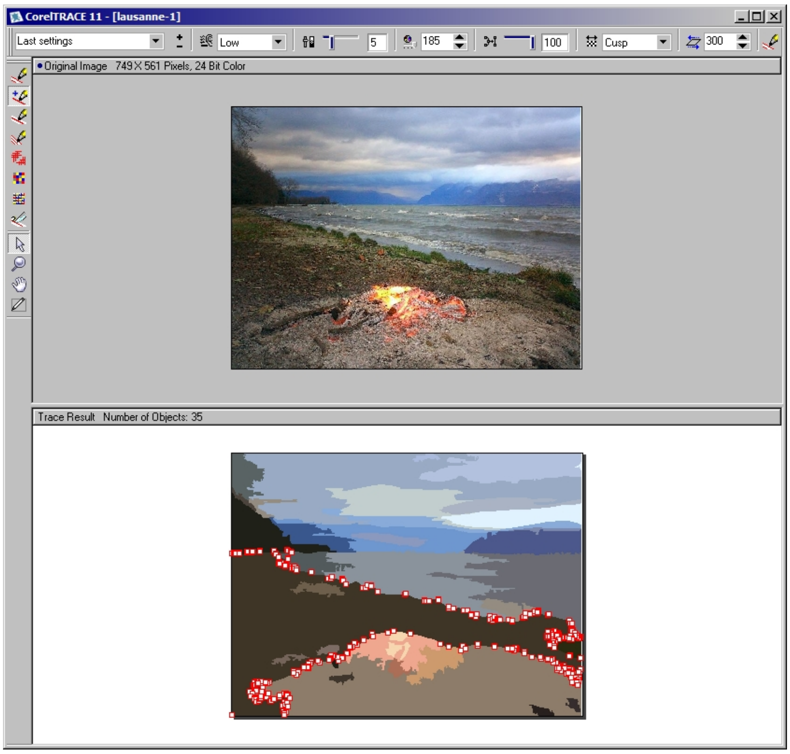
CorelTRACE 11 - [lausanne-1]

Last settings | Low | 3 | 185 | 100 | Cusp | 300

• Original Image 749 X 561 Pixels, 24 Bit Color

Trace Result Number of Objects: 12

Figure 4: The trace result with the Complexity parameter set to 3.



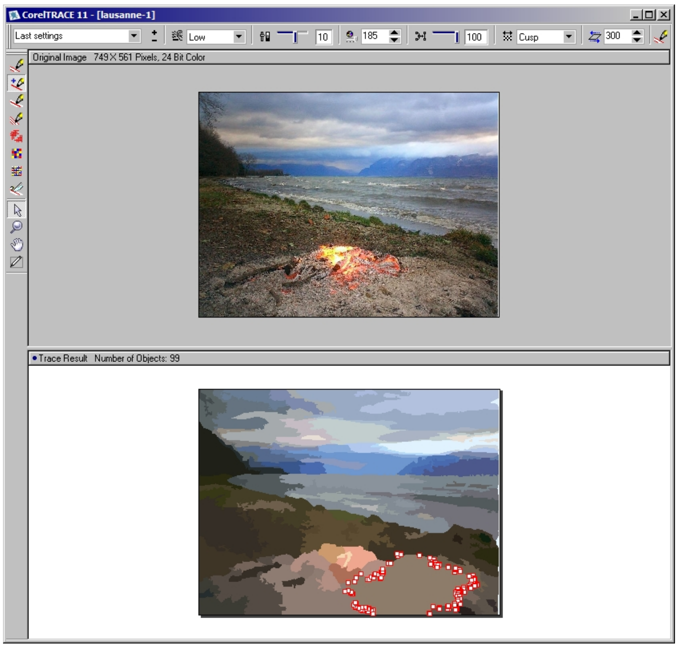
CorelTRACE 11 - [lausanne-1]

Last settings | Low | 5 | 185 | 100 | Cusp | 300

• Original Image 749 X 561 Pixels, 24 Bit Color

Trace Result Number of Objects: 35

Figure 5: The trace result with the Complexity parameter set to 5.



Original Image 749 X 561 Pixels, 24 Bit Color

Trace Result Number of Objects: 99

Figure 6: The trace result with the Complexity parameter set to 10.

Based on this discussion, the trace result with Complexity set to 3 was selected for manual manipulation in CorelDRAW. It contains most of the objects suitable for annotation. Only some of them need manual finalizing by Group and Weld functions. Very small objects like birds were discarded. Finally, the segmented picture was converted to the SVG format.

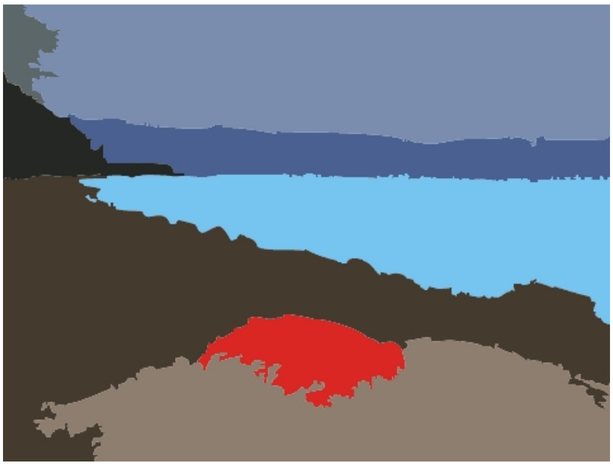


Figure 7: The final result of the semi-automatic image segmentation.

7 Conclusions and Future Work

The detection and annotation of graphical objects in raster images is a complex task requiring a specific approach for each concrete application. In the GATE project, most of the commonly used methods for image segmentation and object detection can be utilized, at least to some extent. To find and implement an appropriate strategy for balancing the choice of the segmentation methods with the manual segmentation and detection is the main goal for the future work.

In the area of the dialogue management, developing and implementing further types of dialogue grammars is an important goal. Suitable procedures and principles defined in Speech Recognition Grammar Specification [26] and Semantic Interpretation for Speech Recognition standards [27] will be utilized to accomplish this task.

References

[1] Edman, P. K.: Tactile Graphics. American Foundation for the Blind, New York, 1992.

[2] Kaczmarek, K.: Electrotactile Display for Computer Graphics to Blind. Research Report 5–R01–EY10019–08, University of Wisconsin, 2004.

[3] Kurze, M.: TDraw: a computer-based tactile drawing tool for blind people. In *Proceedings of the second annual ACM conference on Assistive technologies*, Vancouver, 1996, pp. 131–138.

[4] Satoshi, I.: Computer Graphics for the Blind. ACM SIGCAPH Newsletter Page, 1996, pp. 16–21.

[5] Daunys, G. and Lauruska, V.: Maps Sonification System Using Digitiser for Visually Impaired Children. In *ICCHP 2006*, Berlin, Springer-Verlag, pp. 12-15.

[6] Matta S., Rudolph, H. and Kumar, D. K.: Auditory Eyes: Representing Visual Information in Sound and Tactile Cues. In *13th European Signal Processing Conference*, Antalya, 2005.

[7] Bulatov, V. and Gardner, J.: Making Graphics Accessible. In *3rd Annual Conference on Scalable Vector Graphics*, Tokyo, 2004.

[8] Fredj, Z. B. and Duce, D. A.: GraSSML: Accessible Smart Schematic Diagrams for All. In *Theory and Practice of Computer Graphics*, IEEE Computer Society, 2003, pp. 49–55.

[9] Mathis, R. M.: Constraint Scalable Vector Graphics, Accessibility and the Semantic Web. In *SoutheastCon Proceedings*, IEEE Computer Society, 2005, pp. 588–593.

[10] Kopeček, I. and Ošlejšek, R.: GATE to Accessibility of Computer Graphics. In *Computers Helping People with Special Needs: 11th International Conference*, Berlin, Springer-Verlag, 2008, pp. 295–302.

[11] Kopeček, I. and Ošlejšek, R.: Creating Pictures by Dialogue. In *International Conference on Computers Helping People with Special Needs*, Berlin, Springer-Verlag, 2006, pp. 61–68.

[12] Kamel, H. M. and Landay, J. A.: Sketching images eyes-free: a grid-based dynamic drawing tool for the blind. In *Fifth international ACM conference on Assistive technologies*, 2002, pp. 33–40.

[13] The World Wide Web Consortium - Voice Extensible Markup Language (VoiceXML) Version 2.0, available at *http://www.w3.org/TR/voicexml20*.

[14] Sharon, E., Brandt, A. and Basri, R.: Fast Multiscale Image Segmentation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, South Carolina, 2000, pp. 70–77.

[15] Canny, J.: A Computational Approach to Edge Detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Washington, 1986, pp. 679–698.

[16] Duda, R. O. and Hart, P. E.: Use of the Hough Transformation to Detect Lines and Curves in Pictures. In *Communications of the ACM*, Vol. 15, January 1972, pp. 11–15.

[17] Liu, L. and Sclaroff, S.: Region Segmentation via Deformable Model-guided Split and Merge. In *International Conference on Computer Vision (ICCV)*, July 2001, pp. 98–104.

[18] Reyes, A. and Constantino, C.: Image Segmentation with Kohonen Neural Network Self Organising Maps. In *International Conference on Telecommunications ICT 2000*, Acapulco, Mexico, 2000.

[19] Cootes, T. F., Edwards, G. and Taylor, C. J.: Active Appearance Models. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, June 2001, pp. 681-685.

[20] Beucher, S.: The Watershed Transformation Applied to Image Segmentation. In *Proceedings 10th Pfefferkorn Conference on Signal and Image Processing in Microscopy and Microanalysis*, Cambridge, U.K., September 1991, pp. 299–314.

[21] Reyes-Aldasoro, C. C. and Aldeco, A. L.: Image Segmentation and Compression Using Neural Networks. In *Advances Artificial Perception Robotics CIMAT*, Guanajuato, Mexico, October 2000.

[22] Duda, R. O., Hart, P. E., Stork, D. G.: Pattern Classification, John Wiley & Sons, New York, USA, 2001.

[23] Schlesinger, M. I., Hlaváč, V.: Ten lectures on statistical and syntactic pattern recognition, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[24] Bishop, C.: Pattern Recognition and Machine Learning, New York, Springer-Verlag, 2006.

[25] Taking Corel PowerTRACE X3 for a Test Drive - CorelDRAW Graphics Suite available at *http://www.corel.com/servlet/Satellite/gb/en/Content/1171405233027*.

[26] The World Wide Web Consortium - Speech Recognition Grammar Specification Version 1.0, available at *http://www.w3.org/TR/speech-grammar/*.

[27] The World Wide Web Consortium - Semantic Interpretation for Speech Recognition (SISR) Version 1.0, available at *http://www.w3.org/TR/semantic-interpretation/*.

[28] Tu, Z. W., Chen, X., Yuille, A. and Zhu, S. C.: Image Parsing: Segmentation, Detection and Recognition. In *9th IEEE International Conference on Computer Vision (ICCV)*, Nice, France, October 2003, pp. 18-25.

[29] Spirkovska, L.: A Summary of Image Segmentation Techniques. In *Technical report NASA*, Ames Research Center, 1993.