A New System to Perform Unsupervised and Supervised Classification of Satellite Images from Google Maps

Sergio Bernabé[a] and Antonio Plaza[a]

[a]Hyperspectral Computing Laboratory
Department of Technology of Computers and Communications
University of Extremadura, Avda. de la Universidad s/n
E-10071 Cáceres, Spain

ABSTRACT

In this paper, we describe a new system for unsupervised and supervised classification of satellite images from Google Maps. The system has been developed using the SwingX-WS library, and incorporates functionalities such as unsupervised classification of image portions selected by the user (at the maximum zoom level) using ISODATA and k-Means, and supervised classification using the Minimum Distance and Maximum Likelihood, followed by spatial post-processing based on majority voting. Selected regions in the classified portion are used to train a maximum likelihood classifier able to map larger image areas in a manner transparent to the user. The system also retrieves areas containing regions similar to those already classified. An experimental validation of the proposed system has been conducted by comparing the obtained classification results with those provided by commercial software, such as the popular Research Systems ENVI package.

Keywords: Satellite image classification, Google Maps.

1. INTRODUCTION

The wealth of satellite imagery[1] available in web mapping service applications such as Google Maps*, which now provides high-resolution satellite images from many locations around the Earth, has opened the appealing perspective of performing classification and retrieval tasks via the Google Maps application programming interface (API)† and other external libraries such as SwingX-WS‡. In fact, the introduction of Google's mapping engine prompted a worldwide interest in satellite imagery exploitation. The combination of an easily pannable and searchable mapping and satellite imagery tool such as Google Maps with advanced image classification and retrieval features has the potential to significantly expand the functionalities of the tool and also to allow end-users to extract relevant information from a massive and widely available database of satellite images (the Google Maps service is free for non-commercial use).

Google created the Google Maps API to allow developers to integrate Google Maps into their own websites and applications. By using the Google Maps API or external libraries such as SwingX-WS, it is possible to embed the full Google Maps site into an external website application. Other similar services currently available comprise Yahoo Maps§ and OpenStreetMap¶. The characteristics of Yahoo Maps are similar to those available in Google Maps (although the spatial resolution of the satellite imagery available in Yahoo Maps is generally lower than the resolution of the image data available from Google Maps). On the other hand, the OpenStreetMap follows a different approach. It is a collaborative project aimed at creating a free editable map of the world. Its design was inspired by sites such as Wikipedia. Instead of using satellite images, the map data was built from scratch by volunteers performing systematic ground surveys using a hand-held global positioning system (GPS)

Send correspondence to Sergio Bernabé and Antonio Plaza:
E-mail: sbernabe@alumnos.unex.es, aplaza@unex.es; Telephone: +34 927 257000 (Ext. 51520)

*http://maps.google.com

†http://code.google.com/apis/maps/index.html

‡https://swingx-ws.dev.java.net

§http://maps.yahoo.com

¶http://www.openstreetmap.org

Satellite Data Compression, Communications, and Processing VI, edited by Bormin Huang, Antonio J. Plaza, Joan Serra-Sagristà, Chulhee Lee, Yunsong Li, Shen-En Qian, Proc. of SPIE Vol. 7810, 781010
 · doi: 10.1117/12.863243

	Google Maps	YAHOO! LOCAL Maps	OpenStreetMap
Restrictions of use	No, up to a certain limit	Yes	Yes
Hybrid satellite view	Yes	Yes, low visibility	No
High resolution imagery	Yes	No	No
Zooming levels	Very high quality	High quality	Medium quality
Error correction	Low	Low	Very high
Smoothness in navigation	Very high	High	High
Adaptivity for desktop applications	High	High	High

Figure 1. Comparison between the main functionalities of Google Maps, Yahoo Maps and OpenStreetMap.

unit and a notebook, digital camera, or a voice recorder. These data were then entered into the OpenStreetMap database. More recently, the availability of aerial photography and other data sources from commercial and government sources has greatly increased the speed of this work and has allowed land-use data to be collected more accurately. For illustrative purposes, Fig. 1 shows a comparison between the main functionalities of Google Maps, Yahoo Maps and OpenStreetMap. As shown by Fig. 1, the Google Maps service offers important competitive advantages, such as the availability of high resolution satellite imagery, the smoothness in the navigation and interaction with the system, the availability of a hybrid satellite view which can be integrated with other views (maps, etc.) and adequate adaptivity for general-purpose desktop applications. Despite its competitive advantages, the possibility to perform unsupervised or supervised classification of satellite images[2] is not available in Google Maps, despite image classification is widely recognized as one of the most powerful approaches in order to extract information from satellite imagery.[3–5]

In this paper, we describe a new tool which allows an unexperienced user to perform unsupervised classification of satellite images obtained via Google Maps by means of the ISODATA and k-Means classifiers, followed by spatial post-processing based on majority voting. Selected regions in the classified portion can then be used to train a more sophisticated, supervised classifier (of maximum likelihood type) able to map larger image areas in a manner transparent to the user. Several examples of use, focused on the analysis of different locations around the Iberian Peninsula and comprising different types of information extraction case studies (vegetation, semi-arid environments, urban areas, etc.) are illustrated and thoroughly described.

The remainder of the paper is organized as follows. Section 2 describes the programming libraries used to develop the proposed system and the integration of the different modules. Section 3 presents the unsupervised and supervised classification techniques considered in its implementation. Section 4 describes the integration of the different software modules. Section 5 presents an experimental evaluation of the proposed system which has been conducted by comparing the obtained classification results with those provided by commercial software, such as the popular Research Systems ENVI package. Finally, section 6 concludes with some remarks and hints at plausible future research.

2. LIBRARIES

In this section we describe the different steps that we followed for the integration of different software components in libraries towards the design of our proposed classification system for Google Maps satellite imagery. Our first approach to the design of such system was directed towards the exploitation of the Google Maps API in order to access the satellite imagery. However, we realized that the Google Maps API is mainly intended for exploitation

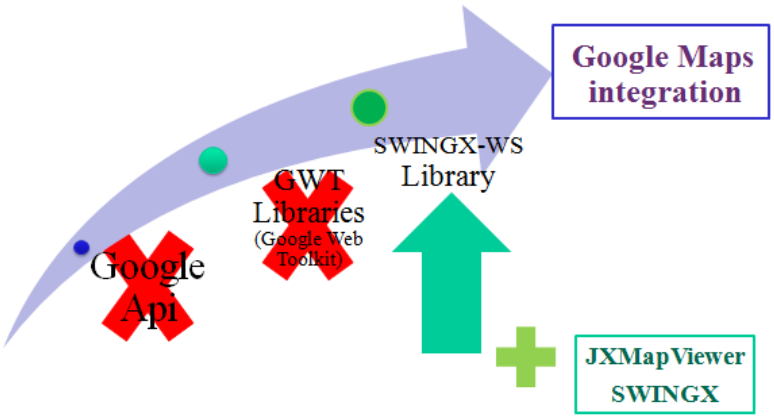


Figure 2. Flowchart describing our interaction with different libraries for accessing the satellite data available in Google Maps and our final choice (SwingX-WS library with the JXMapViewer component).

in web-based applications and our goal was to develop an executable application in the PC. Another approach was to use Google Web Toolkit (GWT)[‖], an open source set of tools that allows web developers to create and maintain complex JavaScript front-end applications in Java. However, the fact that GWT is also intended for web-based applications led us to further discard this toolkit as the baseline package for the development of our tool. Finally, our approach to address the aforementioned issues was to resort to the SwingX-WS library. SwingX-WS attempts to simply the use of web services (in the broad sense) by providing APIs that site on top of existing libraries. It contains a set of JavaBeans for interacting with web services, but it allows embedding of those services into a standard application (this feature fit very well our desired functionality). The initial beans included in SwingX-WS comprise support for several Google web services such as searching news, video, images, and financial data, as well as a generic tile based mapping component. The SwingX-WS beans have been designed with graphical configuration in mind and work very well inside of a JavaBeans aware editor such as NetBeans. The bean that we particularly exploited in the development of our application is `JXMapViewer`, a generic viewer for tile based map servers. For illustrative purposes, Fig. 2 shows a flowchart of our interaction with different libraries for accessing the satellite data available in Google Maps and our final choice, given by the use of SwingX-WS library with the `JXMapViewer` component. In addition to `JXMapViewer`, other additional SwingX-WS and SwingX modules were used in the development of our application. In the following, we provide a brief overview of these additional modules:

- `JXMapKit`. The `JXMapKit` is a pair of `JXMapViewers` preconfigured to be easy to use with common features built in. This includes zoom buttons, a zoom slider, and a mini-map in the lower right corner showing an overview of the map.
- `TileFactoryInfo`. A `TileFactoryInfo` encapsulates all information specific to a map server. This includes everything from the URL to load the map tiles from to the size and depth of the tiles. Theoretically any map server can be used by installing a customized `TileFactoryInfo`.
- `DefaultTileFactory`. Creates a new instance of `DefaultTileFactory` using the spcified `TileFactoryInfo`.
- `GeoPosition`. Provides the geographical latitude and longitude and allows centering of the satellite image in a certain geographic location.
- `Painter`. This API allows developers to be able to customize the background painting of a `JXPanel`. Since many components within SwingX extend `JXPanel`, the developer can implement custom painting on many parts of SwingX.

[‖]http://code.google.com/webtoolkit

Figure 3. Flowchart describing the interaction procedure with Google Maps to obtain the satellite data.

- `CompoundPainter`. `Painters` can be combined together by using the `CompoundPainter`. `CompoundPainter` uses an array to store several `Painters`, and the order in which they should be painted.

To conclude this section, Fig. 3 describes the overall procedure for interacting with Google Maps via the SwingX-WS library to obtain the satellite data. As shown by Fig. 3, a client (user) sends a request to a web server via SwingX-WS. The web server interacts with an application server that provides the map server functionality. Finally, the application server interacts directly with a database from which the extracted information is provided.

3. CLASSIFIERS

The proposed system incorporates functionalities of unsupervised clustering and supervised classification, all followed by spatial post-processing based on majority voting.[6] Unsupervised clustering aims at grouping pixels if feature space, so that pixels belonging to the same cluster are spectrally similar.[2] In our implementation, we have used the well-known ISODATA[7] and k-means[8] algorithms for this purpose. For instance, the ISODATA is a squared-error clustering method. The algorithm starts with a random initial partition of the available pixel vectors in the original image $\mathbf{I}$ into c candidate clusters. It then iteratively optimizes this initial partition so that, on each iteration i, a partition $\mathbf{P}^i = \{\mathbf{P}_1^i, \mathbf{P}_2^i, \cdots, \mathbf{P}_c^i\}$ of the set $\mathbf{I}$ into c clusters is computed, where $\mathbf{P}_k^i = \{\mathbf{X}_{j,k}^i \in \Re^n, j = 1, 2, \cdots, m_k^i\}$ contains the pixels belonging to the k-th component on the iteration i, with m_k^i denoting the number of pixels in $\mathbf{P}_k^i$ and $k = 1, 2, \cdots, c$. With this notation in mind, in which the spatial coordinates associated to the pixels have been omitted for simplicity, the squared error for a given partition $\mathbf{P}^i$ of the original hyperspectral image $\mathbf{I}$ into c clusters is defined as:

$$e^2(\mathbf{P}^i) = \sum_{k=1}^{c} \sum_{j=1}^{m_k^i} \|\mathbf{X}_{j,k} - \mathbf{C}_k\|, \tag{1}$$

where $\mathbf{C}_k$ is the centroid of the k-th cluster. The squared error in Eq. (1) is reduced at each iteration, until a convergence criterion is achieved. A relevant issue for the ISODATA algorithm is how to set the number of clusters c in advance. In our work, this choice is left to the end-user, who can adjust the quality of the clustering by interactively setting this parameter. Once a preliminary segmentation of the original image has been achieved via unsupervised clustering, a supervised procedure can be applied to classify other different areas based on the

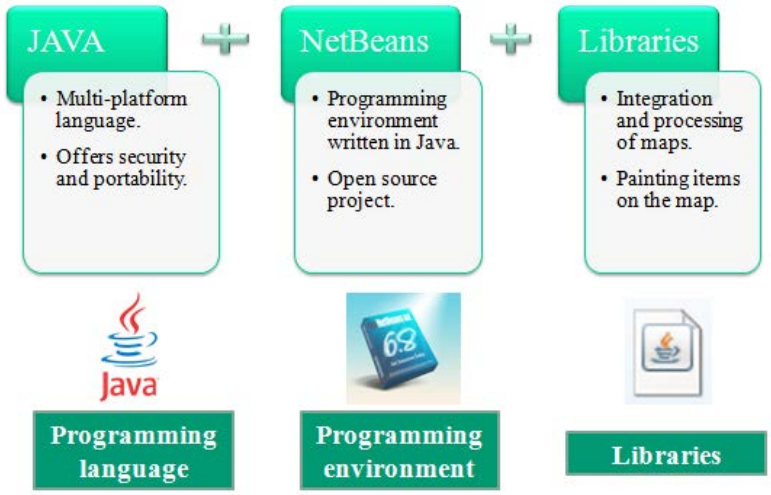


Figure 4. Different software modules used for the development of our application.

training sites selected in a different spatial location. In our tool, this can be accomplished using the minimum distance and maximum likelihood classifiers.[2] Conventional maximum likelihood classification is based on the assumption that the probability distribution for each spectral (color) class is of the form of a multivariate normal model with dimensions which equal the number of color bands. This leads to the following discriminant function:

$$g_i(\mathbf{X}_j) = -\ln|\gamma_i| - (\mathbf{X}_j - \mathbf{C}_i)^T \gamma_i (\mathbf{X}_j - \mathbf{C}_i), i = 1, \cdots, c, \tag{2}$$

where $\mathbf{X}_j$ is a pixel vector (formed by three color components: red, green and blue and the alpha channel in our system), $\mathbf{C}_i$ is the mean vector for the class i, with $i = 1, \cdots, c$, and γ_i is the covariance matrix of class i. A spatial post-processing module can be applied to refine the outcome of the segmentation by simply sliding a square neighborhood window (with sizes ranging from 3×3 to 7×7 pixels) centered in each classified pixel and applying a majority voting procedure in which the central pixel is assigned to the most predominant class in the neighborhood window.

4. INTEGRATION

The final implementation of our system consists of the integration of the different software modules developed (unsupervised and supervised classifiers) with the functionalities provided by the SwingX-WS libraries (provided by the SwingX libraries), in the form of a general-purpose desktop application. For this purpose, we have resorted to the Java programming language (see Fig. 4), which is a multi-platform environment that simplifies porting of our tool to different environments. Specifically, we have resorted to the NetBeans platform (see Fig. 4), which allows applications to be developed from a set of modular software components called modules. In this framework, applications can install modules dynamically and any application can include the update center module to allow users of the application to download digitally-signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again. The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. With the above ideas in mind, Fig. 5 shows an outline of the integration of the different modules, which result in an application called GoogleCBIR. This name refers to our future goal to have a content-based image retrieval system in place based on the interaction of the user with the system.

5. VALIDATION

In this section, we perform an experimental validation of the unsupervised and supervised classification features of our tool tested using satellite images obtained from Google Maps across different locations. The experimental

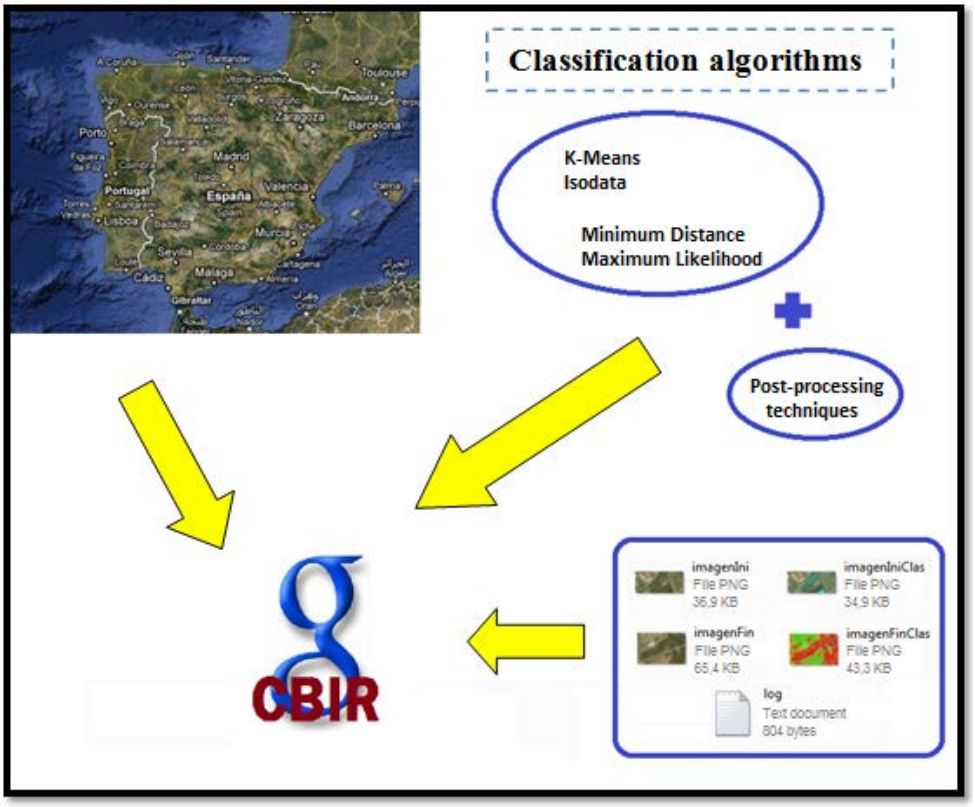


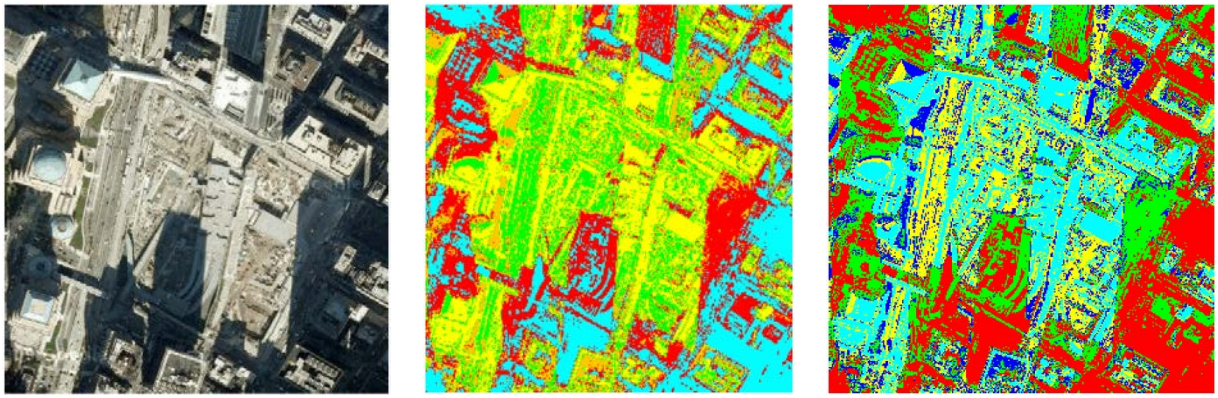
Figure 5. Integration of the different modules in a desktop application (GoogleCBIR) that we plan to extend to the functionality of content-based image retrieval in future developments.

validation of unsupervised (ISODATA, k-means) and supervised (minimum distance, maximum likelihood) classification algorithms has been conducted by comparing the results provided by our implementations with those available in a well-known commercial software package: the Environment for Visualizing Images (ENVI) package distributed by ITT Visual Information Solutions**. In our tests, conducted across four different locations around the world, we adopt exactly the same parameters when running our implementations and those available in the ENVI package, comparing the results in terms of the overall accuracy (OA) resulting from the confusion matrix between the classification maps provided by ENVI and by our tool.

In the following, we present the obtained results in a specific case study focused on classification of satellite images available from the World Trade Center (WTC) area in New York City. Fig. 6(a) shows a satellite image over the WTC extracted using the developed tool from Google Maps engine. The resolution of the image is quite high, with approximately 5 meters per pixel. Fig. 6(b) shows the unsupervised classification result provided by our implementation of ISODATA. Fig. 6(c) shows the unsupervised classification result provided by ENVI's ISODATA. As shown by Fig. 6, the color labels for our implementation and the one available in ENVI are different, but the classification maps are very similar. In both cases, the parameters for both algorithms have been set to exactly the same values, with $c = 5$. For illustrative purposes, Table 1 reports the OA (in percentage) measured after comparing our ISODATA classification map with the one obtained by ENVI. As shown by Table 1, the similarity between both classification maps is very high, achieving an OA close to 90% assuming the ENVI ISODATA map as the ground reference.

In a second experiment, we select a larger image over a different location in New York City [see Fig. 7(a)]. The spatial resolution of this image is approximately of 5 meters per pixel. In order to classify this scene in supervised fashion, we have selected the areas resulting from an unsupervised classification (using ISODATA) of a different zone in New York City and used those areas to train our maximum likelihood classifier using $c = 5$. The

**http://www.ittvis.com/ProductServices/ENVI.aspx



(a) (b) (c)

Figure 6. (a) Satellite image over the World Trade Center area. (b) Unsupervised classification result provided by our implementation of ISODATA. (c) Unsupervised classification result provided by ENVI's implementation of ISODATA.

Table 1. Overall (OA) and individual accuracies (in percentage) after comparing our ISODATA classification map with the one obtained by ENVI for the image in Fig. 6(a).

Shadows #1 (blue)	Shadows #2 (red)	Urban areas #1 (yellow)	Urban areas #2 (green)	Urban areas #3 (orange)	Overall accuracy
78.26	95.49	85.56	90.65	97.39	89.47

resulting classification map is displayed in Fig. 7(b). For illustrative purposes, the classification map achieved by the maximum likelihood algorithm available in ENVI is displayed in Fig. 7(c). In order to obtain this map, the maximum likelihood algorithm in ENVI was trained using exactly the same areas and with the same parameters as those used in our implementation. Again, Fig. 7 reveals that, although the color labels for our implementation and the one available in ENVI are different, the classification maps are very similar. For illustrative purposes, Table 2 reports the OA (in percentage) measured after comparing our maximum likelihood classification map with the one obtained by ENVI. As shown by Table 2, the similarity between both classification maps is very high, achieving an OA above 85% assuming the ENVI maximum likelihood map as the ground reference. These results indicate a high degree of similarity between our implementations of available classification algorithms and those available in ENVI. The main contribution of this work is the possibility of integrating these classifiers with the satellite images derived from Google Maps via the developed tool.

To conclude this section, Fig. 8 shows different views of the developed tool. As shown by the figure, the tool allows selecting an area to be classified, obtaining classification results both in unsupervised and supervised fashion, retaining the classified area at different zoom levels (although the classification is obtained at the

(a) (b) (c)

Figure 7. (a) Satellite image over New York City. (b) Supervised classification result provided by our implementation of maximum likelihood (trained with the areas derived from other zone by ISODATA). (c) Supervised classification result provided by ENVI's implementation of maximum likelihood (trained with the areas derived from other zone by ISODATA).

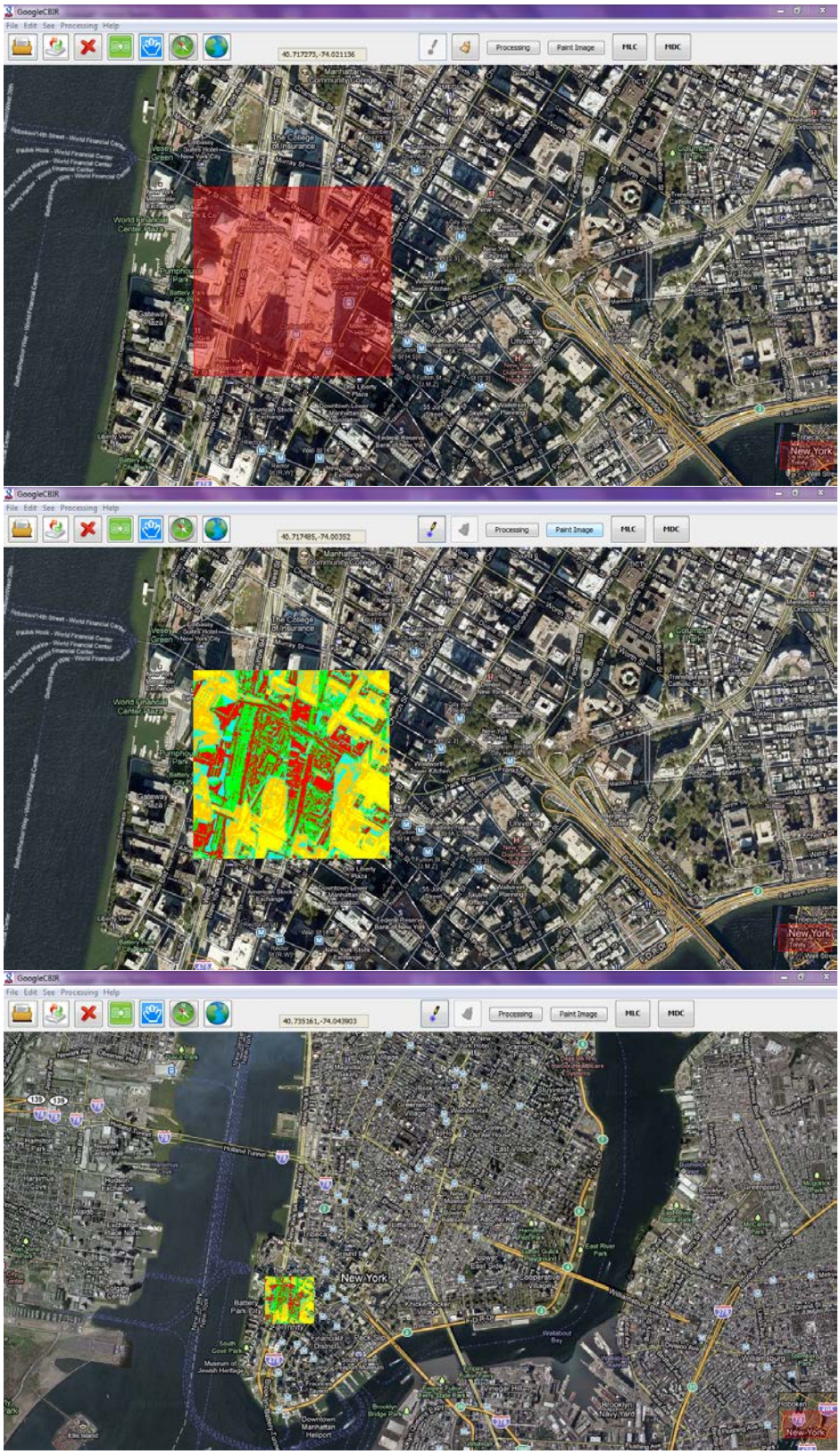


Figure 8. Different views of the developed tool. Selection of an area to be classified in New York City (top). Unsupervised classification result provided by our implementation of ISODATA superimposed on the tool (middle). Zoom reduction retaining the classified area (bottom).

Table 2. Overall (OA) and individual accuracies (in percentage) after comparing our maximum likelihood classification map with the one obtained by ENVI for the image in Fig. 7(a).

Vegetation #1 (green)	Vegetation #2 (blue)	Water (orange)	Urban areas #1 (red)	Urban areas #2 (yellow)	Overall accuracy
71.71	85.60	76.33	94.13	98.07	85.17

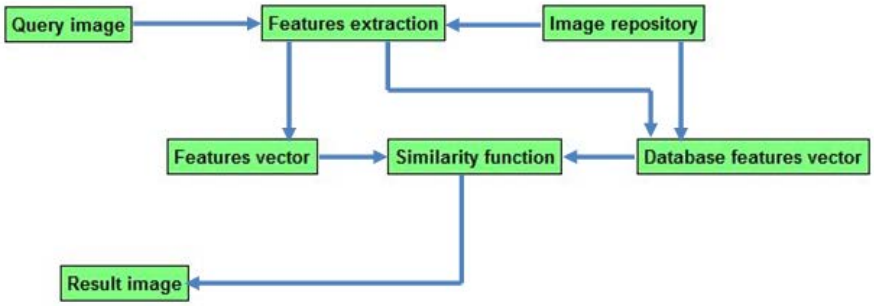


Figure 9. Integration of the different modules in a desktop application (GoogleCBIR) that we plan to extend to the functionality of content-based image retrieval in future developments.

maximum zoom level), and other functionalities not illustrated in Fig. 8 such as spatial post-processing of obtained results for increased spatial consistency, managing of the resulting classification and extracted satellite images, loading/storing of results via file logs which can be saved in a database, automatic positioning in any latitude and longitude coordinates in the entire Google Maps database, overlaying of classification results with different views (satellite, map, hybrid), etc. Overall, we feel that the developed tool incorporates interesting additional functionalities to the Google Maps engine (particularly in the possibility of better exploiting the satellite images available from this tool in different application domains).

6. CONCLUSION AND FUTURE LINES

This paper has described a new application for unsupervised and supervised classification of satellite images from Google Maps. The system has been developed using the SwingX-WS library, and incorporates functionalities such as unsupervised classification of image portions selected by the user (at the desired zoom level) using ISODATA and k-Means, and supervised classification using the Minimum Distance and Maximum Likelihood, followed by spatial post-processing based on majority voting. Our experimental results, conducted by comparing the obtained classification results with those provided by commercial software such as the popular Research Systems ENVI package, reveal that the proposed tool provides classification maps of high similarity with regards to those provided by ENVI for the same satellite imagery, but with the possibility to perform classification of any image portion available in Google Maps engine, both in unsupervised and supervised fashion.

In future developments, we plan to extend the developed tool with the incorporation of content-based image retrieval funcionalities. For that purpose, the strategy will be based on a query system linked to feature extraction from an image repository (Google Maps). The retrieved features (which will comprise shape descriptors, texture features, etc.) will be stored in a database of features and used to compare the feature vector of the input query with those recorded in the database by means of a similarity function, which will provide a result to the end-user in the form of image portions (across different locations) with sufficient similarity with regards to the features in the input query. This strategy is illustrated in Fig. 9.

Finally, another drawback of the proposed tool is the high computational cost of extracting and processing large images (especially at the highest zoom level, which results in large image sizes). To address this issue, we are currently experimenting with different forms of high performance computing architectures.[9–11] The most promising strategy seems to be the parallelization of the different software modules for efficient processing in multiple cores (generally available in modern desktop PCs) as well as in graphics processing units (GPUs) of NVidia type.[12–14] Both mechanisms allow incorporation of high performance computing capabilities at relatively

low cost in order to speed-up the computations performed by the developed tool, which mainly comprise regular image processing-type computations which are quite appealing for parallel implementation.

7. ACKNOWLEDGEMENT

This work has been supported by the European Community's Marie Curie Research Training Networks Programme under reference MRTN-CT-2006-035927 (HYPER-I-NET). Funding from the Spanish Ministry of Science and Innovation (HYPERCOMP/EODIX project, reference AYA2008-05965-C04-02) is gratefully acknowledged.

REFERENCES

1. D. A. Landgrebe, *Signal theory methods in multispectral remote sensing*, John Wiley and Sons, Hoboken, NJ, 2003.
2. J. A. Richards, "Analysis of remotely sensed data: the formative decades and the future," *IEEE Trans. Geoscience and Remote Sensing* **43**, pp. 422–432, 2005.
3. P. Soille, *Morphological image analysis: principles and applications*, Springer-Verlag, Berlin, 2003.
4. J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geoscience and Remote Sensing* **42**, pp. 480–491, 2005.
5. L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive svm for the semisupervised classification of remote sensing images," *IEEE Trans. Geoscience and Remote Sensing* **44**, pp. 3363–3373, 2006.
6. P. Gamba, F. Dell'Acqua, A. Ferrari, J. A. Palmason, and J. A. Benediktsson, "Exploiting spectral and spatial information in hyperspectral urban data with high resolution," *IEEE Geoscience and Remote Sensing Letters* **1**, pp. 322–326, 2004.
7. G. Ball and D. Hall, "ISODATA, a novel method of data analysis and classification," *Tech. Rep. AD-699616, Stanford University* , 1965.
8. J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society, Series C (Applied Statistics)* **28**, pp. 100–108, 1979.
9. A. Plaza, D. Valencia, J. Plaza, and P. Martinez, "Commodity cluster-based parallel processing of hyperspectral Imagery," *Journal of Parallel and Distributed Computing* **66**(3), pp. 345–358, 2006.
10. A. Plaza and C.-I. Chang, *High performance computing in remote sensing*, CRC Press, Boca Raton, 2006.
11. A. Plaza and C.-I. Chang, "Clusters versus FPGA for parallel processing of hyperspectral imagery," *International Journal of High Performance Computing Applications* **22**(4), pp. 366–385, 2008.
12. Y. Tarabalka, T. V. Haavardsholm, I. Kasen, and T. Skauli, "Real-time anomaly detection in hyperspectral images using multivariate normal mixture models and gpu processing," *Journal of Real-Time Image Processing* **4**, pp. 1–14, 2009.
13. J. Setoain, M. Prieto, C. Tenllado, A. Plaza, and F. Tirado, "Parallel morphological endmember extraction using commodity graphics hardware," *IEEE Geoscience and Remote Sensing Letters* **43**, pp. 441–445, 2007.
14. J. Setoain, M. Prieto, C. Tenllado, and F. Tirado, "GPU for parallel on-board hyperspectral image processing," *International Journal of High Performance Computing Applications* **22**(4), pp. 424–437, 2008.