VISUAL ANALYTICS FOR BUILT-UP AREA UNDERSTANDING FROM METRIC RESOLUTION EARTH OBSERVATION DATA

J. Lozano*, M. Quartulli, I. Tamayo, M.Laka and I. Olaizola

Digital Television and Multimedia Services
Vicomtech-ik4
Mikeletegi Pasealekua 57, 20009 Donostia - San Sebastian, Spain
{jlozano, mquartulli, itamayo, mlaka, iolaizola}@vicomtech.org
http://www.vicomtech.org/

KEY WORDS: visual analytics, high resolution, remote sensing

ABSTRACT:

Large scale archives can benefit the application of visual analytics methodologies aimed at characterizing their contents by the effective inclusion of the human analyst in the interpretation loop.
Exploiting the knowledge of users that are not remote sensing experts requires the design of easy to use applications.
Applied analytical reasoning by visual representations involves methodological aspects dealing with both the design of multiple interactive visualizations as well as data representation and transformation considerations.
We present examples of such methodological aspects aiming at the understanding and characterization of metric resolution datasets acquired on urban environments.

1 INTRODUCTION

The latest technological advances have made possible to acquire a multitude of imaging data of the Earth at metric resolutions. With a high level of detail these technologies allow us to detect objects in metric range like a cars or building details. This powerful feature allow us to analyze or detect objects in urban environments. In turn, this level of detail complicates classification tasks. The selection of the best features and of the appropriate classification algorithm is essential for good results. A possible solution for this arduous task, may be the involvement of the user as intelligent agent. User involvement means a development of a interactive interface where the visual analytics concept helps to process the collected information. The user is able to select range of features to obtain a group of interesting images. This idea was implemented via web based interface that allows user to filter features to get the target group of images.
In the following sections we describe a metric resolution urban data set, followed by a description of used features. After this we explain the methodological approach that we use and the designed user interface with some examples, and finally we present some conclusions and future work.

2 METRIC OPTIONAL URBAN DATA

Earth Observation has traditionally focused on environmental monitoring at geographic scales. The main scene classes observable by decametric resolution systems were limited to urban areas, water bodies, forests, agricultural areas, bare soil areas with large extensions.

The detail visible in metric resolution products potentially allows the development of novel applications focusing on mapping at the scale of human phenomena, from detailed environmental dynamics to urban area understanding.

*Corresponding author. This is useful to know for communication with the appropriate person in cases with more than one author

This in turn requires methodologies for being able to deal with the significant increase in scene classes e.g. in urban environments at metric scales. In this contribution, we consider sample metric resolution data acquired on the city of Rome, Italy, by DigitalGlobe systems.

Given the multi-media derivation of the methodology and in an extreme simplification attempt, we disregard multi-spectral information content and only deal with standard tri-stimulus RGB quick-looks instead.

3 PRIMITIVE FEATURES

Image analysis has developed various types of descriptors of different kinds, like color, texture, shape and topology. The first attempt starts with the selection of the simplest descriptors, color. Color is a human subjective sensation of the visible light, depending of the intensity and a set of wavelengths, related with the electromagnetic spectrum. This set of wavelengths define the chrominance values of the visible light for a human. These values are hue and saturation, or the dominant wavelength and purity of color respectively. Color descriptors are the most used features in Content Based Image Retrieval or *CBIR* systems. To select the color descriptors first we need to specify the color space. A color space is a multidimensional space of color components, in our case we work in two different color spaces, RGB and HSV, and dimensions are the three primary colors, red (R), green(G) and blue (B), and Hue (H), saturation(S) and Intensity value (V or I). This approach provides six different histograms which can filter data to obtain the data tiles of interest. Because the R, G, and B components of an image are correlated with each other, object discrimination task with these components is difficult. HSV descriptors are often more relevant. Once color space is selected we select the descriptors. To represent color we employ histogram in both color spaces, RGB and HSV. A histogram describes a distribution of color within a whole image or specific region. Like a pixel-wise feature, the histogram is invariant to rotation, transla-

tion, and scaling of an object but not capture semantic information, so two images with similar histogram can have fully different meaning.

This approach considers the user like a part of selection process, so the facility and usability of different parts of User interface implies simplicity and well organized User Interface.

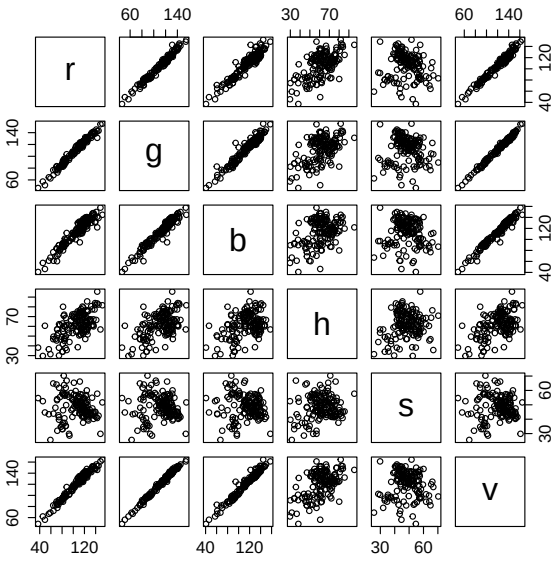

Figure 1: RGB and HSV Histogram Matrix. Each dot represents a image tile in the full product.

4 METHODOLOGICAL APPROACH

(Quartulli et al., 2012) uses a tridimensional viewer in which image tiles are displayed, see Figure 2. The selection task in this viewer was very complicated and is difficult to select interesting images. To facilitate this, we changed the concept and we are going to develop another interface with a gallery of images. In addition to the gallery, the application of visual analysis tools simplifies the selection of interesting image tiles. Therefore, in this first attempt bar charts are applied to display the stored data from the images. The architecture behind the user interface has not changed.

5 ARCHITECTURE

The system architecture is based on a three-tier model.
The data is stored in a NoSQL document database (Plugge et al., 2010) capable of efficient sharding and Map-Reduce job execution.
Communication in between the client user interface and the database is mediated by a highly parallel, asynchronous event-driven web application server. The server implements a RESTful HTTP API allowing web clients to efficiently manipulate the contents of the archive.
The user interface queries the server via the HTTP interface, internally representing the retrieved objects in efficient OLAP-oriented

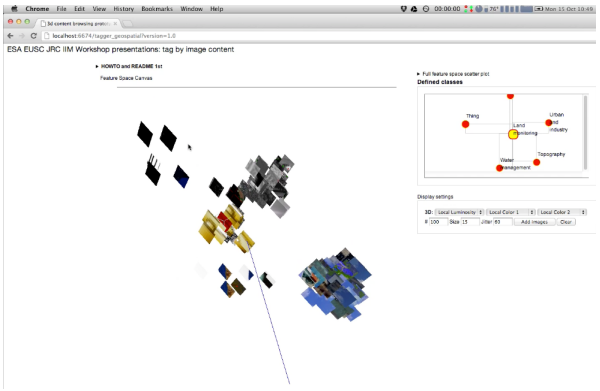

Figure 2: UI based on 3D image tile gallery: User interaction for the selection of elements of interest requires complex navigations skills in a high dimensional space. 2D interfaces are easier to use.

formats.
When the user access the web page, all data needed for visualization are loaded at the same time. Once the data is loaded, the filtering action does not require data reload for the data base, making the iteration more dynamic and responsive.
As seen in Figure 3, the architecture is a classical client server architecture. In the server side MongoDB and NodeJS are used as data base and server. In the client side, Dc.js together with D3.js and Crossfilter.js are the responsible for displaying data on a web page.

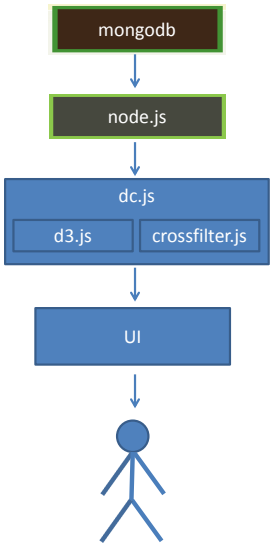

Figure 3: High level System Architecture: Image tile meta data are stored on a NoSQL database served by an event driven asynchronous RESTful interface. Coordinated views with 2D brushing in the client are based on a client side OLAP engine.

6 UI DESIGN

As mentioned in the Abstract, to exploit this application the user is not required to be a expert in remote sensing, so the user interface has to be very intuitive and easy to use.
As you can see in Figure 4, the User Interface has two parts, result panel or Gallery and filters or control panel. At the first time the Gallery shows a random set of image tiles. When the user filters or selects different ranges in the descriptors, the gallery is updated according with the selection made.

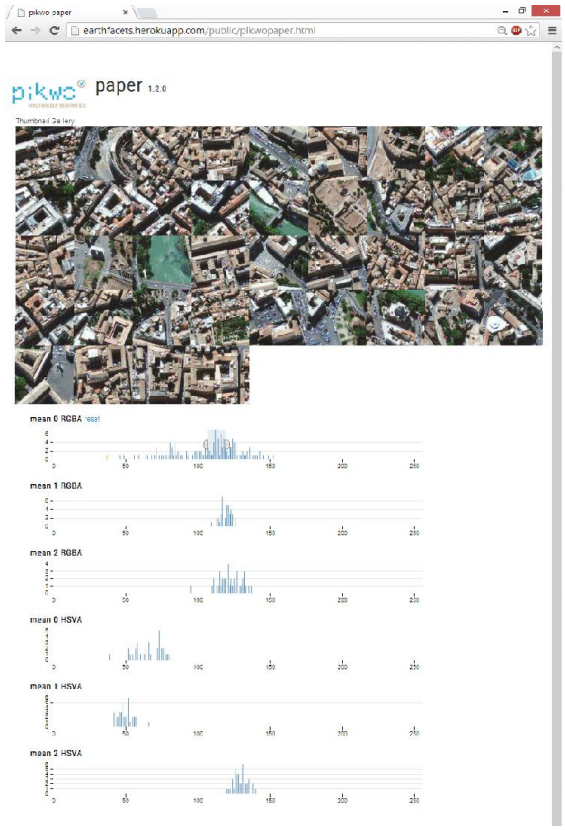

Figure 4: User Interface: a image gallery on top shows a random selection of image tiles. User can interact with the gallery by the bottom half of the interface. This selection panel presents separated histograms for the complete tile set for each one of the calculated features. By clicking and dragging on those histograms the user selects a region of interest in the feature space that can be refined and optimized by moving the limits or the centers of the selections. Each selection operation produces real time update in both the image gallery and in the histograms values displayed. Multiple concurrent selections are allowed to specify a selection hypercube in feature space.

One of the interesting features that the GUI has is the coordinated and interactive functionality that all displays, like Gallery as the Image Descriptors have. When user selects or filters a range of data in one of the descriptors, the gallery and the rest of descriptors update the information displayed in real time. This is possible thanks to Dc.js. Dc.js is a Javascript charting library with native Crossfilter(Crossfilter, Fast Multidimensional Filtering for Coordinated Views, URL: http:
square.github.io, 2012) support and allowing highly efficient exploration on large multi-dimensional dataset. It leverages the D3 (Bostock et al., 2011) engine to render charts in CSS friendly SVG format. Charts rendered using Dc.js(Zhu, 2012) are naturally data driven and reactive therefore providing instant feedback on user's interaction. The main objective of this library is to provide an easy yet powerful Javascript functions which can be utilized to perform data visualization and analysis.
The gallery is based on dataTable element that Dc.js has. The changes have been minimal, the table elements have been replaced for simple HTML DIV and returned data for HTML HREF and image elements.

7 EXAMPLES OF USE

As explained in section 4, Methodological Approach, the Visual Analytics make easier the analysis of huge data stores. In this approach data came from satellite images from the center of Rome, with a 1 pixel per meter resolution. The dimension of image tiles is 100x100 pixels, it is arbitrary measure for this first approach.
If we take a look, to some of these images, we can detect different elements, like an historical and new buildings, trees and gardens, river and roads among other things. These elements can be classes for classification task. That being so, the classification task was greatly complicated, because in the same tile can appear all the elements, we mentioned before and more, that can be classified.
In figure 5, the selected ranges in different histograms result a three images of the Tiber river.

In the next figure 6, the selection results in image tiles containing trees.

Finally in figure 7,the selection results in image tiles containing historical buildings.

8 EVALUATION AND CONCLUSIONS

This first approach confirms the usability of Visual Analytics for Big Data analysis. The functionality and usability of the web interface was proved.
Even so, the User Interface require a new design owing to the dynamic that has the gallery. Dc.js use implies that different elements be updated in real time. A design with the histograms, or filterable features, on the left and the gallery on the right must be more useful and easy to use.
As expected, the handles data in this attempt isn't the best. First, the dimensions of analyzed images are so large, for pixel resolution that we have. A simple tile can contain a lot of elements or classes that can be target. And second, the used descriptors are so basic to distinguish some pictures of other. At the same time as we change the dimension of image tile, the use of texture descriptors starts to make sense, if we have in mind that the roofs of the buildings and vegetable areas, like a forest, follow a similar pattern. This will be considered in an updated version of this system.

REFERENCES

Bostock, M., Ogievetsky, V. and Heer, J., 2011. D3 data-driven documents.

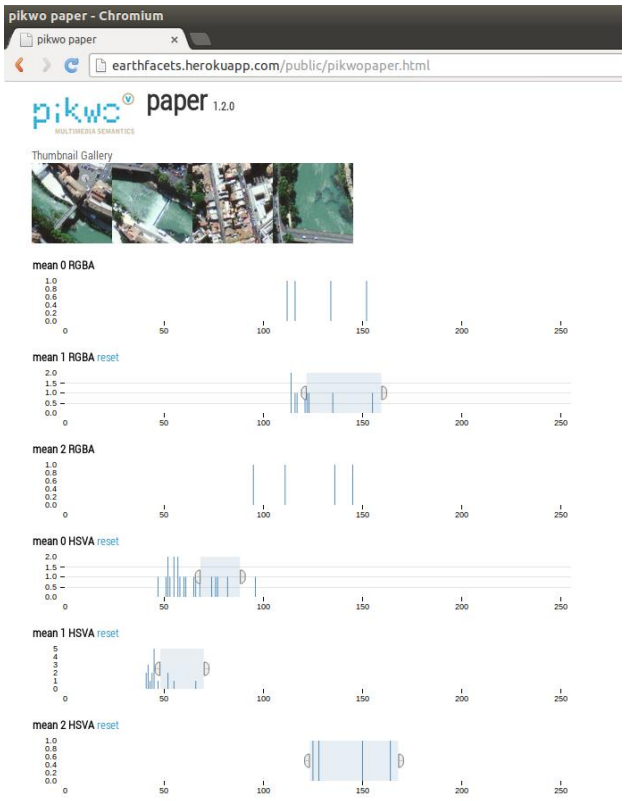

Figure 5: Search/Filtering results for class "River": image tiles are identified as highly luminous areas with significant saturation and strong green component. Note how selection operations have produced an update in both the image gallery and in the histograms.

Crossfilter, Fast Multidimensional Filtering for Coordinated Views, URL: http:
square.github.io, 2012.

Plugge, E., Membrey, P. and Hawkins, T., 2010. Introduction to mongodb. The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing pp. 3–17.

Quartulli, M., Zorrilla, M. and Olaizola, I., 2012. On the image content of the esa-eusc-jrc workshop on image information mining. ESA-EUSC-JRC 8th Conference on Image Information Mining:Knowledge Discovery from Earth Observation Data pp. 70–74.

Zhu, N. Q., 2012. dc.js dimensional charting javascript library, url:http://nickqizhu.github.io/dc.js/.

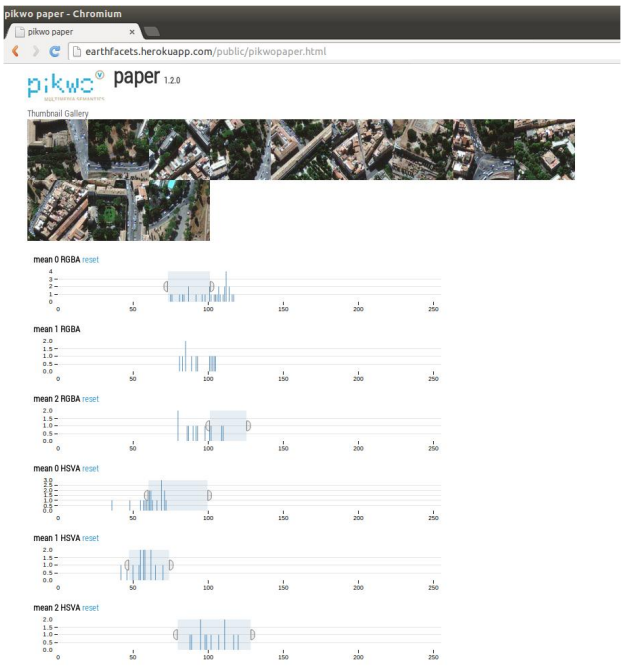

Figure 6: Search/Filtering results for class "Trees": Tree areas are identified by low luminosity areas with high saturation, provably because of the shadows that are visible in them. The selection of a specific Hue segment makes it unnecessary to use the Green component for filtering.

Figure 7: Search/Filtering results for class "Historical Buildings": the selection of specific Hue and luminosity intervals are sufficient for the selection of this very populated image class in our example data set.