MetricSPlat - A platform for quick development, testing and visualization of content-based retrieval techniques

Jose Fernando Rodrigues Junior[1], **Luciana A. M. Zaina**[1], **Luciana A. S. Romani**[2], **Ricardo Rodrigues Ciferri**[3]

[1] Universidade Federal de São Carlos at Sorocada

[2]Embrapa Agriculture Informatics at Campinas

[3]Universidade Federal de São Carlos at São Carlos

{junio,lzaina}@ufscar.br, luciana@cnptia.embrapa.br, ricardo@dc.ufscar.br

Abstract. *The development and testing of content-based data retrieval systems is a time-consuming task. Over the concept of metric space, such systems must integrate the three factors that define an indexing environment. These factors are features extraction, metric structures and distance functions, not to mention a suitable user interface. This integration deviates the work from the real focus of research, suppressing quick experimentation of ideas. In this context, we present the Metric Space Platform (MetricSPlat), a system designed for content-based retrieval enabled with plug-in features. With minimal effort, MetricSPlat substantially speeds up the experimentation of new techniques by providing a well-defined framework aided with interactive data visualization techniques.*

1. Introduction

A major line of research on content-based data retrieval techniques relies on the possibility of similarity queries. Such queries work by searching a dataset tracking for the elements that, according to a specific criterion, are similar to a given data element. This kind of functionality has a large scope of application, from bioinformatics to complex-data storage systems [Rodrigues Jr. et al. 2008]. Similarity queries are useful when the data do not define an order relation. This is the case, for example, in a set of images; different from a set of numbers or letters, one cannot specify an absolute ordering for images. In order to bypass this limitation, metric spaces are the basis of a well-known framework used to extend the exact-searching paradigm, traditionally offered in database systems.

In this work we present the Metric Space Platform, or MetricSPlat for short (`http://gbdi.icmc.usp.br/~junio/MetricSPlat/index.htm`). MetricSPlat is a system that integrates the components that define a metric space so that content-based retrieval can be accomplished in a visual interactive environment, see Figure 1. It uses plug-in features and coding frameworks to permit the fast development and testing of features extraction techniques, distance functions and metric structures. The aim of MetricSPlat is to reduce the burden of raising a content-based retrieval system; at the same time, the system offers visual functionalities to enhance the comprehension of the techniques and of the data.

2. Basic concepts

The most referenced framework for content-based data retrieval is the use of metric spaces. This framework is the basis for organizing datasets from a wide range of domains.

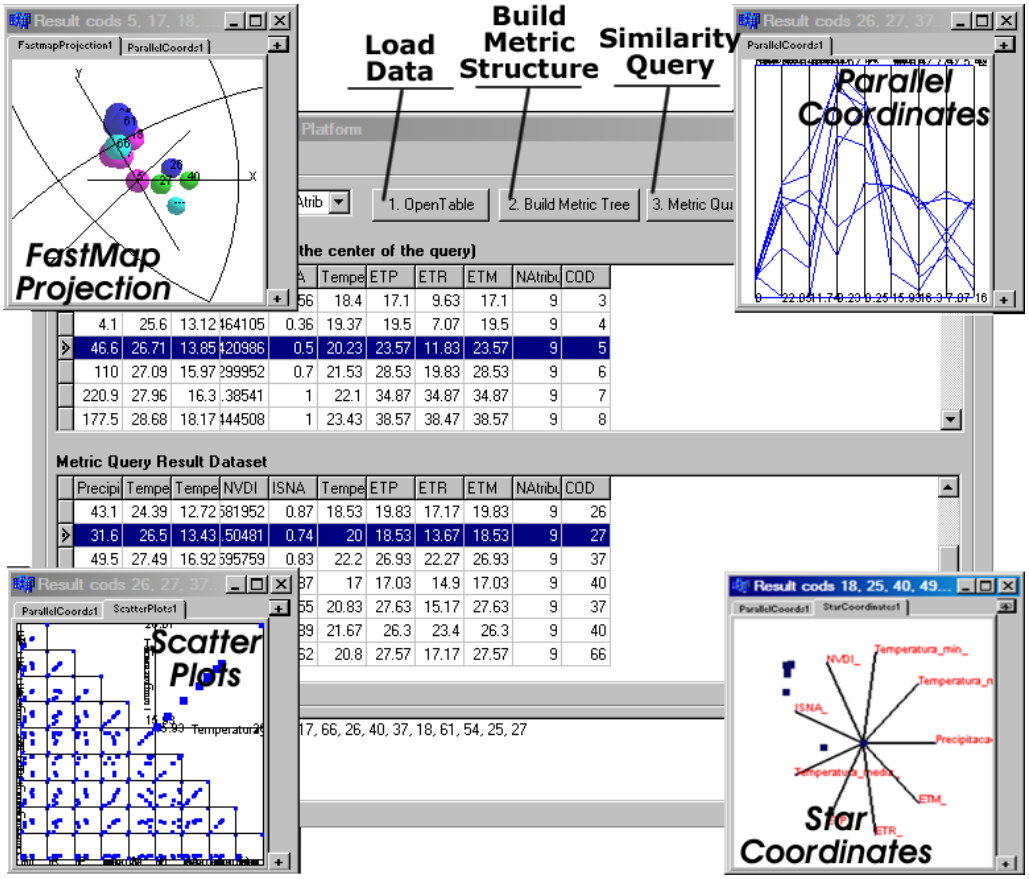

Figure 1. The MetricSPlat platform. Metric space content-based data retrieval aided with data visualization techniques. (a) FastMap multidimensional projection. (b) Parallel coordinates. (c) Scatter plots. (d) Star coordinates.

2.1. Features extraction

In the present work, the first thing, in order to index a set of complex data, is to translate them into a numerical representation. This is called *features extraction* and it corresponds to processing the data so that a vector of n representative numbers is extracted from them. Classical examples, in the case of images, are the color histogram [Felipe et al. 2005] and the coefficients achieved with the Fourier transform [Zhang and Lu 2001]; both of them define a vector of numbers. Along this text, we refer to features extraction in general as a function $f : D \rightarrow \mathfrak{D}$, where D is a domain of specific data, e.g. images, and $\mathfrak{D} \subset R^n$ is an n-dimensional features space.

2.2. Distance function

The second step in order to define a metric space is to establish a similarity measure, or distance function, among the vectors of numbers extracted from the data objects. A trivial way to do this is to consider each numerical feature as an n-dimensional coordinate and calculate the Euclidian distance among the vectors. Other examples of distance functions are the City Block and the Minkovisk distances [Aggarwal et al. 2001].

2.3. Metric space

Once feature vectors and a distance function are specified, a metric space is established. A metric space refers to a set in which the notion of distance among its elements is well-defined. Formally, a metric space is a pair $M =< \mathfrak{D}, \delta() >$, where $\mathfrak{D}$ is the domain of the elements to be indexed and $\delta : \mathfrak{D} \times \mathfrak{D} \rightarrow \Re^+$ is a function that associates a distance to any pair $o_i, o_j \in \mathfrak{D}$. Moreover, the pair $M =< \mathfrak{D}, \delta() >$ is named metric space whenever the function $\delta()$ satisfies the following axioms:

1. **Symmetry**: $\delta(o_1, o_2) = \delta(o_2, o_1)$
2. **Non negativity**: $0 < \delta(o_1, o_2) < \infty$ if $o_1 \neq o_2$ and $\delta(o_1, o_1) = 0$
3. **Triangular inequality**: $\delta(o_1, o_2) \leq \delta(o_1, o_3) + \delta(o_3, o_2)$

2.4. Similarity queries

Over a metric space, it becomes possible to perform similarity queries. That is, given an element of interest – the center of the query – what are the elements of the dataset with smaller distances (higher similarities) to this element. The two basic similarity queries are the *nearest neighbor* – the closest element to the query center – and the *range query* – the elements that are not further than a specific distance.

Definition 1 (Nearest-Neighbor query): Given a query object o_q represented by its features vector $f(o_q)$, and the set of data elements D, the nearest neighbor is the element of D such that $NNQuery(o_q) = \{o_n \in D | \forall\ o_i \in D, \delta(f(o_q), f(o_n)) \leq \delta(f(o_q), f(o_i))\}$. An example of a nearest neighbor query in an image database is: "find the image in D that is the most similar to image A".

Definition 2 (Range query): Given a query object represented by its features vector $f(o_q)$, the set of data elements D, and a maximum search distance r, the answer of the range query is the subset of D such that $RQuery(o_q, r) = \{o_i \in D | \delta(f(o_i), f(o_q)) \leq r\}$. An example of a range query is: "find the images in the database that are at least 80% similar to image A".

2.5. Metric structures

Answering similarity queries corresponds to tracking the nearest elements to a chosen query center, according to the similarity measure being used. It can be done sequentially, calculating the distance from the center to all of the other elements in the dataset or, alternatively, with the aid of a metric structure. A metric structure is a data structure specifically designed to index the elements of a metric space reflecting the distances among them. The benefits of a metric structure come from the reduced number of distance calculations necessary to answer a query. Well-referenced such structures are the Slim-Tree [Traina et al. 2000] and the DBM-Tree [Vieira et al. 2004].

The use of metric spaces in the task of content-based retrieval, then, involves three techniques: features extraction, similarity measure (distance function) and metric structures. Together, they form a framework where different kinds of data objects can be indexed and retrieved over the concept of similarity.

3. The metric space platform components

The test and validation of techniques for content-based data retrieval demands the building of a system that puts together efforts on features extraction, metric data structures and distance functions. This is not a trivial task, but a task that adds extra complexity to the research and development of such techniques. In this section we describe how MetricSPlat defines a framework that is not another *ad hoc* instance of the metric space systematization but, rather, a platform to accommodate new conceptions in an extensible way.

3.1. Features extraction

In order to appropriately use features extraction and have it in a manageable format, MetricSPlat uses a dataset facility. This way, the system can perform similarity queries over any given set of extracted features. To do so, the features must be in a specific tabular format. The format requires a table in which the n first fields correspond to the numerical features. Besides the features, it is necessary to have two extra fields named *NAtributos* and *COD*. Field *NAtributos* tells the software how many of the first fields carry features; field *COD* is an integer index. Along with the software bundle of our system, the user will find a set of datasets from the *UCI Machine Learning Repository* – `http://archive.ics.uci.edu/ml/`; the datasets are in the appropriate format and can suitably demonstrate the functionalities of MetricSPlat.

The system uses a package of database components that works standalone, with no installation, nor driver issues; the package is named *ZeosLib* – `http://sourceforge.net/projects/zeoslib`. In MetricSPlat, it is set to read a file named *Features.mdb* expected to be in the same directory of the *MetricSPlat.exe* file. According to this configuration, any set of extracted features, in text format for example, can be configured to MetricSPlat with the aid of free software suites as, for example, OpenOffice – `http://www.openoffice.org`.

3.2. Metric data structure and distance function

In MetricSPlat, we benefit from an Application Programming Interface (API) of metric structures named *Arboretum* – `http://www.gbdi.icmc.usp.br`. Arboretum provides a set of ready-to-use distance functions and metric structures and it also supports the definition of new instances of these concepts. Its API defines a set of software interfaces to ease the usage and development of the components that support metric spaces and similarity queries. It is the core of the functionalities of MetricSPlat; therefore, any new technique that is supposed to work in MetricSPlat must comply with the Arboretum interface.

In order to make MetricSPlat easily extensible to new functionalities, we have developed it around a Dynamic Linking Library (DLL) framework. This framework is defined in compliance to the Arboretum API; this is done in such a way that new dll files will automatically hook up to the system simply by copying them to the MetricSPlat working directory. In order to create new dlls, the MetricSPlat project provides a customizable compilation subproject, available in the website of the system. With this compilation project, new dll files may be created with minimal effort. It is necessary only to recompile the dll project altering the name of the desired metric structure. MetricSPlat will automatically recognize new dll files making them available for testing and debugging.

Adding a new distance function to MetricSPlat follows the same procedure as that of adding a new metric structure. In fact, metric structures must be provided in combination to distance functions. For example, it is possible to create pairs of the form Slim-Tree-Euclidean or DBM-Tree-City Block.

3.3. Data visualization

Complex data, such as images and text, are the focus of content-based data retrieval. They usually define volumes of features vectors that are not prone to be easily read, due to its cardinality and format [Rodrigues Jr. et al. 2007]. As so, MetricSPlat also offers a set of visualization techniques to permit the visual inspection of such data. These techniques can be used for visualizing either the result set that is returned when similarity queries are performed or either the set of extracted features that define the search space.

Visualizing the result set returned in similarity queries:

- FastMap spatial projection: this technique is based on the dimensionality reduction algorithm named FastMap [Faloutsos and Lin 1995]; it is used to reduce the dimensionality of the features dataset. Once the dataset has its dimensionality reduced to 3 dimensions, it becomes possible to have it in a 3-dimensional plot space. In such representation, one can observe clusters, exceptions and, more important, the spatial distance-based placement of the data. This possibility corresponds to the visual observation of a metric space allowing to see how a similarity query works and what are its results, see Figure 2.

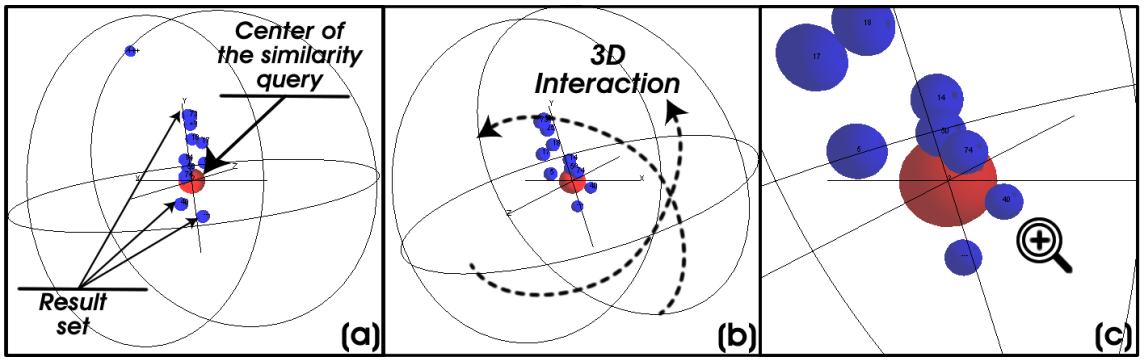

Figure 2. Visualization of a result set in a 3-dimensional environment. (a) The query center as the origin of the projection. (b) 3D interaction. (c) Zooming capabilities.

FastMap is an instance of multidimensional projection, that is, a function $m : \mathfrak{D} \rightarrow \Re^3$ whose goal is to minimize the difference between functions δ and the Euclidian distance d, that is:

$$Argmin_m(\sum_{i,j} |\delta\{(o_i, o_j)\} - m\{(f(o_i), f(o_j))\}|) \quad (1)$$

Visualizing the extracted features that define the search space:

- Parallel coordinates [Inselberg and Dimsdale 1990]: this technique is adequate for characterizing a vector of features the same way as temporal data in the form of vectors, that is, in a signal-like plot;

- Scatter plots: this classical technique can reveal correlations among the features by ploting the dimensions in pairwise fashion;
- Table lens [Rao and Card 1994]: permits the visual presentation of tabular data with the aid of graphical entities, promoting comprehension of the correlation and of the distribution of the data;
- Star coordinates [Kandogan 2000]: this is a technique whose interactive nature permits to define scatter plots with more than two coordinates, augmenting the possibilities of axial plotting.

3.4. Creating new pairs metric structure-distance function

The creation of new pairs consisting of a metric structure and a distance function is the basis of MetricSPlat flexibility. Such pairs are packed into dll files that can be plugged into MetricSPlat and be put into production with no need to recompile the system. Since this is an important feature of the project, the website of the tool (`http://gbdi.icmc.usp.br/~junio/MetricSPlat/index.htm`) offers a dll ready-to-compile project and an illustrative video that shows, step-by-step, how to create a new dll in less than 3 minutes.

4. Conclusions and future work

MetricSPlat, in its present configuration, can furnish the use and comprehension of content-based retrieval techniques with reduced effort. At the same time, it is a visualization tool with several possibilities. Notwithstanding, there are further developments to be achieved in this project. Two first needs are: to have features extraction plug-in capabilities and to have plug-in capabilities that will present the original data that was used in the features extraction process as, for example, the exhibition of images. The first would bring the pre-processing step into the tool, making it a broader repository of techniques; the second would provide more interaction and data management by permitting full data inspection.

Another possible work is to benefit from the visual interactive nature of MetricSPlat in order to deal with content-based retrieval aided with relevance feedback. This kind of methodology is useful for fine-tuning distance functions that gradually will reflect the user perception of the retrieval system. In this context, the user would be able to visually tell the system what are the most and what are the less relevant results returned by the similarity query system.

Aknowledgements Thanks to Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes).

References

Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional spaces. In Bussche, J. V. d. and Vianu, V., editors, *International Conference on Database Theory (ICDT)*, pages 420–434, London, UK. Springer Verlag.

Faloutsos, C. and Lin, K.-I. D. (1995). Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In Carey, M. J. and Schneider, D. A., editors, *ACM SIGMOD International Conference on Management of Data*, pages 163–174, San Jose, CA. ACM Press.

Felipe, J. C., Traina, A. J. M., and Traina Jr., C. (2005). Global warp metric distance: Boosting content-based image retrieval through histograms. In Tsai, J. J. P., Muhlhauser, M., and Takizawa, M., editors, *IEEE International Symposium on Multimedia-ISM2005*, page 8, Irvine, CA. IEEE Press.

Inselberg, A. and Dimsdale, B. (1990). Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *IEEE Visualization*, volume 1, pages 361–370. IEEE Computer Press.

Kandogan, E. (2000). Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *IEEE Symposium on Information Visualization*, pages 4–8.

Rodrigues Jr., J. F., Balan, A. G. R., Traina, A. J. M., and Traina Jr., C. (2008). The visual expression process: Bridging vision and data visualization. In *8th International Symposium on Smart Graphics*, volume 5166, pages 207–215, Rennes, França. Lecture Notes in Computer Science. Berlin Heidelberg : Springer-Verlag.

Rodrigues Jr., J. F., Traina, A. J. M., Oliveira, C. F., and Traina Jr., C. (2007). The spatial-perceptual design space: a new comprehension for data visualization. In *Information Visualization*, volume 6, pages 261–279.

Rao, R. and Card, S. (1994). The table lens: Merging graphical and symbolic representation in an interactive focus+context visualization for tabular information. In *Proc. Human Factors in Computing Systems*, pages 318–322.

Traina, Caetano, J., Traina, A. J. M., Seeger, B., and Faloutsos, C. (2000). Slim-trees: High performance metric trees minimizing overlap between nodes. In Zaniolo, C., Lockemann, P. C., Scholl, M. H., and Grust, T., editors, *International Conference on Extending Database Technology (EDBT)*, volume 1777 of *Lecture Notes in Computer Science*, pages 51–65, Konstanz, Germany. Springer.

Vieira, M. R., Traina Jr., C., Traina, A. J. M., and Chino, F. J. T. (2004). Dbm-tree: A dynamic metric access method sensitive to local density data. In Lifschitz, S., editor, *Brazilian Symposium on Databases (SBBD)*, volume 1, pages 33–47, Brasilia, DF. SBC.

Zhang, D. S. and Lu, G. J. (2001). Shape retrieval using fourier descriptors. In *Intl. Conference on Multimedia and Distance Education*, pages 1–9, Fargo, ND, USA.