

# M-Cube: A Visualization Tool for Multi-dimensional Multimedia Databases

**André Maximo**  
COPPE/UFRJ  
andmax@cos.ufrj.br

**Maria Paula Saba**  
ESDI/UERJ  
msaba@esdi.uerj.br

**Luiz Velho**  
IMPA  
lvelho@impa.br

## ABSTRACT

The last decade was marked by a striking growth on database size and dimension. This increase is noticeable in many areas, ranging from personal data storage to large corporation databases. The size and high dimensionality of these data sets prompt the application of specialized graphical representations rather than tables and generic charts to visualize the data. This specialization limits the representation to a particular type of database, but ensures a better understanding of its data.

In this paper, we present an innovative visualization tool for multi-dimensional multimedia databases, called *Multi-dimensional Cube* (or *M-Cube*). Unlike previous data cubes, which are simple extensions of two-dimensional tables or charts for any type of data, the M-Cube employs a three-dimensional space to visualize multimedia content with great interaction flexibility – the user can change any of the current dimensions by rotation. Our tool is specialized for multimedia contents, that is, text, music, images and videos, allowing the M-Cube to improve both visually and interactively. While the visualization is enhanced by visually rich data elements inside the cube, the interaction provides fast change of dimensions required when exploring multimedia databases.

## Author Keywords

HCI, Visualization Tool, Interaction, Multi-dimensional Database, Information Design.

## ACM Classification Keywords

H.5.2 [HCI] User Interfaces: Graphical user interfaces; Input devices and strategies.

## INTRODUCTION

In the last decade, we witnessed a fast growing in digital data supported by a constant increase in data storage devices as well as larger communication bandwidths. This striking growth appears not only on the size of databases but also on the number of attributes classifying the raw data. The data attributes, or *metadata*, decompose the corresponding data set in dimensions, playing an important role on both fronts: large data sets visualization and querying. The complexity behind these tasks lies on human-computer interfaces (HCI) to employ metadata in a meaningful and consistent way. In this work, we explore both visualization and HCI to build an innovative tool to deal with multi-dimensional multimedia databases.

The visualization activity on databases aids the creation of queries which, in turn, work to produce better visualizations. This recurring process aims to extract significant information from a data set. It has in principle an exploratory context, where the user interacts with the data set searching for a specific content or, conversely, the user wishes to analyze the data in a more generic form. The most common techniques employ relational tables and textual languages in order to carry out these interactions, that is, establish visualization and query, respectively. The result is a poor visualization with difficult interaction, as can be seen in Figure 1.

One of the features which make large database visualization especially challenging is its inherent high dimensionality. Interesting enough, the n-dimensions can be used to avail the visualization itself by means of data classification. The classification is done by projecting data elements in each dimension, which leads to an n-dimensional chart, where n is the number of attributes or database dimensions. For instance, a simple database with 3 attributes could be plotted in a 3-d chart. The chart presents a better database visualization than the table, depending on the data visualized and the exploration process.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2009, April 4–9, 2009, Boston, Massachusetts, USA.  
Copyright 2009 ACM 978-1-60558-246-7/09/04...\$5.00.

```
SELECT *
FROM supplies, location
WHERE supplies.location_id = ...
```

	PAPER	PEN	PENCIL	RUBBER	SCISSOR
RESTAURANT	134	14	32	7	114
HIGH SCHOOL	379	55	21	82	27
COLLEGE	561	72	55	81	52
OFFICE	678	77	40	58	63
STORE	1220	107	25	90	102
LIBRARY	421	11	52	9	41

**Figure 1: Example of a query textual language (upper box) and a table relating office supplies with different client types.**

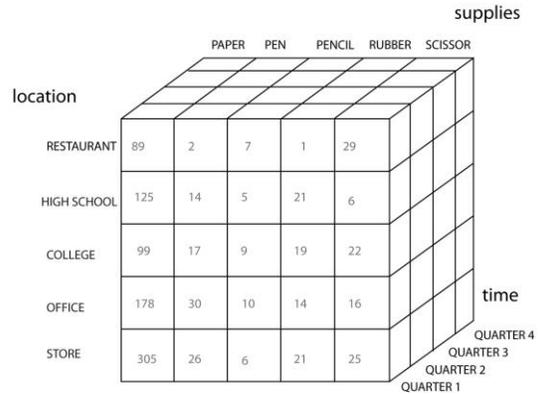
The type of data is an important aspect to analyze when building a database visualization tool. However, the majority of interfaces available ignore the data type, leading to an inadequate visualization. Examples of such interfaces include: table with numbers (see Figure 1) [7], discretized or aggregated; graphs with bar charts, such as histograms [7]; scatter plots with icons or glyphs, varying in color, size, etc. [14]; data cubes [15]; and so on. A usual problem arises when dealing with complex types of data, for instance media data are poorly visualized with generic tools [1]. In this paper, we are interested in visualizing multimedia data inside a relational multi-dimensional database.

One tool often employed in multi-dimensional database visualizations is a dynamic table with numbers in cells, called Pivot Table [7]. These tables can be organized in data cubes, as shown in Figure 2, where each dimension of the relational database can be rotated, or pivoted. Once the pivots are set in rows and columns of the table, the dimensions left are aggregated and the results are shown as numbers or plotted as graphs. One advantage of tables over graphs is the freedom to apply a desirable order to the data, while charts present data in a fixed sequence, depending on the dimension. Nevertheless, the common desirable order is intrinsic to the dimension and thus respected by the chart.

The interaction on pivot tables and data cubes fails to grant the user easy ways to change pivots and make different views of the database quickly. The problem is that it is based on a limited visualization of the data set – relational tables – in order to be generic enough to deal with any type of data. This limitation leads to difficult ways to change pivots, e.g. *typing* or *drag-and-drop* dimensions, and both visualization and interaction become compromised.

In this paper, we present the *Multi-dimensional Cube*, dubbed *M-Cube* (or  $M^3$ ), a visualization tool for multi-dimensional multimedia databases. M-Cube extends the data cube representation by offering a three-dimensional space to visualize and explore multimedia data. In addition to normal actions, such as opening a media data, the M-Cube allows three new interactions:

- **Rotation** – the cube can be rotated for a better view of the data or to change the current dimensions, similar to change pivots in pivot tables;
- **Selection** – parts of the cube edges can be chosen in order to narrow the current view or produce a new visualization, working as queries in the recursive process of exploration;
- **Zooming** – regions inside the cube can be magnified for a proper view of the data, allowing to change fast from a generic analyze to a more specific one.



**Figure 2: Example of data cube visualization.**

M-Cube also employs innovative database visualization, taking advantage of the specialized data type. The data elements inside the cube are displayed representing its type and hinting its content, for example, an image data set is displayed with each data element as the actual picture inside a frame. Other visual improvement appears when grouping or aggregating data elements, different media objects can be grouped making a new object, for instance several music tracks could be aggregated to a song album. Note, however, these improvements can only be achieved with specialization of our visualization tool to one type of data: multimedia content.

The visualization and HCI of the M-Cube are simple and intuitive. The user employs natural actions to interact with the M-Cube interface and have a meaningful and rich graphical response from the visualization. This helps the interaction and the overall exploratory process. M-Cube can be used to analyze a complete multimedia database as well as browse its data searching for a specific content.

The remainder of this paper is organized as follows. In the related work section we discuss previous work on database visualization and interaction techniques. Next, we describe the design principles behind our tool. While, in the following sections we present the M-Cube and its results. Finally, in the last section we conclude providing several future work directions.

## RELATED WORK

The area of multi-dimensional database visualization receives special attention in both ventures: commercial products and research studies. One example on commercial trends is the *Excel*<sup>TM</sup> application [7] from *Microsoft*<sup>TM</sup>. This application employs pivot tables with different projections of large multi-dimensional databases. Each projection defines a new dimension, or pivot, for a row or column of the table. The interaction is made by text-based fields forcing the user to enter new pivots by typing, making the interface difficult to use.

Another example on commercial trends is the *Spotfire*<sup>®</sup> application [10]. This application from *TIBCO*<sup>®</sup> targets to allow interactive analysis of multi-dimensional data. In the *Spotfire*<sup>®</sup>, users can organize data in pages containing a set of restricted visualizations, such as descriptive text areas, details-on-demand and filtering panes. The limitation on visualization drives the user to follow a particular path of data exploration without freedom.

One example on research ventures is the *Multi-dimensional Data Viewer* [12]. This application maps visual objects in a 3D space to a final image by projecting the 3D scatter plot accordingly to a given point-of-view. Users can interact with the plot by rotation, changing the point-of-view and thus the final image. The data elements are displayed as glyphs, which have different visual features (such as size and color), creating layers of representation on top of data types. These layers make the real meaning of the data difficult to distinguish. Another problem arises when trying to choose a good point-of-view; the user may get the wrong data multiple times in the final image.

Another application based in scatter plot graphics is the *Scatterdice* [3]. In this application, users can explore information in a matrix of 2D scatter plots and browse the data in a structured multi-dimensional space. It uses 3D animation to navigate among plots, and apply bounding volumes to create queries, allowing the user to explore the database by clipping the plots. Dimensions in navigation space are reordered to highlight correlations and differences among them. The main disadvantage of this approach is a confusing visualization due to many different scatter plots on the same view.

The *Dynamic Homefinder* application [18] employs an alternative way of interaction. It provides database searching using a more direct manipulation, called dynamic queries. This type of query is made by interactively adjust graphical widgets, such as sliders, to produce better visualizations. It is interesting to note that even though the HCI of the *Dynamic Homefinder* is inadequate, this dynamic technique is applied nowadays as a standard form of interaction, and it is used in data cubes and in our visualization tool.

The *Polaris* system [14] presents an interface for exploring large multi-dimensional databases. *Polaris* is based on the construction of table-based graphical displays through

consecutive queries. It also exploits traditional 2D charts adding to them an algebraic formalism based in Bertin's graphical properties [2]. Users can choose among basic visual principles for data visualization. At this point, the HCI of the system suffers with drag-and-drop widgets, while the visualization is limited by two-dimensional tables and charts.

Stolte et al. [15] describe a multiscale visualization of data cubes, building it on top of their *Polaris* formalism. They apply data and visual abstraction in a general mechanism in order to deal with many types of data sets. The multiscale feature is combined with independent zooming at different scales along one or more dimensions. As the *Polaris* system, it is based on tables and data cubes yielding to a poor visualization and ineffective HCI.

The zooming interaction of the M-Cube is also explored by two consecutive works; *Magic Lens* [16] and *3D Magic Lens* [17]. They use the concept of a magnifying glass to distinguish data objects packed together in a small space. This interaction, in 2D or 3D, is important when dealing with large databases, where there are more regions with many data elements making the visualization difficult. Our M-Cube addresses the problem of large database visualization by using a zooming interaction, as described in the M-Cube section.

One interesting mathematical technique used to improve multi-dimensional database visualization and interaction is the dimensionality reduction. *Hypercell* [11] describes a filtering approach to reduce a database dimension and achieve a simplified interface. It provides tools for extraction of subspaces, called cells, from the original database, reducing the overall complexity and organizing filtered cells as workspaces. Other mathematical-based effort is the *Wolfram Alpha* [21] system, by applying built-in algorithms it allows computational queries to be done on a huge data collection. In this paper, we use the idea of dimensionality reduction but in an interface driven way, improving it with a fast switch of current dimensions.

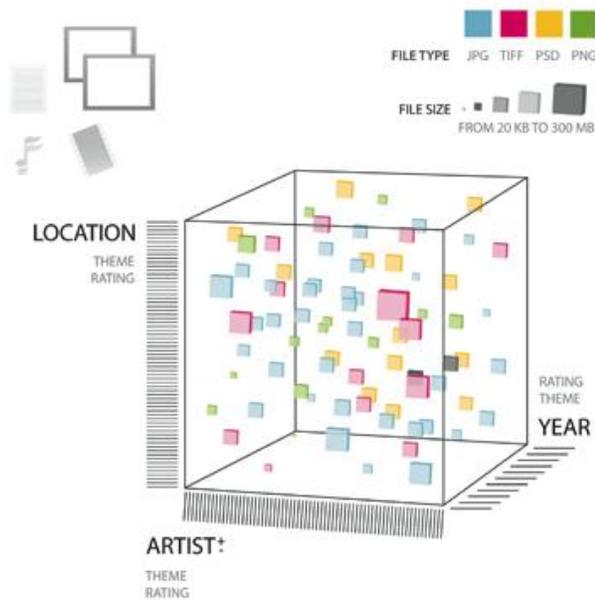
*Filter Flow* [8] explores a visual query language to provide a simple interface through the usage of images. It uses pictorial icons in the visualization and applies filter operators as the result of queries. The visual query paradigm is also explored in this work, and filtering is achieved by combining one or more selections in a certain view with changing of dimensions and others selections in different views.

An important commercial trend is the *iTunes*<sup>TM</sup> [1] from *Apple*<sup>TM</sup>, which implements a table-based interaction to deal with a multi-dimensional music database. Even though it is a personal database, interaction and visualization are compromised by an arduous interface, where the user has to fill attributes in non-intuitive windows. The software tries to evade this problem by offering an artificial intelligence tool to manage the database for the user, going in a complete opposite direction of an effective HCI.

## DESIGN PRINCIPLES

In this section, we present conceptual ideas for the development of the M-Cube: a visualization tool for multi-dimensional multimedia databases. M-Cube employs a 3D space, which is more natural and richer than a 2D table, outlined by the edges of a cube.

The data elements are projected into this space, as normally done in a three-dimensional scatter plot. The projection result is a floating 3D object inside the cube. The tool allows a natural rotation of the space, as it was a real cube, in order to better visualize the data objects. The M-Cube also allows, in the same rotation interface, the switching of the current three dimensions being used to project the data. The user chooses a side dimension, i.e. an attribute not in use, in a preferred axis, rotates the cube with it and the axis become the chosen dimension changing the visualization instantly. For example, the user may choose the *theme* attribute in the *year* axis, shown in Figure 3, rotate the cube from right to left switching the current view to become *location*, *artist* and *theme*.



**Figure 3: A design example of the M-Cube, the current media is images represented by two boxes on the top left corner of the interface.**

The action of choosing in the M-Cube can be accomplished by mouse clicking or using a touchable interface. The latter is better than the former, in the case of the M-Cube, since the gesture related to change pivots, or dimensions, is more natural and intuitive; the user chooses a side dimension by touch and rotates the cube while touching it. On the other hand, the user has the option to touch any region inside the cube (excluding axes and border labels) and apply a gesture to rotate the cube without changing dimensions.

Data elements are normally displayed as glyphs, i.e. objects with different shapes and colors, without any meaning related to the data. For example, the Polaris system [14] allows the user to choose among geometric shapes, such as triangles and circles, to represent data elements, such as cities and states. In the M-Cube shown in Figure 3, we display image elements as boxes with different colors, where each color corresponds to an image file type. On the other hand, the glyph sizes are proportional to the image file size. Moreover, we add a preview visualization of the image inside each box, discussed in the next section, improving the meaning of the data glyph.

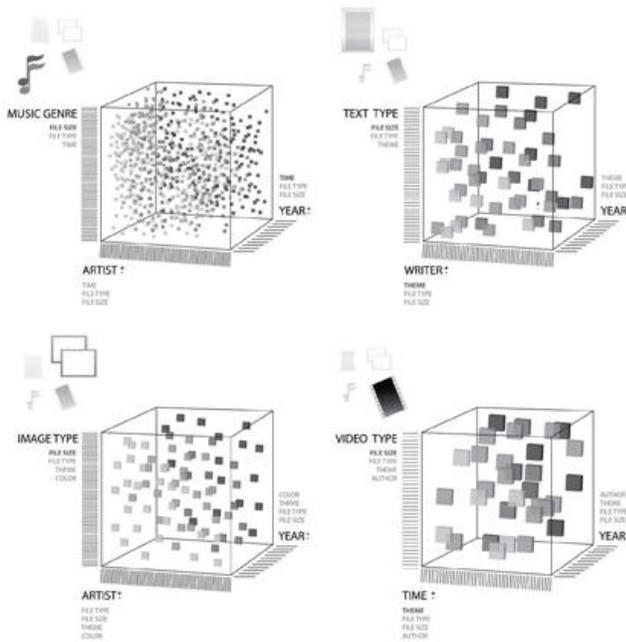
The visual aspects of the glyphs add more dimensions to the original three-dimensional M-Cube. For instance, the image file type and size are encoded to glyph colors and box sizes in Figure 3, adding two new attributes to the visualization. In this example, the M-Cube has 5 dimensions: *location*, *artist* and *year* in the three axes; plus *image type* and *size*.

In addition to viewing rotation and switching of dimensions by touch gestures in the M-Cube, we have two other interaction gestures: zooming and filtering. For these two gestures, it is important to have a multitouch interface, where the touchable screen can recognize more than one touch. In the case of zooming, the user touches with two fingers determining a region of zoom, and pinch the fingers together to zoom in or reverse pinch to zoom out. While filtering, or selection, is done by touching with one or two fingers in each axis, determining a specific attribute value or a range of values between the fingers to make selections on the database. The HCI gestures will be explained in more details in the next section.

The M-Cube is a new interface in both visualization and interaction aspects. It presents a clean interface, so it won't compete with database information, with natural and intuitive interaction. Moreover, the M-Cube features are better achieved when using a high quality screen with multitouch interaction.

## M-CUBE

The  $M^3$  – Multi-dimensional Cube (or M-Cube) – is designed for multimedia databases, providing visualization and interaction in a novel way. There are four types of media: text, music, image, and video. Figure 4 illustrates one M-Cube example for each type of media. The icons on the top left corner of each example show the four media types with the current one bigger than the others. The user can change the current displayed type by choosing one of these four icons. Note that each media type has different associated attributes on the axes. For instance, images and texts are static media, lacking the *time duration* dimension of music and videos.



**Figure 4: The four types of media: music, text, image and video, from top to bottom and left to right.**

The data elements illustrated in Figure 4 are displayed using the same glyph object regardless of the media type. However, depending on the current type, each element has a different border and preview image inside the glyph. For instance, image glyphs have a picture frame and an image preview inside, while video glyphs have a movie style border and a short sequence of the video inside. In the examples shown in this section, we use the same glyph objects as a design representation of any multimedia element. Nevertheless, the final glyph objects for music and images can be seen in the  $M^3$  prototype explained in the next section.

In the case of a music database, we add an audio representation to the glyphs besides the retinal properties, such as color and shape. The preview audio plays when the user interacts with a particular data element, and stops playing when leaving that element. This interaction is different from the opening action, where the user wants to play or look at the entire content of the media data. While the preview interaction is done by touching the element once, the opening action is done by touching it twice.

The action of opening a video data element using the M-Cube is illustrated in Figure 5. The user navigates through the video database, changing dimensions and making filters, until a particular video is found (the white box in the middle). The user chooses to open this video, by double touching it, and the video starts playing.

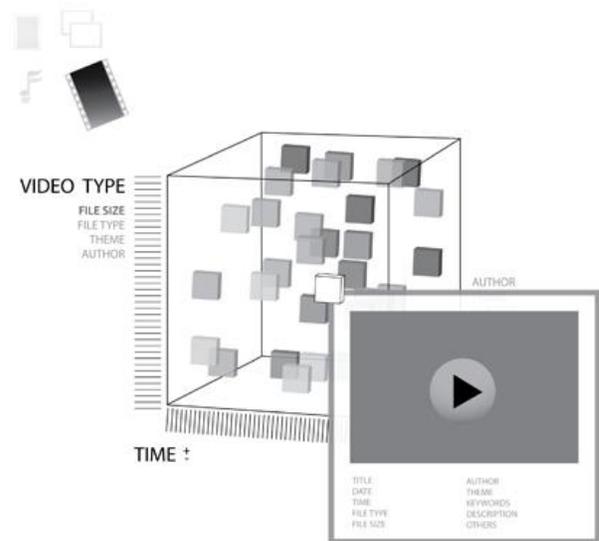
Opening and previewing elements are important actions when dealing with rich and complex content, such as multimedia. In the case where data means quantities or

static values, the user wants to see and analyze the data but not interact with it. Examples of these types of visualization and interaction can be seen in Figure 1 and 2.

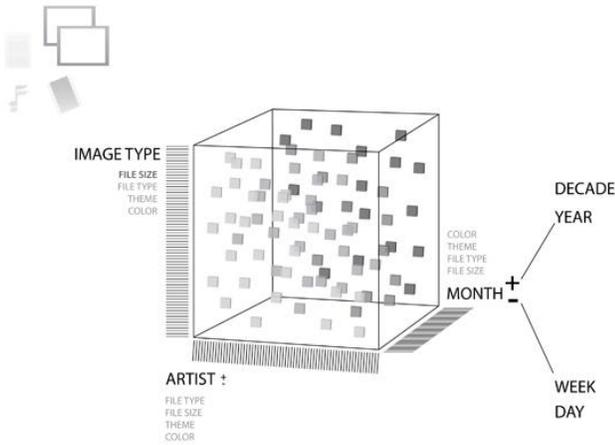
There are many interfaces to handle multimedia data. Operating systems, such as Microsoft™ Windows and Apple™ OS X, exhibit multimedia content in file browsers, organizing the data in tables and relying in different applications to preview and open media. M-Cube introduces a unified interface to both interactions and for any type of media. The visualization is more natural and intuitive than tables showing data elements in rows and attributes in columns. In the M-Cube interface, the attributes are presented on three axes and the data elements are floating objects inside the cube.

The attributes, or dimensions, on each axis have different types of values. For example, the *artist* attribute have name strings as values, while the *date of creation* attribute is identified by dates. Attributes may also have different scales for values. For instance, the date of creation of a media data can be expressed in days, weeks, months, etc.

M-Cube interface allows the user to choose the scale of any dimension having more than one scale. Figure 6 illustrates an M-Cube for images where the user can choose a more refined view for the date axis or a coarser one. The option appears in the form of a plus and minus sign when the user touches the current dimension. These options allow the user to increment, or decrement, the attribute resolution and the M-Cube visualization changes accordingly.

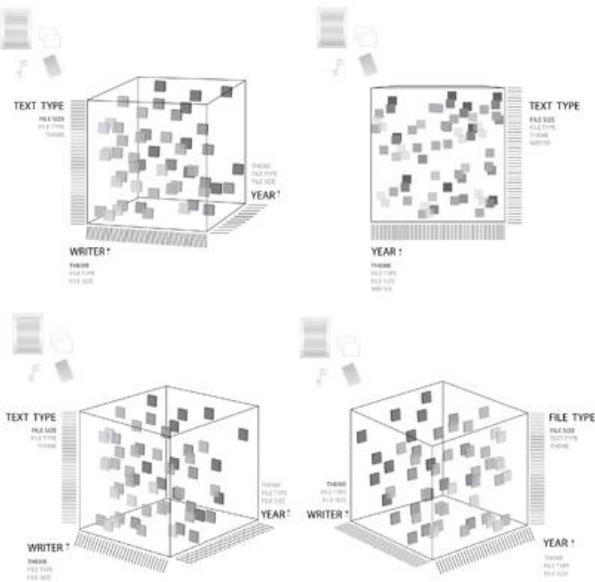


**Figure 5: The action of opening a multimedia data element in a video database.**



**Figure 6: The action of choosing from different scales of the date of creation attribute in an image database.**

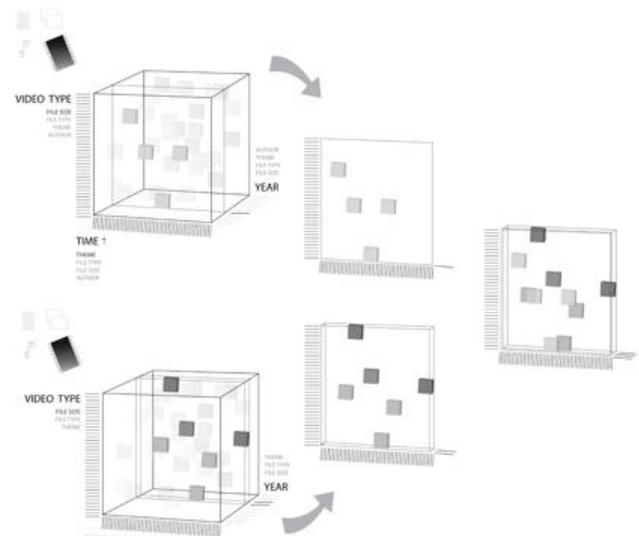
A key feature of using a cube representation to the space is the ability to rotate naturally the space in order to better visualize the data elements. Figure 7 exemplifies a rotation interaction with an M-Cube for text media. The user can manipulate the cube in any direction, making it display the database in preferred faces or transforming the cube into a 2D scatter plot by making one face aligned to the view, as can be seen in the top right corner of Figure 7. The rotation is done by touching any part of the space and gesturing the desired direction to rotate. Note the bottom two M-Cubes in Figure 7, when the user rotates the cube more than a certain angle, making one face more emphasized, the axes change to adapt to this new configuration.



**Figure 7: The action of rotation changes the M-Cube visualization allowing database exploration.**

Using the same rotation gesture, the user can change dimensions. M-Cube interface allows the user to touch one of the side dimensions, such as *color* and *theme* in Figure 6, in a preferred axis and, by rotating the cube while touching the side dimension, the current dimensions change. This interaction makes database exploration easy, since the user rotates the cube to better visualize the current M-Cube and also rotates the cube to change dimensions.

Another important feature of our representation is the selection of attributes by choosing one or more axes to reduce the database visualization. The values on axes can be chosen by intervals or single values. For example, Figure 8 shows two selections, in the first one value is chosen on the *year* dimension (top), and then an interval is chosen on the same dimension (bottom). Each time a selection is made a slice of the M-Cube is created with the elements of the corresponding selection (two cube slices in the middle). The slices are combined forming a selection query, as shown in the right side of Figure 8.



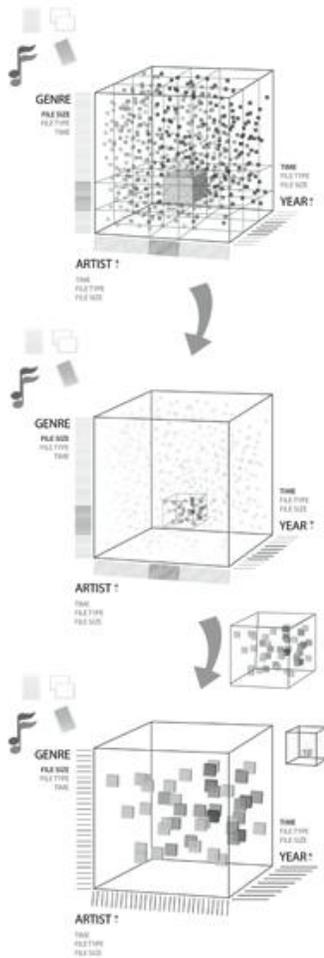
**Figure 8: The action of selecting attribute values in axes reduces the M-Cube visualization.**

The selection action is used to find a particular data element or to make a subset of the database. In the former, the selection improves the rotation action by reducing the number of data elements in the visualization. In the latter, the selection can be used to make playlists when dealing with a music data set, for example.

Building subsets of the database is a common and desirable feature in the case of multimedia data sets. Texts can be organized into folders, music into albums or playlists, pictures and movies into events, such as “vacations”. These subsets can be built using the M<sup>3</sup> tool by the action of selecting augmented by the action of filtering.

The action of filtering permits the user to select multiple axes at the same time, and filter out the M-Cube to reflect the selected axes. Figure 9 shows an example of a large music data set being filtered by a user selection. In a large database, the data elements are very small and difficult to visualize. The filtering action, as the selection, can be used to enhance visualization or build a subset of the data set.

In the example shown in Figure 9, the user selects desirable values in the current dimensions (top): *genre*, *artist* and *year*. The intervals can be selected at the same time using two fingers to touch the start and end values, and different hands to choose more than one attribute. After the selection, the M-Cube animates from the original data elements to the filtered cube (arrows indicate the animation). The result is a new M-Cube with dimensions limited by the intervals specified by the user (bottom). The original M-Cube appears as an icon on the top right corner of the interface, allowing the user to touch it and go back to the original visualization (top cube).

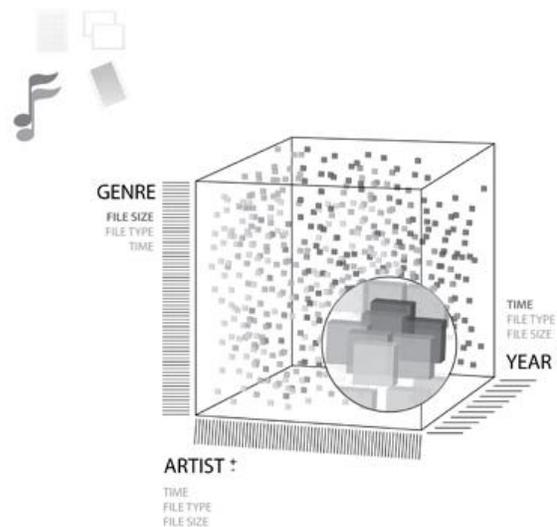


**Figure 9:** The action of filtering uses multiple selections to make a new M-Cube from a portion of the database; the original M-Cube is displayed in the top right corner.

Another way to better visualize the database is the zoom interaction. Large data sets impose a great number of data elements inside the M-Cube, making data glyphs difficult to distinguish. The user can touch with two fingers inside the cube to determine a magnifying region. Note that this gesture is different than touching with two fingers in one of the axes for selection.

Figure 10 illustrates the action of zoom in a large music data set. The circle with bigger data elements inside is a hand lens; the user defines the lens border using two fingers. The region of zoom can be moved by moving the fingers and changing the position of the lens accordingly. This interaction is similar to a cartographer holding a magnifying glass to analyze a map.

In the case of a very large database, this type of zoom can still result in a cluster of many data elements inside the lens. To address this problem, the zooming action allows a second gesture; the user defines the lens border normally, and pinches the fingers together to zoom in the entire  $M^3$  or reverse pinch to zoom out. This interaction discards data elements falling outside the cube as the user zoom in, similar to the action of filtering, and recovers the discarded data elements as the user zoom out.



**Figure 10:** The action of zoom allows the user to distinguish data elements in a dense cluster of glyphs.

The last interaction provided by the M-Cube is a semantic filtering action. In multimedia databases, data can be associated to other elements. For example, lecture texts of a course, the same music played by different artists, pictures taken of the same place, and videos of a movie sequence. M-Cube allows the user to choose one data object and drag it outside the cube, creating a view of related files to the selected data object.

Figure 11 shows an example of semantic filtering in a movie data set. The user chooses a particular data element (the white box on top), drags it outside the cube and another view appears (arrow indicates the change in the visualization). This view shows the semantic features shared by the selected object and the other data elements. It is a reduced set of the original database, since not every data is correlated, and can be visualized as a 2D scatter plot (bottom) or another M-Cube.

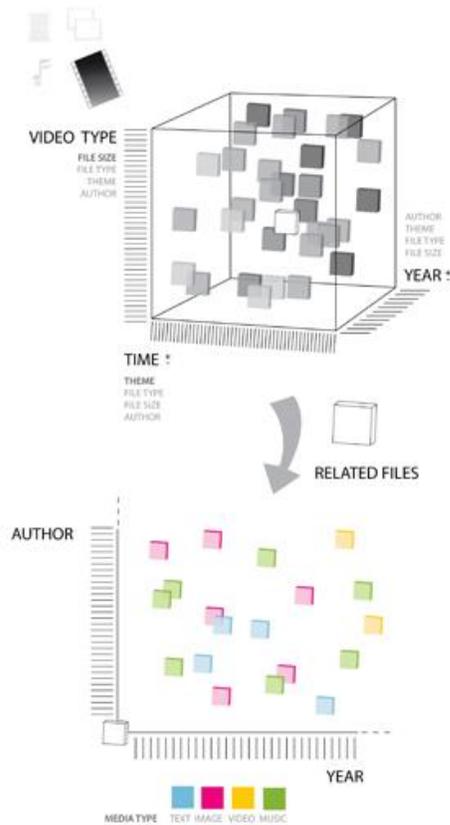


Figure 11: The action of semantic filtering allows the user to explore the correlation among files.

## RESULTS

In this section we present an implementation prototype of the Multidimensional-Cube – M-Cube or  $M^3$ . We build an *interactive table* based on the *reactIVision* framework [19] to support our tool; a display surface which recognizes multiple fingers touches. We choose to start with music data sets, reading the database from the iTunes™ [1] library described in XML. The prototype is part of a previous work called *collecTable* [22], written in *Python* and running on *Linux* operating system.

Figure 12 shows two snapshots of the M-Cube prototype. Note that, even though it is based on a personal database, the music collection can still be large (left snapshot).

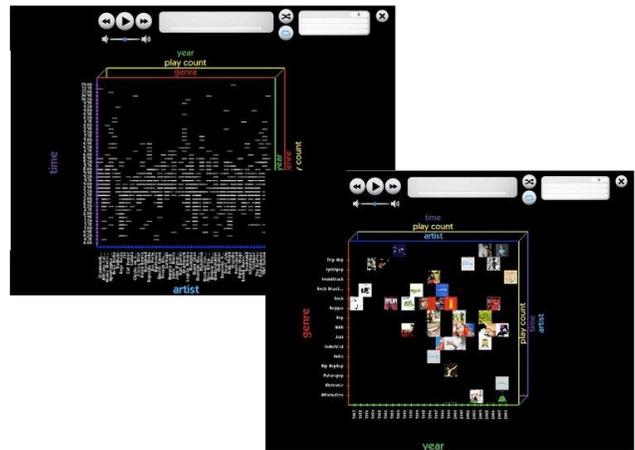


Figure 12: M-Cube prototype for music data sets, visualizing tracks (left) and albums (right).

The prototype implements two types of visualization: music tracks and albums. The first (illustrated in the left of Figure 12) has small gray rectangles as glyphs and is used when tracks of an album does not share the same values in the current dimensions. For instance, the *time* attribute (y-axis in purple) is different for the music tracks inside the same album. Conversely, the second (illustrated in the right of Figure 12) has album arts as glyphs and is used when tracks of the same album share the current attributes. For instance, the *genre* (in red) and *year* (in green) dimensions have the same value for all music tracks inside the same album.

In this first prototype, the M-Cube can be rotated to change its dimensions but the cube ends the rotation facing the user. Figure 13 illustrates a rotation animation of the cube. The prototype allows only one axis to be rotated, or pivoted, at each time. The final goal is to allow rotation in any direction (not only facing the user) and in one or more axes, changing more than one dimension at the same time.

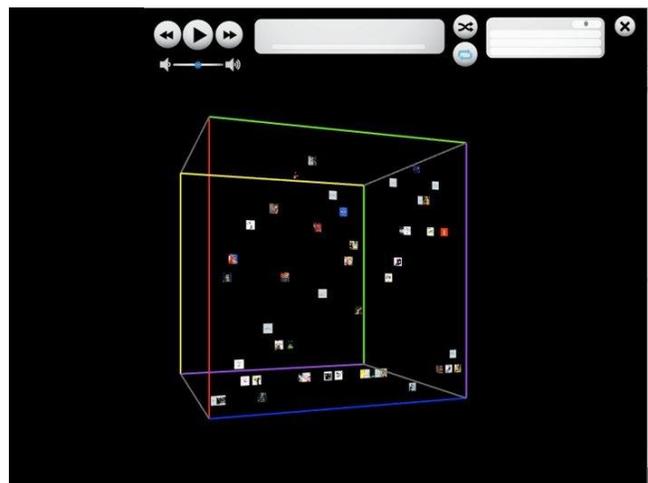
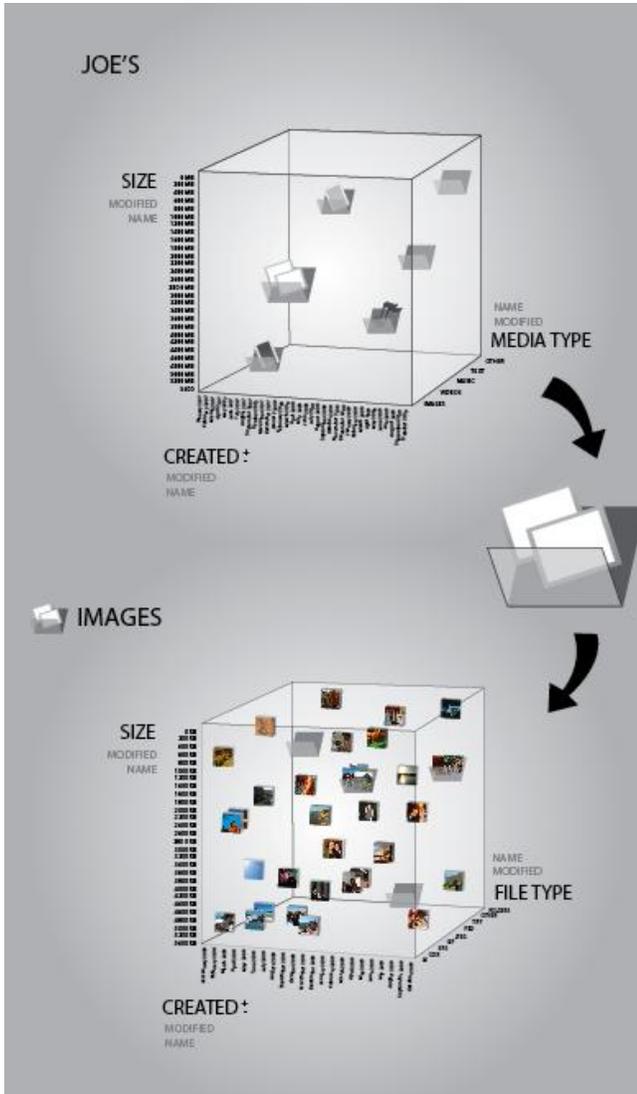


Figure 13: The rotation animation of the M-Cube prototype.

In the second version of the prototype, we build a file browser based on the M-Cube concepts. The user is presented with a base  $M^3$ , and chooses one type of media. After that, the user navigates the selected media database. Figure 14 shows a user named *Joe* choosing images file type to browse (arrows indicate animation after choosing). The resulting M-Cube (bottom) displays *Joe's* pictures as previews inside glyphs and folders storing more pictures. Each folder when opened, using double touch, transforms the current M-Cube, the visualization changes to the display the elements inside the folder.



**Figure 14: The use of M-Cube as a file browser.**

There are still many interaction and visualization ideas to implement. Nevertheless, we believe the prototype acts as a proof of concept and the possibilities that M-Cube offers open a new area on visualization tools for multi-dimensional multimedia databases.

## CONCLUSIONS

We have presented M-Cube, a multi-dimensional visualization tool for multimedia databases. M-Cube introduces new concepts on both visualization and interaction lines. It is based on 3D scatter plots and pivot tables, but improves in both directions. We use specialized glyphs to represent data elements inside the 3D scatter plot, and the rotation action provides great interaction flexibility to change pivots.

M-Cube can be used to visualize any type of media database: text, music, image and video. It can be used as a better visualization tool on top of iTunes™ [1], as demonstrated in the prototype of the previous section. Or as an improved file browser. M-Cube unifies preview and open actions with database visualization in a novel way. It provides a keen visualization of the complete data set, as well as enhanced tools to search for a specific content.

We choose to restrict our visualization tool for multimedia databases in order to benefit from the synergism among this type of data. In this way, M-Cube improves both visually, by rich data elements, and interactively, by specialized interaction techniques.

Our tool is designed for personal use and for large corporation databases. For personal data, we have tested our prototype in different iTunes™ collections. For corporation data sets, we plan to implement the M-Cube in a public institute dedicated to the musician Tom Jobim. This institute has a large collection of data related to Tom Jobim, ranging from letters and music scores to performance and interview videos.

Another path for future work is to implement the M-Cube interface as an add-on of the open source project called *Songbird* [20].

## ACKNOWLEDGMENTS

We would like to thank the Brazilian agency CNPq (National Counsel of Technological and Scientific Development) for the grant provided to the first author.

## REFERENCES

1. Apple iTunes™.  
<http://www.apple.com/itunes>
2. Bertin, J. *Semilogy of Graphics*, Madison, Wis: Univ. of Wisconsin Press, 1983.
3. Elmquist, N., Dragicevich, P., Fekete, J., Rolling the Dice: Multidimensional Visual Exploration Using Scatterplot Matrix Navigation. In *IEEE Transactions on Visualization and Computer Graphics*, Vol. 14, 2008.
4. Keim, D. A., Information Visualization and Visual Data Mining. In *IEEE Transactions On Visualization And Computer Graphics*, Vol. 8, No. 1, Jan-Mar 2002, 1-8.

5. Kreuzeler, M., López, N., Schumann, H., A Scalable Framework for Information Visualization. In *Proceedings of the IEEE Symposium on Information Visualization 2000 Infovis*, 2000, 27.
6. Lee, H., Ong, H., Toh, Eng., Chan, S., A Multi-Dimensional Data Visualization Tool for Knowledge Discovery in Databases. In *Proceedings of Nineteenth Annual International Computer Software and Applications Conference*. IEEE Xplore 1995, 26-31.
7. Microsoft Excel™.  
<http://office.microsoft.com/pt-br/excel/default.aspx>
8. Morris, A. J., Abdelmotv, A. I., El-Geresy, B. A., Jones, C. B., A Filter Flow Visual Querying Language and Interface for Spatial Databases. In *GeoInformatica 8:2*, 2004, 107-141.
9. Pastizzo, M. J., Erbacher, R. F., Feldman, L. B., Multi-Dimensional Data Visualization. In *Behavior Research Methods, Instruments, and Computers*, 34(2), 2002, 158–162.
10. TIBCO Spotfire®.  
<http://spotfire.tibco.com/>
11. Santos, S. R., A Framework for the Visualization of Multidimensional and Multivariate Data. *Ph.D Thesis from The University of Leeds School of Computing Doctor Thesis*, September 2004.
12. Schroeder, W., Martin, K., Lorensen, B., The Visualization Toolkit, *Upper Saddle River*, NJ: Prentice-Hall, 1998.
13. Stone, M. C., Fishkin, K., Bier, E. A., The Movable Filter as a User Interface Tool. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, ACM 1994, 306-312.
14. Stolte, M. C., Tang, D., Hanrahan P., Polaris: a system for query, analysis, and visualization of multidimensional relational databases. In *IEEE Transactions On Visualization And Computer Graphics*, Vol.8, N.1 January-March, 2002, 52-65.
15. Stolte, M. C., Tang, D., Hanrahan P., Multiscale Visualization Using Datacubes. In *IEEE Transactions On Visualization And Computer Graphics*, Vol.9, N.2 April-June, 2003, 176-187.
16. Stone, M. C., Bier, E. A., Pier, K., Buxton, W., DeRose, T. D., Toolglass and Magic Lenses: The SeeThrough Interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993 ACM, 73-80.
17. Viega, J., Conway, M. J., Williams, G., Pausch, R. 3D Magic Lenses. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, 1996 ACM, 51-58.
18. Williamson, C., Schneiderman, B., The Dynamic HomeFinder: Evaluating Dynamic Queries in a Real-Estate Information Exploration System. In *15th Ann Int'l SIGIR '92/Denmark-6/92*, 1992 ACM, 338-346.
19. Kaltenbrunner, M., Bencina, R. reactIVision: A Computer-Vision Framework for Table-Based Tangible Interaction. In *TEI'07: Proceedings of the 1st international conference on Tangible and embedded interaction*, 2007 ACM, New York, NY, USA, pp. 69-74.
20. Songbird.  
<http://getsongbird.com/>
21. Wolfram Alpha.  
<http://www.wolframalpha.com>
22. Maximo, A., Saba, M. P., Velho, L., collectable: uma Interface Natural para Coleções de Música, In *Proceedings of 4o CIDI, Infodesign 2009*.