Visualizing Abstract Information using Motion Properties of Data-Driven Infoticles

Andrew Vande Moere [*a], Kuk Hwan Mieusset [a], Markus Gross [b]
[a] Chair for CAAD, Swiss Federal Institute of Technology Zurich, Switzerland
[b] Computer Graphics Laboratory, Swiss Federal Institute of Technology Zurich, Switzerland

ABSTRACT

This paper presents a novel exploratory information visualization technique that allows users to analyze time-varying characteristics of large datasets within immersive virtual reality environments. This metaphor represents data objects as particles, coined infoticles, which are placed inside a three-dimensional scene. Forces correspond to specific data value conditions and influence matching infoticles according to the rules of Newtonian mechanics. In addition, infoticles are driven by a set of local behavior rules that react upon successive data updates, hereby generating distinct emergent motion typologies which are visually interpretable by users. These data patterns can be detected dynamically by observing the spatial transformations of infoticle streams, or statically, by interpreting the shapes of individual pathlines. This visualization method exploits the qualities of immersive virtual reality technology as it combines the characteristics of behavior generation and motion perception with the concepts of spatial awareness and stereoscopic vision. Infoticles are useful in visualizing time-varying characteristics of large, dynamic datasets because of their cognitively distinguishable and interpretative animation properties. The generation and evolution of infoticle patterns are based upon empirically defined grammatical rules. These visualization principles are demonstrated using the access logs of an internal knowledge document management website of a global consultancy company.

Keywords: information visualization, exploratory data analysis, particles, motion, behavior, immersive virtual reality.

1. INTRODUCTION

This paper describes a novel exploratory data analysis metaphor that maps time-varying data updates onto the three-dimensional movement and the kinetic behavior of data-driven particles, coined infoticles. Recognizable dynamic pattern typologies emerge that represent specific data update characteristics and corresponding data value alterations. In contrast to approaches that use force-placement algorithms to spatially cluster similar data entities, the infoticle concept is based upon the motion properties of particles. Infoticle movements are generated by a simple set of local rules that react upon the continuous update of time-varying data values, out of which global behaviors emerge that are visually perceivable by users. Some simple flow visualization techniques to enable users to interpret the resulting trajectories dynamically, by observing the spatial transformations of infoticle streams, or statically, by evaluating the shapes of infoticle pathlines. The data-driven particle concept was originally developed for immersive virtual reality environments, as it exploits the unique properties of spatial awareness and stereoscopic depth. Special attention has been given to the notion of user engagement[1], or the connection of interaction design with enjoyable user experience.

The infoticle method is presented as a visualization metaphor rather than a model, to denote the open-endedness, incompleteness, and inconsistent validity of metaphoric comparisons versus the explicitness, comprehensiveness and validity of a model. Models are designed to represent some target domain, whereas metaphors are clearly designed to invite comparisons and implications. We believe that infoticles are useful in visualizing large, time-based datasets because of their interpretative motion properties and their ability for a high degree of user interaction. Our current application prototype is developed in close collaboration with a global engineering consultancy firm. This company employs Intranet websites as a knowledge management tool to communicate and continuously keep track of relevant issues. However, people at the company were dissatisfied with current web log analysis tools and expressed their interest to interactively explore the site's performance in geographically spreading expert knowledge.

[*] moere@arch.ethz.ch

2. RELATED WORK

To date, only few information visualization applications exist that represent the evolution of time-varying datasets. The 3-D Visual Data Mining system (3DVDM) is one of the few examples that merge time-varying data visualization with virtual reality technology, generating three-dimensional scatterplots that are able to morph between static states, and using point vibration as an informational cue[2]. In contrast, not the objects moving but instead recognizable motion typology characteristics carry informational meanings in the data-driven particle concept. The Control Project extensively uses the concept of data flow to visualize adaptive database query streams, as gradually representative data objects are streamed to the visualization system to generate a meaningful approximate result[3].

2.1. Immersive Information Visualization

The very first approaches to use three-dimensional space to browse and locate information were inspired by the notion of spatial databases[4]. In such virtual worlds, database objects can be distributed in meaningful locations, so that a viewpoint becomes a query, and browsing resembles information spaceflights. Immersive virtual reality (IVR) is typically used for the visualization of scientific data[5], but also for abstract datasets[6]. Many examples have shown that this technology is suitable for collaboration while working with massive amounts of data[7].

2.2. Particles

Particle systems are a technique within the field of computer graphics for creating a wide range of complex visual effects, and comprise different modeling techniques, rendering methods and animation types[8]. Particles are also widely used in scientific visualizations, especially in Computational Fluid Dynamics (CFD). This field mainly deals with the representation of unsteady, three-dimensional flows, by either showing moving particles or by tracing particles that are injected in the fluid[9], which are then rendered in e.g. immersive virtual reality environments[10].

Groups of particles, combined with external forces or internal relationships, are able to convey complex behaviors[11], a phenomenon which has already been demonstrated in some other information visualization applications. For instance, particles have been used to debug visualization systems, by representing attributes and relationships graphically instead of algorithmically[12], to target and highlight features of interest in the dataset by spraying so-called smart paint particles[13], to visualize the interests of employees by the clustering movements of swarming fish[14], and to characterize network communication performance by swarm behavior typology[15]. Similarly, the infoticle concept exploits this behavior generation capability by combining a set of globally valid Newtonian point forces with data-driven local rules.

2.3. Force-Placement

Although data-driven particles integrate some CFD ideas, they are not injected in fluids, but are instead attracted by external point forces according to Newtonian mechanics. Force placement and spring-embedded algorithms are capable of generating so-called undirected graphs[16], and are used in several virtual reality applications[17]. In most force-directed visualization systems, spring stiffnesses correspond to pre-computed similarity measures, out of which an energy minimum is calculated that reveals multidimensional relations in terms of spatial neighborhoods. In contrast, infoticles behave independently and are unaware of any data similarities, do not need dedicated calculation time to build up the scene, are capable to represent time-varying datasets, and never reach a static state of equilibrium. Furthermore, infoticle representations differ from static or morphing scatterplots in that they are not mapped in Cartesian space and utilize motion typologies and pathline shapes instead of spatial coordinate positions to convey information.

2.4. Motion & Stereoscopic Vision

Stereoscopic depth is the most powerful, pre-attentive visual feature as compared to, for example, color intensity and hue, can be used to overcome the effects of conjoin[18], and significantly increases the size of a graph that can be understood[19]. Search tasks, which are often required during explorative data analyses, are completed faster in immersive environments than on normal desktop machines[20]. Other experimental research showed that users perform better on statistical data visualization tasks, such as cluster analysis, but worse on interaction effectiveness, in an immersive virtual reality than on a traditional desktop[21].

Motion is generated through a process called animation, in which form and structure evolve through a certain development over time, hereby conveying the feeling of kinetics or dynamics. Well-designed motion metaphors can be aesthetically appealing, and are able to attract attention and maintain motivation. Motion is a powerful pre-attentive feature[18], and according to psychological studies, is able to convey meaning, intentions, and high-level emotions[22], and can thus describe how different abstract elements are related. Michotte[23] discovered that humans group distinct points with equal speed and parallel directions. Morrison et al.[24] presented a comprehensive investigation on the effectiveness of animated graphics in conveying visual or spatial information, stating that motion does not always automatically benefit information transfer: similar to static representations, only carefully designed and appropriate graphics seem to perform better. In fact, animations often violate the second principle of good graphics, the apprehension principle: animations are often too complex or too fast to be correctly perceived, and continuous events are sometimes conceived as sequences of discrete steps[25]. Notably, the use of interactivity may overcome both these disadvantages, and the animation should be implemented as a smooth and continuous process.

Bartram[26] describes the use of two-dimensional motion for information display and its potential for filtering and brushing actions. She concludes that similar motions are effective in perceptual grouping and motion type is an excellent discriminating feature. Consequently, as positional mapping and motion generation are obviously contradictory visualization cues, novel visualization metaphors are required that focus on dynamic behavior generation instead of static coordinate interpretation. In effect, the generation of motion typologies by behavior functions, which themselves can typically only be determined by trial-and-error experimentation, creates systems of graphic objects which appear to behave purposefully by mimicking complex spatial relationships[27].

In relation to our proposed visualization technique, these studies thus suggest that users should be capable to perceive larger amounts of points in stereoscopic environments, cluster similar spatial trajectories, and recognize and interpret different smooth and interactive motion typologies that are generated by empirically defined behavior functions.

3. INFOTICLE METAPHOR

The infoticle concept corresponds with that from our earlier publication[28], which described the possibilities of interactive, data-driven particle streams in immersive virtual reality environments. Since then, a live database connection has been implemented that is capable to gradually stream large datasets, and the local behavior rules have been fine-tuned. As the next paragraphs will demonstrate, these adaptations determine various global, emergent animation behaviors, resembling that of a *space galaxy*.

3.1. Concept

The scene consists of a set of static forces and a collection of particles, called infoticles. Each infoticle represents a unique data object, retrieved from a remote database. A typical data object consists of a fixed list of data attributes that contain specific data values, and practically resembles a row from a database table. During the visualization process, an infoticle is spatially attracted to those set of forces within the scene that contain corresponding data value conditions. This simple interaction results into a change of both the infoticle direction and speed, as the force attraction follows the rules of Newtonian mechanics. All the data objects are continuously updated via a live database connection, according to the simulation timeline. In practice, a parallel process continuously keeps track of the current application timeframe, and sends sequential queries to the database to collect and cache the updated data objects that are present in the next database timeframe. As these data objects typically contain altered data values, the according infoticles become attracted by different sets of forces. As a result, data value changes are translated into spatial variations of infoticle trajectories.

3.2. Structure

Next to the normal particle attributes such as position, direction, speed and color, each unique infoticle contains:

- **Data Object**. Each infoticle has a pointer to a unique programmatic data object that is retrieved from the database. This data object can contain different data attribute values that are all valid for the active application timeframe.
- **Time-Average Force**. As the infoticle continuously changes its direction towards those forces that contain equal data values, it visualizes only the data values of the active timeframe. In order to consider the data value evolutions within previous timeframes, the average force represents the average of all the forces the infoticle was attracted to, weighted in time.
- **History List**. A history list consists of past positions, speeds and data values, needed to calculate the infoticle pathline and to reproduce the spatial, historical trajectory when the application timeline is animated back- and forwards.

3.3. Behavior
Several experiments have been conducted to determine the most effective infoticle-force interaction rules that result in easily perceivable and interpretable spatial behaviors. Each infoticle is controlled by the following set of behavior rules.

- **Tool Influence**. Adapt speed and direction to the laws of Newtonian mechanics and the current average force.
- **Drag**. Gradually reduce speed and brightness, expressing the passed time since the last data update.
- **Orbit**. If the distance to the average force is smaller than the predefined value, start orbiting around it. This rule avoids that infoticles ‘crash’ into the center of the force.
- **Update Data**. If the data object has changed for the current application timeframe: speed up, highlight brightness, calculate new average force position and direct towards it.

3.4. System Update
The system updates the active date and time, queries the database for the next database timeframe, and stores the new data objects in the shared memory once they have arrived. At a certain predefined rhythm, the system acts as follows.

- **Query**. Load and cache new data objects for the next timeframe. Share this memory range with the infoticles.
- **Update**. Save current positions and changed data values of all infoticles in their respective history lists.
- **Switch**. Check for every infoticle if a new data object has arrived and switch to it. Remove old data object from cache.

3.5. Simulation
Time-varying datasets can be visualized either using pre-computed or on-the-fly computed animation. Pre-computed animation typically results in simulations in which the interaction is limited to passive spatial navigation. On-the-fly computed animation offers users the experience of smooth dynamics that react to real-time interaction. This concept requires large computing resources, since frames must be continuously recalculated in response to changes in both incoming data streams and user interactions. On-the-fly computed animation is the only feasible method for information visualizations and virtual reality environments, as direct user interaction is a crucial requirement of both fields.

The infoticle metaphor concept consists of a direct mapping mechanism between a unique data object and a data-driven particle. For time-varying datasets, the data values of such a data object change over time. In practice, this means that a database table contains several entries of the same data identifier, at many different points in time. Consequently, a *data update* can be imagined as the sequential change of data values to a successive timeframe of a specific data object. Two different timeframes are needed to accurately simulate time-varying data values. The exact durations are defined at the visualization initialization, but can be dynamically changed during the visualization simulation by the user.

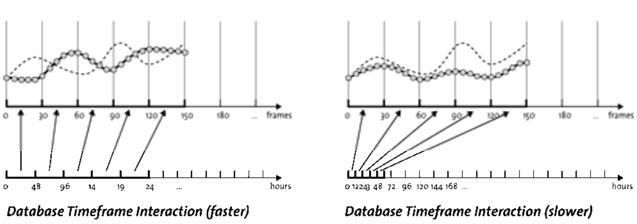


Figure 1: Database timeframe alteration.

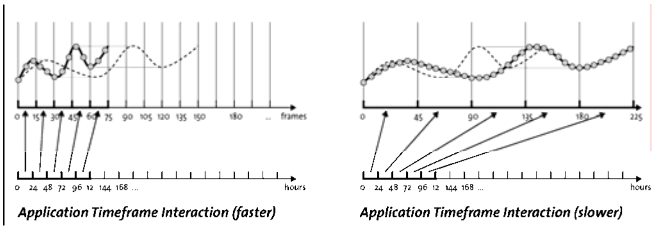


Figure 2: Application timeframe alteration.

- **Database Timeframe**. The database timeframe describes the start and end points between which the data that is streamed from the database to the visualization system needs to be positioned in time, and is typically measured in physical time units. In practice, the visualization system queries the database for all the data that is situated within the active database timeframe. At every *application timeframe* step, the database timeframe is sequentially shifted to the next timeframe and new data is collected. As shown in Figure 1, resizing the database timeframe enables the analysis of data patterns at different time granularities, but also influences the total time needed to represent the complete dataset. The database timeframe duration is also directly related to the quantity of the acquired and streamed data during each application timeframe, and therefore has a considerable effect on the application computing performance.

- **Application Timeframe**. The simulation or application timeframe denotes the rhythm at which the application retrieves the next batch of data objects for the following database timeframe. In practice, it contains a specific number of frames after which the visualization system retrieves the new data objects of the next database timeframe. Each subsequent application timeframe corresponds to a unique, sequential database timeframe, and both are continuously updated in parallel. When the application timeframe is changed during real-time simulation, different, shorter visual patterns appear due to the gradual and force-adaptive nature of the infoticle system. However, application timeframe alterations are particularly useful when animating the simulation back- or forwards in time, as it exactly recreates the infoticle trajectories at different simulation speeds and thus enables the rapid skipping of unimportant sequences or the careful evaluation of events in slow motion, as illustrated in Figure 2.

Determining the relationships between data object values and their evolution in time is not trivial, as it is unknown how, when and what quantities of the data are altered in each database timeframe and are streamed to the visualization system at each application timeframe. Therefore, data acquisition (data querying, organizing and caching) and visualization simulation procedures are separated and executed in parallel. As a result, the real-time infoticle animations become not interrupted by computing expensive algorithms such as database calls, memory caching or data look-up processes.

3.6. Average Force

Most dynamic datasets contain reoccurring data objects within single database timeframes. In fact, multiple data entries of single data objects cannot be avoided if the duration of the database timeframe is larger than the smallest time granularity of the database entries. Therefore, the infoticle metaphor needs to handle effectively the reoccurrence of data objects within database timeframes. For quantitative data, the infoticle system calculates the average of all data object values that are present in parallel within the actual database timeframe and stores these in a single representative infoticle. Alternatively, for unstructured data, an average force is computed that takes into account the frequency of unstructured data values within a single timeframe and the spatial positions of the corresponding forces within the scene.

$$c_i = c_x \Rightarrow f_x = f_x + 1 \rightarrow \left. \begin{pmatrix} c_i \in \Delta t_{database} \\ x \in [0,1,..,n] \end{pmatrix} \right\} \Rightarrow \vec{F}_{average} = \frac{\sum_x^n f_x . \vec{F}_x}{\sum_x^n f_x}$$

First, all frequencies of each retrieved data value c_i that correspond to a conditional force value c_x of a force $\vec{F}_x$ within the timeframe $\Delta t_{database}$ need to be determined. Then, an averaged force of parallel reoccurring data values (not yet time-averaged), for an infoticle world with n forces is calculated.

4. INTERFACE

The infoticle concept differs from Cartesian-mapped visualization systems such as scatterplots[29], as users do not need to spatially orientate themselves in relation to three-dimensional coordinate axes, but instead rely on relative force-infoticle distances and pathline shapes. This method applies the qualities of stereoscopic vision, which has proven to increase the amount of comprehensible information and improves the perception of three-dimensional distances. These immersive aspects thus help users to overview and keep track of the thousands of moving points, a task that would be difficult to perform on normal displays. Figure 3 demonstrates that the whole interface is built up in human proportions and uses no space occluding menu widgets, resulting in a full physical immersion of users within the visualization space.

4.1. Dimensionality

As the whole infoticle application is placed inside the virtual, three-dimensional world itself, the interface design has to express a recognizable distinction between the information representation and the offered interaction features. Therefore, the infoticle metaphor categorizes element functionality by representation dimensionality. For instance, all interface elements, including forces, are represented by two-dimensional billboard icons that are rotated towards the user. Each visualization element that supports users to comprehend the emergent patterns, such as trace ribbons, is inherently three-dimensional, provoking human cognitive capabilities to interpret the shapes and curvatures. Lastly, the infoticles themselves can be considered as four-dimensional mathematical objects placed inside three-dimensional space that are animated over time.

Figure 3: A user immersed in a virtual reality cave and powerwall virtual reality environment.

4.2. Interaction

A six-dimensional mouse controls a virtual spotlight cursor that reveals detailed data labels of the infoticles that are in its view cone. This interaction device also controls the navigation, which is similar to a track-ball interface with zoom and pan control. Users are able to query the data by manipulating all the elements that are present in the scene: e.g. forces can be selected, dragged and grouped together, so that new infoticle constellations emerge. Users can also manipulate the direction and speed of the simulation timeline, so that sudden events can be replayed in slow-motion, or unimportant time sequences can be rapidly skipped. By observing the animation of moving particles, the causality of events, turbulent fields and fluctuating densities can be detected. Moving infoticles give a good indication of the direction and velocity magnitude of infoticle groups, while the static pathlines are suited to visualize the spatial history, time-dependent character and long-term data value evolution of individual infoticles.

4.3. Pathline Ribbon

The most common visualization motion tracing method generates spatial curves or splines that sequentially connect all points in space which the infoticles have traversed. The visualization of such curves is a widely used technique in the area of scientific visualization, as these line representations are able to provide information about the character of unsteady, time-dependent flows. An infoticle ribbon is typically constructed by two parallel Catmull-Rom splines that smoothly pass exactly trough the coordinate values previously stored within the infoticle history list. Consequently, the pathline ribbon represents the approximate historical infoticle path. The infoticle method uses ribbons instead of lines, as they convey more spatial qualities and facilitate the representation of extra data dimensions (see Figure 4). The exact width size represents a specific quantifiable and time-varying data attribute value, defined by the application designer, such as the frequency of the data object within a database timeframe. The color denotes the original data typology, while the decreasing transparency represents the direction of the infoticle history. Text labels along the pathline denote the historical data values at those specific points in space and time that the infoticle has traversed. As will be demonstrated later, the overall ribbon curvature represents the different motion typologies and data update characteristics an infoticle was subjected to and thus inherently contains various perceivable informational values.

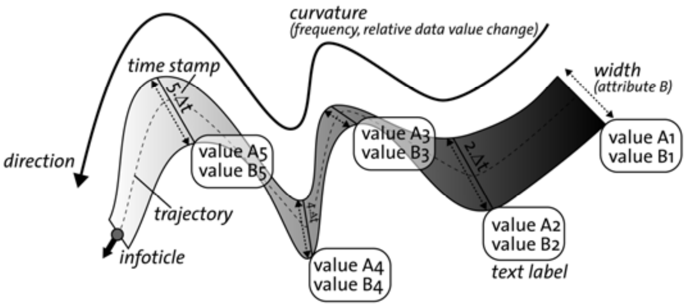


Figure 4: Pathline ribbon features.

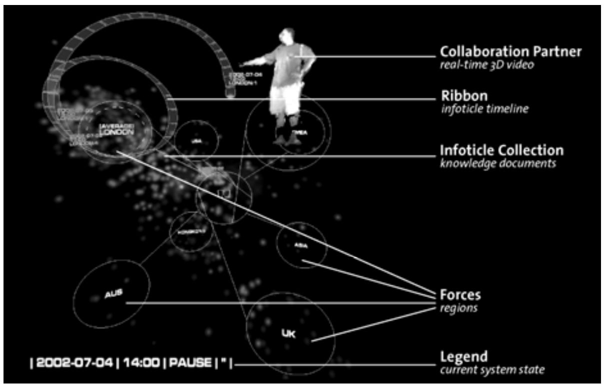


Figure 5: Infoticle scene overview.

5. APPLICATION

5.1. Intranet File Usage Visualization

Our current application prototype visualizes the Intranet file usage of a global engineering consulting company which comprises approximately 7000 employees. This firm has created internal communities of practice, so-called *skill networks*, which consist of staff members who are specialized in a specific field of expertise, such a structural engineering, fire engineering, acoustics, etc., or have a professional interest in being a member of that community. Although these experts share a common corporate identity, they are often spread out geographically throughout the various international offices. To enable an effective communication of project experience and specialized knowledge, each skill network is represented by a website that is posted on the corporate Intranet, with the primary goal to geographically distribute the knowledge that is embedded in its content. All IP-numbers have been categorized in seven different regions, containing an approximate equal number of employees. The Intranet files are divided in three categories, each represented by a different color: orange document files (e.g. Word, PDF), purple structural files (e.g. HTML, ASP), and red image files (e.g. JPEG, GIF). This particular categorization has been chosen to analyze the differences between the web page functionalities and the specific knowledge documents they give access to.

Each unique Intranet file is represented by a single infoticle. All infoticles initially swirl around the center of the scene. Seven forces, depicting the different geographic regions, are placed around this center and are represented by simple circular icons. A status console, placed at the bottom of the screen, reveals the timeline state and the database timeframe that is currently being loaded by the visualization system (see Fig.5). The system retrieves all the file access log entries that are present within a certain database timeframe (e.g. 1 hour: 2002-07-02, from 9:00 until 10:00). If the same data value (e.g. "USA") for a certain file (e.g. "index.html") appears more than once within this timeframe, they become accumulated for the average force calculation and the total number is mapped unto the width of the pathline ribbon.

At first glance, the use of relatively sophisticated virtual reality technology for visualizing this kind of dataset might seem disproportionate. Yet, this dataset is implemented primarily as a manageable, real-world source for experimentation, with the goal to explore the infoticle metaphor effectiveness and detect its potential features for information display in immersive environments. In the following descriptions, a data entry found within the web logs is considered as a single 'use' of the corresponding document, although theoretically this relationship does not always hold: users might have just clicked on the document, or looked at it so briefly that assuming this data entry as an effective usage would be an overestimation. Also, the following infoticle scenario should not be confused with network visualizations, in which typically the physical transmission of unique electronic packets is represented. Instead, these infoticles denote the time-varying and geographical sharing of centrally stored knowledge documents.

5.2. Visual Patterns

As a direct result of the infoticle behavior rules specifications, different spatial behaviors can be visually distinguished. These distinct patterns can be discriminated in several modes, either by interpreting the visual features of static trace shapes or by observing dynamic infoticle behaviors. Because of obvious analogies, these phenomena are uniquely identified with terms taken from the world of physics, astrophysics and astronomy.

- **Transfer**. Whenever the data values of an infoticle have significantly altered, it directs towards a new average force, generating a straight line trajectory. This pattern is only an intermediate state.
- **Global**. The time-averaged force calculations that are based upon the document's access history produce several large, linear infoticle clusters that spatially connect pairs of forces. These groups consist of documents that are shared extensively by two or more regions. Time-varying changes in volume and density can be easily detected.
- **Time-Related**. Access patterns of simultaneously highlighted infoticles that suddenly change their motion pattern are often driven by the international time zones, week days and office hours, and can clearly be perceived.
- **Star**. A star infoticle represents a document that has recently been accessed, just reached the average force and started to orbit around it (Fig. 6). During this process, it is slowed down by the continuous drag influence and remains on a relative wide distance from the average force. A star infoticle is very sensitive to any further direction or speed alteration by the fragile nature of its circular trajectory.
- **Electron**. Due to the continuous speed decrease, a star infoticle becomes increasingly attracted towards the average force. As a result, small clusters of fast moving infoticles emerge in the force center (Fig. 7). These infoticles generally represent files that are rarely accessed, and consequently might contain non-shareable, outdated or 'redundant' knowledge.

- **Comet**. An infoticle that is accessed more than once by a single region, or same set of regions, within a short period of time. Due to the considerable increase in speed and the absence of any significant force-induced direction change, the resulting trajectory becomes elliptical and quasi-periodic (Fig. 8). Comet behavior often is caused by non-shareable or regional documents or even by unexpected repeated flaws in the access logging mechanism.
- **Burst**. A cluster of multiple infoticles that change their direction simultaneously and travel in the same direction. An infoticle burst generally consists of documents that are conceptually related in content, such as PowerPoint presentations saved for the web, or groups of Adobe Acrobat PDF documents of the same project or subject.
- **Quark**. An infoticle that behaves erratically, changing directions multiple times within a short time span (Fig. 9). This chaotic behavior is easy to spot, as it visually 'pops out'. Quark infoticles represent files that are effectively and often shared across geographical regions. Typical quark infoticles are portal documents, or individual documents that became popular by internal advertisements.

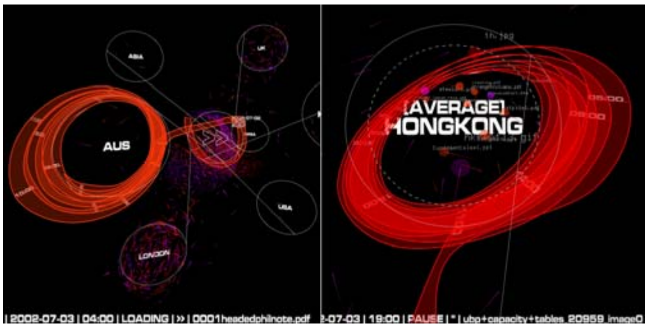


Figure 6: Star pattern.

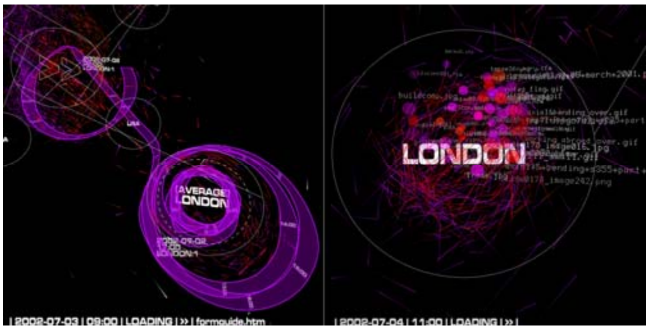


Figure 7: Electron pattern.

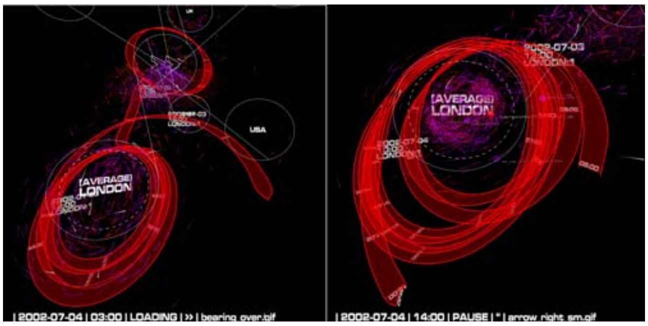


Figure 8: Comet pattern.

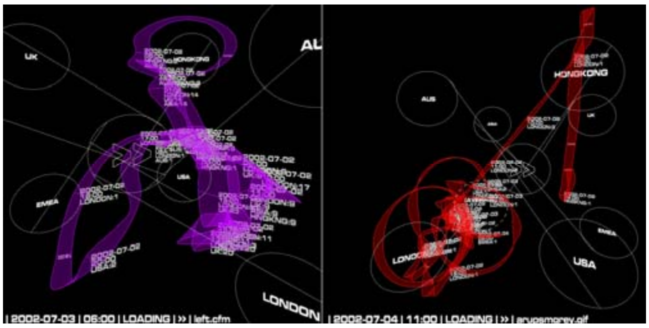


Figure 9: Quark pattern.

6. ANALYSIS

6.1. Influences

The infoticle visualization method differs fundamentally from traditional information mapping techniques which translate data values directly and singularly into static visual cues, such as spatial position, color and shape. As illustrated in Figure 10, the infoticle behavior is instead determined by the following parallel, data-driven influences.

- **Data Value**. The exact data object values determine the relative spatial location within the scene, because the infoticle is attracted by, and thus is in the vicinity of, the corresponding average force.
- **Data Value Change**. The relative data value change in comparison to previous historical values influences considerably the gradual evolution of spatial behaviors. For instance, depending on the relative data value update, a star infoticle is followed by an electron (no change), star (significant change) or a comet (equal change) pattern.
- **Data Update Frequency**. Because it is unknown whether and how often a data object is updated within each database timeframe, the data update frequency also determines some infoticle behavior characteristics. A fast sequence of relative small data value changes might not alter the data pattern typology but some of its more detailed visual features. For instance, a comet infoticle that becomes updated to equal data values will stay in a comet state, but might exhibit a more eccentric orbit path. Also, fast data update frequencies typically produce more irregular infoticle behaviors and tracing artifact shapes in comparison to slow, stable updates.

6.2. Dynamic

An infoticle representation can be perceived and analyzed in contextually different situations, offering distinct conceptual dataset interpretations. The infoticle metaphor is primarily meant to be perceived dynamically, as it represents the continuous data updates in a faster pace than the originally stored database timeline. The infoticle behavior rules inherently determine specific sequential pattern emergence evolutions. Figure 11 illustrates how the different infoticle patterns are interrelated when the system is in a dynamic state. For instance, the comet and electron patterns are evolutionarily derived from the star pattern after a specific amount of time. A comet can be perceived by spotting a sudden event, while electrons are formed through a slow, incremental process. The star pattern always depends on the transfer phenomenon, and is triggered by a specific force distance condition. Table 1 also indicates the start and end conditions and whether specific patterns can be detected by considering individual or global infoticle behaviors during sudden events or long-term evolutions.

One of the most unique capabilities of the infoticle system is its power to effectively represent various data update characteristics, or the ‘characteristics of change’, instead of exact data values. Table 2 demonstrates the emerging patterns under different time-varying features, and, for instance, the patterns that appear as a result of ‘noisy’ data or time-dependent data value alterations that are either relatively equal in size or happen at equal points in time.

Pattern	Infoticle Quantity	Pattern Before	Trigger Action	End Pattern	Other Visual Features
Transfer	single	any	sudden event → *directional change*	any	straight line
Global	multiple	stars, comets, electrons	slow evolution	diffusion	high, stable density, parallel directions
Time	multiple	any	sudden event	any	directional change, color change
Star	single	transfer	force distance (orbit)	transfer electron comet quark	stable circular trajectory , slow speed
Electron	single	star	slow evolution (drag, force distance)	comet quark transfer	circular trajectory, spinning, high speed
Comet	single	star electron	sudden event (speed increase)	any	elliptical trajectory, high speed
Burst	multiple	multiple transfers	sudden event	any	equal directions, same speed
Quark	single	sequential transfers	multiple sudden events	star	chaotic, high speed

Table 1. Infoticle pattern evolution.

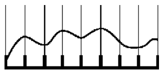
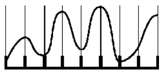
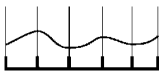
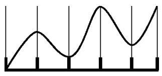
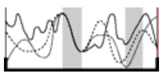
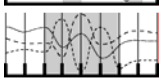
	Data Update Typology		Visualization
	Update Frequency	Data Value Change	Pattern
	high	low	*comet*
	high	high	*quark*
	low	low	*electron*
	low	high	*transfer→star*
	any	equal	*burst*
	equal	any	*time*

Table 2. Time-varying characteristics patterns.

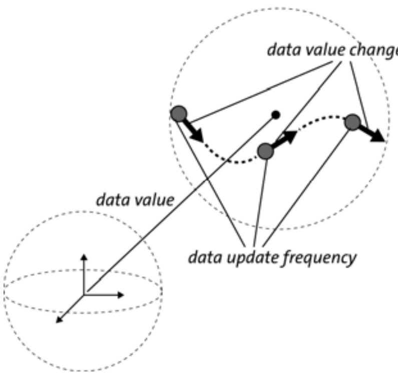


Figure 10: Pattern determinators.

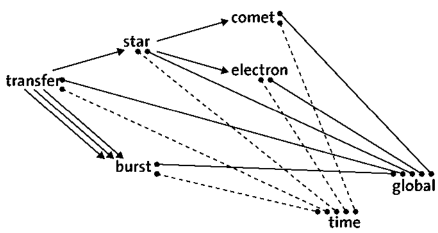


Figure 11: Pattern evolution.

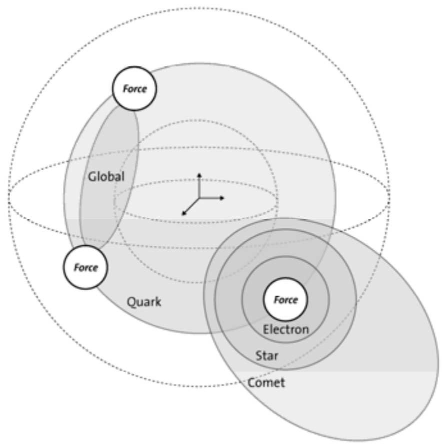


Figure 12: Pattern spatiality.

6.3. Static

An infoticle representation can also be analyzed in a static state, hereby conveying fundamentally different data patterns as from the previously mentioned dynamical analysis. Generally, the frozen visualization world is visually searched for infoticle clusters that are in close proximity to one another or are heading for an almost identical spatial direction, as such patterns typically denote a certain degree of data similarity. Figure 12 shows where specific patterns can be sought within the visualization world. The static state is also suited to compare the different pathline ribbon shapes.

7. RESULTS

7.1. Implementation

Creating a virtual reality experience requires both the design and programming of virtual elements, interaction effects and sounds, suitable physics simulation algorithms and intuitive user interfaces. As few authoring tools are currently available to support the handling and modeling of virtual simulations for immersive environments, many technical issues have to be solved on a high conceptual level. In fact, this work is built upon the blue-c Application Programming Interface (bcAPI) [30]. The blue-c API is built on top of the SGI OpenGL Performer API, which itself offers a scene graph and real-time rendering system[31], and uses parts of its process and shared memory management for process synchronization and communication. The majority of the infoticle code is implemented using the SGI OpenGL Performer software library functionalities, while the blue-c API is primarily employed for virtual reality hardware configuration, spatialized sound simulation and interaction processing.

The infoticle system is able to visualize an Intranet log file of several months, containing a total of about 400.000 Intranet log entries, generated by about 12.000 unique files. The visualization updates the data values in a time-ordered way, as it retrieves and visualizes new data objects within a database timeframe of one hour with a tempo of about one second. Infoticles adapt to the continuous change of average forces and spatial behaviors in a stable manner. The dataset can be analyzed by observing the individual (comet, quark) or global (star, electron, burst) behaviors. The different visual patterns can be detected both by detecting dynamic behaviors and by searching in specific spatial areas of interest within the scene. Although the movements and events are purely driven by a continuous data updates, the process of pattern detection and pathline interpretation has some analogy with scientific flow visualizations, in which e.g. distinctions are made between different sorts of flows. As users are able to manipulate the system's timeline and can visualize three separate ribbons simultaneously, smaller trajectory comparisons and causality tests can be made as well.

7.2. Evaluation

The infoticle visualization was able to uncover data patterns that were previously unknown. The document type color coding unmasked relatively more structural files as quarks, proving that the website itself was accessed by most regions, in contrast to the document files, and so the knowledge. Many documents enjoyed a time-limited popularity, a phenomenon that is probably related to actual projects or hot research topics at those points in time. A very significant discrepancy was detected between the knowledge sharing between specific pairs of regions. Some caching mechanism within the Adobe Acrobat Reader plug-in seems to generate significant mistakes within the log mechanism, transforming the corresponding knowledge documents automatically and wrongfully as very popular in traditional log rankings. The different visual patterns were easily perceived by users as document usage categories. Test users had little problems with spatial orientation, a problem often seen in immersive virtual reality applications, and especially liked the ability to explore the dataset in a direct way, the time direction and speed manipulation, and the feeling of being fully immersed 'inside' the data without being disturbed by menus or widgets. The participating company especially appreciated the non-quantitative and still interactive approach of the application, instead of automatically produced and probably faulty popularity rankings consisting of cryptic document names.

The application seemed to produce a high degree of user engagement, as people liked to use it for longer periods of time and expressed their appreciation for the overall interface design. Users especially appreciated the direct and simple user interaction, which stimulated the exploration of the dataset at different levels of detail while maintaining a contextual view. In contrast, some users, mostly inexperienced in the world of computing, seemed to be visually overwhelmed, probably by the combination of the immersive virtual reality technology and the continuously dynamic visualization features. It was also noticed that mostly persons without any virtual world experience expressed problems with effective navigation and interaction.

7.3. Dataset

An ideal infoticle dataset consists of a reasonably large quantity of data objects of which a significant percentage is updated during each database timeframe, containing only few data dimensions or a limited amount of possible unstructured data values, and may include some degree of data inaccuracies (~noise). Infoticles can visualize most time-varying characteristics (e.g. similarity in frequency, long- and short-term evolutions, relative data value alterations, etc.) of both quantitative and unstructured datasets. Typical infoticle datasets contain time-stamped entries of specific reoccurring dynamic data objects with constant data attributes. Whether this dataset is collected beforehand and thus represents the tendencies of an historical timeline or is continuously streamed and stored in the database in real-time by some sort of external update process does not play a substantial role for the proposed visualization methodology.

7.4. Conclusion

This application prototype demonstrates that the data-driven particle concept is particularly suited to detect so-called odd-performers or the very small, unexpected percentage of data that is 'different', as these visually 'pop-out' either by their motion behavior (e.g. erratic), easily recognizable pathlines, or unique directional (e.g. away from center) or spatial positions (apart from other infoticles). Furthermore, this method also visualizes global patterns by spatial directional and behavioral clustering, as those infoticles with similar motion patterns (e.g. circular orbits), directions and velocities (e.g. bursts), or close proximities are visually grouped. This application proves that infoticles enable the exploration of relative large amounts of data along a timeline in different simulation speeds and duration granularities, while keeping a contextual overview over the whole dataset at each point in time.

8. DISCUSSION & FUTURE WORK

This paper has presented a novel method of exploratory information visualization that exploits the motion properties of data-driven infoticles, by using the qualities of spatial awareness and stereoscopic vision of immersive virtual environments. This metaphor has been applied to analyze a large, time-based dataset retrieved from corporate Intranet logs. The driving local rules have been described that react upon the continuous data value updates, and showed how they result into different infoticle behavior typologies. As a result, different visually perceivable infoticle categories can be recognized, e.g. dynamically, by motion characteristics, and statically, by interpreting pathline shapes. Users are also capable to model the force constellations within the scene, and change the time simulation direction and speed.

Several other scenarios were proposed and implemented during the development process, as some users wanted to follow the time-dependent performance of their own created documents in relation to the whole document collection, or vice versa, employees wanted to discover those documents that seemed to be important for their regional office. Our current application was originally developed to effectively use the unique stereoscopic and immersive characteristics of virtual reality environments, but a stripped-down, simplified two-dimensional infoticle application can be imagined that uses less sophisticated hardware resources, and so might solve the overwhelming impressions that some users have expressed. To demonstrate the versatility of this information visualization method, other possible application scenarios in fundamentally different conceptual fields, such as real-time stock quote data streams or financial visualizations will be evaluated.

ACKNOWLEDGEMENTS

This work is accomplished within the ETH-Zurich blue-c project[32], in which four academic chairs developed a novel, networked tele-immersive virtual reality theater. We would like to thank Prof. Gerhard Schmitt of ETH-Zurich, and Alvise Simondetti, Chris Luebkeman of ARUP and the whole ARUP Research & Development group for their input and active support. Also thanks to Martin Naef, Edy Lamboray and the whole blue-c team for their continuous help.

REFERENCES

1. B. Laurel, *Computers as Theatre*, Addison-Wesley, Reading, Massachusetts etc., 1991.
2. H. R. Nagel and E. Granum, "A Software System for Temporal Data Visualization in Virtual Reality", in *Software System for Temporal Data Visualization in Virtual Reality, Workshop on Data Visualization for large data sets and Data Mining*, Augsburg, Germany, 2002.

3. J. M. Hellerstein, A. Chou, C. Hidber, C. Olston, V. Raman, T. Roth, and P. J. Haas, Interactive Data Analysis: The Control Project, *IEEE Computer* 32 (8), 51, 1999.
4. C. Fields and N. Negroponte, "Using New Clues to Find Data", in *Third International Conference on Very Large Data Bases*, IEEE-CS and ACM, Tokyo, Japan, pp. 156-158, 1977.
5. A. van Dam, A. S. Forsberg, D. H. Laidlaw, J. J. LaViola, and R. M. Simpson, Immersive Vr for Scientific Visualization: A Progress Report, *IEEE Computer Graphics and Applications* 20 (6), 26-52, 2000.
6. M. Chalmers and P. Chitson, "Bead: Explorations in Information Visualization", in *ACM Conference on Research and Development in Information Retrieval (ACM SIGIR)*, ACM Press, Copenhagen, Denmark, pp. 330-337, 1992.
7. J. Leigh, A. Johnson, T. A. DeFanti, and M. Brown, "A Review of Tele-Immersive Applications in the Cave Research Network", in *IEEE Virtual Reality*, Houston, Texas, pp. 180-188, 1999.
8. W. T. Reeves, Particle Systems-a Technique for Modeling a Class of Fuzzy Objects, *Computer-Graphics* 17 (3), 359-76, 1983.
9. W. Merzkirch, *Flow Visualization*, Academic Press, New York, 1987.
10. S. Bryson and C. Levit, "The Virtual Windtunnel: An Environment for the Exploration of Three-Dimensional Unsteady Flows", in *IEEE Visualization*, IEEE, San Diego, California, pp. 17-24, 1991.
11. C. W. Reynolds, Flocks, Herds, and Schools: A Distributed Behavioral Model, *Computer Graphics* 21 (4), 25-34, 1987.
12. P. Crossno and E. Angel, "Visual Debugging of Visualization Software: A Case Study for Particle Systems", in *Visualization (VIS'99)*, IEEE Press, San Francisco, California, United States, pp. 417- 420, 1999.
13. A. Pang and K. Smith, "Spray Rendering: Visualization Using Smart Particles", in *IEEE Visualization*, IEEE, San Jose, California, pp. 283-290, 1993.
14. G. Proctor and C. Winter, "Information Flocking: Data Visualisation in Virtual Worlds Using Emergent Behaviours", in *Virtual Worlds 98*, Springer, Paris, France, pp. 168-176, 1998.
15. B. A. Kadrovach and G. B. Lamont "A Particle Swarm Model for Swarm-Based Networked Sensor Systems", in *ACM symposium on applied computing*, ACM Press, Madrid, Spain, pp. 918-924, 2002.
16. P. Eades, A Heuristic for Graph Drawing, *Congressus Numerantium* 42, 149- 160, 1984.
17. S. Pettifer, J. Cook, and J. Mariani, "Towards Real-Time Interactive Visualisation in Virtual Environments: A Case Study of Q-Space", in *International Conference on Virtual Reality 2001*, Laval, 2001.
18. K. Nakayama and G. H. Silverman, Serial and Parallel Processing of Visual Feature Conjunctions, *Nature* 320 (6059), 264-265, 1986.
19. C. Ware and G. Franck, "Evaluating Stereo and Motion Cues for Visualizing Information Nets in Three Dimensions", in *ACM Transactions on Graphics*, pp. 121-140, 1996.
20. R. Pausch, D. Proffitt, and G. Williams, "Quantifying Immersion in Virtual Reality", in *International Conference on Computer Graphics and Interactive Techniques*, ACM, pp. 13-18, 1997.
21. L. Arns, D. Cook, and C. Cruz-Neira, "The Benefits of Statistical Visualization in an Immersive Environment", in *Virtual Reality*, IEEE, pp. 88-95, 1999.
22. F. Heider and M. Simmel, An Experimental Study of Apparent Behavior, *American Journal of Psychology* 57, 243–259, 1944.
23. A. Michotte, *The Perception of Causality*, Basic Books, New York, 1963.
24. J. B. Morrison, B. Tversky, and M. Bétrancourt, "Animation: Does It Facilitate Learning", in *AAAI 2000 Spring Symposium Smart Graphics*, Stanford, CA, USA, pp. 53-60, 2000.
25. B. Tversky, J. B. Morrison, and M. Betrancourt, Animation: Can It Facilitate?, *International Journal of Human-Computer Studies* 57 (4), 247-262, 2002.
26. L. Bartram, *Enhancing Information Visualization with Motion*, Ph.D. Thesis, Simon Fraser University, 2001.
27. T. C. Lethbridge and C. Ware, Animation Using Behavior Functions, in *Visual Languages and Applications*, Ichikawa, T., Jungert, E., and Korfhage, R. R. Plenum Press, New York, 1990, pp. 237-252.
28. A. Vande Moere, "Infoticles: Information Modeling in Immersive Environments", in *International Conference on Information Visualisation (IV02)*, London, England, pp. 457-461, 2002.
29. B. Shneiderman, Dynamic Queries for Visual Information Seeking, *IEEE Software* 11 (6), 70-77, 1994.
30. M. Naef, E. Lamboray, O. Staadt, and M. Gross, "The blue-c Distributed Scene Graph", in *IPT/EGVE 2003*, ACM Press, Zurich, Switzerland, pp. 125-133, 2003.
31. J. Rohlf and J. Helman, "Iris Performer: A High Performance Multiprocessing Toolkit for Real-Time 3d Graphics." in *ACM SIGGRAPH*, pp. 381-395, 1994.
32. M. Gross, S. Würmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. Van Gool, S. Lang, K. Strehlke, A. Vande Moere, and O. Staadt, "blue-c: A Spatially Immersive Display and 3d Video Portal for Telepresence", in *ACM SIGGRAPH 2003*, San Diego, CA, California, pp. 819-827, 2003.