Time-Varying Data Visualization using Information Flocking Boids

Andrew Vande Moere*
Key Centre of Design Computing and Cognition, University of Sydney

Abstract

This research demonstrates how principles of self-organization and behavior simulation can be used to represent dynamic data evolutions by extending the concept of information flocking, originally introduced by Proctor & Winter [1], to time-varying datasets. A rule-based behavior system continuously controls and updates the dynamic actions of individual, three-dimensional elements that represent the changing data values of reoccurring data objects. As a result, different distinguishable motion types emerge that are driven by local interactions between the spatial elements as well as the evolution of time-varying data values. Notably, this representation technique focuses on the representation of dynamic data alteration characteristics, or *how* reoccurring data objects change over time, instead of depicting the exact data values themselves. In addition, it demonstrates the potential of motion as a useful information visualization cue.

The original information flocking approach is extended to incorporate time-varying datasets, live database querying, continuous data streaming, real-time data similarity evaluation, automatic shape generation and more stable flocking algorithms. Different experiments prove that information flocking is capable of representing short-term events as well as long-term temporal data evolutions of both individual and groups of time-dependent data objects. An historical stock market quote price dataset is used to demonstrate the algorithms and principles of time-varying information flocking.

CR Categories: H.5 [Information Systems]: Information Interfaces And Presentation (I.7); I.2.11 [Artificial Intelligence] Distributed Artificial Intelligence

Keywords: time-varying information visualization, artificial life, 3D information visualization, motion, boids.

1. Introduction

Emergence is the process of deriving some new and coherent structures, patterns and properties in a complex system. Emergent phenomena are typically generated by a specific mechanism of simultaneous interactions between all the elements of a certain system over time that sense a situational awareness of their environment. Accordingly, the research presented in this paper is based upon the assumption that typical information visualization techniques such as spatial clustering could be driven by data-dependent interaction rules. These are valid within the visualization space itself, for instance by determining dynamic behaviors of visual elements depending on their relative spatial position and the data values of the elements in their environment. Information visualizations that are based upon such behavior rules could possibly lead to novel techniques that would perform in ways that exceed what they have been explicitly programmed to do. Such emergent data representations would act in *self-organizing* ways, displaying meaningful data patterns both locally, by interactions between nearby entities, and globally, by generating recognizable emergent behaviors of groups of entities.

Consequently, such emergent principles could be integrated within time-varying (also called *dynamic*, *time-dependent*, *time-variant*, *time-based* or *temporal*) information visualizations, as they are capable of representing both instant events as well as continuing data evolutions. Such visualizations have the potential to be used for exploratory tasks, in which users typically have little knowledge about the temporal characteristics or in which experts wish to have a drastically different look at the dataset.

This paper is mainly based on the mathematical flocking simulation of birds or fish, normally used for computer graphics real-world simulation effects, such as in popular movies [2]. These swarming movements are generated by behavior rules introduced by Reynolds [3], who successfully modeled the movements of so-called *boids* (or bird-objects) within a flock. A possible functional explanation for this social flocking behavior describes how animals at the edge of the herd are more likely to be selected by predators [4]. Accordingly, boids would 'selfishly' attempt to move as close to the center of the herd as possible, and thus, in visualization terminology, cluster. Such boids act as *agents*: they are situated, viewing the world from their own perspective rather than from a global one, and their actions are determined by both internal states as well as external influences.

Flocking behavior has already been explored for information display by Proctor and Winter [1], who showed how the clustering movements of swarming fish are able to represent the static relationships of interest of employees, based upon a simple, static similarity weight matrix. This paper extends this *information flocking* concept with several novel features, such as time-varying data simulation, live database querying, continuous data streaming, local data caching, real-time data similarity evaluation, data-dependent behavior rules, automatic shape generation and more stable flocking algorithms. As a result, this visualization technique is able to generate dynamic visual patterns that are based upon long-term as well as short-term temporal similarities within time-varying datasets. In addition to the spatial clustering mechanisms, it introduces a set of global and local motion typologies as novel visual cues for dynamic data representation.

2. Background

2.1. Time-Varying Data

Time-varying datasets contain collections of data objects that are altered in time, due to continuously executed, time-dependent *data updates*. A *data update* can be imagined as the sequential change of a data value (e.g. stock market price) to a successive timeframe (e.g. daily) of a specific data object (e.g. a company). A data update can be classified according to its specific *time-dependent behavior*, e.g. continuous or discontinuous, by its

* andrew@arch.usyd.edu.au

IEEE Symposium on Information Visualization 2004
October 10-12, Austin, Texas, USA
0-7803-8779-1/04/$20.00 ©2004 IEEE

frequency of change, e.g. regularly or unpredictably event-based, or by its *relative data value change*, e.g. noisily or significantly.

Time-varying information visualization differs considerably from traditional static data representations of fixed datasets. Often, its users are not concerned with exact data values currently valid (e.g. stock market quote prices), but rather are interested in *how* data values evolve in time (e.g. a company performing significantly better than the day before, regardless of the exact stock market quote price) or in the context of the whole dataset (e.g. a company quote performing better than all others on a single day). Only a few approaches exist that are capable of representing such time-varying characteristics alterations over time, and mostly are limited to a sequence of static data representations.

2.2. Time-Varying Data Visualization

The following list compares alternative, commonly used methods for visualizing time-varying datasets.

Static State Replacement. These techniques represent data value updates by *instantly* replacing a static world with another version, mostly because the visualization generation requires considerable calculation efforts. The disadvantage of this approach is that the continuous sequences can be ineffectively perceived as discrete steps [5].

Time-Series Plots. These approaches typically employ time-series plotting, connecting sets of static states that are mapped in space and time with simple curves, stacks or timelines, such as stock market chart line diagrams, web usage bar charts, or line and river metaphors [6].

Static State Morphing. Temporal data can be filtered by directly changing the selected time period, a method also called *dynamic queries* [7]. Static state morphing techniques interpolate the visualization between different fixed states that accurately represent the data values within discrete time intervals. Notably, it is the object's motion and not the nature of the motions that carries the informational values. These techniques also differ from information flocking because they require some pre-computation of the static states, and thus are unable to visualize real-time data.

Control Applications are like *online aggregations* and can be classified as anytime algorithms that can produce a meaningful approximate result at any time during their execution. Instead of requiring a dedicated amount of time to build up the scene, a control project gradually streams representative data objects to the visualization system [8].

Equilibrium Attainment. Force-directed diagrams and self-organizing maps show many conceptual similarities with information flocking, as they are internally controlled by local interactions and only reach a state of equilibrium after a specific adaptation time [9]. However, most force-directed methods visualize static datasets and require pre-computed data similarity matrices to determine the spring strengths between pairs of points. Notably, force-directed visualizations only represent informational values by individual distance differences, and do not generate sets of recognizable behaviors as the motion characteristics carry no specific meaning.

2.3. Cognitive Science

Motion is generated through a process called *animation*, in which form and structure evolve through a certain development over time, thereby conveying the feeling of *kinetics* or *dynamics*. Well-designed motion metaphors can be aesthetically appealing, are able to attract attention, maintain motivation and facilitate comprehension, learning, memory, and communication. Mostly, computer-generated motion is used to demonstrate change through a process of *morphing*, or to grasp the awareness of participants by displaying unexpected events.

Normally, animated objects follow predefined paths that are based on various mathematical equations or constraints. However, motion can also be controlled by so-called *behavior functions*, which determine the dynamic behavior of objects based on the status of other objects in the environment rather than interpolating spatial positions between fixed control points. Such behaviors functions generally consist of so-called *local* rules. These rules are local in two senses: each individual member contains its own set of rules and the future state of a member only depends on its neighbors. The goal of this approach is to generate interpretatively rich and unexpected behaviors that seem to be intentional, by provoking the perception of causality, animacy and initiative.

Michotte [10] suggests that causal relationships are perceived directly when certain simple animation techniques are used, such as launching, entraining and triggering. This phenomenon proves the potential of the rich expressive vocabulary of motion for information visualization purposes [11]. Lethbridge and Ware [12] used simple behavior functions based on distance, velocity and direction to model complicated relationships such as pulling, pushing, chasing, escaping, repulsion, collision and anticipation. Such *behavioral animation* techniques employ local rules to determine the dynamic motion of actors in a rule-based system. Basically, a database is made up of a set of *cause-and-effect* rules which the animated objects, usually called actors, must follow. To a certain extent, an autonomous character determines its own actions, as it has some ability to 'improvise'. Similar motion typologies are perceived as grouped, a phenomenon also called *temporal grouping*, and relates to the appearance and locations of the elements, the proximity in time and similarity of motions [13]. Bartram and Ware [14] proved that motion typology has a strong and effective perceptual grouping effect for information display.

Although initially the use of motion might seem to contradict mapping data on 'fixed' coordinates, it has great potential for novel visualization techniques that are independent of spatial positioning and are instead solely based on dynamic properties, for instance those created by emergent behavior generation algorithms.

2.4. Motion-Based Data Visualization

The 3-D Visual Data Mining system (3DVDM) generates three-dimensional temporal data scatterplots that morph between two static states (Nagel and Granum, 2002). Motion is used to denote change from one time state to another, next to the use of *point vibration* as an informational cue, in terms of amplitude, frequency and phase in all three dimensions. Similarly to the information flocking approach, this method solves time-varying data evaluation and visualization performance issues through distributed computing and shared memory parallelism.

The boid concept has already been employed to simulate a realistic mathematical model of dynamic formations in order to characterize swarms with respect to implementation parameters of wireless, ad-hoc network communications systems [15]. This research demonstrated how the boid technique can be used to analyze the network communication link establishment performance by characterizing different sorts of swarm behavior, such as ordered or chaotic, tight or loose, and global or regional.

The emergent behavior of flocks can also be used as an effective search strategy for performing an exploratory geographical analysis, much like detecting visual clusters in large collections of points [16]. It is an example of a new kind of social intelligence model, called Particle-Swarm Optimization (PSO), that, like other evolutionary computation algorithms, can be applied to problem solving, learning and optimization problems in the fields of system design, pattern recognition, biological system modeling, signal processing, decision making, simulation and identification, and so on [17]. In the field of scientific visualization, *swarming agents* have been applied in three-dimensional virtual reality environments to simultaneously represent and solve specific constraints of computational fluid dynamics problems [18].

3. IMPLEMENTATION

Each boid represents a stock market company. The data object of a boid is subjected to a continuous stream of updated data values (i.e. new stock market prices). The goal of the information flocking metaphor is to visualize the characteristics of this update process through an automatic interpretation process.

3.1. Infoticle

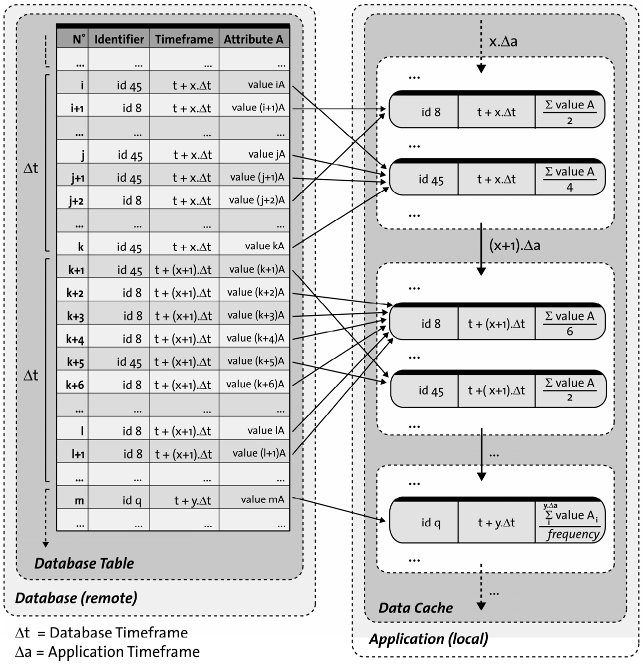
The information flocking technique was implemented on top of the so-called *infoticle* (information–particle) visualization engine [19]. The infoticle technique is a rule-based visualization tool that represents each reoccurring data object within a time-varying dataset with a single particle positioned in three-dimensional space. The system is capable of querying and streaming sequential subsets of a large data collection from a remote database to the visualization engine, and can evaluate the incoming data value alterations in real time. These dynamic data changes then trigger appropriate behavior rules, which in turn determine the spatial behaviors of the corresponding infoticles.

In practice, each boid is represented by a colored, curved line with a gradually decreasing transparency, spatially connecting the three-dimensional positions it has passed through. The boid color depicts positive or negative data value changes. At a specific predefined rhythm, each boid stores its current position and data values within a history list, thus defining the control points and color coding for its line representation. Technically, these traces are generated by the so-called *Catmull-Rom spline* algorithm, which is capable of mathematically representing curves that connect a series of points at irregular intervals. This representation technique was chosen because it generates smooth and stable curves that pass through all defining control points, are continuous in the tangent direction and have second-degree curvatures that change linearly over the length.

Notably, the visualization can be shown on immersive virtual reality installations, which require special considerations regarding spatial orientation, navigation and user interaction. However, this paper will not deal with these specific issues, as the main contributions can be considered independently from the presentation medium.

3.2. Simulation

The system is capable of retrieving, processing and visualizing a continuous stream of data values in real time. The fact of whether this data flow is generated by a separate, remote process, for instance by a live stock market feed, or whether it is retrieved from a database with historical data values plays no significant role.



N°	Identifier	Timeframe	Attribute A
...	...	...	...
i	id 45	t + x.Δt	value iA
i+1	id 8	t + x.Δt	value (i+1)A
...	...	...	...
j	id 45	t + x.Δt	value jA
j+1	id 45	t + x.Δt	value (j+1)A
j+2	id 8	t + x.Δt	value (j+2)A
...	...	...	...
k	id 45	t + x.Δt	value kA
k+1	id 45	t + (x+1).Δt	value (k+1)A
k+2	id 8	t + (x+1).Δt	value (k+2)A
k+3	id 8	t + (x+1).Δt	value (k+3)A
k+4	id 8	t + (x+1).Δt	value (k+4)A
k+5	id 45	t + (x+1).Δt	value (k+5)A
k+6	id 8	t + (x+1).Δt	value (k+6)A
...	...	...	...
l	id 8	t + (x+1).Δt	value lA
l+1	id 8	t + (x+1).Δt	value (l+1)A
...	...	...	...
m	id q	t + y.Δt	value mA
...	...	...	...

Figure 1. Database and application timeframe update process.

A time-varying dataset contains a given set of reoccurring data objects in which the data values change over time. Each data entry has a specific timestamp, which enables the system to query and order subsets of data according to delimiting database and application timeframes, as illustrated in Figure 1.

Database Timeframe determines the duration, measured in physical time units, within which the data that is streamed from the database to the visualization system needs to be positioned. At every application timeframe step, the database timeframe is sequentially shifted and new data is collected. Differences in database timeframe durations, or dataset *time granularities*, result in fundamentally different visual patterns. The database timeframe duration is also directly related to the data quantity that has to be streamed from the database to the visualization, thereby directly affecting the application's performance.

Application Timeframe denotes the rhythm at which the application retrieves the next batch of data objects for the following database timeframe. In practice, it describes the 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 database timeframe, and both are continuously updated in *parallel*. Differences in simulation timeframe durations result in dissimilar boid patterns, as these receive either more or less time to adapt to the newly created situation. For instance, a too short application timeframe would prevent the boids from clustering and leave them in a constantly chaotic state, whereas a too long application timeframes would cause boids to act slowly and less information-generated.

The real-time data similarity evaluation requires the handling of multiple occurrences of data objects within a single database timeframe. As each boid can represent only a single data value for each application timeframe, averaged data values need to be calculated on the fly whenever the database timeframe is longer than the actual data granularity of the stored dataset and multiple equal data object entries exist (see Fig.1).

3.3. Database

The used dataset accumulates the daily opening, closing, high and low prices and the corresponding trade volumes of historical stock market quotes within one year, totaling about (±500 companies x ±200 working days) 12.631 data entries. It was acquired from a public website that collects the historical stock market prices of the 500 Standard & Poor's Index Directory [20], which represents a sample of 500 leading companies in the most important industries of the U.S. economy.

The boid flocking method uses a database instead of a static data storage for its versatile and efficient data querying and retrieval features. Accordingly, the database timeframe can be altered by way of adapting the database query if users want to analyze the data in different time granularities (e.g. analyzing weekly averaged instead of daily stock market quotes). Databases can be queried for data subsets on the fly, so that different data attributes of the same dataset can be analyzed in various levels of detail. In addition, the calculation-intensive effort of browsing, searching, ordering and organizing large quantities of data is transferred to well-established, optimized database algorithms. In addition, a database can be easily changed, extended or updated, even in real time, by parallel, external processes running independently from the visualization application.

Querying, communicating and caching datasets are typical calculation intensive processes. However, the boid flocking method is highly dependent on the correct interpretation of continuous motion typologies, which thus should never be visually interrupted. Therefore, the boid application is controlled by two different processes that run in parallel, preferably on a dual-processor machine. The data process is responsible for querying and caching the data from the remote database, while a separate process renders the scene, calculates the data dependencies and according boid positions. To further optimize the data handling procedures, the data caching process already collects data subsets of the next database timeframe.

3.4. Behavior Rules

The direction and speed of each boid *A* with position $\vec{p}_A$, is dependent on all the boids *X* with position $\vec{p}_X$ in its neighborhood. Following flocking rules apply, which are executed for each single boid and at each single frame.

Collision Avoidance. Pull away before crashing with other boids nearby. If the distance between the boids is within the collision avoidance range d_{CA}, calculate a directional vector pointing away, of which the strength is inversely proportional to the distance between the boids.

$$\left\|\vec{p}_X - \vec{p}_A\right\| \leq d_{CA} \Rightarrow \vec{v}_{CA} = \sum_X \frac{\left|\vec{p}_X - \vec{p}_A\right|}{\left\|\vec{p}_X - \vec{p}_A\right\|}.$$

Velocity Matching. Attempt to move with about the same speed as the neighbors in the flock. If the distance between the boids is smaller than the velocity matching threshold, boid A should attempt to take over the direction of boid X.

$$\left.\begin{array}{l}\left\|\vec{p}_X - \vec{p}_A\right\| \leq d_{VM} \\ \left\|\vec{p}_X - \vec{p}_A\right\| > d_{CA}\end{array}\right\} \Rightarrow \vec{v}_{VM} = \sum_X \vec{v}_X$$

Flock Centering. Attempt to move toward the center of the flock as the boid perceives it. If the distance between $\vec{p}_X$ and $\vec{p}_A$ is smaller than the flock centering limit, boid X should try to direct itself towards boid A.

$$\left.\begin{array}{l}\left\|\vec{p}_X - \vec{p}_A\right\| \leq d_{FC} \\ \left\|\vec{p}_X - \vec{p}_A\right\| > d_{CA}\end{array}\right\} \Rightarrow \vec{v}_{FC} = \sum_X \left|\vec{p}_X - \vec{p}_A\right|$$

These three traditional flocking principles are solely based upon the relative spatial positions of the pairs of boids, and are extended with two separate data-dependent clustering rules. These rules rely on the relative change q_{change} in the data values from the previous database timeframe ($q_{previous}$) to the current one ($q_{current}$). Only one of both rules is valid for each pair of boids *A* and *X*, depending on whether the similarity between the two data alterations is below a specific threshold $t_{threshold}$. In addition, a weight factor w_{DS} between boid A and X is introduced, that is proportional to their data alteration similarity and will influence the strength of the resulting attraction or repulsion force.

$$\frac{q_{current}(A) - q_{previous}(A)}{q_{previous}(A)} = q_{change}(A)$$

$$q_{change}(A, X) = \left|q_{change}(A) - q_{change}(X)\right|$$

$$w_{DS}(A, X) = \frac{\left|q_{threshold} - q_{change}(A, X)\right|}{q_{threshold}}$$

Data Similarity. Attempt to stay close to those boids that have experienced a similar data value evolution during the current database timeframe. The data alteration similarity, in this case the relative change of stock market prices, is determined by calculating the difference between the data value evolutions of the boids. The strength of the attracting force is proportional to the distance between the boids and the similarity between the two data value evolutions.

$$\left.\begin{array}{l}\left\|\vec{p}_X - \vec{p}_A\right\| < d_{DS} \\ q_{change}(A, X) < q_{threshold}\end{array}\right\} \Rightarrow \vec{v}_{DS} = \sum_X w_{DS}(A, X).\left|\vec{p}_X - \vec{p}_A\right|.\left\|\vec{p}_X - \vec{p}_A\right\|$$

Data Dissimilarity. Attempt to stay away from boids with dissimilar data values in the current database timeframe. This rule is similar to the previous data similarity influence, except that the repulsion force and the distance between the boids are inversely proportional.

$$\left.\begin{array}{l}\left\|\vec{p}_X - \vec{p}_A\right\| < d_{DD} \Rightarrow \vec{v}_{DD} \\ q_{change}(A, X) > q_{threshold}\end{array}\right\} = \vec{v}_{DD} = \sum_X w_{DS}(A, X).\frac{\left|\vec{p}_X - \vec{p}_A\right|}{\left\|\vec{p}_X - \vec{p}_A\right\|}$$

$0 < w_{CA}, w_{VM}, w_{FC}, w_{DS}, w_{DD} < 1$ are weights applied to Collision Avoidance, Velocity Matching, Flock Centering, Data Similarity and Data Dissimilarity behaviors respectively. $\vec{v}_A, \vec{v}_{CA}, \vec{v}_{VM}, \vec{v}_{FC}, \vec{v}_{DS}, \vec{v}_{DD}$ denote the accumulated velocity vectors.

$$\left\|\vec{v}_x\right\| > 1 \Rightarrow normalize(\vec{v}_x) \rightarrow x = \{CA, VM, FC, DS, DD\}$$

$$\vec{v}_A = -w_{CA}.\vec{v}_{CA} + w_{VM}.\vec{v}_{VM} + w_{FC}.\vec{v}_{FC} + w_{DS}.\vec{v}_{DS} - w_{DD}\vec{v}_{DD}$$

It was chosen to include two separate data-dependent behavior rules instead of one, to generate a more controllable flocking behavior that can be changed in several terms of data similarity instead of a single repulsion/attraction factor. By also keeping the traditional behavior rules, the general flocking organization influences can still be stabilized or adapted.

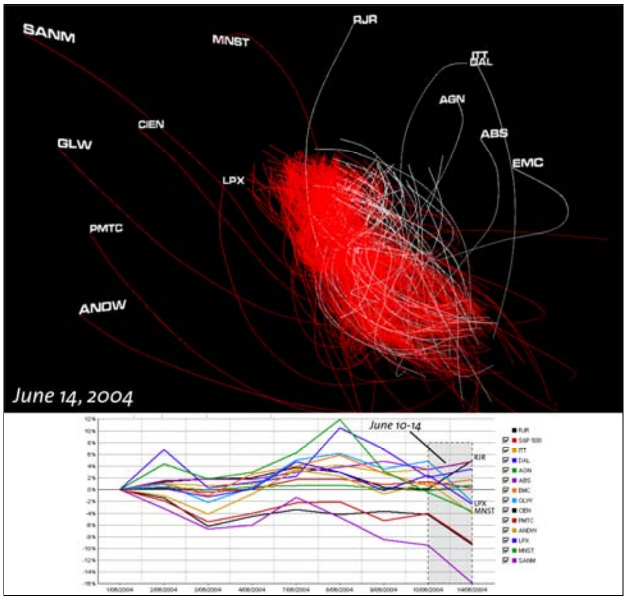


Figure 2. Individual boids separating from the main flock.

In general, the Flock Centering and Velocity Matching weighting factors are small, to avoid any fast directional movement of the whole flock as a whole, and to provide the data similarity rules enough freedom to invoke the clustering tendencies.

The behavior of a flock never attains equilibrium, and expresses continuing characteristics but also unpredictable patterns [21]. Accordingly, it is necessary to stabilize this behavior as much as possible by fine-tuning both the different weighting factor values and the threshold distances. The exact numerical values of these variables are determined through a process of trial and error, as the application designer is mostly unable to foresee the exact outcomes of the simultaneously applied local interactions.

3.5. Shape

The formal language of enclosed figures like blobs, circles, bars, lines, crosses and arrows, suggests certain physical properties that cognitively provoke compelling conceptual interpretations. Furthermore, enclosed figures suggest the possibility of *containing* certain elements, thus effectively separating those elements from others. Accordingly, the information flocking method uses blob shapes to denote data similarity and to support the perception of spatial clusters.

Three-dimensional lines are generally rendered with a constant thickness regardless of the distance from the viewer, so that the representation contains few effective depth perception cues and an accurate spatial overview is made difficult. By enclosing boids within shapes, the visual perception of global and directional trends is enhanced. In effect, a user is able to recognize the visual impact of a single three-dimensional shape more rapidly than navigating around spatial clusters of spatially separated boids. In addition, shapes reduce the amount of points to be observed, and shift the attention from individual movements to a more *formal* interpretation. These shapes also facilitate the detection of individual outlying boids that might have been overlooked otherwise: points that lay outside of cluster centers typically form bumps or significant bulges that are visually more predominant. In addition, the shape surface also reproduces the colors of the boids it conceals, generating highlighted zones of winning and losing stock market quotes on the outskirts of the flock.

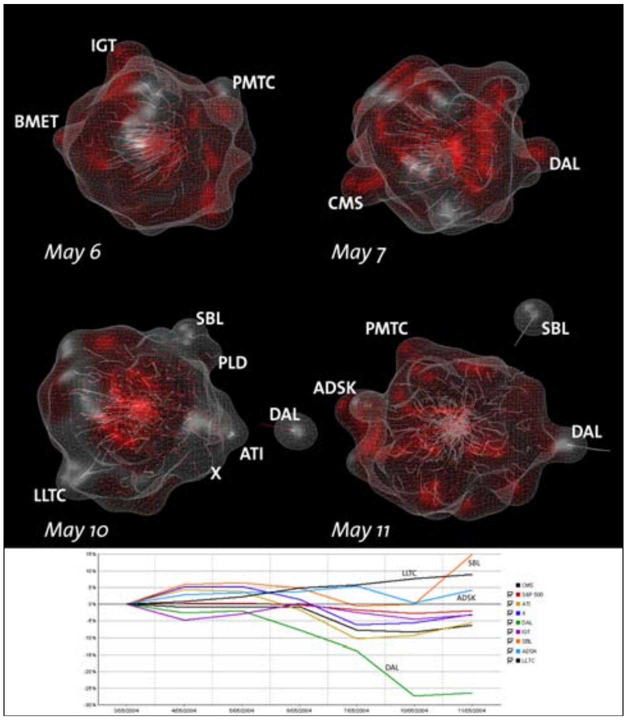


Figure 3. Boid shape generation and evolution.

These shapes are generated by a specific surface mesh algorithm called *marching cubes*. This method is able to generate closed, three-dimensional surfaces that span a number of three-dimensional points, in this case the collection of boids. The implementation is based upon the work of an improved marching cubes method called *implicit surface polygonizer*, which defines the cube divisions dynamically using algorithms instead of a look-up table [22]. This mesh generation technique cannot be executed at each single frame because of computing performance limitations, and so is triggered at time intervals or by user action.

4. RESULTS

4.1. Dynamic patterns

The current information flocking method is capable of clustering groups of boids that experience similar data alterations. Figure 2 shows how sudden significant events are depicted by expulsed individual boids that leave the main flock. The dominant red color shows that the majority of the stock market was exposed to negatively changing stock market prices. Those companies that have a significantly different price change become separated into two directions: those winning (in white and on the right), and those losing (in red and on the left). The parallel directions of the lines denote a similar price change for that database timeframe. As MNST, RJR and LPX experience slightly different price changes in comparison with those in the sub-flocks on that day, their boids move in directions that are not perfectly in parallel.

Figure 3 shows the shapes generated for a simulation timeframe of four days. It demonstrates how shapes can support the perception of the outskirts of the flock, and the estimation of the relative distances. Very significant price changes of individual companies result in the expulsion of the according boids. More subtle spatial self-organization occurs within the main flock: CMS becomes the head of the group at May 7, as its price remained almost constant and experienced a price fall well after that date.

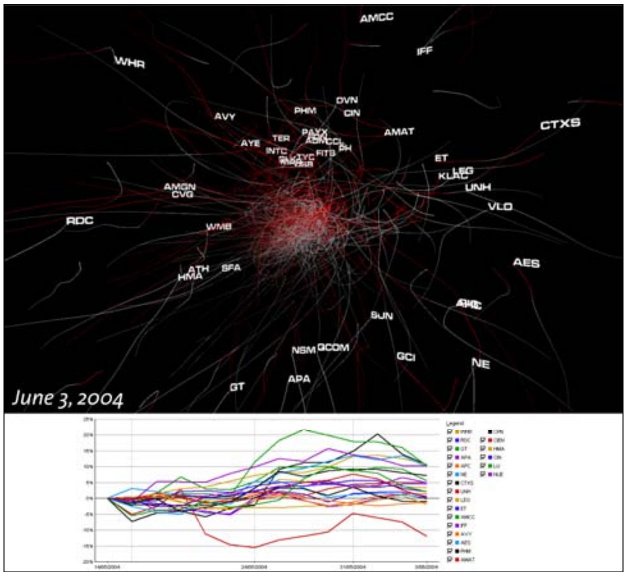


Figure 4. Periphery flock view.

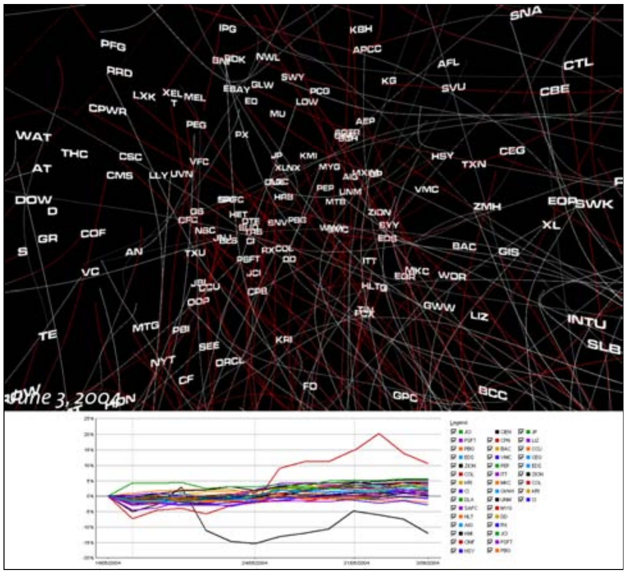


Figure 5. Flock center view.

Although the similar price evolving companies are at the back of the flock, their initial spatial situation makes it impossible to pass the main group immediately, due to the overwhelming collision avoidance forces. One can also perceive smaller bulges of the shape surface, denoting boids at the outer sides of the flock. Although humans almost automatically attempt to classify these formal inconsistencies, experience has shown it is not efficient to do so: often, bulges are generated through the random nature of the general flocking behavior instead of data-driven influences. Therefore, it is more effective to observe the dynamic behavior of individual or groups of boids, as the direction and speed generally do correspond accurately to direct data evolution similarities.

In contrast with Figure 2, which showed short-term events of small subgroups, and Figure 3, which demonstrated how flocks are capable of reorganizing after a few application timeframes, Figure 4 and 5 illustrate long-term group behavior and clustering of the whole flock, here from May 14 until June 3. Both figures illustrate the exact same flock: Figure 4 analyzes the price performance of the boids at the outskirts of the flock, which have clearly experienced large and random price differences, whereas Figure 5 focuses on the flock center, which is clustered closely together and represents equally, steadily changing data entities. Many other short-term dynamic behaviors occur that are difficult to illustrate in static images, such as implosions, explosions (as illustrated in Fig.6), and sudden global direction changes. Often these behaviors are caused by extreme chaotic or, on the contrary, very regular data value changes for the majority of the dataset. Figure 7 demonstrates how a relatively large subgroup of boids is dynamically generated around the data of May 13th, and becomes separated from the main flock because of a significantly similar change in stock market price around that date.

In effect, all the boid figures demonstrate how information flocking boids use directional expulsion as a behavioral technique to correlate unique and outlying dynamic patterns within the time-varying dataset. This method is valid both for individual as well as groups of boids, and is capable of clustering similar data entities in terms of short-term significant events in comparison with the whole dataset as well as long-term data value evolutions. Note that parallel (and not necessarily overlapping) stock price quote chart lines induce the boid clustering tendency.

4.2. Static patterns

It was expected that the *formality* of the boid shapes would correspond with the data update characteristics, so that, for instance, irregular forms would result out of chaotic, significant data alterations, while sphere-like objects would arise out of stable or slowly developing stock market quote prices. However, it was discovered that such interpretations could only be made by a continuous observation of the changing shapes, and not from the perception of separate static entities. Nevertheless, the boid shapes do enable a more efficient recognition of the flock's overall constellation, individual expulsions and its spatial division between periphery and center.

Another way of analyzing a 'frozen' flock collection, is to consider the directionality of the boids: at specific time intervals one can observe a clearly uniform directed movement, in contrast to sudden 'implosions' that often denote very chaotic price evolutions within the majority of the dataset.

As all boids are represented by splines, users are able to perceive the past trajectory of the boids, and evaluate its curvature and directionality relative to the whole flock. In effect, the boid traces act as static, visual artifacts representing the historical motion typologies. In addition, information flocking can also be interpreted by considering both spatial clustering and boid directionality: boids that are in close proximity to one another and at the same time are moving in a parallel direction represent some data update similarity for the active database timeframe.

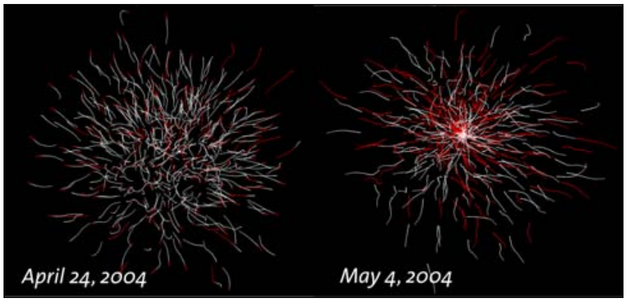


Figure 6. Boid explosion and implosion behaviors.

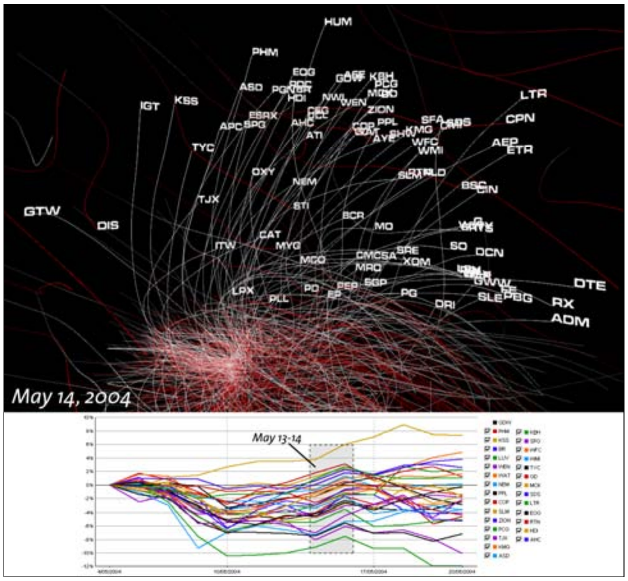


Figure 6. Boid subgroup separation.

5. Evaluation

5.1. Advancements

The research presented in this paper has applied following extensions on the original information flocking method [1].

Time-Varying Dataset. The original paper mentioned the possibility of time-varying datasets, however did not describe any specific implementation results. In fact, one might note some discrepancy between the inherently dynamic nature of boids that continuously move and adapt (and in fact constantly reform spatial clusters), and the representation of inherent static data. In contrast, our implementation is able to adapt to rapid and even chaotic data update sequences in real time.

Behavioral Clustering. Whereas the original research relied solely on the spatial clustering of similar data points, time-varying boids are able to demonstrate dynamic data-dependent motion behaviors on a global level, for instance representing calm (directional movement) or chaotic (implosive, erratic movement) stock market periods or long-term price quote change evolutions (periphery orbiting versus center flocking). In effect, both the boid motion typology and the spatial position relative to the flock and its neighbors carry informational meaning.

Real-Time Evaluation. The infoticle method does not require pre-calculated similarity matrices denoting the relationships between all the individual data entries, but determines the data similarity during execution time. Consequently, this method is capable of representing real-time datasets and inherently different dataset typologies simultaneously.

Algorithmic Alterations. In contrast to the first approach, the current boid approach has not removed the original Flock Centering rule out of the flocking algorithms because it enables a more effective global and time-varying coherence of the general flock behavior. This algorithmic difference between the two methods is most probably caused by solving the continuous directional and spatial 'stress' imposed by the repetitive data update process, which evidently requires more rule-based dynamic stability and thus more organizing power over the whole behavior generation.

Dissimilarity Repulsion. Two, instead of one, extra factors extend the original boid algorithm by also considering data dissimilarity as a separate, individual influence. Alternatively, dissimilarity could not be evaluated at all, or be merged into a single (negative similarity) data influence vector. However, adding this influence separately enforces the clustering and especially the de-clustering tendency of the boids considerably, a factor which is important when the boids have to regroup and show an interpretable constellation within the time allocated by the application timeframe.

5.2. Initialization

The spatial initialization state of the boid collection does not play a significant role, as the quantity and continuity of local boid interactions will randomize the initial setting during the first application timeframes. However, a visualization that initiates from different points on the database timeline will inevitably result in different long-term flocking behaviors, as the boids have not been exposed to the same time-varying 'experience' and thus will be positioned differently within the flock. However, significant short-term events will be represented correctly, as these depend much less on past data evolutions.

Flocking behavior has the natural tendency to move to specific, unexpected directions, and therefore has the inherit risk of disappearing by floating off the screen space. Although the current prototype minimizes this risk by selecting a well-considered flock centering weighting factor, appropriate navigation and interaction means are still required to follow and analyze the flock from different distances and directions.

5.3. Equilibrium Attainment

The behavior of a flock never attains equilibrium [21]. The information flocking technique uses a dynamic and evolutionary process to produce a visual representation that is constantly subjected to a state of *flux*. Distances between boids with equal directions or behaviors denote a degree of data similarity. Spatial direction or behavior inequality means that those boids are subject to different data update histories. However, these proximity measures need to be considered in relation to the phenomenon of the true *representation equilibrium*. Immediately after each data update, the complete boid collection is transformed into a state of extreme instability, as all boids with changed data values are influenced by new internal spatial behavior specifications. Consequently, a specific *equilibrium adaptation time* $t_{equilibrium}$ is required to reach a true, stable representation. A state of equilibrium is attained when all boids have reached their destination within a stabilized flock, and subsequently have acquired a constant dynamic behavior pattern. In theory, such a true representation can be analyzed in a static state without the risk of misinterpretation. In practice however, the application timeframe is mostly shorter than the required equilibrium adaptation time. This subjects the resulting representation with a continuous level of stress, which causes fast and efficient dynamic adaptations. In practice, the required equilibrium time span $t_{adaptation}$ is directly related to a single boid i which is positioned at the greatest distance d_i from its mates with similar data values, and is traveling with velocity v_i to its final mates destination T_i.

$$t_{equilibrium}(i) = \frac{d_{T_i}}{v_i} \rightarrow t_{equilibrium} < t_{application}$$

This equilibrium adaptation time is effectively utilized to visualize similarity in long-term data history evolutions. Boids that represent similar data values over time, and are consequently already close together, receive relatively more equilibrium adaptation time to cluster closely in a equilibrium than those that were subjected to a considerable amount of different data values within this timeframe, and are further away as a result of this. As shown in Figures 4 and 5, the continuous stress and instability influences that are continuously imposed on the boid collection generate an effective long-term separation within the flock between stable and significantly changing time-varying data objects.

5.4. Time

One should note that this method does not offer an immediate representation of a dataset at a certain point in time, but rather requires a specific time simulation to calculate the time-varying data similarities and evolutions out of which the spatial flock constellation arises. Whereas sudden individual boid expulsions are generally easily spotted and denote exceptional changes in the stock quotes of the corresponding companies, collective behaviors typically require slightly longer observation time spans to be effectively perceived. In effect, information flocking pattern recognition is based upon an uninterrupted comparison of altering animated clusters and shapes generated by groups of boids, instead of analyzing the spatial composition of a three-dimensional world. However, a static boid representation can still be analyzed on the level of spatial clustering, by comparing both the directions and proximities of nearby boids. However, the boid behaviors offer little cues about the reasons behind the data dependencies or evolutions, requiring the use of alternative tools or information sources to analyze the specific data patterns in more detail.

6. Conclusion

This paper has described the algorithms and principles that drive the information flocking method, and how they can be applied to a real-world and almost chaotic dataset. It has demonstrated a working prototype which is capable of handling 500 boids in real-time, generating a dynamic information visualization that depicts the evolutionary data similarities within a time-varying data collection over different time periods. It has shown how the original information flocking principles of spatial clustering are extended with dynamically generated motion typologies to represent similarities in time-varying datasets, or *how* data evolves over time, both for individual and groups of data objects. Further research should prove whether data-driven boids are capable of also qualitatively representing the so-called *delta* or change in time-varying datasets.

Information flocking could be made more user-friendly by depicting the degree of data dependencies within a flock of user-selected boids by color or by automatically labeling thematic subgroups within the flock during the visualization simulation itself. Furthermore, users could be offered menus or tools to enable the adaptation of boid dependencies and timeframe durations during runtime, making the visualization more interactive and explorative. We are currently evaluating the information flocking technique with other, less chaotic, time-varying datasets, such as lexicalized concepts from language-based human communications during creative design meetings.

Acknowledgements

This research was accomplished within the framework of the blue-c project at the Swiss Federal Institute of Technology (ETH-Zurich), which developed novel technologies for tele-immersive virtual reality installations. We would like to thank Prof. Gerhard Schmitt and Prof. Markus Gross for their continuous support and ideas, and Kuk Hwan Mieusset and Martin Naef for their programming support. All stock market quote graphs are based upon the stock market visualization applet offered by MSN MoneyCentral at http://moneycentral.msn.com.

References

[1] G. Proctor and C. Winter. Information Flocking: Data Visualisation in Virtual Worlds Using Emergent Behaviours. In *Virtual Worlds 98*, pages 168-176, Springer, 1998.

[2] A. Stanton and L. Unkrich. Finding Nemo. USA, Buena Vista Pictures / Walt Disney Pictures, 101 mins, 2003.

[3] C.W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*, 21(4), pages 25-34, 1987.

[4] W.D. Hamilton. Geometry for the Selfish Herd. *Journal for Theoretical Biology*, 31, pages 295-311, 1971.

[5] B. Tversky, J.B. Morrison, and M. Betrancourt. Animation: Can It Facilitate? *International Journal of Human-Computer Studies*, 57(4), pages 247-262, 2002.

[6] S. Havre, et al. Themeriver: Visualizing Thematic Changes in Large Document Collections. *Transactions on Visualization and Computer Graphics*, 8(1), pages 9-20, 2002.

[7] B. Shneiderman. Dynamic Queries for Visual Information Seeking. *IEEE Software*, 11(6), pages 70-77, 1994.

[8] J.M. Hellerstein, et al. Interactive Data Analysis: The Control Project. *IEEE Computer*, 32(8), pages 51, 1999.

[9] P. Eades. A Heuristic for Graph Drawing. *Congressus Numerantium*, 42, pages 149-160, 1984.

[10] A. Michotte. *The Perception of Causality*. Basic Books, New York, 1963.

[11] C. Ware, E. Neufeld, and L. Bartram. Visualizing Causal Relations. In *IEEE Symposium on Information Visualization (Infovis'99)*, pages 39-42, 1999.

[12] T.C. Lethbridge and C. Ware. Animation Using Behavior Functions. In: T. Ichikawa, E. Jungert, and R.R. Korfhage, eds. *Visual Languages and Applications*. Plenum Press, New York, pages 237-252, 1990.

[13] P. Kramer and S. Yantis. Perceptual Grouping in Space and Time: Evidence from the Ternus Display. *Percept Psychophys*, 59(1), pages 87-99, 1997.

[14] L. Bartram and C. Ware. Filtering and Brushing with Motion. *Journal of Information Visualization*, 1(1), pages 66-79, 2002.

[15] B.A. Kadrovach and G.B. Lamont. A Particle Swarm Model for Swarm-Based Networked Sensor Systems. In *ACM symposium on applied computing*, pages 918-924, ACM Press, 2002.

[16] J. Macgill and S. Openshaw. The Use of Flocks to Drive a Geographic Analysis Machine. In *International Conference on GeoComputation*, 1998.

[17] J. Kennedy and R.C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, 2001.

[18] P. Brewster. Advanced Interfaces for Virtual Environments. In *NASA Workshop on the Knowledge Integrating Virtual Iron Bird*, 2004.

[19] A. Vande Moere, K.H. Mieusset, and M. Gross. Visualizing Abstract Information Using Motion Properties of Data-Driven Infoticles. In *Conference on Visualization and Data Analysis 2004 (IS&T/SPIE Symposium on Electronic Imaging)*, SPIE, 2004.

[20] SWCP. Historical Data for S&P 500 Stocks. 2003, URL, http://kumo.swcp.com/stocks.

[21] J. Toner and Y. Tu. Flocks, Herds, and Schools: A Quantitative Theory of Flocking. *Physical Review E.*, 58, pages 4828-4858, 1999.

[22] J. Bloomenthal. Polygonization of Implicit Surfaces. *Computer Aided Geometric Design*, 5, pages 341-355, 1988.