



**HAL**  
open science

## Visual Sedimentation

Samuel Huron, Romain Vuillemot, Jean-Daniel Fekete

► **To cite this version:**

Samuel Huron, Romain Vuillemot, Jean-Daniel Fekete. Visual Sedimentation. IEEE Transactions on Visualization and Computer Graphics, 2013, IEEE Transactions on Visualization and Computer Graphics, 19 (12), pp.2446-2455. 10.1109/TVCG.2013.227 . hal-00846260

**HAL Id: hal-00846260**

**<https://inria.hal.science/hal-00846260>**

Submitted on 27 Sep 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Visual Sedimentation

Samuel Huron, Romain Vuillemot, and Jean-Daniel Fekete, *Senior Member, IEEE*

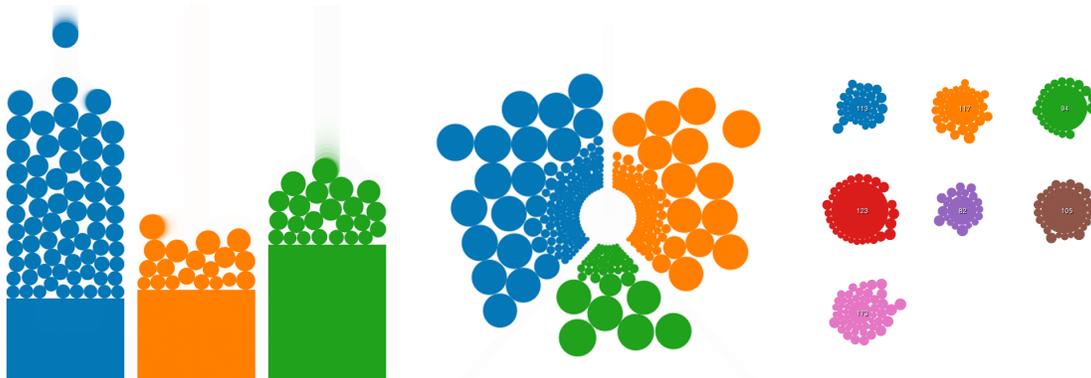


Fig. 1. The Visual Sedimentation metaphor applied to a bar chart (left), a pie chart (center), and a bubble chart (right).

**Abstract**—We introduce *Visual Sedimentation*, a novel design metaphor for visualizing data streams directly inspired by the physical process of sedimentation. Visualizing data streams (e. g., Tweets, RSS, Emails) is challenging as incoming data arrive at unpredictable rates and have to remain readable. For data streams, clearly expressing chronological order while avoiding clutter, and keeping aging data visible, are important. The metaphor is drawn from the real-world sedimentation processes: objects fall due to gravity, and aggregate into strata over time. Inspired by this metaphor, data is visually depicted as falling objects using a force model to land on a surface, aggregating into strata over time. In this paper, we discuss how this metaphor addresses the specific challenge of smoothing the transition between incoming and aging data. We describe the metaphor’s design space, a toolkit developed to facilitate its implementation, and example applications to a range of case studies. We then explore the generative capabilities of the design space through our toolkit. We finally illustrate creative extensions of the metaphor when applied to real streams of data.

**Index Terms**—Design, information visualization, dynamic visualization, dynamic data, data stream, real time, metaphor

## 1 INTRODUCTION

This paper introduces *Visual Sedimentation*, a novel design metaphor inspired by the physical process of sedimentation. This process is the result of objects falling due to gravitational forces and then aggregating over time into compact layers. We show how the physical properties of the sedimentation process can be applied to the design of effective visualizations of data streams through metaphor: new items are the equivalent to falling objects, animated by virtual forces, and aggregating over time in area charts.

Data streams are sequences of typed objects, and are very common with social networks updates, tweets, emails, network logs, RSS feeds, or updates in distributed version control systems. The challenges of visualizing data streams match many characteristics of the physical sedimentation process: data *appear* at unpredictable times, *accumulate* until they are processed, and need to be kept in *aggregated* form to provide historical and contextual information over time. Designing visualizations to convey those various stages is not straightforward as many visual representations may have to be displayed simultaneously. We are specifically interested in smoothing the transition between the data stream’s focus—recent data—and the context—older data.

In this paper, we define the sedimentation process as it appears in

- Samuel Huron is with INRIA and IRI. E-mail: samuel.huron@cybunk.com.
- Romain Vuillemot is with INRIA. E-mail: romain.vuillemot@inria.fr.
- Jean-Daniel Fekete is with INRIA. E-mail: jean-daniel.fekete@inria.fr.

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

nature, followed by an explanation of how it applies as a metaphor for data streams visualizations. To facilitate the design of visualizations using this metaphor, we implemented a supporting toolkit; we applied this toolkit to design and implement case studies of both classical and novel visualizations based on real-life datasets. Building on the success of these case studies, we then explore the Visual Sedimentation design space by modifying the toolkit parameters and creating a group of variant visualizations. This process of deconstructing the design space has provided us with creative results that we present. We finally address technical issues related to the implementation of the Visual Sedimentation metaphor before discussing related and future works.

## 2 SEDIMENTATION

*Sedimentation* is commonly understood as the deposition of a solid material from air or water. Biologists define it as “*the tendency for particles in suspension to settle out of the fluid in which they are entrained, and come to rest against barrier*” [40]. Geologists extend this definition to “*deposits from glacial ice and those materials collected under the impetus of gravity alone, as in talus deposits, or accumulations of rock debris at the base of cliffs*” [6].

Many factors contribute to the sedimentation process; we focus here on several elements that are applicable to our metaphor. The central element of the sedimentation process is the **sediment** itself: rock, dust or particles that vary in composition, size and weathering stage (Figure 2, center). External **physical forces** are applied to sediments transporting them according to the forces’ directions and magnitudes. Furthermore, depending on weathering and collisions, sediments may split or become compressed. The process that compresses sediments over time is called **decay**. Sediments end their journey as they accumulate on each other and on **barriers**, settling and aggregating into sediment-



Fig. 2. Illustrations of real life sedimentations. Fossils [2] (left), types of sediments [20] (middle), Mississippi riverbed [15] (right).

tary rocks; this process is called **flocculation**. Sedimentary rocks may separate into distinctive layers or **strata**, with the oldest strata on the bottom and the youngest on the top. These strata reflect the different nature of sediments, time periods, or compression over time with **deformations** in organic or liquid-like shapes.

As sedimentation is part of a global physical environment, physical forces also play a role in this process and the deposit of visible traces. For example, glaciers leave **trails** of heterogeneous sediments in their wake. Similarly, riverbed deposits show historical meanders in a river's floodplain (Figure 2, right); finer grain sediments deposits in suspension visually mark the maximum water level, an example of naturally-occurring **contours**. Fluids may also leave traces of past interactions between its flow and the sediment deposits with **ripple** marks. Sedimentation not only leaves visual cues or traces of the past, but also captures snapshots of living ecosystems through fossils.

Humans are familiar with the visible parts of the sedimentation process, such as mountain formations, dunes or riverbeds occurring at the surface of the Earth. The underground portion of sedimentation (*i. e.*, strata) previously had only been revealed to us as exposed cliff faces. During the industrial revolution, digging and coring techniques were developed that brought sedimentary layers to the surface. It was not until the late eighteenth century that James Hutton revealed a complete understanding of the laws that govern the formation of sediment on the Earth [25]. Strata became a familiar concept when cut representations in cartography became incorporated into our standard educational material. Our work builds upon the general populace's widespread understanding of sedimentation.

### 3 VISUAL SEDIMENTATION

Visual Sedimentation applies the metaphor of the process of sedimentation to the visualization of data streams. The following section explains how sedimentation concepts can be mapped to their visualization counterparts, particularly: tokens (corresponding to sediments), layout, physical forces, and aggregated areas.

#### 3.1 Tokens

A *token* is a visual mark representing a data item arriving from the stream. For the purposes of this paper, each data item appears at a particular time  $t$ , and carries attributes (*e. g.*, size, weight, price); one of these attributes is categorical and is considered as the token's category. The token may be as graphically rich as needed in order to visually express the data it represents. Each token undergoes four *states* during its lifetime (Figure 3):

- *entrance* into the scene from an *entrance point*;

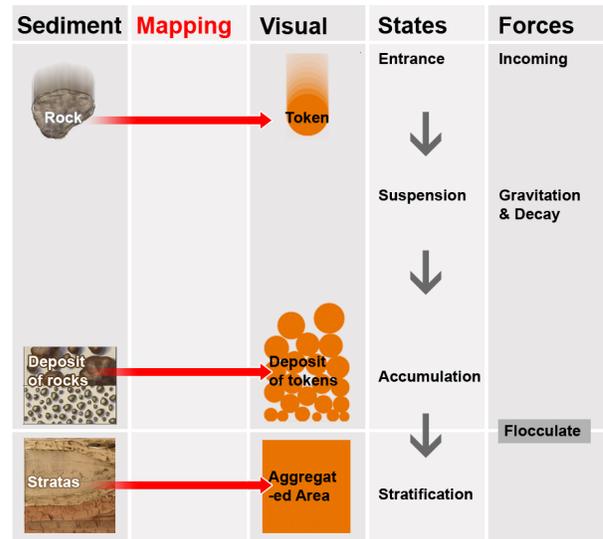


Fig. 3. Summary of the different states of tokens in the Visual Sedimentation metaphor.

- *suspension* while falling to the ground;
- *accumulation* on the ground or on top of previous tokens;
- *decay* and finally merging or *flocculating* into an aggregated areas.

The token's visual appearance can change during its lifetime to reflect its current *state*, but it should remain recognizable and identifiable through some consistent visual encoding to clearly keep its identity or *category* until it flocculates.

#### 3.2 Layout

The *layout* roughly corresponds to a two-dimensional geological cross-section of strata layers in which the tokens are shown in their multiple forms. The layout is comprised of four components that satisfy these constraints:

- *Walls* are separators used to control tokens' trajectories and to separate tokens according to possible categories while maintaining their arrival order visually.
- The *ground* is the baseline or barrier on which tokens accumulate and generate deposits over time.
- *Aggregated areas* lie under the ground and reflect the history of tokens previously accumulated underground.
- *Containers* are compound physical elements made of walls and ground. They are semi-closed shapes that contain categorized tokens.

#### 3.3 Forces

*Physical forces* specify the dynamics of tokens, for example their trajectories and reactions to collisions with other tokens or obstacles. They indirectly impact the tokens' states as their position changes and form decays. Properties of physical forces are specified as follows:

- *Gravity* moves tokens from their entrance point to the ground; it is probably the most intuitive and predictive type of force. Furthermore, token dynamics under gravity forces can be influenced by friction, such as fluid viscosity or dry friction.
- *Decay* simulates the weathering of a token. It is effectively an aging function that begins to work once tokens start to accumulate. The application of this force can produce different effects on tokens, including shrinking, flattening, or darkening.
- *Flocculation* is the transformation of individual tokens into a continuous aggregated area. This transformation can be rendered with various visual effects.

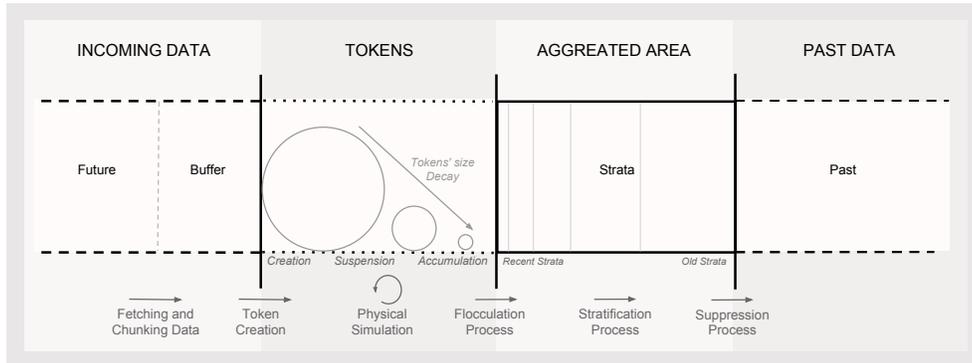


Fig. 4. Summary of the design space. Time goes from right to left: 1) Data streams chunked and stored in a buffer for further reading, 2) Tokens appear from an entrance point, 3) Physical forces transport and shrink tokens until they flocculate, 4) Strata are tokens aggregated into layers.

### 3.4 Aggregated Areas

The final state undergone by tokens is *aggregation* with previous tokens into *strata* (Figure 3, bottom). *Stacked layers* represent a period of time, with a height proportional to the number of flocculated tokens. Layers are stacked but visually separated by a line or a different visual encoding. *Deformation* is caused by physical forces that change the shape and thickness of the layer, and its visual representation (e.g., color or texture).

## 4 VISUAL SEDIMENTATION TOOLKIT

To facilitate the exploration of the Visual Sedimentation design space, we implemented an open-source toolkit [23], written in JavaScript to run in modern web browsers. It provides templates for using the common charts described in the next section. For example, the bar chart on the left in Figure 1 can be programmed with the parameters details shown in Listing 1. The toolkit reads parameters following the design space described in section 3 to configure the visual sedimentation engine. The capabilities of the toolkit can be extended through a programming API to control the data sources, visual encodings, dynamic behavior, layout, and interactions.

The Visual Sedimentation toolkit relies on two existing libraries: the physics engine Box2DWeb [11] to manage force-based animation and collision, and the visualization toolkit D3 [5] for implementing the visual encodings of the aggregated areas and their deformations. Token management is handled by the physics engine in a dynamic fashion, while the aggregated areas are handled using D3 according to the strata model and created once for all when the visualization starts.

The Visual Sedimentation toolkit relies on tokens being discrete, however data streams can carry continuous data or discrete high volume-rate data produced at high-speed, both of which can be unsuitable for most types of visualization. To use visual sedimentation with such data streams, the programmer will need to first split the continuous or fast stream into discrete chunks that the user will be able to see and understand. Furthermore, when the data flow rate is irregular or bursty, it may need some regulation; the toolkit provides some pre-processing facilities to smooth the stream throughput.

The toolkit's parameters are summarized in Table 1. They can be set individually to explore numerous design variations. To help designers get started, we predefined a series of templates that can be loaded by setting the `chart` type to `type: 'StackedAreaChart'` for bar charts. Other charts (pie and bubble charts) can be loaded in the same way and then be customized to meet the designers' needs.

However, not all parameter values will produce visualizations compliant with the sedimentation metaphor. Parameter setting should be done carefully since parameters may depend on or influence each other. For example, changing the incoming point position will impact the accumulation location, forcing tokens to land on the wrong part of a chart. To ensure that the resulting visualization will comply with the metaphor, designers should observe the following high level guidelines that are not enforced by the toolkit:

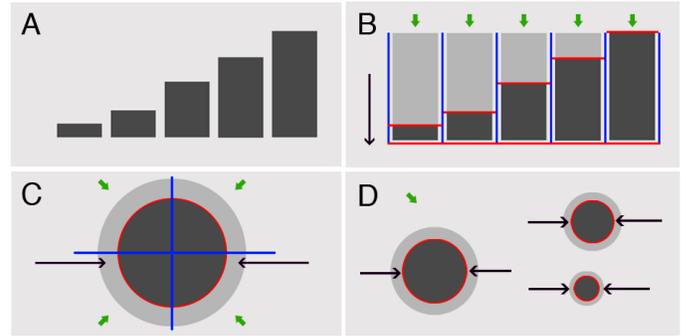


Fig. 5. Template schemas: Blue lines represent the *walls*; red lines the *ground*; green arrows the *entrance point*; black arrow the *gravity* direction; light grey shapes the *deposit area*; and dark grey shapes the *aggregated area*. (A) A common bar chart and (B) its template; (C) the pie chart template; (D) the bubble chart template.

Table 1. Parameters of the Visual Sedimentation Toolkit

(t) TOKEN	(a) AGGREGATED AREA	(f) FORCE	(l) LAYOUT
1 - Visual encoding	1 - Geometric primitive	1 - Gravity	1 - Wall
2 - Incoming point	2 - Stacked geometric primitive	2 - Decay	2 - Ground
3 - Trajectory	3 - Trail	3 - Flocculation	3 - Container
4 - Target	4 - Pixel		
5 - Impulse	5 - Isocontours		
6 - Physical parameters	6 - None		

1. The accumulation area and the aggregated area should look continuous at their interface;
2. The trajectory of tokens in the accumulation area should be consistent with the orientation of the aggregated areas (the perceived "time direction"), at least locally around their interface;
3. The encoding of strata sizes in the aggregated area should decrease monotonically starting at the interface following the "time direction";
4. To support the metaphor, the flocculated area and the aggregated area should be visually consistent around the interface.

```

1 mySettings = {
2   width: 300, height: 300,
3   chart: {
4     type: 'StackedAreaChart', ...
5   }
6   data: {
7     model: [
8       {label: "Column A"}, ...
9     ],
10    strata: [

```

```

11     [ {initValue: 100, label: "Bar A"} ], ...
12   ],
13   stream: {
14     provider: 'generator', refresh: 10000/4
15   },
16 }, // data:
17 sedimentation: {
18   token: {
19     size: {original: 6, minimum: 2}
20   },
21   aggregation: {type:"stacked"},
22   suspension: { decay: {power: 1.01} },
23   flocculation:{...}
24 }, // sedimentation:
25 };
26 var barChart = $("#myDivChartContainer")
27   .vs(mySettings)
28   .data('inputstream');

```

Listing 1. Bar Chart implementation based on a template

## 5 CASE STUDIES

We implemented a series of examples and cases studies using the Visual Sedimentation toolkit to demonstrate which parameters combinations can be effectively used in real settings. The Visual Sedimentation toolkit not only facilitates the implementation of the sedimentation metaphor; it allows its exploration in a guided way by parametric exploration of the design space.

### 5.1 Common Area Charts with Visual Sedimentation

The toolkit supports the application of the visual sedimentation metaphor to common area charts such as bar charts, pie charts and bubble charts. We explain how these charts are specified and how to control their visual encoding.

#### 5.1.1 Bar Chart

A simple way to use a bar chart with Visual Sedimentation (see Figure 1, left), is to consider each vertical bar as an aggregated area, with a height proportional to the number of tokens stratified. Walls are added to separate the chart into bins, each containing a different category of tokens (Figure 5 B). The top of the bar becomes the sedimentation ground where incoming tokens will bounce and deposit before they flocculate. The upper area of the bin, above the bar, becomes a container which is to be filled up by incoming tokens. Tokens fall from an entrance point that can be right above each bin, moved by gravity. To accommodate entrance points and tokens in suspension, the maximum bar height should remain below the top of the bin containers.

#### 5.1.2 Pie Chart

Similarly, a pie chart with Visual Sedimentation (see Figure 1 center) considers each sector as an aggregated area, where the angle is proportional to the number of stratified tokens. Just as with bar charts, walls are added between sectors. The ground is represented as an arc in each sector that forms a container delimited by two walls crossing at the center. The arc radius should be small enough to leave room for entrance points. Tokens enter in the containers from an entrance point that can be in front of each sector, and move towards the center of the chart, as though tokens were gravitationally drawn to by the pie center.

#### 5.1.3 Bubble chart

A bubble chart with Visual Sedimentation is slightly different than the other two types of area charts (see Figure 1, right). There is one center per token category, organized in a grid. Aggregated areas are discs with a radius proportional to the number of tokens stratified. Bubble charts have no walls; the ground is the external circle of each disc where tokens are deposited. Tokens can enter from any point; they are gravitationally drawn to the center of the category to which they belong.

Note that for these 3 charts, the perceived size of the areas is slightly perturbed by the tokens that float outside and their non-linear aggregation over time. Value retrieval tasks cannot be performed accurately but comparisons between area sizes remain valid. If accurate value retrieval is needed, a corrected aggregated area can be drawn translucently over the deposit and aggregated areas.

### 5.2 Tweet Monitoring and Social TV

The original inspiration for the Visual Sedimentation metaphor is Bubble-t [39], a bar chart where tokens represent tweets (Figure 6A), and token categories are the candidates for the 2012 French presidential election mentioned in the tweets. While it lacks aggregated areas, it does have other components of Visual Sedimentation: each tweet enters from a single entrance point, where they appear as text for two seconds; each tweet then morphs into a token, propelled by an initial force into category bins. Each token displays the icon from the Twitter user who sent it. Tokens do not decay as they would become too small to recognize the user icons. Flocculation consists in removing the last 10 tokens every minute.

Bubble-t was designed to monitor the Twitter users' perspective on the 2012 French presidential elections; it was selected among the finalists of the DataViz Google Elections 2012 competition. We received very positive feedback: 1800 tweets referenced its URL, and its average visit duration is 4:21, far longer than similar pages. We thus decided to further explore why it was so engaging.

We extended Bubble-t into Bubble-TV for Social TV tweets [24], by adding aggregated areas as bar charts and the count of aggregated tweets underneath. Extra styling is added to match the graphical style of the TV show that made use of it. The positive feedback we received again convinced us that the underlying physical metaphor was engaging, versatile, and could be used in multiple cases with varying styles but a consistent behavior.

### 5.3 SediClock

SediClock is a hybrid of an hourglass and a water clock (Figure 6C). Like an hourglass, SediClock uses sand particles as tokens to depict seconds. Like a water clock, it simulates a liquid filling-up a translucent recipient. SediClock encodes minutes and longer time units as strata. Tokens representing seconds flocculate each minute; therefore the minute container is filled with the latest count of seconds from the starting time. In addition to second tokens, SediClock also shows time in 200ms units using smaller tokens. Interestingly, these extra tokens have no consequence on the layout: they fill up the empty space left between the stacked second tokens when they fall in the container.

### 5.4 StackOverflow Questions Visualization

StackOverflow<sup>1</sup> is a popular website where users ask technical questions and get answers from the Web community; the questions are freely tagged by users for categorization and search [35]. Questions and answers flow in rapidly: as of the submission of this paper of this article, the ten most popular tags are associated with 428,345 to 160,826 questions and the site receives 20 to 500 new questions every day. Applying Visual Sedimentation to this stream, questions map to tokens and the 10 most popular tags. Our visualization uses the classical bar chart layout. To visually match StackOverflow's logo (Figure 6D), we use flat rectangles instead of circles as token shapes. To force the falling rectangular tokens to remain approximately flat when the physics engine could make them rotate, we limited the height of the deposit area above the bars and tuned the force model. To match the logo style we also set the decay function to not change the shape of tokens but instead darken their color over time. Consequently, there is no flocculation and tokens are removed after their decay time.

To control the data flow rate, a buffer containing the last 100 questions is loaded when the application is launched. This buffer is asynchronously filled, while each second, a question is taken out of the buffer and enters the visualization.

<sup>1</sup><http://stackoverflow.com/>

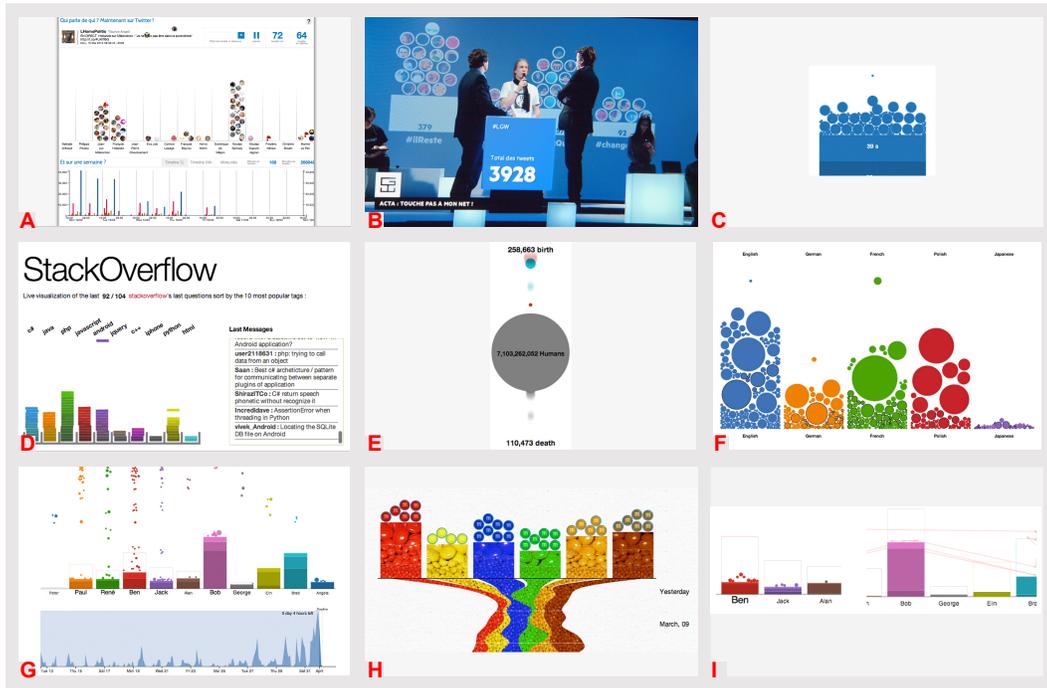


Fig. 6. Summary of the case studies: (A) Bubble-T seminal applications.; (B) Bubble-TV.; (C) SediClock: a hybrid between an hourglass and a water clock.; (D) Most popular questions asked on StackOverflow, the categories being the 10 most popular tags.; (E) Birth, death and population screen capture at 5h15 pm.; (F) WikiFlow.; (G) Edits of text and code before the deadline of VisWeek 2012 using SediVN.; (H) SediMMS.; (I) Label size reflecting the recent activity (bottom left) and lines between tokens showing changes on common documents in SediVN.

## 5.5 Real-Time World Population

Worldometers<sup>2</sup> is a website that provides real-time statistics on global issues such as demography, ecology, health, and economics, as counters updated in real time. Their real-time counters are streams of numbers using a specified unit, updated every millisecond; unfortunately the scale and context of their numerical data is difficult to understand without a proper visualization.

We used Visual Sedimentation to visualize the real-time world population, using the aggregated area to represent the living population. Unlike the previous examples, birth tokens enter and quickly flocculate to the circle representing the world population, while grey death tokens leave the disc, fall down, and vanish (Figure 6E).

Due to the high frequency of updates, we do not use any accumulation; the speed of tokens during suspension conveys the scale of the counters clearly showing that the birth rate is higher than the death rate.

## 5.6 WikiFlow

Wikipedia<sup>3</sup>, the free encyclopedia online, manages each of its languages as a different Web site, resulting in differences in the number of articles, coverage, style, and frequency of edits. We apply Visual Sedimentation to create the WikiFlow visualization, showing real-time edits made in five languages: English, German, French, Polish, and Japanese (Figure 6F). Languages correspond to categories, page edits correspond to tokens with a size proportional to the edit's length.

Interactions with tokens allow users to get details on demand: hovering on a token shows the page name, author, and summary of the edit. After edit details are shown, a black outline is drawn around the token. Clicking on the token opens the edited page. For managing the updates, new data is buffered and appear every 30 seconds to give users enough time to detect changes.

## 5.7 SVN Commits Replay and Visualization

We extracted edit logs from our lab's SVN repository, reflecting the writing and programming of 11 researchers; relating to activity both on articles and programming during the 20 days preceding the VisWeek 2012 deadline. To visualize this data stream, we designed *SediVN*, a Visual Sedimentation bar chart where researchers correspond to categories, edits correspond to tokens with a size proportional to the edit's length, and the sequence of edits is replayed in the original order at higher speed (Figure 6G). Tokens had a 24-hour lifetime before flocculating into three strata that encoded the token count during the past 24 hours, the previous week and the overall period of time since the beginning of the recording, respectively. Since data was known in advance, the maximum bar heights (representing the total quantity of edited text for one researcher) were shown as bar contours that fill up when tokens flocculated. To provide more control and an overview, we added a static visualization under the animated bar chart showing the distribution of edits over time during the total 20-day period; it clearly shows the increase of activity in the run-up to deadline. The progression of the SediVN animation is depicted as a temporal slider, with a darker translucent background from the starting time (left) to the current time. The current time position can be interactively moved to replay the animation, jump to interesting times, or control the animation speed.

As tokens may appear simultaneously, we reinforced token arrival notifications both visually and with sound. Visually, we increased the label size of the bin that will fall into. The size then decreased slowly until another token arrives. Thus, label sizes reflect the recent activity of each bin (Figure 6I, left). Each entering token makes a short audible beep, with a different pitch associated with each bin category. This effect is designed to evoke the vocal pitch of different researchers working in the same room; it also expresses accurately how frequently edits can occur. All these feedback modes are reminiscent of a *ripple* effect.

We received feedback from many researchers within and outside our team; they commented that they could feel the heat of the deadline by watching SediVN. However, they sought more information relating

<sup>2</sup><http://www.worldometers.info/>

<sup>3</sup><http://www.wikipedia.com/>

to collaborations. We made an adjustment to SediVN by adding lines between tokens that are related to the same edited files. Since the lines also added visual clutter, they only appeared when hovering the mouse over a particular researcher’s bin (Figure 6I).

## 5.8 Twitter Popularity of M&M’s Colors

During two months, we recorded 19,852 tweets containing both the word “M&M’s” and a color name among red, yellow, blue, green, orange and brown. The SediMMS application visualized them in a Visual Sedimentation bar chart with colors as categories (Figure 6H).

Each tweet is a token, represented as a real-looking M&M’s. They initially appear as their tweet’s text in the upper part of the chart and then morph into the well-known M&M’s rounded shape. This visual effect enforces continuity between the textual and graphical representations. During initial design feedback, a user suggested that the visual transformation should give a momentum to the tokens. Therefore, we added an initial force with a random magnitude to simulate a propelling force. To show the strata, we used a background texture of M&M’s at multiple scales to simulate token accumulations. Lower-level strata showed denser M&M’s using a scaled-down texture to express compression over time.

One major design issue was how to display of all the recorded tweets. Standard stratification applied independently for each category made it difficult to compare the evolution of strata sizes. To preserve the strata alignment, each bar was rotated by 90 degrees at the end of each day and stacked, resulting in a streamgraph [8]. This created an organic shape that showed the varying trends of M&M’s colors over time. We advertised the visualization through Twitter, explained the streamgraph to users as the temporal **trail** left by the bar chart over time.

## 6 DECONSTRUCTING THE METAPHOR

The aforementioned case studies are based on pre-configured range parameters from the toolkit. In these standard cases, the aggregated area always defines the visualization template and tokens fall and flocculate within these areas.

Despite the abundant literature on designing user interfaces (*e. g.*, [34]), there are few accepted methodologies in information visualization to explore design spaces. In this work, we adhere to the design methodology described by Buxton [7]: starting from our initial experience on Bubble-t and Bubble-TV (subsection 5.2), we generalized the design space by creating the Visual Sedimentation metaphor. We then applied the design space to several examples that are instantiations of the metaphor (section 5). We now further extend the metaphor by deconstructing it. We will progressively modify toolkit parameters that imitate physical-constraints and are implicit in the sedimentation metaphor; physical constraints are not necessarily required. This deconstruction is compelling for two reasons: first, from a design standpoint, it extends the possibilities of Visual Sedimentation to richer and more creative solutions; second, from a technical standpoint, it adapts Visual Sedimentation to simulate conditions that go beyond a literal physical metaphor. Moreover the design space exploration is also useful to provide some guidelines for designers and developers.

### 6.1 Methodology

To explore the design space, we relied on the parametric space provided by the toolkit (Table 1). One simple way to explore it would be to vary all the parameters systematically and observe the results. However, this is not possible due to the strong constraints linking the parameters. Instead, we created variations by focusing on three parameters: force, layout, and aggregated area. We created a large number of designs with variations of these parameters, selected the “best” samples, and iteratively improved them. The criteria we used to select the best samples were: consistency with the metaphor, readability of the mapping function, and interesting or surprising results.

### 6.2 Variations on Force-Based Behavior

The pie chart, bar chart, and bubble chart standard templates (Figure 5) are provided to the designer as starting points to produce variations

(Table 1, a[1,2], l[1,2,3]; f[1,2,3]). The three first variations (Figure 7A, 6E and 6I) are based on changing some parameters of these templates. The *Bar Chart without boundaries* (Figure 7A) uses the general container but without the walls that indicate a column structure (Table 1, f[2,3]). The same process was applied to (Figure 7E). In this case, the three categories show the same dataset. Because the categories are color coded, the chart remains meaningful, although less readable than the original. Because the number of categories may be unknown at start-up time, we found this chart useful in allowing new category creation or removal. Despite the missing wall constraints, and even if tokens can organize freely around their landing position, the overall result remains close to a regular chart with the exception of a few tokens. In these two charts, we have also removed aggregated areas (Table 1, a[6]); the resulting token accumulation encodes information beyond the chronological order of the tokens, showing trends in the data stream. For example, a large number of green tokens are overflowing near the center of the pie chart, indicating that there was an abundance of green tokens early in the data stream.

*Heap Charts* (Figure 7I) are constructed like a bubble chart but with no container and a distributed entrance point. Because the tokens have full freedom, they even out in a circle; they produce a temporal fisheye effect, as tokens shrink and decay over time. We discuss the mechanisms to preserve the metaphor in subsection 6.3.

Removing the ground and gravity in a bar chart results in a *Silo Chart* (Figure 7B, Table 1, f[1], l[2]). Having no ground removes the physical force of gravity (Table 1, f[1], l[3]), allowing the infinite spread of tokens. The chart preserves the separation between categories, and still provides a potentially infinite space for the tokens: only walls direct the tokens’ placement inside infinitely long strips. No gravity means that as the tokens enter through their entrance point, they push away older tokens, resulting in a growing circular shape. As the tokens decay over time, the outer ring of the chart ends-up containing smaller tokens than the center, making the more recent tokens more prominent. Multiple contiguous silo charts can be compared visually as the tokens fill the strips additively.

The metaphor can also be deconstructed by *perforating* layouts with holes that enable tokens to escape through the holes and prevent unexpected overflows. A *Drop Chart* (Figure 7C) is a bar chart with a hole on the ground that is filled with tokens from the top and emptied at the bottom. A second container underneath is progressively filled with the falling tokens. This chart allows token paths to be specified, showing the time windows before the tokens flocculate.

Varying the container shapes produces a wide variety of layouts, sometimes requiring the force models to be adapted. Another variation is to assemble various shapes as obstacles to slow down and control the path followed by the tokens (Table 1, l[1]). In the *Typo Chart* (Figure 7, L), tokens follow predefined paths that are typographic characters or signs. In this example, tokens draw the letters “V”, “I”, and “Z” as they are constrained by invisible walls. These letters are drawn as tunnels in front of the entrance points. They are filled with tokens from the top and emptied from the bottom. Despite the high level of expressivity in this example, it is difficult to compare the quantities and arrival time of tokens when the containers have different shapes.

### 6.3 Variations on Aggregated-Area Layouts

We have found three possible strategies for creating the area chart: 1) generate an area chart from scratch with the support of the D3 toolkit, 2) generate a layout from the flocculated tokens, and 3) generate a layout from the tokens’ trails.

The first strategy is provided by the default templates for the standard aggregated areas (Table 1, a[1,2]). It can either use extensions of templates to produce more complex area charts (as in SediMMS), or use new variations as in Figure 7G and 6H. To be consistent with the metaphor, the generated area chart should follow the requirements described in section 4, but the toolkit cannot enforce these requirements: automatically generating area charts consistent with the tokens’ behavior and with the requirements remains an open research question.

For the second strategy, the toolkit makes it possible to use the tokens’ dynamic behavior against physical barriers to generate shapes

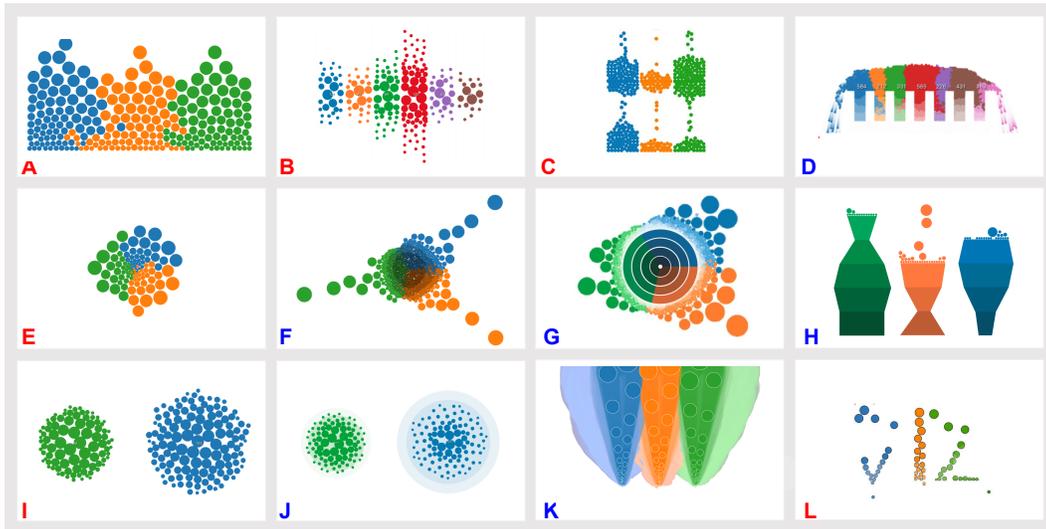


Fig. 7. Summary of the case studies: (A) “Bar Chart” without walls.; (B) “Bar Chart” without ground and gravity; (C) “Drop Chart”, constraining tokens to flow inside identical complex shapes.; (D) A bar chart *overflow* due to the limited container capacity. (E) “Pie Chart without walls”; (F) The previous chart with a pixel aggregated area at 2 minute mark; (G) The previous chart with circular strata; (H) “Hoodoo Chart”.; (I) “Heap Chart” where tokens arrive from the center.; (J) “Heap Chart” where the deposit and aggregated areas are overlaid.; (K) “Trail Chart” where the tokens leave trails; (L) “Typo Chart” constraining tokens to the shape of the letters “VIZ”.; Red letters are pre flocculation variations and blue letters are post flocculation variations.

that can be used in the area chart. For example, the flocculation process can be further simulated to generate future generations of flocculated tokens that will constitute the aggregated area (Table 1, a[4]); the toolkit allows exploration of several parametric configurations for that strategy. This is shown in Figure 7F where a pie chart is generated by the composition of flocculated tokens which, instead of disappearing, fill pixel by pixel the aggregated area at the center of the chart. Fading the color of post-flocculated tokens visually creates different strata levels (Table 1, a[4,2]). Interestingly, this mechanism results in a rendition similar to Figure 7G which is based on the distribution of flocculated tokens, but requires the full execution of the simulation and may not result in a chart consistent with the flocculation stage.

The third strategy is to not only provide a visual persistence once the token is settled, but also while it is in motion starting from its incoming point. *Trail Chart* (Figure 7B; Table 1, a[3]) shows the trails of every single token trajectory, with the thickness decreasing as the token decays. This results in overlapping strips where color encodes the token’s dimension and fades out after a certain amount of time. Overlapping trails may thus indicate trends thanks to transparency. Large trails indicate that a data burst happened, which is not visible anymore once the tokens become deposited in the chart. Trails have the same properties as strata as they show previous tokens’ activity. They can even systematically be used in any type of chart as a way to add motion blur and facilitate token tracking, as well as to capture screenshots of charts (Figure 7, I). Once they disappear, tokens may also leave a trail, which also fades away over time resulting in distinct horizontal layers similar to strata (Figure 7, E, I).

In addition to aggregated areas, dynamically generating strata according to token positions can be done by adding *isocontours* (Table 1, a[5]): area boundaries containing the tokens just before they flocculate. These boundaries provide a persistent visual footprint of maximal radii during a sliding time window (*e. g.*, last minute, hour, and day). Isocontours explicitly encode the maximum value of their perimeter of deposits, they are useful for visual comparison tasks. In Figure 7, J, we display them low opacity so as to keep them both visible because they share the same visual space as tokens. Isocontours can only be used when the aggregated areas are generated in a way consistent and continuous with time, unlike *e. g.*, Figure 7, F. Just like obtaining consistent isocontours, obtaining aggregated areas suitable to perform specific reading tasks is not straightforward since the aggregated area depicts the number of flocculated tokens modulated by

their age. To address that issue, Figure 7, H visualizes the total number of tokens as bar heights and number per time window as bar widths, allowing comparisons between categories.

## 7 TECHNICAL AND DESIGN IMPLICATIONS

We now relate and comment on *technical* and *design* observations from our experience in the implementation of the previous case studies and extended designs. These issues arise because, while the physical world is mostly self-regulated (except occasional catastrophes), the digital world is not. Therefore, implementations should take measures to adjust to scale variations in an adequate way. These scale variations can relate to the tokens rate, the token sizes, the deposit time, or flocculation time.

To mitigate these issues, Visual Sedimentation uses a physics engine that behaves in understandable ways when pushed over the tokens capacity. Very much like a dam overflow, tokens overflow their containers and walls but continue to behave consistently so the metaphor is still valid to some extent. However, visualizations can become cluttered and unpredictable, sometimes in creative ways but not under the designer’s control. We discuss the current performances of the toolkit and its mechanisms to improve control.

### 7.1 Performance

As the primary purpose of the toolkit is to enable the exploration of the metaphor’s design space, its current implementation has not yet been optimized for performance. Still, we tested it and identified strategies to improve its performance with different parameters using the following configuration: MacBookPro, processor 2.4Ghz Corei7, 8GB Ram, Mac Os X 10.7.5, Chrome Version 27.0.1453.93. We ran the test on the default chart of the library (a  $300 \times 300$  pixel canvas with a 3-bin bar chart). We recorded the number of frames per second (fps) as a measure of visual rendering quality, and also the number of collisions per second (cps) as we hypothesized that this could be a bottleneck for fps drops. We implemented various scenarios (see results Figure 8) with multiple strategies to keep the fps consistent. All scenarios are based on the same range of the number of incoming tokens per second (tps), from 3 to 105 (tps).

Two factors seem to account for the drop of performance: 1) increases in cps, and 2) triggers of the token aggregation function (which requires JavaScript code processing by the browser). Beyond 541 cps



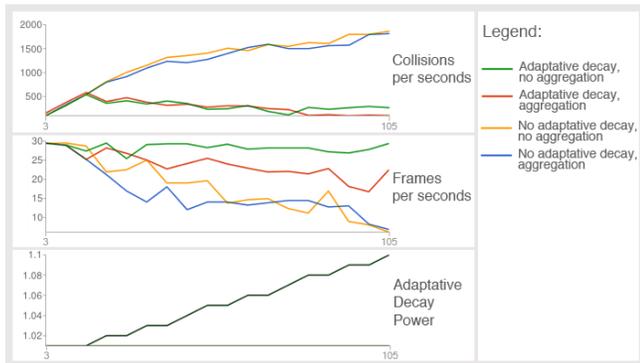


Fig. 8. Summary results of the Visual Sedimentation toolkit. The horizontal axis represents the number of tokens per second, from 3 to 105.

(measured with 15 *tps*), the refresh rate drops under 25 *fps*, which is below the standard quality for animation. To limit the number of collisions per second and maintain a decent frame rate some strategies can be adopted. We call these strategies *adaptive* as they are activated according to dynamic parameters such as *tps* or *cps*.

The first strategy is to change the decay rate according to the number of *tps* to limit the number of tokens in the deposit area and, indirectly, the number of collisions. The second strategy is to delay the aggregation shape update to a limited number of tokens. In this type of configuration the simulation can go beyond 25 *fps* for more than 105 *tps*. The pitfall of these two strategies is that they introduce a variation of the encoding according to the data flow rate. To avoid confusion for these adaptive solutions we recommend that designers indicate any change of encoding to the user.

## 7.2 Bursty Data Streams

The main mechanism for managing bursty data streams is buffering tokens to control their entrance rate. Buffering implies adding delay or forcing a constant interval between token, creating lag between what the data stream sends and what the visualization displays.

Visualizing tokens immediately after a new data arrives from the stream is the simplest and most sensible strategy, when possible. Variations in token rate become visible, for example, in SediVN, an increasing edit rate as the deadline approaches. However, SediVN tunes the token sizes and gravity forces optimally for the stream because it replays recorded data and can prepare in advance.

For online cases where the data stream rate is unknown, we adopted an hourglass buffering and reading strategy: each token is displayed for a short time before it flocculates. This is the case for Bubble-t, Bubble-TV, StackOverflowChart, WikiFlow and SediMMS. This short time allows users to read the token data and it also ensures the deposit area does not overflow (Figure 7, I). Other strategies include enlarging the deposit area, increasing the suspension time, and decreasing the flocculation time. These parameters affect the look and feel of the visualization and are therefore design trade-offs. Alternatively, modifying the buffering strategy has no impact on the look and feel but on the faithfulness of the interface: if it pretends to be near real time but it is severely delayed, the interface is not faithful for the user. A simple way to fix the problem consists of showing a count of buffered items or a visual feedback representing them, such as drops ready to fall. Interactive control can be added to flush all the buffered tokens.

## 7.3 Tokens Deposit and Strata Visual Comparison

Comparing stream throughput between categories requires choosing deposit and aggregated areas that are easy and accurate to compare.

In Bubble-t, as containers share a *common ground* and no decay, the comparison is facilitated, in contrast to Bubble-TV, SediMMS and SediVN where strata elevate the ground level, making the comparison task difficult. To solve this problem, the deposit areas could share an aligned ground level from which the deposit would grow up and

the aggregation would grow down. This is the design choice made in SediMMS.

In addition to positions, visual variables encoding tokens such as size and shape impact the comparison, and even the overall readability since tokens filling speed is not uniform. For example, during one of the first iterations of StackOverflowChart, tokens were horizontal rectangles ( $4 \times 12$  pixels) in a container twice as wide. This configuration gave too much freedom for token accumulation in containers and resulted in unexpected stacking positions. By reducing the container's width, the tokens became ordered in the next generation.

## 7.4 Token Provenance and Destination

Tokens may arrive from several entrance points, forcing users to scan many deposit areas to detect new incoming tokens. An appropriate visual encoding facilitates the *provenance* tracking of tokens. Adding labels for each entrance point is the natural solution, but is redundant with existing labels of the charts. Alternatively, using an arbitrary entrance point with no label has led to unexpected interpretations. In SediVN, a user suggested making the tokens fall from the top of the web page, as it seems more natural than the tokens arriving from nowhere. Indeed, users tend to associate visual elements surrounding the entrance point with the origin of token data. If details about tokens are provided before token creation (*e. g.*, tweet text) then a visual effect expressing causality has to transform the details into tokens: in Bubble-t, tokens are propelled in the direction of their destination column, in SediMMS a smooth transition changes the tweet's text before it gets to its destination bin.

Deciding on the destination of tokens is not straightforward either. For example, in Bubble-t, the tokens are thrown similarly as a basketball shoot from a single entrance point to the final container. The angles and forces applied to the tokens were manually tuned in order to create aesthetically pleasing token trajectories before entering the container, as well as to prevent collisions with other tokens and to reach the correct container. Having an interactive tool to test the parameters would have helped tremendously.

## 8 RELATED WORK

Visual Sedimentation is a design metaphor applied to the visualization of streaming data. We structure the discussion of related work into three parts: metaphors, visualization of streaming data, and visual encoding of token-oriented data in art and design.

### 8.1 Metaphors in Visualization

Using analogy and metaphor is a common and effective human process, as described by Gentner [19]. Metaphor is used to support design in several areas including information visualization [42], computer supported collaborative work (*e. g.*, [29]), and more generally computer human-interaction (*e. g.*, [10]).

There are several well-known metaphors already used in visualization and HCI, such as Furnas's generalized Fisheye views [17], Perlin and Fox's zoomable user interfaces [33], and Bier et al.'s Tool-glasses and Magic Lenses [4]. Interestingly, Visual Sedimentation can be presented as an application of Furnas's generalized Fisheyes where tokens have a degree of interest function (DOI) related to time. These metaphors have sometimes been deconstructed and extended too: Magic Lenses by Eccentric Labels [14] and Color Lenses [13] where the focus area remains a lens but the visual effect extends to its surrounding or to the whole display.

### 8.2 Streaming Data Visualization

Abundant work exists on visualizing time-oriented data [1]. In this section we will focus on dynamic data, which are data updated over time, and in particular data streams: unbounded sequences of data elements continuously transmitted. The topic of data streams is popular in several computer science disciplines such as database systems [3], machine learning [18], and information visualization [12, 32, 41].

Due to its novelty in information visualization, the vocabulary related to data stream visualization is not standardized. Norton et al. [32] used the term "streaming graphics". Ben Fry [16] proposes "Organic

information design” for a certain type of dynamic data visualization. Finally the term “dynamic visualization” seems to have become accepted, as it is used by several authors [12, 41] who present dynamic visualization as visualization for dynamic data, including streaming.

Norton et al. introduce in their paper a new approach of visualization called streaming graphics [32] to display streaming data in real time. They provide several preliminary guidelines to design streaming visualizations, depending on whether the time-scale is continuous or discrete; they describe update strategies, and real-time versus replay. However, they discuss graphical constraints and solutions instead of metaphors or design rules.

Fry [16] highlights the fact that not only the value of data could change over time, but also its structure. For this reason, Fry insists on providing consistency through an adaptive visualization to help the user maintaining a consistent mental model. His thesis is much more general than Visual Sedimentation but does not introduce metaphors or systems as we do.

Cottam et al. [12] present a taxonomy for understanding how data changes influence the interpretability of visual representations. This taxonomy classifies dynamic visualization in a two dimensional matrix defined by spatial and retinal categories. According to the taxonomy, Visual Sedimentation can be described as a CREATE&DELETE × MUTABLE SCALE with an *immediate* identity property. CREATE&DELETE means that “elements may be created or deleted in response to incoming data.” MUTABLE SCALE means that “updates may change the representation of an element and the mapping function itself.” This taxonomy cell is described as “This [] category of visualization techniques can present the current data state with the highest degree of fidelity. However, it is the weakest for comparisons across time [...] Despite this restriction, visualizations in this category can communicate step-by-step changes in a complex space. Short term comparisons can be made when changes are made slowly and vary a small number of elements at once (this is further helped by following animation guidelines [22]).” The identity property relates to how elements (tokens for us) retain their identity from users’ point of view; it is *immediate* according to the taxonomy because the identity is maintained through “careful design choices.” The taxonomy describes accurately the characteristics of Visual Sedimentation.

A large number of techniques to visualize streaming data have already been published. Vande Moere’s work is the closest to ours: he uses the metaphor of *Flocking Boids* (flocking bird objects) [31] to capture emerging patterns in dynamic data. While his technique relies on a complex algorithm to find similar streaming items, Visual Sedimentation uses very simple properties of the items coupled with a physics simulation for generating a trajectory. The items managed by Vande Moere are curves while our items are tokens which can take any area shape. Collision between items is managed in both systems, but only Visual Sedimentation uses data aggregation over time.

Whisper [9] is a system visualizing in real-time the spatio-temporal traces of tweets diffusion according to topics. Whisper shares some elements with Visual Sedimentation, such as the “Topic discs”, which are equivalent to our tokens. It also uses a force model to animate the visualization. Whisper is specialized to trace the diffusion of topics, which is a particular task that is contrary to Visual Sedimentation, and is not for and particularly suited to using with spatial data.

Krstajic et al. [27] present a technique to produce incremental time-series visualization. They introduce two major concepts which highly inspired our work: 1) the need to emphasize incoming elements and provide a context on their past, and 2) the use of a decay function to express a distortion over time. In Visual Sedimentation, we apply the same concept of the decay function to reduce the tokens over time, but we use it differently to produce several possible types of visual encodings, including classical bar charts.

Finally the closest approach to our metaphor is the exploration by Viegas et al. [38] of geological metaphor. In their work, they compress and display data as strata over time. Still, as far as we know, the full exploration of the sedimentation metaphor from falling objects, as the result of gravitational forces, to their aggregation into compact layers over time is new. However, the idea of using forces to drive the accu-

mulation of items thus forming an area is not new in itself. The HCI community has explored this modality in relation with icons and tangibles [30, 36]. We can also identify a trend in several design [37] and Art [21, 28] works. We report on the works closest to Visual Sedimentation from both of these fields.

### 8.3 Art and Design

In art and design, the process of accumulating elements to express quantity is broadly spread. In the following section we collect some examples that explicitly use accumulation to represent an area.

“The dumpster” [28] is a good example of this aesthetic trend. This work visualizes a collection of 20,000 blog posts, extracting posts related to breakups of romantic relationships over the course of one year. “The dumpster’s” visual encoding uses “bubbles” (similar to our tokens) for presenting each romantic behavior collected in blog posts. These bubbles are organized according to physical forces. The “bubble visualization” is linked to a pixel based visualization which provides an aggregated view over the year but does not provide a transition between the aggregated area and tokens.

“We feel fine” [21] is a well-known website which defines itself as “An exploration of human emotion, in six movements”. Like “The dumpster”, this website presents some extraction of blog-posts containing emotional words collected over the world. To show these elements in a visualization, the author, Jonathan Harris, uses rounded shapes that are spatially organized around templates such as a bar chart. This organization allows a quantitative analysis and possible serendipitous findings made possible by pointing on shapes to see details about the blog they represent.

Jordan [26] explains in his talk: “I am trying to take the raw language of data to a more universal language that can be felt.”. In his artworks series “Running the Numbers”, he produces monochromes based on accumulation of elements to express statistics about these elements. For instance, to present the number of gun-related deaths in the US in 2004 (29,569), he painted 29,569 handguns on a canvas. Some of his canvases have a layout, some do not. This work shows how tokens could be highly adapted to communicate emotions.

“US debt” [37] is a famous infographic of the US debt, based on the accumulation of \$100 bills. This infographic video uses a storytelling process to provide a sense of scale between a unit token (100 dollars) commonly known and the amount of US Debt at that time, 122.1 trillion Dollars, not familiar to anyone. To fill the gap between the unit and the total sum, no aggregation process is used but a viewport zooms out and large objects are displayed to allow visual comparisons. However, the aggregation is only achieved by scaling down since the purpose of the infographic is to always relate to the size of \$100 bills to convey the scale of the debt.

All these examples use non-aggregated elements to provide visual information on data. Some of them group elements in a higher-density area to produce a nesting. Half of the examples that group the items in a density area use a diagram template.

## 9 CONCLUSION AND FUTURE WORK

We introduced a new design metaphor, directly inspired by the physical process of sedimentation, *Visual Sedimentation*. Using the metaphor, meaningful chunks of data are visualized as tokens, which enter the visualization at an entrance point. These tokens remain in suspension while they fall to the ground, moved by a virtual gravitational force; they accumulate on the ground, decay, and finally flocculate into aggregated areas or strata.

We demonstrated how this metaphor can be applied to a wide variety of data streams, including twitter topics, social media sites such as StackOverflow or Wikipedia, or for monitoring user activity in versioning system repositories.

We also showed how the metaphor reacts when adjusting design parameters that otherwise connect the visualization to the physical constraints of real-world sedimentation processes. This metaphor opens-up a design space with a lot of expressive potential, as demonstrated by our examples.

Furthermore, we created an open-source toolkit that enables designers to quickly implement novel visualizations using the Visual Sedimentation metaphor. This toolkit facilitates implementation, and allowing us to explore the design space opened by Visual Sedimentation. This design space is restricted to the part of the metaphor that the toolkit implements, but it is already very large and has allowed us to create a wide spectrum of creative and unexpected visualizations.

Visual Sedimentation opens-up new perspectives and expressivity when visualizing data streams. Future work includes enriching the toolkit by implementing more simulations of the physical process of sedimentation, which will further enrich the expressive power of the toolkit. Providing a graphical builder would tremendously help implementers to design new configurations and quickly find working parameters and strategies for their data streams.

## ACKNOWLEDGMENTS

We thank the CINEGIFT DGCIS project and ANR Periplus for their funding; and France Televisions Innovation, Allo Cine and Google who hosted some of our case studies. We also thank the IRI team: Nicolas Sauret, Raphael Velt, Yves Marie Haussonne, Vincent Puig; as well as Lindsay MacDonald, Jagoda Walny, Laurence Fyfe and Lora Oehlberg for their help in improving the article. Finally, thanks to the reviewers and paper chairs for their valuable comments which greatly improved the paper's quality.

## REFERENCES

- [1] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visualizing time-oriented data—a systematic view. *Comput. Graph.*, 31(3):401–409, June 2007. 8
- [2] ASTR 1210 (O’Connell) Study Guide. Last access: 10 March 2013. <http://www.astro.virginia.edu/class/oconnell/astr121/guide12-s02.html>. 2
- [3] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *PODS ’02*, pages 1–16. ACM, 2002. 8
- [4] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: the see-through interface. In *SIGGRAPH ’93*, pages 73–80. ACM, 1993. 8
- [5] M. Bostock, V. Ogievetsky, and J. Heer. D<sup>3</sup> data-driven documents. *TVCG*, 17(12):2301–2309, 2011. 3
- [6] Britannica. Sedimentation (Geology). Britannica. Last access: March 2013. <http://global.britannica.com/EBchecked/topic/532291/sedimentation>. 1
- [7] B. Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Interactive Technologies. Elsevier Science, 2010. 6
- [8] L. Byron and M. Wattenberg. Stacked graphs — geometry & aesthetics. *TVCG*, 14(6):1245–1252, Nov. 2008. 6
- [9] N. Cao, Y.-R. Lin, X. Sun, D. Lazer, S. Liu, and H. Qu. Whisper: Tracing the spatiotemporal process of information diffusion in real time. *TVCG*, 18(12):2649–2658, dec. 2012. 9
- [10] J. Carroll, W. Kellogg, and R. Mack. *Interface metaphors and user interface design*. 1987. 8
- [11] E. Catto. Box2d, 2010. <http://box2d.org/>. 3
- [12] J. A. Cottam, A. Lumsdaine, and C. Weaver. Watch this: A taxonomy for dynamic data visualization. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 193–202, oct. 2012. 8, 9
- [13] N. Elmquist, P. Dragicevic, and J.-D. Fekete. Color Lens: Adaptive Color Scale Optimization for Visual Exploration. *TVCG*, pages 795–807, Jan. 2011. 8
- [14] J.-D. Fekete and C. Plaisant. Excentric labeling: dynamic neighborhood labeling for data visualization. In *CHI ’99*, pages 512–519. ACM, 1999. 8
- [15] H. N. Fisk. Geological Investigation of the Alluvial Valley of the Lower Mississippi River. Last access: 10 March 2013. <http://pruned.blogspot.fr/2005/06/geological-investigation-of-alluvial.html>. 2
- [16] B. J. Fry. Organic information Design. Master’s thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2000. 8, 9
- [17] G. W. Furnas. Generalized fisheye views. In *CHI ’86*, pages 16–23. ACM, 1986. 8
- [18] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy. Mining data streams: a review. *SIGMOD Rec.*, 34(2):18–26, June 2005. 8
- [19] D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, 1983. 8
- [20] Geotechnical Services Information. Last access: 10 March 2013. [http://www.globalspec.com/learnmore/engineering\\_services/geotechnical\\_services](http://www.globalspec.com/learnmore/engineering_services/geotechnical_services). 2
- [21] J. Harris. We feel fine. Last access: March 2012. <http://www.wefeelfine.org/>. 9
- [22] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *TVCG*, 13:1240–1247, November 2007. 9
- [23] S. Huron and R. Vuillemot. Visual Sedimentation. Last access: 10 March 2013. <http://www.visualsedimentation.org/>. 3
- [24] S. Huron, R. Vuillemot, and J. D. Fekete. Bubble-TV: Live Visual Feedback for Social TV Broadcast. In *CHI 2013 Workshop: Exploring and enhancing the user experience for television*, Paris, France, Apr. 2013. 4 pages. 4
- [25] J. Hutton. Theory of the earth. *Transactions of the Royal Society of Edinburgh*, I(II):209–304, 1788. 2
- [26] C. Jordan. Chris Jordan: Turning powerful stats into art. Last access: March 2012. <http://www.youtube.com/watch?v=f091Q8Q1iKE>. 9
- [27] M. Krstajic, E. Bertini, and D. Keim. Cloudlines: Compact display of event episodes in multiple time-series. *TVCG*, 17(12):2432–2439, Dec. 2011. 9
- [28] G. Levin. The dumpster. Last access: March 2013. <http://www.flong.com/projects/dumpster/>. 9
- [29] P. Mambrey. Chapter 9 metaphors as requirement analysis tools: The market metaphor in cscw system design. In M. T. Dan Shapiro and R. Traunmuller, editors, *The Design of Computer Supported Cooperative Work and Groupware Systems*, volume 12 of *Human Factors in Information Technology*, pages 135–150. North-Holland, 1996. 8
- [30] R. Mander, G. Salomon, and Y. Y. Wong. A “pile” metaphor for supporting casual organization of information. In *CHI ’92*, pages 627–634. ACM, 1992. 9
- [31] A. V. Moere. Time-varying data visualization using information flocking boids. In *IEEE Symposium on Information Visualization (InfoVis 2004)*, pages 97–104, 2004. 9
- [32] A. A. Norton, M. A. Rubin, and L. Wilkinson. Streaming graphics. *Statistical Computing and Graphics Newsletter*, 12(1):11–14, 2001. 8, 9
- [33] K. Perlin and D. Fox. Pad: an alternative approach to the computer interface. In *SIGGRAPH ’93*, pages 57–64. ACM, 1993. 8
- [34] B. Shneiderman and C. Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, 2010. 6
- [35] Stackoverflow Tags. Last access: 10 March 2013. <http://stackoverflow.com/tags>. 4
- [36] B. Ullmer, H. Ishii, and R. J. K. Jacob. Token+constraint systems for tangible interaction with digital information. *ACM Trans. Comput.-Hum. Interact.*, 12(1):81–118, Mar. 2005. 9
- [37] A visualization of United States debt. Last access: March 2013. <http://demonocracy.info/>. 9
- [38] F. Viégas, E. Perry, J. Donath, and E. Howe. Artifacts of the presence era: visualizing presence for posterity. In *ACM SIGGRAPH 2004 Sketches*, SIGGRAPH ’04, pages 19–. ACM, 2004. 9
- [39] R. Vuillemot, S. Huron, R. Velt, and Y.-M. Haussonne. Bubble T : Et si twitter votait ? <http://dev.fabelier.org/bubble-t/>. 4
- [40] Wikipedia. Sedimentation. Wikipedia. Last access: March 2013. <http://en.wikipedia.org/wiki/Sedimentation>. 1
- [41] P. C. Wong, H. Foote, D. Adams, W. Cowley, and J. Thomas. Dynamic visualization of transient data streams. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 97–104, oct. 2003. 8, 9
- [42] C. Ziemkiewicz and R. Kosara. The shaping of information by visual metaphors. *TVCG*, 14(6):1269–1276, 2008. 8