Real-time Visualization of Streaming Text with Force-Based Dynamic System

Jamal Alsakran, Kent State University
Yang Chen, UNC - Charlotte
Dongning Luo, UNC - Charlotte
Ye Zhao, Kent State University
Jing Yang, UNC - Charlotte
Wenwen Dou, UNC - Charlotte
Shixia Liu, Microsoft Research Asia

ABSTRACT

An interactive visualization system, STREAMIT, enables users to explore text streams on-the-fly without prior knowledge of the data. It incorporates incoming documents from a continuous source into existing visualization context with automatic grouping and separation based on document similarities. STREAMIT supports interactive exploration with good scalability: First, keyword importance is adjustable on-the-fly for preferred clustering effects from varying interests. Second, topic modeling is used to represent the documents with higher level semantic meanings. Third, document clusters are generated to promote better understanding. The system performance is optimized to achieve instantaneous animated visualization even for a very large data collection. STREAMIT provides a powerful user interface for in-depth data analysis. Case studies are presented to demonstrate the effectiveness of STREAMIT.

1 INTRODUCTION

Advanced technologies (e.g. mobile phone and internet) have greatly increased the quantity and accessibility of text documents in human society. Massive documents are generated at a significant speed, e.g., from daily, hourly, or minutely emails, messages, webs, broadcasts, and TVs. They have introduced an urgent need for efficient storage, processing, and analysis of such constantly growing text collections. Recently, visualization tools have been successfully applied in processing and analyzing text data.

A stream text collection constantly evolves as new documents are continuously generated and published. Unlike traditional text database, the quantity and representation (e.g. keywords or topics) of the documents are not known in advance. Visual exploration of real text streams is a challenging task. First, text streams continuously evolve. Visualization aids should be provided to trace the temporal evolution of existing topics, monitor emerging topics, and examine the relationships between them. Second, a visualization system should process a text stream without pre-scanning the whole stream or assuming a priori knowledge. Third, a visualization system should allow users to interactively change their information seeking focus at any time and receive immediate feedback. Such interactivity is a decisive factor for a visual analytic system in real applications where domain users usually do not know the text streams in advance. Fourth, a visualization system should scale to large volumes of text streams and respond to their evolution in real time.

In this paper, we design a novel dynamic visualization system, named **STREAMIT**, for exploratory applications of text streams. The paper is expanded from our previous work [2] by introducing topic modeling, automatic cluster discovery, and enhanced visualizations. The system is based on a dynamic force-directed simulation into which documents are continuously inserted. Each document is represented as a mass particle that moves inside a 2D visualization domain. A potential energy is defined by pairwise text similarity between documents. Minimizing the total potential energy of the system moves similar document particles closer and drives away dissimilar ones, which are achieved by attractive and repulsive forces between particles. Consequently, an equilibrium state of the particles visually depicts the data clusters and outliers at a particular moment. The system automatically adjusts its visual output with newly injected document particles. The dynamic procedure of this change is critical for reducing change blindness when new patterns emerge. This system has the following features to enable effective text stream visualization:

Continual evolvement: This physical model is well-suited to visualize text streams for continuous depiction and analysis of growing document collections, where the dynamic nature is simulated through the dynamic behavior of particles. Text documents enter the system at any time and automatically join clusters of related collections. In the meantime, the particles already inside the system travel continuously under the impact of new particles. The visual structures hence gradually evolve without abrupt changes that break the mental picture users already formed. Erratic motion of particular particles (e.g., moving from one cluster to another cluster) may reveal outliers or significant new trends. This is advantageous to existing static or time-window based visualization approaches, which depict only stationary data patterns or the sporadic transitions between these patterns.

Dynamic processing: Text documents are typically represented and manipulated through the vector of keywords. Existing methods usually pre-calculate similarity between documents from their predefined constituent keywords. Instead, we develop dynamic keyword vectors that upgrade adaptively from the incoming documents. Essentially, our system does not require a scan of the whole collection before visualization. The visualization parameters and functions can be managed with respect to the temporal context.

Interactive exploration: We propose a *Dynamic Keyword Importance* that presents the significance of a keyword at a certain time. It reflects user interest so that similarity is updated on-the-fly, and consequently the visualization artifacts. For instance, keywords with increasing importance will make documents having those keywords aggregated closer.

Scalability: Our system works well with topic modeling techniques, such as the LDA model [3]. They summarize the documents using a set of probabilistic topics, while the topics are described by a probability distribution of keywords. By using topics, we reduce the operating space from a large number of keywords to a much lower dimensional feature space, which can be easily used to investigate thematic variation. In addition, we generate clusters on-the-fly from the force-based results, where a Delaunay triangulation combined with graph cut are applied by directly utilizing geometric features. This dynamic clustering function enables easy visualization of cluster growth, split, and merge for better knowledge discovery.

Performance optimization: We optimize our method by introduc-

ing a similarity grid that helps new particles quickly reach their preferred location. Moreover, our particle system is inherently parallel for direct GPU acceleration, which achieves fast speed for a very large number of documents.

2 Related Work

Many text visualization systems use similarity-based projection to help users get insights from large text collections. IN-SPIRE [11] uses multidimensional scaling (MDS) to map documents with similar contents close to each other, and thus form "galaxies" or "mountains" in the displays. A point placement approach is proposed in [9] to build a hierarchy of the documents and project them as circles. Our approach is different in that it uses a dynamic similarity-based projection system to depict text streams.

Related to our aim to handle continuous incoming text streams, TextPool [1] produces a visual summary that clusters related terms as a dynamic textual collage. Unlike our method, it visualizes very recent stream content as a partially connected graph, which is "not for analyzing any significant portion of stream history". Besides, the graph represents salient terms of the stream instead of the documents. Wong et al. [12] dynamically visualize stream data in a moving time window using incremental data fusion. Newly arrived data items are inserted into existing layout when the error of the similarity placement is smaller than a given threshold. Once the threshold is exceeded, the whole layout is recalculated. Interactive exploration and user control are not addressed in [12]. Eventriver [8] processes incoming documents on the fly using a dynamic keyword processing and an incremental text clustering algorithm where individual documents are not visible from the overview of the stream. However, in our approach, individual documents can be examined within the global temporal and similarity context. Hetzler et al. [6] visualize text collections in a 2D projection space with fresh and stale documents visually distinguished. They apply IN-SPIRE [11] to a dynamic document flow. When new documents are added, the existing vocabulary content is adjusted and the visual result is regenerated. However, the method does not show the animated transition of the view. In comparison, our system reveals the evolvement of the stream in fine details with controllable transient animations.

Our algorithm employs Force-Directed Placement (FDP) for visualizing dynamic documents. FDP [5] has $O(N^3)$ complexity which urges researchers to improve its computational performance. Restrictions are imposed on the force calculations to a subset of the entire data, which could possibly lead to misleading approximated results. Unlike these methods working on static high-dimensional data, our approach is among the first efforts to visualize text streams using force-directed placement. Furthermore, to make correct dynamic behavior, we avoid reducing force computation scope on only a portion of the particles. Instead, we use a spatial division of the visualization domain for fast locating the appropriate initial position of particles. More importantly, we fully utilize the parallel nature of the simulation algorithm by GPU acceleration which achieves a dramatic speedup.

3 System Overview

The infrastructure of STREAMIT is illustrated in Figure 1. Continuously incoming text documents are visually presented to users in a dynamic 2D display. Users can explore the documents and clusters based on keywords or topics in the dynamic display. Furthermore, we employ tag clouds and present a novel spiral view to visualize and analyze clusters in a global view. The semantics of the clusters are examined in keyword clouds, and the titles of individual documents are displayed as labels. Users can discover emerging patterns on-line by monitoring the real-time display. They can also examine the temporal evolution of historical data through animations that playback the stream evolvement over time. A set of interactions are

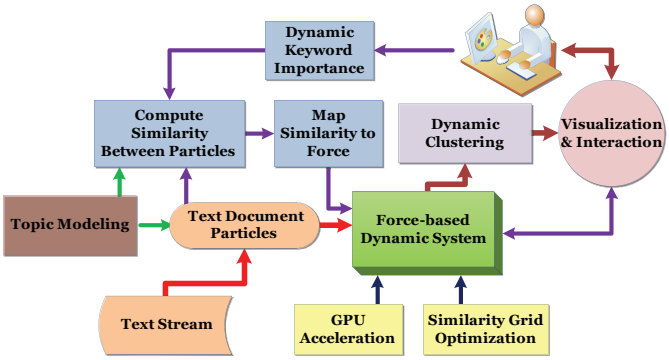


Figure 1: STREAMIT system overview.

provided for users to manipulate the visual structure of the display on-the-fly such as through varying keyword importance.

4 Force-based Dynamic System

4.1 Particle Potential

Documents are presented as mass particles inside the 2D domain with their velocity and acceleration following Newton's law of motion. Each pair of particles has a potential energy Φ_{ij}:

$$\Phi_{ij} = \alpha(|\mathbf{l}_i - \mathbf{l}_j| - l_{ij})^2, \tag{1}$$

where α is a control constant, and $\mathbf{l}_i$ and $\mathbf{l}_j$ are the positions of two document particles p_i and p_j, respectively. While $|\mathbf{l}_i - \mathbf{l}_j|$ represents the Euclidian distance of the two particles, and l_{ij} is their ideal distance computed from similarity. Hence, this pair potential function models the deviation of the two particles from their ideal locations, which is achieved at zero potential.

4.2 Particle Similarity

An optimal layout is determined by the definition of l_{ij}. l_{ij} is obtained from the pairwise similarity computed from their keywords as:

$$l_{ij} = 1 - \delta(p_i, p_j), \tag{2}$$

where $\delta(p_i, p_j) \in [0,1]$ is the cosine similarity between document particles p_i and p_j. With this formula, those documents with large similarity will have a smaller ideal distance, l_{ij}, and move closer for clustering in the visualization.

4.3 Force-directed Model

A global potential function is the sum of the pairwise energy:

$$V(\mathbf{l}_1, ..., \mathbf{l}_N) = \sum_i \sum_{j>i} \Phi_{ij}, \tag{3}$$

where N is the particle number, and $\mathbf{l}_1, ..., \mathbf{l}_N$ represent the current locations of these particles. The potential of the system is minimized to an equilibrium state that provides a global optimized placement of these particles. A numerical simulation is performed to achieve the optimization by minimizing the global potential with a sequence of simulation time steps. At each time step, the minimization leads to forces acting on each particle:

$$F_i = -\nabla_{\mathbf{l}_i} V(\mathbf{l}_1, ..., \mathbf{l}_N), \tag{4}$$

which attracts or repulses particles from each other. From Newton's law:

$$F_i = m_i a_i \tag{5}$$

where m_i is the mass. We compute the particle acceleration as:

$$a_i = \frac{2\sum_j \alpha(|\mathbf{l}_i - \mathbf{l}_j| - l_{ij})}{m_i}, \tag{6}$$

which is used to update the location of the particle, p_i, at each simulation time step. While every particle no longer moves (in numerical computing, the displacement smaller than a threshold ξ), the system is optimized to its best visual layout.

Algorithm 1 Dynamic Simulation Algorithm

```
Set the maximum displacement D as a large value
while D > ξ do
  for i = 0 to N - 1 do
    for j = i + 1 to N do
      F_i += 2 * α(|l_i − l_j| − l_ij)
    end for
  end for
  for i = 0 to N - 1 do
    a_i = F_i / m_i
    update the position of this particle
    update maximum displacement D of all particles
  end for
end while
```

Algorithm 1 describes the basic computing procedure, where we assume every particle has the same unit mass. The constant α is an empirical parameter used to control the force ($\alpha = 0.01$ in the case studies), so that the numerical simulation is stable, i.e., all the particles will not totally move out of the 2D domain or be squeezed to the center of this domain.

5 Advanced Features for Interactive Exploration and Scalability

The evolving force-based system successfully generates and presents document clusters, as well as outliers, for dynamic visualization. It automatically creates temporal visual output from continuously inserted documents. We further develop techniques for advanced data exploration.

5.1 Dynamic Keyword Importance

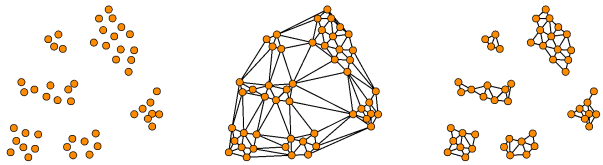
Keywords are vital words that frequently occur in a document. The similarity $\delta(p_i, p_j)$ is typically computed by predefined formula, e.g. cosine similarity, from the keyword vector of documents p_i and p_j. However, stream text collections usually span a long period of time. For a real world stream, one keyword might excessively appear for a period of time and then fade out, while another one might frequently pop up during the entire period of time. While users typically do not have knowledge about the incoming documents, they will change their focus of interest along the stream evolvement. Consequently, the definition and computation of similarity should instead be a function of time and adjusted by user input.

To address the challenge, we propose Dynamic Keyword Importance in addition to the computation of $\delta(p_i, p_j)$, which interactively enables the users to manipulate the significance of keywords at any time. The classic cosine similarity can be improved as:

$$\delta(p_i, p_j) = \frac{\sum_{k=1}^{K}(w_{ik}I_k)(w_{jk}I_k)}{\sqrt{\sum_{k=1}^{K}(w_{ik}I_k)^2 \cdot \sum_{k=1}^{n}(w_{jk}I_k)^2}} \tag{7}$$

where I_k is the importance of keyword k, K is the number of keywords, and w_{ik} is the weight of keyword k in the document p_i. The classic cosine similarity can be considered as a special case where $I_k = 1$. All the K weights form the keyword vector of this document. The length of the current vector is dynamically updated, so that our system can handle data streams not prerecorded. The weight of keywords is calculated as:

$$w_{ik} = O_{ik} * \log_2 \frac{N}{n_k} \tag{8}$$



(a) Particles (b) Triangulation (c) Graph cut

Figure 2: Cluster generation with triangulation and graph cut.

where O_{ik} is the occurrence of keyword k inside the document i, N is the total number of documents, and n_k is the number of documents that contain the keyword k inside N. The inverse document frequency factor $\frac{N}{n_k}$ favors the keywords concentrated in a few documents of a collection, in comparison to the keywords with similar frequency but are prevalent in the whole collection [10].

Users can freely modify the keyword importance through the visual interface, where frequent keywords are presented in an ordered list. Furthermore, the importance can also be determined automatically by the system as follows (See Section 7.1):

$$I_k = a * O_k + b * (te_k - ts_k) + c * n_k. \tag{9}$$

Here, O_k is the occurrence of keyword k in the current existing documents, te_k is the last time it appears, and ts_k is the first time it appears. $(te_k - ts_k)$ makes the importance larger for aged keywords. n_k makes the importance larger for the keywords appearing in a large number of different documents. Here, a, b, and c are positive constants satisfying $a + b + c = 1$. They are selected to determine how the three factors are preferred. In our experiments, we use $a = 0.3$, $b = 0.3$, and $c = 0.4$. Users indeed can define their preferred keyword importance in a variety of functions for different purposes.

5.2 Topic-based Visualization

A document is usually represented as a vector of keywords. The capability of STREAMIT in revealing clusters may be impaired when the dimensionality of the keyword space is too high, due to the lack of data separation in such a high dimensional space. Recent topic modeling techniques, such as the LDA (Latent Dirichlet Allocation) model [3], reduce the keyword-document space to a much lower feature space that is not only intuitive to interpret, but also captures most of the variance in the corpus. In particular, topic modeling automatically represents the documents using a set of probabilistic topics. The topics are described by a probability distribution over keywords.

We use the LDA model [3] to extract topics from a large document archive that is highly related to the text stream to be visualized, such as its historical archives. Each extracted topic is associated with a set of keywords highly related to the stream. During the ongoing visualization, STREAMIT dynamically examines whether an incoming document contains the extracted topics according to its keywords. The document is then represented by a vector of probable weights of topics it contains. All the aforementioned calculation, visualization, and interactions based on keywords can be applied to the topics. The benefits are significant: since the number of topics is much less than the number of keywords, the documents are better clustered; since the topics are at a higher semantic level than keywords, it is easier for users to understand the clusters generated (see Section 7.3). For example, interdisciplinary proposals that cover multiple topics, which are difficult to be identified in the keyword-based approach, can be easily detected in the topic based approach.

5.3 Dynamic Clustering

The distribution of document particles in the 2D space allows users to visually identify clusters of documents with similar semantics. However, based on individual particles, it is difficult to conduct cluster-level operations, such as selecting all documents in a cluster and examining the semantics of a cluster. To address this problem, STREAMIT automatically discovers clusters from the evolving geometric layouts, so that they can be explicitly presented and operated through the visual interface. Moreover, the visualization can display a text stream in the cluster level to reduce clutter and enhance scalability.

5.3.1 Cluster Generation

At a moment, a group of document particles can be considered forming a semantic cluster with the following definition:

Definition: *If particles s and t are in a cluster, there must at least exist a path between s and t, which connects a sequence of particles s,* p_0, p_1, ..., p_c, *t with pairwise line segment,* $s \rightarrow p_0$, $p_0 \rightarrow p_1$, ..., $p_c \rightarrow t$. *The maximum length of all the connected segments is smaller than a predefined threshold* ζ.

Here we use the single linkage rule in defining clusters, which considers connected components (with respect to ζ) as one cluster. We discover such clusters directly from the 2D geometric layout. In particular, a typical agglomerative algorithm can be applied to partition all particles into clusters: starting with N particles forming N clusters, repeatedly merging two clusters according to the distance between the nearest neighbors of them. This straightforward approach has $O(N^2)$ complexity and does not utilize the geometric layout of the particles. A drawback of this approach is that the resultant clusters are only represented by individual particles and no topological information is provided. Since an effective visualization should show the spatial areas of these clusters distinctly, a computational geometry method has to be invoked to create a simple polygon from the particles of each cluster. To address this problem, we propose to use Delaunay triangulation and graph cut in generating clusters. A similar approach has been used in spatial data mining [7]. The algorithm is shown in Algorithm 2. The graph cut partitions the particles into disjoint sets (i.e. connected components) that represent the semantic clusters. Figure 2 illustrates the cluster generation process.

Algorithm 2 Creating Clusters From Triangulation

Step 1: Apply Delaunay triangulation for all particles in the system;

Step 2: In the created graph (i.e., the triangle mesh), cut the edges whose length is larger than ζ.

Our method has the complexity of $O(NlogN)+O(E)$ for N particles, with $O(NlogN)$ for Delaunay triangulation and $O(E)$ for browsing all E edges in graph cut. Since in Delaunay triangulation the maximal number of edges is $3N-6$, $O(E) \sim O(N)$. Therefore, our method achieves $O(NlogN)$, better than the agglomerative algorithm.

5.3.2 Cluster Evolution

The created clusters merge and split over time when new documents arrive or keyword importance is changed. Such evolution of the clusters are critical for knowledge discovery and should be tracked and visualized. Therefore, we propose a method to track cluster evolution. In particular, each cluster is given a distinct ID value for identification and each particle carries the ID of its cluster. To manage the cluster identification in a context-aware way, we compute a *preferred ID list* for each new cluster. The list ranks the IDs carried by all the particles inside this cluster before the update, according to the number of the particles with the same ID. The largest new cluster is first assigned the top ID in its list. Then we iteratively choose a cluster according to the cluster size (i.e., the number of particles). Each cluster is given one ID following the order of its preferred list, if the ID has not been assigned to other clusters. If all ID choices are occupied, this cluster is given a new ID that did not appear in the previous step. In this way, large clusters have the tendency to keep their contextual information from previous time.

Each ID is associated with a color, which is assigned to the cluster with that ID (see Section 6). To avoid color clutter, we set a threshold K of the number of significant clusters. The largest K clusters are considered significant and displayed with background halos in the colors assigned to them. All the other clusters are displayed without background halos (see Figure 6).

6 Visualization and Interaction

6.1 Visualization

STREAMIT has a main window, an animation control panel, a keyword/topic table, and a set of document tables (see Figure 3):

Main Window: The main window (top left of Figure 3) visually presents the movement of the particles through an animated 2D display. Each document particle is represented by a pie. The similarities among the documents are reflected by the closeness of the positions. The pie position dynamically changes to reveal the temporal evolution of the stream. A grey scale is used to indicate the age of the documents, namely the older a document is, the darker its color (see Figure 3). The size of pies can be proportionally mapped to an attribute of the documents. Moreover, the size and transparency of pies can be adjusted to lessen any clutter that might be introduced as the number of documents grows. Keywords of interest are represented by pie sectors and their colors can by assigned by users.

Animation Control Panel: STREAMIT buffers recent documents falling into a moving time window (named the buffer window) that is larger than the moving window of currently displayed documents. Users can playback the animation within the buffer window to examine the temporal and semantic evolution of the buffered stream in detail. Users can change the size of the buffer window to explore a longer or shorter time period. An animation control panel is used to control the playback (see Figure 3(3)). The users can move the slider to start the animation from any moment and they can pause the display to examine a moment of the stream or change parameters such as keyword importance.

Keyword and Topic Tables: STREAMIT provides keyword information in a keyword table which is updated dynamically (see Figure 3(1)). It lists all the keywords characterizing the documents currently displayed, their frequencies, importance, and colors. Users can sort this table to find frequent and important keywords. They can also change the keyword importance or colors. When topic modeling is used, topics will replace keywords in the table and the significant keywords describing each topic will be represented.

Document Tables: Users can click a tab to show one of four document tables (see Figure 3(2)). They display the titles, authors, and timestamps of the following documents respectively: (1) all buffered documents; (2) all documents that are displayed in the main window; (3) documents selected by users; and (4) document clusters generated by the system or created by users. The users can sort the documents by their authors or timestamps. They can also click a title to reach the full text of a document.

6.2 Labeling

Labels revealing semantic contents of a collection are desired in text visualization systems. Titles of the documents contain rich semantic information in a condensed manner and thus STREAMIT uses titles as labels of the documents. Severe clutter can be generated if titles of all documents are displayed. We develop a novel labeling algorithm to provide the most recent semantic information with user-controllable clutter levels. In particular, documents are

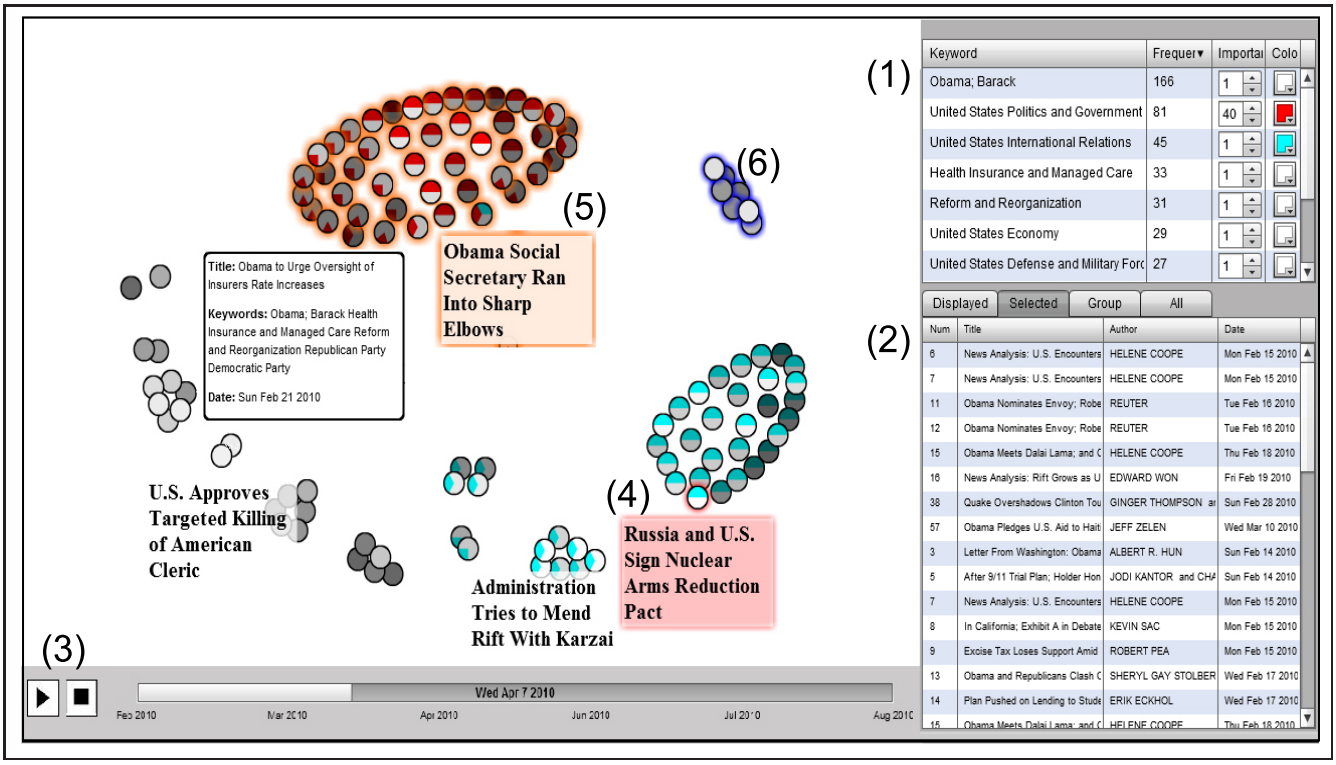


Figure 3: STREAMIT interface. The left part is the visualization view of text streams, and the right part includes keyword table, document tables and parameter controls.

divided into groups according to a dissimilarity threshold. Within each group, the dissimilarities among the documents are less than the threshold. Only one document, namely the most recently arrived document, is labeled in each group. By interactively changing the dissimilarity threshold, users can control the label clutter. A newly arrived document is either assigned to an existing group or forms a new group. Thus no labels will be changed except the label of the group affected. This is an important feature to keep the temporal consistency among adjacent displays. The newest injected document will always be labeled, which is usually desired in text stream visualization. Figure 3 shows the automatic labeling results. The newest injected document and its label are highlighted by red (see Figure3(4)) while the selected documents and their labels are highlighted by orange (see Figure3(5)).

Labels and particles may overlap when a large number of documents are displayed. STREAMIT displays labels on the top of particles and allows users to interactively change the transparency of their background. An opaque background makes the labels easy to read and semi-transparent background (Figure 3) allows users to examine particles hidden by the labels. Users can turn off all the labels and they can also turn on/off an individual label by clicking it.

6.3 Visual Representation of Clusters

After the documents are automatically divided into clusters (Section 5.3), their outlines are represented by the triangle meshes. Background halos are displayed in the mesh area for significant clusters in the assigned colors. Figures 6(A)-(B) show how the background haloes allow users to track the clusters during the dynamic visualization. The system also allows users to explore clusters in a less cluttered spiral view where their temporal trends can be examined [4]. Figure 6(C) shows the spiral view of 12 clusters from a NSF award collection. Each spiral is a time axis located at the center of a cluster in the original 2D display. The documents of the cluster, displayed as pie charts, are mapped to the spiral according to their time stamps. Users can thus learn the temporal trends of the cluster by observing the distribution of the pie charts on the spiral. In topic modeling view, pies are colored with two colors (red and yellow in Figure 6(C)) to indicate how documents are related to the general topics of the cluster. The ratio between the red area and the yellow area in a pie indicates the number of the most shared keywords (or topics) in the cluster a document contains, against other keywords (or topics) in this document.

Users can quickly examine an unknown cluster through a keyword cloud triggered by selecting the cluster. It displays the most significant semantic information of the cluster, namely the titles of the most recently arrived documents and the keywords with the highest TF-IDF (Term Frequency-Inverse Document Frequency) weights. Figures 7(A)-(B) show the keyword clouds of two clusters, respectively. The keywords are displayed below the titles, whose sizes indicate their weights. Users can interactively set the colors of the background, titles, and keywords. By clicking a keyword in a cloud, users can select all documents with this keyword from the cluster.

6.4 Interaction

STREAMIT allows users to interactively manipulate the visualization according to varying interests. It also allows users to search, track, and examine documents.

6.4.1 2D Display Manipulation

Adjusting Keyword Importance: Users can adjust the keyword importance to emphasize particular contents of interest and receive immediate response (Figure 4).

Grouping and Tracking Documents: User-selected groups or automatically computed clusters can be highlighted by halos in user

assigned colors, which promotes easy document tracking in the dynamic display. Figure 3(5) and (6) show two groups in orange and blue, respectively.

Browsing and Tracking Keywords: Users can assign colors to keywords of interest to track them. A document pie conveys the colors of traced keywords. The size of a color section is proportional to the weight of the keyword in the document. Users can investigate keyword and document relations and track the evolution of relevant topics in this way (Figure 4). The users can also click a keyword of interest in the keyword table. All documents containing the clicked keywords are highlighted by halos. The users can sweep the keyword table in this way to find keywords of interest.

Setting Moving Windows: Users can interactively change the length of the moving window, i.e., investigating period, of currently displayed documents.

6.4.2 Document Selection

Manual Selection: Users can manually select documents from the document tables, or use a rubber band selection by dragging the mouse. The selected documents will be highlighted by halos (Figure 3). Their information will also be displayed in the selected document table (Figure 3(2)).

Example-based Selection: Users can use the current selection as examples and select documents that are within a distance range to them. The range is easily controlled to select similar documents.

Keyword-based Selection: Users can select multiple keywords from the keyword table (see Figure 3(1)), and then the related documents are automatically selected and highlighted (see Figure 3(5)).

Cluster-based Selection: Users can click the background halo of a cluster or its spiral to select all documents in it. They can also click a keyword in a keyword cloud to select all documents containing the keyword in the cluster.

Shoebox: In the dynamic environment, users may want to focus on the temporal evolution and examine the selected documents later. They can easily send the selected documents into a shoebox, which can be examined in full text later.

7 Case Studies

We present three case studies in this section. Documents in prerecorded collections are sorted by their time stamps and fed into STREAMIT with an interval of a few seconds to simulate a fast evolving text stream. Topic modeling and dynamic clustering are demonstrated in the third case study.

7.1 Exploring Barack Obama News

We explore a text stream of 230 New York Times news about Barack Obama reported between Jul. 19 and Sep. 18, 2010. The keywords are given tags that come with the news. In each document, the occurrences of the keywords are assigned to a value of one. The buffer window covers the whole stream. Keyword importance is automatically assigned by the algorithm described in Equation 9.

Figure 4(A) shows the display on Aug.13, 2010, where 136 news articles are represented. On Aug.13, 2010, we notice that keywords such as "Politics and Government", "International Relations", "Defences Military", and "Terrorism" have high frequency values according to the keyword table. We assign them distinct colors to track the articles characterized by them as shown in Figure 4(A).

We increase the importance of the keyword "International Relations". Figure 4(B) shows that the articles containing this keyword are attracted closer than in Figure 4(A). We easily select them using a rubber band selection and find in the shoebox that they contain keywords such as "China", "Terrorism" and "Afghanistan War".

We want to focus on "Afghanistan War" and "Terrorism" since most of these news articles are recently inserted (with lighter darkness). We click the keyword "Afghanistan War" to select the related articles and create a new group named "war" for them. We also highlight the group in pink halos (Figure 4(B-2)). We create another group for "Terrorism" in the same way and highlight it in orange halos (Figure 4(B-3)). Then we continue to play the animation and track the evolution of these groups. Figure 4(C) shows the visualization when all the news articles are displayed. We notice that the cluster shown in Figure 4(B-3) gets much bigger. We also notice that there is a recent news article (Figure 4(C-4)) that stands in-between it and the cluster shown in Figure 4(B-2). It is related to both "Afghanistan War" and "Terrorism" (see Figure 4(C-4)). We select this article and read it in full detail by clicking the circle.

Table 1: A table of topics.

Topic	Descriptive Keywords
Topic 2	*data;mine;cluster;graph;biology;analysis;discovery*
Topic 6	*image; scene; model; recognition; language; shape; speech*
Topic 12	*biological, protein; genom; search; gene; sequence; patent*
Topic 13	*video; motion; asl; 3D; camera; sign; dance*
Topic 15	*image; speech; haptic; display; impair; auditory; graphic*
Topic 16	*query; database; data; xml; stream; edu*
Topic 19	*data; workflow; privacy; management; web; metadata*

7.2 Exploring NSF Award Abstracts

We explore 1000 National Science Foundation (NSF) IIS award abstracts funded between Mar. 2000 and Aug. 2003 as a text stream. The time-varying funding behavior is critical in understanding research and administrative trends. Each document was automatically characterized by a set of keywords and its corresponding pie size is proportional to the funding amount of the project.

Figure 5 shows several snapshots of the dynamic visualization. Figures 5(A) and (B) show the stream in two adjacent months. We notice that multiple large projects started from the second month. We pause the animation, select items of interest, and then examine them in detail. From the shoebox, we observe that the keywords "Management" and "Database" appear in many of these project abstracts. We highlight the keyword "Management" in red and the keyword "Database" in green. We also increase their importance values so that we can observe the relevant abstracts easier (Figure 5(B)). Although some abstracts contain both keywords (Figure 5(B-1)), there are many other abstracts that contain only one of them. We pull back the animation to the previous month (Figure 5(A)) to examine the temporal evolution of these topics. When the stream further evolves, we observe that IIS continuously supported projects with these keywords (Figure 5(C)).

In Figure 5(C), we highlight all projects containing the keyword "sensor" by halos. The node with a halo indicated by the arrow (Figure 5(C-2)) is a potential transformative proposal since it is far away from the other projects with halos. We examine this abstract in detail and find that it is a project about just-in-time information retrieval on wearable computers.

7.3 Exploring NSF Award Abstract with Topic Modeling and Dynamic Clustering

We explore the same NSF data with the topic modeling and dynamic clustering. It reveals how these features significantly increase the scalability of STREAMIT. Figure 6(A) can be compared to Figure 5(B) to demonstrate the difference between the topic-based visualization and keyword-based visualization. They show the same set of documents at the same month.

It can be seen that a large number of documents that seem not related to other documents actually belong to clusters in the higher semantic level. Table 1 shows a list of the involved topics in Figure 6(A). Topics 16 (red pie sections in Figure 6(A)) and 19 (green pie sections in Figure 6(A)) contain the keywords "Database" (red pie sections in Figure 5(B)) and "Management" (green pie sections in Figure 5(B)), respectively. The semantics of the related clusters

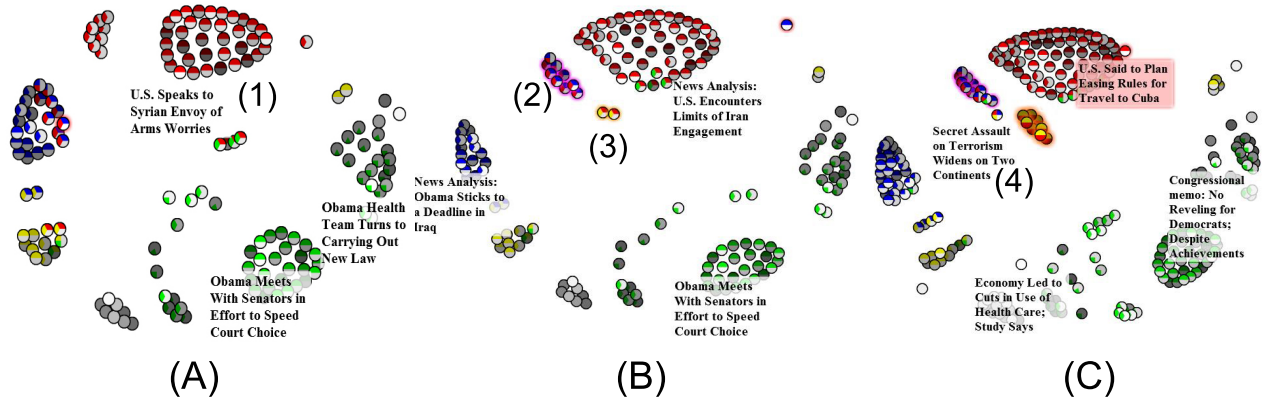


Figure 4: Barack Obama news. (A) Aug. 13, 2010, 136 news articles; (B) after increasing importance of "International Relations"; (C) Sep. 18, 2010, 230 news articles. Keyword colors: "Politics" - green, "International Relations" - red, "Terrorism" - yellow, and "Defense and Military" - blue.

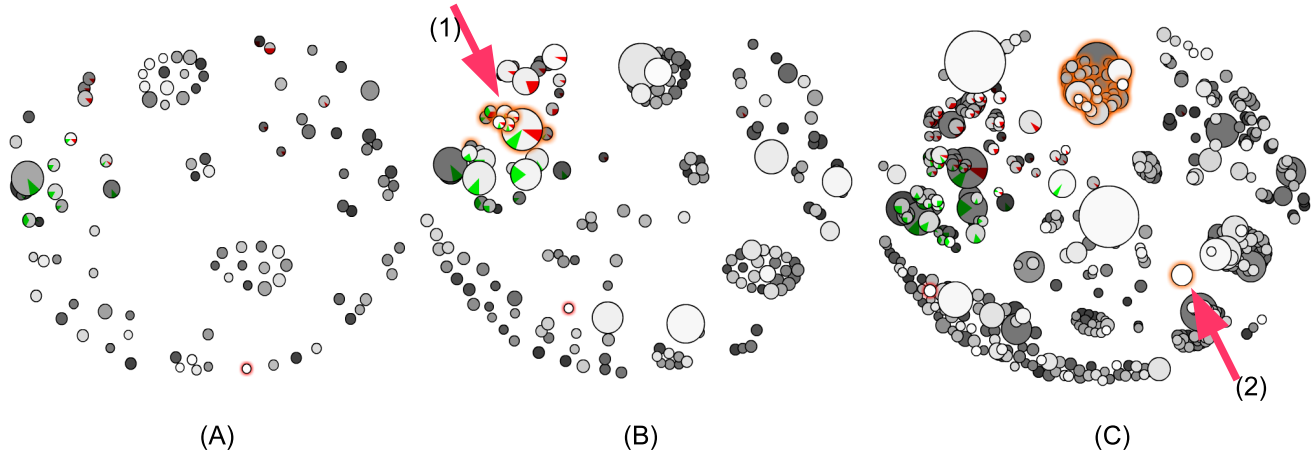


Figure 5: NSF award collections. (A) Aug. 1, 2000, 95 research projects; (B) Sep. 1, 2000, 172 research projects; (C) Mar. 15, 2002, 672 research projects. Keyword colors: "Management" - green, "Database" - red.

is easier to understand in the topic-based visualization than in the keyword-based visualization.

Figure 6(A)-(B) demonstrate dynamic cluster evolution. In Figure 6(A), we can see a handful of clusters. The cluster (A)-(1) mainly about Topic 15 and the cluster (A)-(2) mainly about Topic 6 merge into one larger cluster (B)-(3). In Figure 6(B), we observe two newly formed clusters: the cluster (B)-(4) mainly about Topic 13 and the cluster (B)-(5) mainly about Topic 12. The visual effects help users identify critical data variation. Users can discover semantic details of the evolving clusters by examining their keyword clouds. Figure 7 shows the keyword clouds for the two new clusters, namely (B)-(4) and (B)-(5), respectively.

Figure 6(C) displays the spiral view of clusters at the same moment with Figure 6(B). Each spiral represents a cluster: (C)-(6) represents (B)-(4) and (C)-(7) represents (B)-(5). Two or three most significant topic names of a cluster are displayed below its spiral. Users can also examine details from the keyword clouds. Each pie on a spiral represents a document. The red area of a pie indicates how the document is related to this cluster's major theme. In Figure 6(C), we can discover that large pies (i.e., large projects with funding amount over 1 million project) typically have small red area, indicating that they are probably interdisciplinary projects. On the contrary, small projects usually involve fewer topics and agree more with the clusters.

8 Performance Optimization

Algorithm 1 is an $O(N^2)$ approach and Algorithm 2 is an $O(NlogN)$ approach. We seek to optimize the performance of our system by applying parallel computing and similarity grid to improve its scalability for large data sets.

8.1 GPU Acceleration

Our computational algorithm is inherently parallel at each simulation step. Hence, we accelerate the computation on graphics hardware with CUDA implementation similar to an N-body problem [1]. N particles execute their force-placement algorithm simultaneously as individual threads distributed to a grid of CUDA blocks. Each thread accesses and updates the particle's position from the information of last step loaded into the shared memory of the blocks.

As shown in Table 2, we achieved very good performance on an NVidia NVS 295 GPU with 2GB memory for large scale data sets, compared with an Intel Core2 1.8GHz CPU with 2GB RAM. We conducted a few experiments using text streams from the New York Times news. For each frame, the simulation ran multiple steps with

[1] *http.developer.nvidia.com/GPUGems3/gpugems3_ch31.html*

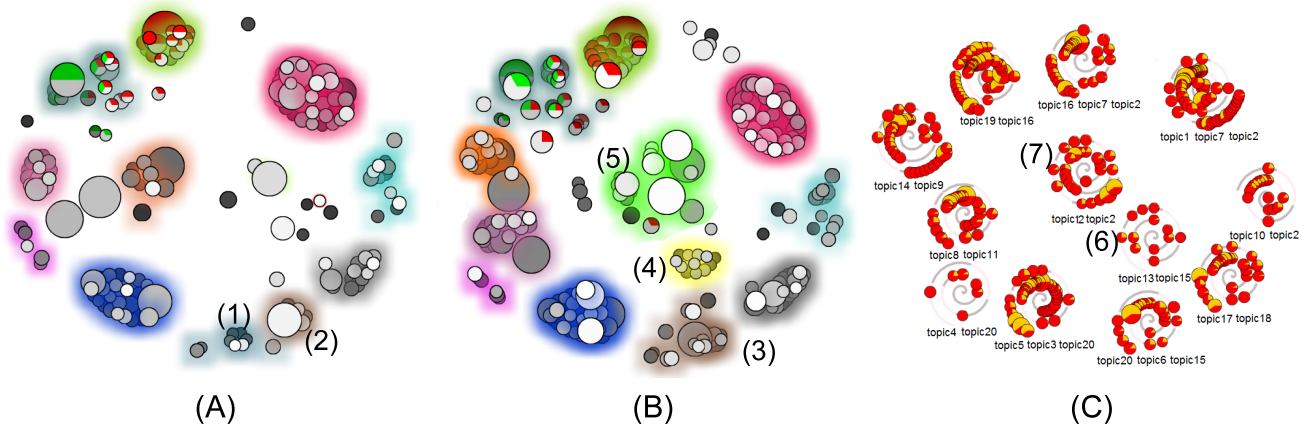


Figure 6: NSF award data with topic modeling. (A) Sep. 1, 2000, 172 research projects. (B) Sep. 15, 2001, 330 research projects. (C) Spiral view of (B). Topic color: topic 16 - red. Topic 19 - green.

Table 2: STREAMIT performance on CPU and GPU (in milliseconds) with selected text streams of New York Times news

Document Time Period	Number of Documents in System	Number of Keywords in System	Ave. Simulation Time Per Frame		GPU/CPU Speedup	Maximum Simulation Time		Avg. of Simulation Steps Per Frame
			CPU (ms)	GPU (ms)		CPU (ms)	GPU (ms)	
Feb.13 - Aug.18, 2010	6157	5057	540	34	17.9	4350	230	173
Aug.1 - Oct.31, 2006	7100	1059	620	41	15.1	9070	480	177
Jul.1 - Aug.31, 2010	10205	2036	986	53	15.9	11030	610	200
Synthetic Data set	15000	2000	1020	65	15.7	13070	682	196

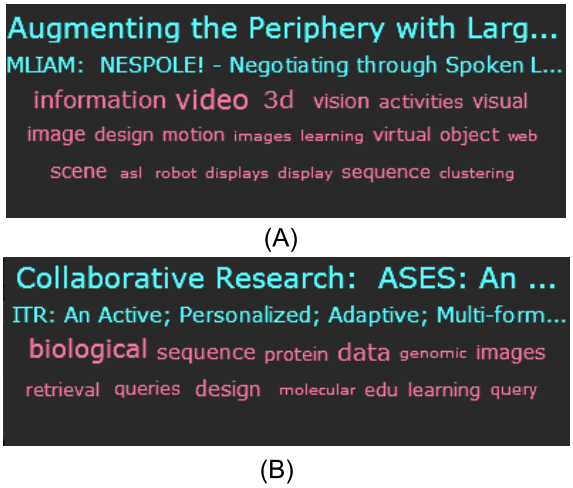


Figure 7: Keyword clouds. (A) For cluster of Fig. 6(B)-(4); (B) For cluster of Fig. 6(B)-(5)

the preset minimum threshold ξ at 10^{-4}. On average, real time running performance was achieved by the GPU acceleration at a frame rate around 25-30 frames per second. It was above 15 times faster than the CPU version. The maximum simulation time after each document insertion on the GPU was less than a second, which was sufficiently fast considering the relatively slower response time for human perception and analysis of the visualization update. In addition, we tested STREAMIT on a synthetic data set with around 15,000 documents and 2,000 keywords. The results showed that our system worked well for such a large text stream on the GPU.

Table 3 reports the GPU/CPU performance of triangulation-based dynamic clustering. With GPU acceleration, the cluster generation does not impose much extra overhead on the system.

Table 3: Dynamic clustering performance (in milliseconds)

Number of Documents in System	Ave. Time Per Frame		Max.Time Per Frame	
	GPU (ms)	CPU (ms)	GPU (ms)	CPU (ms)
6157	317	1270	334	2875
7100	324	1546	334	3484
10205	330	2544	341	5625
15000	332	4247	356	9423

Table 4: Performance optimization obtained by employing similarity grids on a data set of 7100 documents

Similarity Grid Size	Average Number of Simulation Steps
None	225
20×20	207
50×50	177
100×100	182
200×200	186

8.2 Similarity Grid

The initial positions of document particles significantly affect the computational cost. We employ a similarity grid to ensure that new documents are roughly inserted within the proximity of similar documents. The grid divides the 2D visualization domain into rectangular cells with a given resolution. Each cell actively maintains a special keyword vector consisting of the average keyword weights computed from the documents inside the cell. For a new document, we first compute its similarity with this special keyword vector of the grid cells to find the most similar one, and then place this document at the center of that cell. An appropriate resolution will provide a good acceleration while it cannot be too large due to the extra overhead.

Table 4 shows the average number of simulation steps required by 7100 documents with different grid sizes. A 50×50 grid decreased the simulation steps per frame to 78% of the steps needed without the grid. Meanwhile, the execution time was reduced with the same ratio. We used a 50×50 grid for our experiments reported in Table 2.

8.3 Discussion

The performance optimization makes our system applicable in a monitoring setting for live streams. The New York Times news are produced continuously, averaging 3 documents per hour and a maximum of 8 documents per hour at the peak time. The minimum interval between consecutive arrival is around 1 minutes. A capable real-time visualization system should handle newly inserted items faster than this minimum interval. From Table 2, the maximum simulation time of the CPU computing is a few seconds. With GPU acceleration, the handling time is further reduced to less than one second. Therefore, our system can be effectively employed for this news stream with many thousands of documents accommodated in the display for analysis. Note that the capability is provided with an ordinary consumer PC and graphics card. Our system is being upgraded to handle even larger text streams with advanced GPUs. It is important not to overwhelm the users with the flood of information. Our system allows users to manipulate the simulation speed, and users can pause the system and save clusters/documents for further investigation. We will also utilize the unbalanced text streaming speed in future improvement.

9 Conclusion

We have presented a new visual exploration system, STREAMIT, for text streams. The system employs a physical framework to cluster and dynamically analyze incoming documents. A new *Dynamic Keywords Importance* helps users interactively manipulate the importance of keywords for different visualization results. Topic modeling is incorporated into our system and a triangulation based clustering helps better visualization. Furthermore, the system is equipped with powerful interactive tools and accelerated on consumer GPUs, consequently providing fast simulation, immediate response and convenient control.

In future work, we will integrate STREAMIT with online topic modeling techniques to visualize text stream with frequently evolving topics. Furthermore, we will conduct user studies to further assess the feasibility of the system. We also plan to apply STREAMIT to a variety of real life applications.

Acknowledgment

This work is in part supported by NSF grant IIS-0915528, IIS-0916131, NSFDACS10P1309 and Ohio OBR. We thank the anonymous reviewers for helpful reviews and Zhi Yuan for improving the system.

References

[1] C. Albrecht-Buehler, B. Watson, and D. Shamma. Visualizing live text streams using motion and temporal pooling. *IEEE Computer Graphics and Applications*, 25(3):52–59, June 2005.

[2] J. Alsakran, Y. Chen, Y. Zhao, J. Yang, and D. Luo. Streamit: Dynamic visualization and interactive exploration of text streams. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 131–138, 2011.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[4] J. Carlis and J. Konstan. Interactive visualization of serial periodic data. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 29–38, 1998.

[5] T. Fruchterman and E. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164, Nov. 1991.

[6] E. G. Hetzler, V. L. Crow, D. A. Payne, and A. E. Turner. Turning the bucket of text into a pipe. In *Proceedings of IEEE Symposium on Information Visualization*, page 12, Washington, DC, USA, 2005. IEEE Computer Society.

[7] I.-S. Kang, T.-w. Kim, and K.-J. Li. A spatial data mining method by delaunay triangulation. In *ACM GIS*, pages 35–39, New York, NY, USA, 1997. ACM.

[8] D. Luo, J. Yang, M. Krstajic, J. Fan, W. Ribarsky, and D. Keim. Eventriver: An event-based visual analytics approach to exploring large text collections with a temporal focus. In *IEEE Transactions on Visualization and Computer Graphics*, To appear.

[9] F. Paulovich and R. Minghim. Hipp: A novel hierarchical point placement strategy and its application to the exploration of document collections. *IEEE Transaction on Visualization and Computer Graphics*, 16(8):1229–1236, Nov. 2008.

[10] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[11] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information for text documents. *Readings in information visualization: using vision to think*, pages 442–450, 1999.

[12] P. C. Wong, H. Foote, D. Adams, W. Cowley, and J. Thomas. Dynamic visualization of transient data streams. *IEEE Symposium on Information Visualization*, 0:13, 2003.

Jamal Alsakran *is a PhD candidate at the Department of Computer Science at Kent State University. His research is focused in multidimensional and text visualization, and visual analytics.*

Yang Chen *is a PhD student in the Department of Computer Science at University of North Carolina at Charlotte. His research interests include visual analytics and information visualization.*

Dongning Luo *is a PhD student in the Department of Computer Science at University of North Carolina at Charlotte. His research interest is information visualization.*

Wenwen Dou *is a Ph.D. candidate at University of North Carolina at Charlotte. Her research is in the areas of visual analytics and human-computer interaction.*

Ye Zhao *is an assistant professor in the Department of Computer Science at Kent State University. His research interests include natural phenomena modeling, data visualization and visual analytics. He received a PhD in computer science from Stony Brook University.*

Jing Yang *is an associate professor in the Computer Science Department at the University of North Carolina at Charlotte. Her research interests include visual analytics and information visualization. She has a PhD in computer science from Worcester Polytechnic Institute.*

Shixia Liu *is a lead researcher at Microsoft Research Asia. Her research interests include visual text analytics and visual social network analysis.*