Using Visual Stability to Support Search Efficiency and User Experience in Dynamic Graph Drawings

Alfredo Ramos, Tilman Göhnert, Irene-Angelica Chounta and H. Ulrich Hoppe
Department of Computer Science and Applied Cognitive Science
University of Duisburg-Essen
Lotharstr. 65, 47057 Duisburg, Germany
Email: {ramos, goehnert, chounta, hoppe}@collide.info

***Abstract*—In this paper, we present algorithms to compute the node partition of a foresighted graph layout. We aim to produce a dynamic graph drawing that allows the users to track actors in a dynamic social network, supporting a satisfying user experience along with an efficient visual search and recognition of elements. The drawings produced by the algorithms were evaluated through a study, which combined the use of questionnaires, an eye-tracking device and a model we developed to measure the visual stability of dynamic graph drawings. The results suggest dynamic graph drawings achieving visual stability by placing multiple actors in the same position are confusing and are not efficient in the visual search. On the contrary, introducing a restriction to prevent too many actors to occupy the same position produces the desired dynamic graph drawing.**

***Keywords*—graph visualizations, dynamic social network, visual stability, questionnaires, eye-tracking.**

I. Introduction

The use of dynamic graph drawings to visualize the evolution of networks has become a popular topic in recent years. Such visual representations are often produced by capturing a successive sequence of "snapshots" from the network timeline using a prespecified time interval [1]. A graph drawing is computed separately for each snapshot and the sequence is presented to the user in a predefined other. Despite the simplicity of the method, drawings produced with the previous strategy suffer some limitations at the moment of tracking actors in a dynamic social network. Each snapshot distributes differently the actors and relations on the canvas. Thus, it is very likely that their position will change as the dynamic social network is explored. Furthermore, the actors or relations can leave the dynamic social network without prior information and affect the structure of the drawing to the point of distracting the user.

In this paper, we present algorithms to compute the node partition of a foresighted graph layout [2]. Our approaches aim to produce a dynamic graph drawing whose characteristics allows the users to track actors in a dynamic social network; supporting a satisfying user experience along with an efficient visual search and recognition of elements. The drawings produced by the algorithms were evaluated through a study involving 20 users. The participants were requested to track members of a software development community over five periods of time in the drawings produced by our algorithms. The study combined the use of questionnaires to gather information about the user experience; an eye-tracking device to obtain a better insight regarding the visual search and a mathematical model we developed to measure the visual stability of dynamic graph drawings. The results suggest dynamic graph drawings achieving visual stability by placing multiple actors in the same position are confusing and are not efficient in the visual search. On the contrary, introducing a restriction to prevent too many actors to occupy the same position produces the desired dynamic graph drawing.

II. Related Work

A. Layout adjustment techniques and the mental map

Dynamic graph drawings created from a sequence of snapshots, often minimize the changes on their structure as the users explore a dynamic social network. Much of the interest in such feature is due to the *mental map*. The mental map is the "structural cognitive information a user creates by observing the layout of a graph" [3] and helps to improve the orientation of the user on drawing [4] along with the recognition of elements [5]. Modern dynamic graph drawings incorporate strategies to preserve the mental map. For instance, *Circular Layouts* [6] provide a shape that remains constant during the exploration of the dynamic network. *Anchoring methods* utilize the number of times an actor appears in the dynamic social network to calculate a weight. The weigh is assigned to the node in the dynamic graph drawing and as the layout is processed the heavy nodes maintain their position on the canvas [7]. Other approaches like the *Foresighted Graph Layout* (FGL) [2], aggregate the actors and relations of the dynamic social network into a global layout named *Super Graph*. The super graph is used as the input of a reduction process called *reduced graph animation partitioning* (rGAP), because it requires more space to be drawn. The *lifetime* is an attribute describing the appearance of an actor or relation in the dynamic social network. A set of "containers" called partitions are used to store entities of the same type. All those actors with disjoint lifetimes can be placed into a same *node component* and the set of all node components form the *node partition*. Likewise, all those relations with disjoint lifetimes can be placed into a same *edge component* and the set of all edge components form the *edge partition*. The rGAP guarantees that the actors of the dynamic social network will always appear in the same position. Still, a position can contain multiple actors at different points in time. The partitions obtained rGAP form the foresighted graph layout, a drawing that possesses the minimal number of changes on its structure during the exploration of a dynamic social network. In other words, which can be perceived as visually stable.

B. *Visual stability in dynamic graph drawings*

In a previous study [8], we evaluated algorithms to produce dynamic graph drawings. These drawings possessed characteristics related to theories of visual stability [9], [10], [11]. Among the evaluated algorithms, we selected a *Circular Layout* (CL) because it provides a constant circular shape but changes the positions of the actors during the exploration of a dynamic social network. A *Circular Super Graph* (CSG) was also included in our study. The algorithm was selected because it assigns a fixed position to each actor appearing in the dynamic social network, but lacks of a constant shape. A *Foresighted Circular Layout* (FCL) was the last of the algorithms included in our study. The approach was selected because it assigns a fixed position to each actor appearing in the dynamic social network and also the drawing presents a minimal number changes on its structure. Still, multiple actors can occupy the same position at different points in time. We decided to utilize circular drawings to study the same shape under different conditions. A group of 15 students was involved in our study. The participants were requested to track members of a software development community over five period of times in the drawings produced by previous algorithm. The study combined the use of f questionnaires to gather information about the user experience; an eye-tracking device to record the eye movements performed by the participants and a mathematical model we proposed to measure the visual stability of dynamic graph drawings.

The result of the study suggested the *CL* was not suitable for tracking actors in a dynamic social network. The model of visual stability detected an offset in the drawing, because the positions of the actors were always changing. This characteristic was received negatively by the participants, who reported that it was difficult to track the requested actors. The visual search was also affected negatively by this characteristic, even though, the recognition of elements obtained its best performance with this technique. On the other hand, the *CSG* presented some improvements. The model of visual stability detected an absence of offset because each actor in the dynamic social network was assigned to a fixed position. Nevertheless, the model detected only a few nodes and edges were maintained on the canvas during the exploration of the dynamic social network. As consequence, the participants were distracted as they tracked the requested actors. Despite this fact, the visual search obtained its best performance with this technique but it was the opposite case for the recognition of elements. Lastly, the *FCL* also presented some improvements. The model of visual stability also detected an absence of offset due to the use of fixed positions. Additionally, the number of nodes and edges maintained on canvas was considerably higher because multiple actors occupied the same position at different points in time. As result, the participants were able to track the actors without distraction but were confused to observe multiple actors in the same position. Despite this fact, the visual search along with the recognition of elements obtained the second best performance with this technique. Based on this findings, we plan to modify the computation of the node partition executed in the foresighted graph layout; aiming to produce a dynamic graph drawing that allows the users to track actors in a dynamic social network supporting a satisfying user experience along with an efficient visual search and recognition of elements.

III. The Model of Visual Stability for Dynamic Graph Drawings

Wasserman and Faust [12] define a **graph** as a structure in the form $g = (V, E)$, where V is a set of vertices and E is a set of edges $E \subseteq V \times V$. According to Tamassia [13], a graph drawing d of a graph g consists of a mapping of each vertex v from g to a distinct point $P(v) = (v_x, v_y)$ of a plane and each edge (u, v) from g to a simple Jordan Curve with end points $P(u)$ and $P(v)$. A straight line drawing is a drawing in which every edge is mapped to a straight line segment; more formally, a straight line drawing is an injective function $f : v \in V \to (v_x, v_y) \in \mathbb{R}^2$. These definitions are the foundations of our model of visual stability. Nonetheless, we extended the notion of a graph drawing to cover algorithms like the foresighted graph layout in which multiple vertices are mapped to the same position.

We consider a **graph layout** as a mapping of each vertex v of g to a **vertex logical position** with the function f_{pv}:

$$f_{pv} : V \to P$$

where P is a set in the form $\{p_1, p_2, p_3, ...p_n\}$. The vertex logical positions can be mapped to a two-dimensional Euclidean Space with the function f_{sv}:

$$f_{sv} : P \to \mathbb{R}^2$$

$$f_{sv}(p) = (f_{sv}(f_{pv}(v)) = (x, y)$$

The edges (u, v) of g can also be mapped to an **edge logical position**, using the function f_{pe}:

$$f_{pe} : E \to P \times P,$$

$$f_{pe}(e) = f_{pe}((u, v)) = (f_{pv}(u), f_{pv}(v))$$

Each edge logical position can be mapped to a straight line drawing on the Euclidean Space with the function f_{se}:

$$f_{se} : P \times P \to \mathbb{R}^2 \times \mathbb{R}^2,$$

$$\begin{aligned} f_{se}(p_u, p_v) &= (f_{sv}(p_u), f_{sv}(p_v)) \\ &= (f_{sv}(f_{pv}(u)), f_{sv}(f_{pv}(v))) \\ &= ((x_1, y_1), (x_2, y_2)) \end{aligned}$$

Based on these functions, we define a **graph drawing** $d(g)$ as a mapping of the elements of a graph g to the Euclidean Space in the form:

$$d(g) = (f_{sv}(f_{pv}(V(g)), f_{se}(f_{pe}(E(g))))$$

According to Diehl [2], a **dynamic graph** is a sequence $G = [g^1, g^2, g^3, ..., g^n]$ of graphs with $G^i = (V^i, E^i)$. Given that a graph drawing is a mapping of the elements of g to the two-dimensional Euclidean Space we say:

- Let f^i_{pv} be the mapping of a vertex v of g^i to a vertex logical position.
- Let f^i_{sv} be the mapping of a vertex logical position of of g^i to the Euclidean space.
- Let f^i_{pe} be the mapping of an edge e of g^i to an edge logical position.

- Let f^i_{sv} be the mapping of an edge logical position of g^i to the Euclidean space.

Thus, a **dynamic graph drawing** is a successive sequence of drawings in the form:

$$D(G) = [d(g^1), d(g^2), d(g^3), ..., d(g^n)]$$

Each element $d(g^i)$ of $D(G)$ represents a mapping of the vertices $v \in V(g^i)$ and the edges $e \in E(g^i)$ to the Euclidean Space where:

$$d(g^i) = (f^i_{sv}(f^i_{pv}(V(g^i))), f^i_{se}(f^i_{pe}(E(g^i))))$$

A. Dimensions of the model

A dynamic graph drawing in the form $D(G) = [d(g^1), d(g^2), d(g^3)..., d(g^n)]$ is considered to be visually stable if the changes during a transition from $(d(g^i)$ to $d(g^{i+1})$ are minimal or null. The model of visual stability introduces nine dimensions to verify such condition.

1) Vertex set stability: The **vertex set stability** or **VS**, calculates the percentage of vertices from (g^i) that are present in g^{i+1} with the function:

$$VS = \frac{|V(g^i) \cap V(g^{i+1})|}{|V(g^i) \cup V(g^{i+1})|}$$

The values obtained for the **VS** are in a range from 0 to 1. Values closer to 0 suggest only a few vertices from g^i are present in g^{i+1}, while values closer to 1 indicate the opposite case.

2) Vertex set drawing stability: The **vertex set drawing stability** or **VDS**, calculates the percentage of **vertex logical positions** from $d(g^i)$ that are present in $d(g^{i+1})$ with the function:

$$VDS = \frac{|f^i_{pv}(V(g^i)) \cap f^{i+1}_{pv}(V(g^{i+1}))|}{|f^i_{pv}(V(g^i)) \cup f^{i+1}_{pv}(V(g^{i+1}))|}$$

The values obtained for the **VDS** are in a range from 0 to 1. Values closer to 0 suggest only a few vertex logical positions from $d(g^i)$ are present in $d(g^{i+1})$, while values closer to 1 indicate the opposite case.

3) Edge set stability: The **edge set stability** or **ES**, calculates the percentage of edges from g^i that are present in g^{i+1} with the function:

$$ES = \frac{|E(g^i) \cap E(g^{i+1})|}{|E(g^i) \cup E(g^{i+1})|}$$

The values obtained for the **ES** are in a range from 0 to 1. Values closer to 0 suggest only a few edges from g^i appear in g^{i+1}, while values closer to 1 indicate the opposite case.

4) Edge set drawing stability: The **edge set drawing stability** or **EDS**, calculates the percentage of **edge logical positions** from $d(g^i)$ that are present in $d(g^{i+1})$ with the function:

$$EDS = \frac{|f^i_{pe}(E(g^i)) \cap f^{i+1}_{pe}(E(g^{i+1}))|}{|f^i_{pe}(E(g^i)) \cup f^{i+1}_{pe}(E(g^{i+1}))|}$$

The values obtained for the **EDS** are in a range from 0 to 1. Values closer to 0 suggest only a few edge logical positions from $d(g^i)$ appear in $d(g^{i+1})$, while values closer to 1 indicate the opposite case.

5) Vertex set degree change: The **vertex set degree change** or **VDC**, calculates the variations on the degree centrality of those vertices from g^i that are present in g^{i+1} with the function:

$$VDC = \frac{\sum_{v \in V(g^i) \cap V(g^{i+1})} |\frac{C^i_d(v)}{|V(g^i)|} - \frac{C^{i+1}_d(v)}{|V(g^{i+1})|}|}{|V(g^i) \cap V(g^{i+1})|}$$

where C^i_d stands for the **degree centrality** [12] of a vertex v of g^i. The values for the **VDC** are in a range from 0 to 1. Values closer to 0 suggest a minimal change on the degree centrality for those nodes appearing in g^i and g^{i+1}, while values closer to 1 indicate the opposite case.

6) Vertex set drawing neighborhood change: The **vertex set drawing neighborhood change** or **VDNC**, calculates the variations on the number of connections for those vertex logical positions from $d(g^i)$ that are present in $d(g^{i+1})$ with the function:

- Let $PV^i = f^i_{pv}(V(g^i))$
- Let $PE^i = f^i_{pe}(E(g^i))$
- Let $V^* = PV^i \cap PV^{i+1}$

$$VDNC = \frac{\sum\limits_{v \in V^*} \left(|\sum\limits_{e \in PE^i} \frac{f_c(v,e)}{|PE^i|} - \sum\limits_{e' \in PE^{i+1}} \frac{f_c(v,e')}{|PE^{i+1}|}| \right)}{|V^*|}$$

where f_c is a function in the form

$$f_c : P \times (P \times P) \rightarrow [0, 1]$$

$$f_c : (p1, (p2, p3)) = \begin{cases} 1, & \text{if } p1 = p2 \\ 1, & \text{if } p1 = p3 \\ 0, & \text{otherwise} \end{cases}$$

The values obtained for the **VDNC** are between 0 and 1. Values closer to 0 suggest a small change on the number of connections for those vertex logical positions appearing in $d(g^i)$ and $d(g^{i+1})$, while values closer to 1 indicate the opposite case.

7) Vertex set drawing active positions: The **vertex set drawing active positions** or **VDAP**, calculates the percentage of **vertex logical positions** that are active in $d(g^i)$. The indicator was designed for those dynamic graph drawings using a global layout, rather than techniques computing a sequence of independent drawings. The **VDAP** operates in the following way:

$$VDAP = \frac{|f^i_{pv}(V(g^i))|}{|\cup^n_{i=1} f^i_{pv}(V(g^i))|}$$

In case the drawing technique computes a sequence of independent drawings, the **VDAP** is defined as:

$$VDAP = \frac{|f^i_{pv}(V(g^i))|}{|f^i_{pv}(V(g^i))|} = 1$$

The values obtained for the **VDAP** are in a range between 0 and 1. Values closer to 0 indicate a small number of vertex logical position are active in the current drawing, while values closer to 1 indicate the opposite case.

8) Edge set drawing active positions: The **edge set drawing active positions** or **EDAP**, calculates the percentage of **edge logical positions** that are active in $d(g^i)$. The indicator was designed for those dynamic graph drawings using a global layout, rather than techniques computing a sequence of independent drawings. The **EDAP** operates in the following way:

$$EDAP = \frac{|f_{pe}^i(E(g^i))|}{|\cup_{i=1}^n f_{pe}^i(E(g^i))|}$$

In case the drawing technique computes a sequence of independent drawings, the **EDAP** is defined as:

$$EDAP = \frac{|f_{pe}^i(E(g^i))|}{|f_{pe}^i(E(g^i))|} = 1$$

The values obtained for the **EDAP** are in a range between 0 and 1. Values closer to 0 indicate a small number of edge logical position are active in the current drawing, while values closer to 1 indicate the opposite case.

9) Graph drawing offset: The **graph drawing offset** or **GDO**, calculates the changes on the coordinates of those vertex logical positions that have been mapped to the Euclidean Space. The **GDO** is defined as: Let $f_{mv}^i = f_{sv}^i(f_{pv}^i(V(g^i))$ be the mapping of the vertices of $d(g^i))$ to the Euclidean space.

$$GDO = \sum\nolimits_{v \in V(g^i) \cap V(g^{i+1})} dist(f_{mv}(v)^i, f_{mv}(v)^{i+1})$$

where $dist(f_c(v), f_f(v))$ refers to the *euclidean distance* between two points. The values obtained for the **GDO** are higher or equal to 0. Lower values of **GDO** suggest minimal variations on the coordinates of those vertex logical positions present in in $d(g^i)$ and $d(g^{i+1})$, while higher values indicate drastic changes on the coordinates of these elements.

IV. Algorithms to Compute the Node Partition of a Foresighted Graph Layout

The basis of the algorithms we have developed to compute the node partition of a foresighted graph layout is the *lifetime matrix* or *LTM*, a structure containing all the information about the appearance of the actors in the dynamic social network. The LTM defined as follows:

- Let a dynamic graph/social network G be a sequence $G = [g^1, g^2, g^3, ..., g^n]$ of graphs with $G^i = (V^i, E^i)$.
- Let all the actors in the dynamic graph/social network be defined by the set $A = \{a | a \in \cup_{i=1}^n V(G^i)\}$

Therefore, the LTM is a matrix in the form:

$$\texttt{LTM} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & ... & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{24} & ... & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{34} & ... & a_{3n} \\ a_{41} & a_{42} & a_{43} & a_{44} & ... & a_{4n} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ a_{m1} & a_{m2} & a_{m3} & a_{m4} & ... & a_{mn} \end{bmatrix}$$

where $m = |A|$, $n = |G|$ and where a matrix element a_{ij} stands for a property of actor number *i* at time slice *j*. Such property differs for each algorithm we have developed. In addition, we refer to a single row of the LTM as *lifetime vector* or *LTV* in the description of our approaches.

A. Flickering reduction algorithm

One of the algorithms considers the nodes of a dynamic social network drawing as ”bulbs”, which are constantly alternating between the presence or absence of light. The change on the luminosity is given by the lifetime of the actors in the dynamic social network. Actors with longer lifetimes illuminate the bulb for longer periods of time, while actors with shorter lifetimes do the opposite case. However, the lifetime of the actors in a dynamic social network tends to be discontinuous which makes impossible to maintain the bulbs constantly illuminated. As consequence, the bulbs produce ”flickering” and affect the visual stability of the dynamic graph drawing.

The *flickering reduction algorithm* was developed to reduce as much as possible the ”flickering” from the dynamic graph drawing. The algorithm detects those actors with discontinuous lifetimes and uses them as the starting point to compute the node partition. In a first instance, we initialize all the positions a_{ij} of the LTM by placing the value of 1 to indicate the appearance of an actor or 0 to indicate his absence. The flickering reduction algorithm introduces two indicators to detect the actors with discontinuous lifetimes. The *flickering index* is defined as the number of times an actor changes from an appearance to a disappearance and vice versa over all the periods of time that form the dynamic social network. The *gap index* is defined as the number of inactive periods an actor presents over all the periods of time that form the dynamic social network. The selected index is used to build a *priority queue*, containing the ”flickering” produced by each actor ordered from highest value to lowest.

As next step, we proceed to compute the node partition. We select the actor whose index matches the highest value available in the priority queue. A new component is instantiated and the selected actor is placed in this location. At the moment, the component is affected by the ”flickering”. Therefore, we search for candidate actors who can be placed in the same component to reduce the ”flickering”. The main condition is that a candidate actor must have a disjoint lifetime with all the actors in the component. If the condition is satisfied, the candidate actor is placed in the component. Otherwise, the search process continues. In case it is not possible to find more candidates, the actors in the component are marked as ”processed”. These actors are not allowed to be selected again as candidates, since they have been used to reduce the ”flickering”. The process of building the components of the node partition is repeated until the last value of the priority queue has been computed.

B. Degree Stabilization Algorithm

The *degree stabilization algorithm* was developed to minimize the changes to the connections of the nodes in the drawing during the exploration of the dynamic social network. In a first instance, we initialize all the positions a_{ij} of the LTM by placing the *degree centrality* [12] of an actor to denote a presence or -1 to indicate an absence.

With the LTM initialized, we proceed to compute the new partition. An actor is selected arbitrarily and it is placed into a new component. The selected actor is used to attach two metrics to the component. On one hand, the *average degree* *centrality* stands for the average number of connections a node maintains during the exploration of the dynamic social network. This metric is calculated using the LTV of the actor in the component, but only considering the average value of those positions where the degree centrality is different from -1. On the other hand, the *average degree gradient* stands for the average number of connections a node loses during the exploration of the dynamic social network. This indicator is calculated using the LTV of the actor in the component, but only considering the average of the differences in the degree centrality for those positions different from -1.

As next step, we look for candidate actors to place in the same component. These actors must have a disjoint lifetime with all the actors in the component and must contribute to stabilize the connections of the nodes in the dynamic graph drawing. An candidate actor is considered to contribute if the absolute value of the difference between his average degree centrality and the average degree centrality of the component, is lower or equal than the average degree gradient of the component. If the restrictions are not satisfied, the search process continues. Otherwise, the candidate actor is placed in the component and the LTVs of the actors stored in this location are used to update the attached metrics. This last restriction makes the average degree gradient to decrease each time an actor enters the component. As result, the component becomes more restrictive and prevents too many actors to occupy the same position. In addition, the average degree centrality becomes more stable which minimizes the changes to the connections of the nodes in the dynamic graph drawing. In case we are not able to find more candidates, the actors in the component are marked as "processed" and are not allowed to be selected again as candidates. The process described above is repeated until all the actors in the dynamic social network have been assigned to a component.

V. METHOD OF STUDY

We have evaluated the dynamic graph algorithms produced by our algorithms with a study. We combined the use of questionnaires, an eye-tracking device and the model of visual stability. The questionnaires focused on the user perception and experience with the different drawings; the eye-tracking device recorded the eye movements of the participants, while the model measured the visual stability of the drawings. The research hypothesis was that a *a dynamic graph drawing with a moderate level of visual stability supports a satisfying user experience along with an efficient visual search and recognition of elements*.

A group of 20 students from the fields of applied cognitive science and computer science participated in our study. The participants had a basic knowledge regarding network analysis techniques, but were not familiar with theories of visual stability nor the mental map. The study presented three scenarios; each one illustrated a dynamic social network using the developed algorithms. *Scenario 1*, illustrated a dynamic social network with the *Flickering Reduction Algorithm* employing the *flickering index* (FRA-FI). *Scenario 2*, also illustrated a dynamic social network with the *Flickering Reduction Algorithm* but employing the *gap index* (FRA-GI). Lastly, *Scenario 3* made use of the *Degree Stabilization Algorithm* (DSA). We used circular drawings to study the same shape under different conditions. Additionally, the order of execution for the scenarios was assigned randomly to each participant. There was no time limit for the completion of the scenarios. However, the average duration per user for all three scenarios, was 30 minutes.

The participants of the study were asked to complete a series of tasks during which they had to track three actors appearing as nodes of a network graph. The dynamic social network was extracted from the developer mailing list of the Asterisk open source project and contained five snapshots; each one representing a month of communication between the developed (cf. [14] for a more detailed description of the data set). By the end of the tasks, the participants were requested to write a description of their experience with the different drawings and rate them using the questionnaires. The questions were rated with a Likert scale of five points, where the value of 1 indicated a strong disagreement; the value of 3 neither disagreement nor agreement and the value of 5 a strong agreement. The criteria under evaluation were divided in two categories. The positive criteria reflect better results with higher values on the Likert scale. On the contrary, the negative criteria reflected better results with lower values on the Likert scale. The criteria under evaluation are illustrated in Table I.

Criteria ID	Description
C1	The drawing technique uses a fixed coordinate system.
C2	The drawing technique is useful to track persons in a dynamic graph.
C3	The changes on the drawing are noticeable.
C4	The drawing technique requires some effort to locate a person.
C5	The addition or removal of entities from the drawing is distracting.
C6	The addition or removal of relations from the drawing is distracting.

TABLE I: Positive and negative items evaluated by the questionnaires.

The information recorded by the eye-tracking device was analyzed using several metrics (Table II). We proposed two settings for the analysis. The first setting applied the metrics to the overall drawing area. The second setting applied the metrics to the areas of interest (AOIs), i.e. the areas where the actors were located. The AOIs maintained a fixed position in all time slices for each drawing technique, because the algorithms do not change the positions of the nodes on the canvas. Moreover, we compared the eye-tracking metrics for the three scenarios and looked for statistical significance with a Wilcoxon test. This strategy allows us to determine which algorithm is more efficient in terms of visual search and recognition of elements.

VI. ANALYSIS AND RESULTS

A. The model of visual stability

The model of visual stability was computed for the flickering reduction algorithm using the flickering index (FRA-FI), the gap index (FRA-GI) and the degree stabilization algorithm (DSA). We kept the order of the transitions between the drawings, simulating the navigation through the dynamic social network. In addition, we calculated the average value for every dimension of the model to obtain the overall visual stability of the pictures observer by the participants. As it is illustrated in Table III, *FRA-FI* and *FRA-GI* presented the lowest values in

Eye-tracking metric	Description
Number of fixations	Less fixation imply a better recognition of elements [15]
Duration of fixations	A shorter duration implies a faster recognition of elements [15], [16]
Number of saccades	Less saccades imply an efficient search [15]
Saccadic duration	A shorter duration implies a faster search [15]
Saccadic amplitude	A shorter amplitude implies an efficient search [15]
Scan path length	A shorter length implies a more direct search [15]
Scan path duration	A shorter duration implies a faster search [15]
Spatial density	A lower density implies less scattered search patterns [15], [17]
Time to first fixation	A shorter time implies a faster location of an element for the first time [15], [18]
Fixations on target	Less fixations suggest low accuracy on the search [15]
Fixations/Saccade ratio	A higher ratio implies more time spent on the recognition and search of elements [15]

TABLE II: Eye-tracking metrics employed in our study

terms of active positions. Both approaches presented as active 41% (0.41) of the vertex logical positions and up to 14% (0.14) of the edge logical positions (EDAP), during the exploration of the dynamic social network. Contrarily, *DSA* presented as active 42% (0.42) of the vertex logical positions (VDAP) and up to 13% (0.13) of the edge logical positions (EDAP). The dynamic graph drawings did not changed the distribution of the actors on the canvas. Therefore, the graph drawing offset (GDO) was reduced to 0.

A more drastic change was noticed in the other dimensions of the model. The *FRA-FI* along with *FRA-GI* obtained a vertex set stability (VS) of 17% (0.17) and an edge set stability (ES) of 2% (0.02). Despite this fact, the dynamic graph drawings created by both strategies obtained a vertex set drawing stability (VDS) of 53% (0.53) and an edge set drawing stability (EDS) increased of 6% (0.06). On the other hand, the *DSA* obtained a vertex set stability (VS) of 17% (0.17) and an edge set stability (ES) of 2% (0.02). However, the dynamic graph drawing had a vertex set drawing stability (VDS) of 39% (0.39), while the edge set drawing stability (EDS) was of 4% (0.04). The three algorithms obtained a 41% (0.41) of vertex set degree change and vertex set drawing neighborhood change (VDNC) of 39% (0.39).

B. Questionnaires

The data collected from the questionnaires provided better insight about user experience in the three scenarios. We computed the average value of the user ratings per criterion, which ranged within [1, 5] with a mean value of 3. The average user rating for criteria 1 and 2 is illustrated in Figure 1. For these criteria, a rating of 5 is perceived as the best rating while a rating of 1 is perceived as the worst. The average user ratings for criteria 3 to 6 are illustrated in Figure 2. For these criteria, a rating of 5 is perceived as the worst result and a rating of 1 is perceived as the best.

In *Scenario 1*, we used the *FRA-FI* to illustrate the dynamic social network. The average duration of the scenario (including the completion of all three tasks) was *10:38* minutes. The participants noticed the actors of the dynamic social network always appeared in the same position (C1 = 4.15). Furthermore, the addition and removal of elements from the drawing

Dimensions of the model	FRA-FI	FRA-GI	DSA
VDAP vertex set drawing active positions	0.41	0.41	0.42
EDAP edge set drawing active positions	0.14	0.14	0.13
GDO graph drawing offset	0	0	0
VS vertex set stability	0.17	0.17	0.17
VDS vertex set drawing stability	0.53	0.53	0.39
ES edge set stability	0.02	0.02	0.02
EDS edge set drawing stability	0.06	0.06	0.04
VDC vertex set degree change	0.41	0.41	0.41
VDNC vertex set drawing neighborhood change	0.39	0.39	0.39

TABLE III: Average visual stability calculated for the three drawings under study.

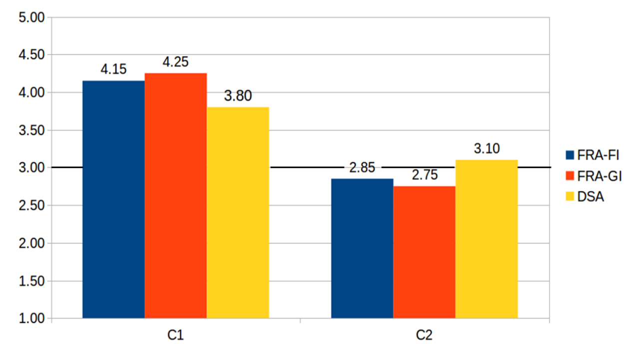

Fig. 1: Results of the user questionnaires for positive criteria (C1, C2)

was not distracting (C5 = 2.85, C6 = 2.50). Despite this fact, the participants were aware of the changes in the dynamic social network and required some effort to locate the requested actors (C3 = 3.40, C4 = 3.30). In general, the participants considered the dynamic graph drawing produced by the *FRA-FI* as not useful for tracking actors in a dynamic social network (C2 = 2.85). Regarding the description about the experience of the participants, the minimal number of changes to the structure of the drawing was received positively but observing multiple actors to occupy the same position was confusing.

In *Scenario 2*, we used the *FRA-GI* to illustrate the dynamic social network, The average duration for the completion of all tasks was *09:06* minutes. The participants were able to recognize that the actors of the dynamic social network always appeared in the same position (C1 = 4.25). Furthermore, the addition and removal of elements from the drawing was not distracting (C5 = 2.95, C6 = 2.20). Despite this fact, the participants were aware of the changes in the dynamic social network and required some effort to locate the requested actors (C3 = 3.60, C4 = 3.30). In general, the participants considered the dynamic graph drawing produced by the *FRA-GI* as not useful for tracking actors in a dynamic social network (C2 = 2.75). Regarding the description about the experience of the participants, once again the minimal changes to the structure of

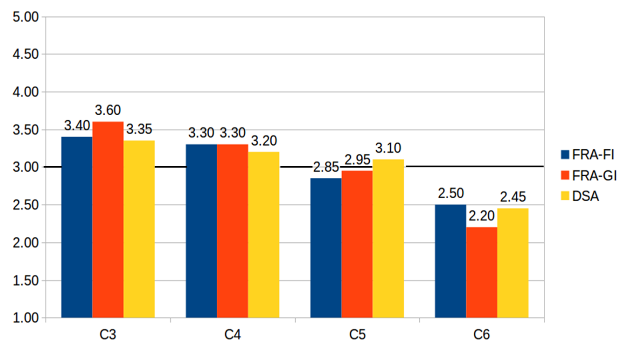

Fig. 2: Results of the user questionnaires for negative criteria (C3-C6)

the drawing was received positively. Still, observing multiple actors to in the same position was confusing.

In *Scenario 3*, we used the *DSA* to illustrate the dynamic social network. The average duration for the completion of the tasks was *8:55* minutes. Once again, the participants noticed that the actors of the dynamic social network always appeared in the same position (C1 = 3.80). Furthermore, the process to locate the requested actors required less effort (C4 = 3.20) despite the addition and removal of elements from the drawing produced some distraction (C3 = 3.35, C5 = 3.10). In general, the participants considered the drawing produced by the *DSA* useful to track actors in a dynamic social network (C2 = 3.10). Regarding the description about the experience of the participants, the changes on the structure of the drawing were noticeable and introduced some distraction. Yet, observing fewer actors to appear in the same position was less confusing for the participants, who executed the tasks without problems.

C. *Eye-tracking metrics for the overall drawing area*

The data collected from the eye-tracking device provided additional information about the visual search and the recognition of elements. As it is illustrated in Table IV, only three of the eye-tracking metrics presented statistical significance on the overall drawing area. The number of saccades presented its lowest value in the drawing produced by the *FRA-GI* (OSAC1 = 58.250), followed by the *DSA* (OSAC1 = 66.190) and the *FRA-FI* (OSAC1 = 70.730). We found the number of saccades of the *FRA-GI* to be statistically significant in comparison to the *FRA-FI* ($[Z = -1.904, p = .057]$). This suggests the drawing produced by the *FRA-GI* makes the user to perform less eye movements during the visual search.

On the contrary, the shortest saccadic amplitude was observed in the drawing produced by the *FRA-FI* (OSAC2 =136.566). The approach was only followed by the *DSA* (OSAC2 = 157.754), since the *FRA-GI* obtained the longest saccadic amplitude (OSAC2 = 168.012). The values obtained for the *FRA-FI* were found to be statistically significant in comparison to the *DSA* ($[Z = -2.501, p = .012]$) along with the *FRA-GI* ($[Z = -1.680, p = .093]$). This suggests the *FRA-FI* offers the shortest distance between eye movements during the visual search.

On the other hand, the lowest spatial density was observed in the drawing produced by the *FRA-GI* (OSD1 = 0.357). The approach was followed by the *FRA-FI* (OSD1 = 0.374) and the *DSA* (OSAC2 = 0.416). We found the spatial density of the *DSA* to be statistically significant in comparison to the *FRA-GI* ($[Z = -2.671, p = .008]$) along with the *FRA-FI* ($[Z = -2.336, p = .019]$). This suggests *FRA-FI* together with the *FRA-GI* offer less scattered search patterns.

Eye-tracking metric	FRA-FI	FRA-GI	DSA
OSAC1 Number of saccades	70.730	58.250	66.190
OSAC2 Saccadic amplitude in pixels	136.566	168.012	157.754
OSD1 Spatial Density	0.374	0.357	0.416

TABLE IV: Average value of the eye-tracking metrics for the overall drawing area

D. *Eye-tracking metrics for the areas of interest*

As it is illustrated in Table V, there was more activity occuring on the areas of interest (AOIs). The lowest number of fixations was noticed in the *FRA-GI* (AFIX1 = 1.740), followed by the *FRA-FI* (AFIX1 = 3.783) and the *DSA* (AFIX1 = 4.796). This suggest the *FRA-GI* requires less fixations to recognize an element. The shortest duration fixations was observed in the *FRA-GI* (AFIX2 = 0.094), followed by the *FRA-FI* (AFIX2 = 0.136) together with the *DSA* (AFIX2 = 0.148). This finding suggest the *FRA-GI* requires less time to recognize an element.

The lowest number of saccades was observed in the *FRA-GI* (ASAC1 = 0.646). The approach was followed by the *FRA-FI* (ASAC1 = 1.80) along with the *DSA* (ASAC1 = 2.546). This indicates the *FRA-GI* needs less eye movements to lock on a specific element. The shortest duration of saccades was observed in the *FRA-GI* (ASAC2 = 0.002) and the approach was followed by the *FRA-FI* (ASAC2 = 0.010) together with the *DSA* (ASAC2 = 0.012). This indicates the *FRA-GI* needs less time to lock on a specific element. The shortest saccadic amplitude was observed in the *FRA-GI* (ASAC3 = 0.350) and the approach was followed by the *FRA-FI* (ASAC3 = 1.233) along with the *DSA* (ASAC3 = 1.589). This finding indicate the *FRA-GI* covers less distance between eye movements to lock on a specific element.

The lowest number of fixations on target was detected in the *FRA-GI* (AFOT1 = 0.027). The approach was followed by the *FRA-FI* (AFOT1 = 0.053) along with the *DSA* (AFOT1 = 0.067). This findings suggest the *FRA-GI* together with the *FRA-FI* do not have a good accuracy on the search in comparison to the *DSA*. The shortest time to first fixation was detected with the *FRA-GI* (ATFF1 = 1.273) followed by the *DSA* (ATFF1 = 2.184) and the *FRA-FI* (ATFF1 = 2.566). The results suggest the *FRA-FI* together with the *DSA* offer the fastest time to lock on an element for the first time.

VII. Discussion

According to the model of visual stability, the drawings produced by the *FRA-FI* along with the *FRA-GI* have a minimal number of changes on their structures during the exploration of the dynamic social network. This characteristic

Eye-tracking metric	FRA-FI	FRA-GI	DSA
AFIX1 Number of fixations per AOI	3.783	1.740	4.796
AFIX2 Duration of fixations per AOI	0.136	0.094	0.148
ASAC1 Number of saccades per AOI	1.80	0.646	2.546
ASAC2 Duration of saccades per AOI in seconds	0.010	0.002	0.012
ASAC3 Saccadic amplitude per AOI in pixels	1.233	0.350	1.589
AFOT1 Fixations on target	0.053	0.027	0.067
ATFF1 Time to first fixation in seconds	2.566	1.273	2.184

TABLE V: Average value of the eye-tracking metrics for the areas of interest

allowed the participants of the study to track the requested actors without major distractions. Nonetheless, it was reported for both algorithms that allowing several actors to occupy the same position is confusing and the search process requires more effort. Certainly, the *FRA-FI* and the *FRA-GI* offer the best performance in terms of the recognition of elements along with less disperse search patterns. Despite this fact, the accuracy on the visual search has its worst performance with these algorithms. This suggests that the users identify faster an element which are actually not looking for. Therefore, we consider the drawing produced by the flickering reduction algorithm not suitable for tracking actors in a dynamic social network.

In contrast to the other approaches, the drawing produced by the *DSA* presents more noticeable changes on its structure during the exploration of the dynamic social network. Such characteristic introduced some distraction as the participants tracked the requested actors. Nevertheless, it was reported that the drawing was less confusing because fewer actors were appearing in the same position. In addition, the search process required less effort. The *DSA* presented a good performance regarding the eye-tracking metrics and obtained the best accuracy on the visual search. These finding suggest the users are able to identify the desired element. Thus, we consider the drawing produced by the *DSA* supports a satisfying user experience along with an efficient visual search and recognition of elements.

VIII. Conclusions and future work

In this paper, we presented algorithms to compute the node partition of a foresighted graph layout. Our approaches aim to produce a dynamic graph drawing that allows the users to track actors in a dynamic social network, supporting a satisfying user experience along with an efficient visual search and recognition of elements. The results suggest dynamic graph drawings achieving visual stability by placing multiple actors in the same position, produce less distraction but are confusing and reduce the efficiency of the visual search. On the contrary, using a restriction to prevent to many actors to occupy the same position, produces a less confusing dynamic graph drawing and improves the efficiency of the visual search. As future work, we plan to visualize the evolution of cohesive subgroups using the degree stabilization algorithm.

References

[1] S. Bender-deMoll and D. A. McFarland, “The art and science of dynamic network visualization,” *Journal of Social Structure*, vol. 7, 2006.

[2] S. Diehl, C. Görg, and A. Kerren, “Preserving the mental map using foresighted layout,” in *Data Visualization 2001*, ser. Eurographics, D. Ebert, J. Favre, and R. Peikert, Eds. Springer Vienna, 2001, pp. 175–184.

[3] H. C. Purchase, E. Hoggan, and C. Görg, “How important is the “mental map”?–an empirical investigation of a dynamic graph layout algorithm,” in *Graph drawing*. Springer, 2007, pp. 184–195.

[4] P. Eades, W. Lai, K. Misue, and K. Sugiyama, *Preserving the mental map of a diagram*. International Institute for Advanced Study of Social Information Science, Fujitsu Limited, 1991.

[5] D. Archambault and H. Purchase, “Mental map preservation helps user orientation in dynamic graphs,” in *Graph Drawing*, ser. Lecture Notes in Computer Science, W. Didimo and M. Patrignani, Eds. Springer Berlin Heidelberg, 2013, vol. 7704, pp. 475–486.

[6] J. M. Six and I. G. Tollis, “A framework for circular drawings of networks,” in *Graph Drawing*. Springer, 1999, pp. 107–116.

[7] C. Erten, P. J. Harding, S. G. Kobourov, K. Wampler, and G. Yee, “Graphael: Graph animations with evolving layouts,” in *Graph Drawing*. Springer, 2004, pp. 98–110.

[8] A. Ramos, I.-A. Chounta, T. Göhnert, and U. Hoppe, “Exploring visual stability in dynamic graph drawings: A case study,” Submitted to the International Conference on Advances in Social Networks Analysis and Mining (ASONAM) 2015.

[9] G. W. Mcconkie and C. B. Currie, “Visual stability across saccades while viewing complex pictures,” *Journal of Experimental Psychology: Human Perception and Performance*, vol. 22, no. 3, pp. 563–581, 1996.

[10] C. Colby, M. Goldberg *et al.*, “The updating of the representation of visual space in parietal cortex by intended eye movements,” *Science*, vol. 255, no. 5040, pp. 90–92, 1992.

[11] D. Melcher and C. L. Colby, “Trans-saccadic perception,” *Trends in cognitive sciences*, vol. 12, no. 12, pp. 466–473, 2008.

[12] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*. Cambridge university press, 1994, vol. 8.

[13] R. Tamassia, *Handbook of Graph Drawing and Visualization*. CRC Press, 2007.

[14] S. Zeini, T. Göhnert, T. Hecking, L. Krempel, and H. U. Hoppe, “The impact of measurement time on subgroup detection in online communities,” in *State of the Art Applications of Social Network Analysis*. Springer, 2014, pp. 249–268.

[15] J. H. Goldberg and X. P. Kotval, “Computer interface evaluation using eye movements: methods and constructs,” *International Journal of Industrial Ergonomics*, vol. 24, no. 6, pp. 631–645, 1999.

[16] M. A. Just and P. A. Carpenter, “Eye fixations and cognitive processes,” *Cognitive Psychology*, vol. 8, no. 4, pp. 441–480, 1976.

[17] L. Cowen, L. J. Ball, and J. Delin, “An eye movement analysis of web page usability,” in *People and Computers XVI-Memorable Yet Invisible*. Springer, 2002, pp. 317–335.

[18] M. D. Byrne, J. R. Anderson, S. Douglass, and M. Matessa, “Eye tracking the visual search of click-down menus,” in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 1999, pp. 402–409.