Graph Drawing with Morphing Partial Edges

Kazuo Misue[✉] and Katsuya Akasaka

University of Tsukuba, Tsukuba, Japan
`misue@cs.tsukuba.ac.jp`, `akasaka@vislab.cs.tsukuba.ac.jp`

Abstract. A partial edge drawing (PED) of a graph is a variation of a node-link diagram. PED draws a link, which is a partial visual representation of an edge, and reduces visual clutter of the node-link diagram. However, more time is required to read a PED to infer undrawn parts. The authors propose a morphing edge drawing (MED), which is a PED that changes with time. In MED, links morph between partial and complete drawings; thus, a reduced load for estimation of undrawn parts in a PED is expected. Herein, a formalization of MED is shown based on a formalization of PED. Then, requirements for the scheduling of morphing are specified. The requirements inhibit morphing from crossing and shorten the overall time for morphing the edges. Moreover, an algorithm for a scheduling method implemented by the authors is illustrated and the effectiveness of PED from a reading time viewpoint is shown through an experimental evaluation.

Keywords: Graph drawing · Partial edge drawing · Morphing edge

1 Introduction

The partial edge drawing (PED) of a graph is a variation of a node-link diagram that is a visual representation of a graph. In PED, a link is drawn, which is a partial visual representation of an edge; that is, a part of the link is omitted, and then intersections of links are eliminated. Therefore, PED can reduce visual clutter of node-link diagrams. An experimental evaluation by Bruckdorfer et al. shows that PEDs reduces errors and provides higher accuracy when reading graphs than traditional node-link diagrams; however, longer reading time is required [6].

We propose a morphing edge drawing (MED), which is a PED that changes with time. In MED, links are morphed between partial and complete drawings. Therefore, reduced loads are expected to infer undrawn parts in a PED. However, the effect depends on the scheduling of morphing edges. We designed a scheduling algorithm that did not unnecessarily cause links to cross. Then, we performed a user study to evaluate the effectiveness of MEDs by implementation. The contributions herein are as follows: Proposal and formalization of MED. Setting scheduling requirements. Proposal of algorithm for scheduling morphing. Evaluation of MED via user study.

2   Partial Edge Drawing

Let $G = (V, E)$ be a simple undirected graph and let $\Gamma(G) = (\Gamma(V), \Gamma(E))$ be a drawing of G, where $\Gamma(V) = \{\Gamma(v)|v \in V\}$ and $\Gamma(E) = \{\Gamma(e)|e \in E\}$. Let $\Gamma(G)$ be a traditional straight-line drawing. Let the drawing $\Gamma(v)$ of a node $v \in V$ be a small disk placed at a position p_v and let $\Gamma(e)$ of an edge $e \in E$ be a straight-line segment between two nodes (disks) incident to the edge. That is, $\Gamma(e) = \{s \cdot p_w + (1 - s) \cdot p_v | s \in [0, 1]\}$ when $e = (v, w)$. We call $\Gamma(G)$ a *complete edge drawing (CED)* because it draws every straight-line representing an edge completely.

We express the partial drawing of an edge $e = (v, w)$ as a function $\gamma_e : [0, 1]^2 \to 2^{\Gamma(e)}$ shown in Exp. (1).

$$\gamma_e(\alpha, \beta) = \begin{cases} \{s \cdot p_w + (1 - s) \cdot p_v | s \in [0, \alpha] \cup [\beta, 1]\} & \text{for } \alpha < \beta \\ \Gamma(e) & \text{for } \alpha \geq \beta \end{cases} \quad (1)$$

That is, $\gamma_e(\alpha, \beta)$ of edge e comprise the parts that remain after removing the corresponding parts (α, β) from $\Gamma(e)$ when the entire $\Gamma(e)$ corresponds to the interval $[0, 1]$. Each of the remaining continuous parts is called a *stub*. The parameters α and β, which determine the stub lengths, are *partial edge parameters*. When $0 < \alpha$ and $\beta < 1$ for γ_e, the part to be deleted is not the end of $\Gamma(e)$; two stubs remain at the two nodes incident to the edge e. These are called a *pair of stubs*.

Drawing $\Gamma_{PED}(G) = (\Gamma(V), \Gamma_{PED}(E))$ is a *partial edge drawing (PED)* if for all edges $e \in E$, α_e and β_e are given, and at least an edge $e_1 \in E$ exists with $\alpha_{e_1} < \beta_{e_1}$, where $\Gamma_{PED}(E) = \{\gamma_e(\alpha_e, \beta_e)|e \in E\}$. When $\alpha_e = 1 - \beta_e$, i.e., the lengths of a pair of stubs are the same, the drawing is a *symmetric PED (SPED)*. The smaller parameter α_e is the *stub-edge ratio*. If the stub-ratios for all edges are the same δ, the drawing is a δ-*symmetric homogeneous PED (δ-SHPED)*.

Herein, we assume that $\Gamma(G)$ is given in advance and stubs may have intersections.

3   Related Work

Becker et al. conceived a drawing concept in which only half the links are used to reduce the visual clutter during the development of a tool called SeeNet [2]. Parallel Tagcloud, developed by Collins et al., adopts a method similar to PED [9]. Although Parallel Tagcloud is an extension of Tag Cloud, it can be regarded as a hierarchical layout of directed graphs; thus, it is useful against visual clutter caused by crossing of links. They can avoid drawing intersections by representing links as straight lines without drawing in the middle of the links.

Bruckdorfer et al. formalized the PED in their study [5]; our formalization of PED in Sect. 2 is a modified version of their formalization. They added continuity of the omitted parts of links as a condition, we have incorporated this into the formalization herein. Moreover, although they focused on a layout without

crossing stubs in PED, this study allows stub crossings. Burch et al. applied PED to directed graphs using tapered links [8]. Schmauder et al. applied PED to weighted graphs by representing weights with edge colors [14].

Bruckdorfer et al. performed a comparison between CED[1] and 1/4-SHPED on reading performance of graphs [6]. Although the statistical significance was not shown, from their chart that visualizes the experimental results, in the task of reading graphs (for adjacency check of two nodes or search for adjacent nodes), we can guess that 1/4-SHPED is slightly more accurate than CED; however, the response time of 1/4-SHPED is longer. Binucci et al. conducted a more detailed evaluation to reveal that SHPED has high accuracy in reading graphs within SPED [3]. Burch examined the effect of stub orientation and length on graph reading accuracy [7].

Blass et al. avoided using arrows to facilitate grasping high-dimensional transitions in the state transition diagram and proposed moving the dashed pattern with animation [4]. Holten et al. compared the recognition accuracy of graphs in various edge drawing methods, such as tapered links and curved links, including animation [11]. They showed that the recognition accuracy of the graph is high by representation using animation. Romat et al. attempted to extend the design space using animation of edge textures [13]. The proposal herein can be considered as an application of animation to graph drawing, especially to drawing edges. However, the purpose is not to express the orientation, but to improve the reading accuracy and efficiency for graphs.

4   Morphing Edge Drawing

Let T be a set of times. Then, function $\mu_e : T \to 2^{\Gamma(e)}$, which determines a partial drawing of edge e for time $t \in T$, a *morphing function*. A dynamic drawing $\Gamma_{MED}(G) = (\Gamma(V), \Gamma_{MED}(E))$ of graph G with morphing functions is a *morphing edge drawing (MED)*, where $\Gamma_{MED}(E) = \{\mu_e | e \in E\}$ is a set of morphing functions.

Then, a function $\rho_e : T \to [0,1]^2$, which determines the partial edge parameters for a time $t \in T$, is a *ratio function*. The morphing function μ_e can be constructed as $\mu_e(t) = \gamma_e(\rho_e(t))$ using the ratio function.

4.1   Symmetric MED

When all ρ_e for all $e \in E$ satisfies $\rho_e(t) = (\delta_t, 1 - \delta_t)$ $(0 \le \delta_t \le 1/2)$ for all $t \in T$, we get SPED at any time. Thus, such a ratio function is a *symmetric ratio function*; furthermore, if a MED is composed of symmetric ratio functions, it is referred to as a *symmetric MED (SMED)*. As the two values obtained by a symmetric ratio function depend on each other, we can define the function as $\rho_e : T \to [0, 1/2]$ without ambiguity.

Morphing of edge e extending from stub-edge ratio δ_e to η_e and then shrinking to δ_e is expressed by a symmetric ratio function ρ_e, expressed as Exp. (2), where

---

[1] They call it the traditional straight-line model (TRA).

t_0 is the start time of the morphing, l is the length of $\Gamma(e)$, and s is the speed of the tips of the stubs (*morphing speed*). Here, let the morphing speed be constant. Figure 1 shows the graph of the function.

$$\rho_e(t) = \begin{cases} \delta_e & \text{for } t \leq t_0 \text{ or } t_2 < t \\ \delta_e + (t - t_0)s/l & \text{for } t_0 < t \leq t_1 \\ \eta_e - (t - t_1)s/l & \text{for } t_1 < t \leq t_2, \end{cases} \tag{2}$$

where t_1 is the time when the stub-edge ratio becomes η_e, and t_2 is the time when the stub-edge ratio returns to δ_e. Using one-way travel time $d_1 = (\eta_e - \delta_e)l/s$, $t_1 = t_0 + d_1$ and $t_2 = t_0 + 2d_1$.
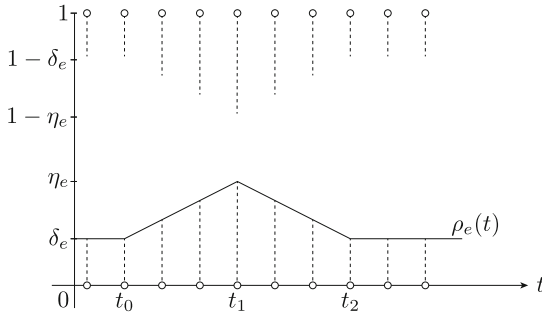

Fig. 1. Graph of ρ_e. Each pair (top and bottom) of dashed lines represent a pair of stubs expanding and contracting.

If the ratio functions have the same δ and η for all edges, the drawing is a (δ, η)-*symmetric homogeneous MED* $((\delta, \eta)$-*SHMED)*. Note, this homogeneity does not always mean synchronicity. Drawings by (δ, η)-SHMED may not be SHPED at any time.

When $\eta = 1/2$, we omit η, like δ-*symmetric homogeneous MED* (δ-*SHMED)*. In the drawing by δ-SHMED, a pair of stubs of edge e becomes $\Gamma(e)$ at a certain moment. Intuitively, the drawing by δ-SHMED changes between δ-SHPED and CED. However, a moment when it becomes CED does not always exist.

5   Scheduling of Morphing

We have set requirements to design the scheduling of morphing of all edges as follows:

R1: Morphing Does Not Make Crossings. To maintain the reading accuracy, visual clutter should be minimized. Therefore, morphing should not result in new crossings among stubs. However, if another edge exists that crosses a stub with a stub-edge ratio δ, then the crossing is inevitable. As mentioned earlier,

rearrangements to avoid such crossings are beyond the scope of this paper. The requirement is to avoid crossings in the center area undrawn by δ-SHPED, (we refer to these areas as *blank areas*).

R2: Shorten Morphing Time for All Edges. The time taken for a viewer to focus on a stub should be minimized before morphing of the stub. We do not know in advance which stubs the viewer will focus on. Therefore, it is necessary to repeat morphing of all edges, and it is necessary to shorten the total morphing time of all edges.

5.1   Morphing Group

First, two non-crossing edges do not generate new crossings of stubs at any timing when morphing, i.e., they can morph simultaneously and independently. However, two edges that do not intersect may not be able to morph independently, depending on the relationship with other edges (see Fig. 2).
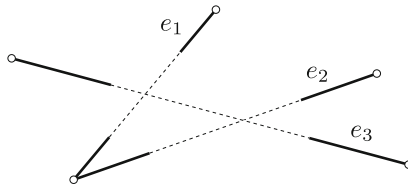

Fig. 2. Dependency between edges. The edges e_1 and e_2 do not intersect, but both intersect e_3, so they cannot be morphed independently. Dotted lines represent the omitted parts (blank areas).

A set of edges where the timing of morphing may affect each other is a *morphing group*. Different morphing groups can be scheduled independently. To determine the morphing groups, another graph is generated from the graph to be drawn. To avoid ambiguity, we will express the newly generated graph and its components as ⟨graph⟩, ⟨node⟩, and ⟨edge⟩. Let each edge be a ⟨node⟩. Suppose that there is an ⟨edge⟩ between ⟨nodes⟩ (i.e., edges) that intersect each other in a blank area. ⟨Nodes⟩ (i.e., edges) included in each connected component of the ⟨graph⟩ generated in this manner constitute a morphing group.

As two edges belonging to different morphing groups do not intersect, it is possible to schedule them independently in units of morphing groups. Hereafter, we describe scheduling of morphing of edges included in a morphing group.

5.2   Sequential Morphing

To prevent new stub crossings from being generated by morphing of two edges intersecting in blank areas, entry into a blank area should be exclusive. That

is, the safest scheduling, satisfying requirement R1 (morphing does not result in crossings), is to perform edge morphing sequentially. However, with such simple scheduling, R2 cannot be satisfied.

5.3   Packing Morphing Intervals

Assume there are two intersecting edges e_1 and e_2, as shown in Fig. 3. If the scheduling is such that when a stub of edge e_1 has been stretched and contracted to the crossing point, a stub of e_2 extends to the crossing point, then no intersection will occur. Let the time it takes for the stub of e_1 to contract to the crossing point after it starts morphing be t_1 and the time it takes for the stub of e_2 to start morphing and then extend to the crossing point be t_2. If the stub of edge e_2 starts morphing $t_1 - t_2$ after the stub of e_1 starts morphing, no crossing occurs. In addition, when morphing is repeated alternately, e_1 will start morphing again next to e_2.
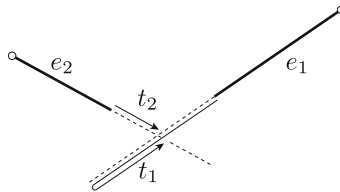

Fig. 3. Packing morphing intervals.

5.4   Parallel Morphing

Even if edges belong to the same morphing group, morphing of two non-intersecting edges may be performed simultaneously. By appropriately morphing parallelly, it is possible to shorten the total morphing time of all edges. For example, as edges e_1 and e_2 in Fig. 2 are not intersecting, their morphing can be parallelized if we can find adequate timing to avoid crossing with e_3. This can reduce the overall time.

5.5   Algorithm for Finding Morphing Start Time

In scheduling morphing, we decided to determine the start time from longer edges. Assuming that every stub has the same morphing speed, the longer the edge, the longer one cycle of morphing takes. By determining the start time from longer edges, while a long edge is morphing, the morphing of short edges that do not intersect with it can be embedded in the same time range.

Given a morphing group E (a set of edges), Algorithm 1 determines the morphing start time $t_s(e)$ for all edges $e \in E$. The algorithm determines the

morphing start time in descending order of edge length. It checks the timing of every morphing stub of all edges in $C(e)$ that intersect with edge e, and allows the morphing start time of edge e to be the earliest time that does not result in intersection with the morphing stubs.

The $r_1(e, c)$ and $r_2(e, c)$ appearing in the algorithm represents the first and last time of the time range, respectively, when the start of the morphing of edge e is prohibited to avoid crossing with edge c. They are described as Exp. (3) and Exp. (4).

$$r_1(e, c) = t_s(c) + t_p(c, e) - t_r(e, c) \tag{3}$$
$$r_2(e, c) = t_s(c) + t_r(c, e) - t_p(e, c), \tag{4}$$

where $t_p(e, c)$ is the time it takes from the start of morphing of the stub of edge e to the first passing (passing while stretching) at the crossing point with edge c (cf. $t_p(e_2, e_1) = t_2$ in Fig. 3), and $t_r(e, c)$ is the time it takes from the start of morphing to the second passing (passing while shrinking) at the crossing point (cf. $t_r(e_1, e_2) = t_1$ in Fig. 3).

Algorithm 1. Determining the start time of morphing

Input: E — Set of edges included in a morphing group
Output: Start time is determined for all edges of E
1: **function** FINDSTARTTIME(E)
2:     **for** e in $sortByLength(E)$ **do**
3:         $I \leftarrow \{(r_1(e, c), r_2(e, c)) | c \in C(e) \wedge t_s(c)$ is defined.$\}$
4:         $t_s(e) \leftarrow earliestSpace(I)$
5:     **end for**
6: **end function**

Function $earliestSpace(I)$ yields the smallest value not included in the time ranges (intervals) in a given set I. If each pair (r_1, r_2) included in the set I is regarded as an interval $[r_1, r_2)$ of real numbers, function $earliestSpace(I)$ is defined as Exp. (5). We calculate $earliestSpace(I)$ using Algorithm 2. Note that T is a nonnegative real number in Algorithm 2.

$$earliestSpace(I) = \min \left\{ \left(\bigcup_{r \in I} r \right)^c \right\} \tag{5}$$

6 Evaluation Experiment

To investigate the effectiveness of MED, we conducted a comparative experiment with three types of visual representations: CED, 1/4-SHPED, and 1/4-SHMED.

Algorithm 2. Finding time when morphing can start

Input: I — Set of time ranges (pairs of times) during which morphing should not start

Output: Earliest time morphing can start

```
 1: function EARLIESTSPACE(I)
 2:     t ← 0
 3:     for (r₁, r₂) in sortByStartTime(I) do
 4:         if r₂ < t then
 5:             continue
 6:         else if t < r₁ then
 7:             return t
 8:         else
 9:             t ← r₂
10:         end if
11:     end for
12:     return t
13: end function
```

6.1 Hypothesis

We made the following hypothesis.

H1 1/4-SHMED requires less time to read a graph than 1/4-SHPED
H2 1/4-SHMED is more accurate at reading graphs than CED.

6.2 Tasks

We designed the following tasks to test the above hypotheses.

T1 Check if the two highlighted nodes are adjacent (connected by an edge).
T2 Select all the nodes to which the highlighted node is adjacent.

For T1, as shown in Fig. 4, a graph in which two nodes are highlighted is displayed. Participants respond by pressing "Y" or "N" on the keyboard. When creating sample graphs, the number of crossings of the edges connecting two nodes were set to 8 or 16 when the nodes were adjacent.

For T2, as shown in Fig. 5, a graph in which one node is highlighted is displayed. Node selection is performed using a trackpad. When clicked, the pointed node is selected and turns orange. Participants can also cancel the selection by clicking again. Answers are confirmed by pressing the Enter key. When creating sample graphs, we selected nodes to be highlighted such that the number of adjacent nodes to it were 3, 6, and 9. Furthermore, we set the average number of intersections of the edges of interest to be within 7.9–8.1 and the average of lengths of the edges to be in the range of 3.3–3.7 cm on the screen to ensure that the task difficulty was not excessively low or high. In this experiment, we assumed that the distance between the participant's eyes and the screen was 40 cm.
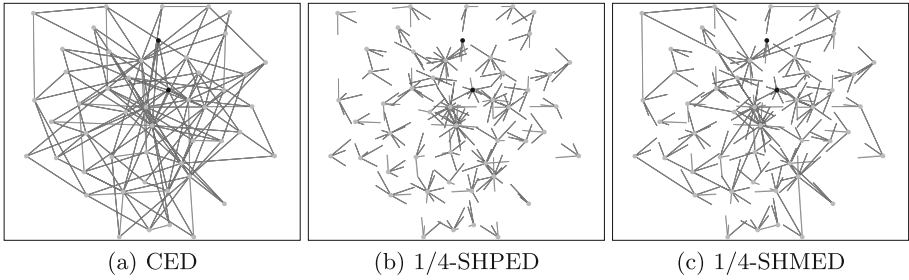
(a) CED   (b) 1/4-SHPED   (c) 1/4-SHMED

Fig. 4. Examples of visual representations used in T1

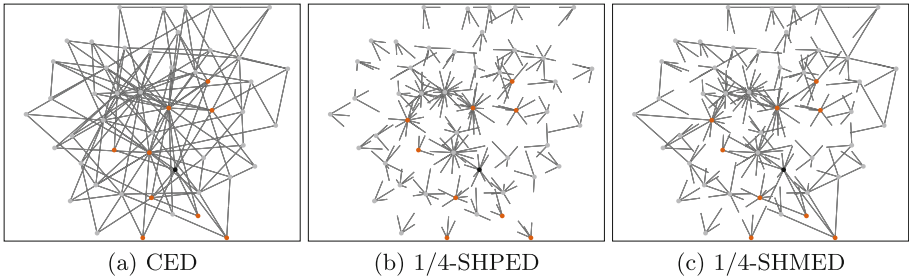(a) CED   (b) 1/4-SHPED   (c) 1/4-SHMED

Fig. 5. Examples of visual representations used in T2

6.3   Graphs Used for the Experiment

We used the Barabási-Albert model [1] as a guideline to create a graph with 50 nodes and 144 edges. We used the Fruchterman-Reingold algorithm [10] to determine the layout of the graph.

6.4   Morphing Speed

We set the morphing speed of each stub as $10°/s$. If morphing is too fast, the human eye cannot track it. Conversely, if it is too slow, reading efficiency is reduced. We derived the speed based on Robinson's experiment [12] such that it is human eye-trackable while being as fast as possible. However, we set a minimum one-way travel time 300 ms to make capturing morphing stubs easy.

6.5   Experimental Settings

We used a MacBook Pro 2017 (screen size 13.3 inches, screen resolution 1440×900) for the experiment. We set the display area to 1000×800 so the graph can be viewed without scrolling.

The participants in this experiment were 12 students (4 university students and 8 graduate students).

6.6   Experimental Procedure

The following procedure was used to conduct the experiment:

1 Overall explanation
2 Visual representation #1
    2-1 T1 practice (one question)
    2-2 T1 actual (nine questions)
    2-3 T2 practice (one question)
    2-4 T2 actual (nine questions)
    2-5 Questionnaire for visual representation #1
3 Visual representation #2 (flow similar to visual representation #1)
4 Visual representation #3 (flow similar to visual representation #1)
5 Questionnaire for whole experiment

We varied the order of presenting visual representations from each participant to eliminate the effects of order. Therefore, visual representations #1, #2, and #3 differ depending on the participant. We assigned two participants for each of the six (= 3!) orders.

6.7   Response Time

Figure 6 shows the distribution (boxplots) of response time (in millisecond) for each task and representation method. In both tasks, the average response time was the lowest for CED and highest for 1/4-SHPED. As the 1/4-SHMED is located in the middle, an improvement in the reading time for 1/4-SHPED can be expected. From the Shapiro-Wilk test ($\alpha = 0.05$), the time taken either task did not follow a normal distribution. Therefore, we performed multiple tests using the Friedman and Holm methods. Tables 1 and 2 show the test results for the response time for T1 and T2, respectively. As shown in Table 1, a significant difference was observed between the representation methods, i.e., 1/4-SHMED can shorten the time taken to confirm the adjacency between nodes, compared to 1/4-SHPED (H1). In contrast, no significant difference was found between the representation methods with respect to the response time of T2.

Table 1. Test result of response time of T1

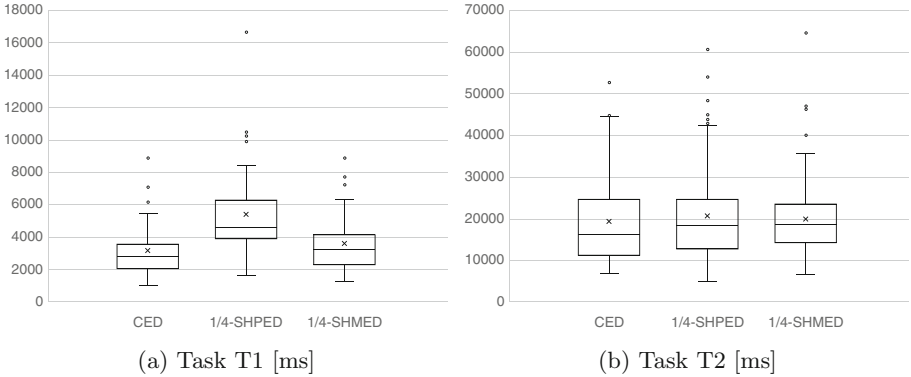| Comparison	Test result (p value)	Significance level
CED vs 1/4-SHPED	2.035e−7	<0.0167
CED vs 1/4-SHMED	0.0343	<0.0500
1/4-SHPED vs 1/4-SHMED	0.0011	<0.0250

(a) Task T1 [ms]   (b) Task T2 [ms]

Fig. 6. Distribution of response time

Table 2. Test result of response time of Task T2

Comparison	Test result (p value)	Significance level
CED vs 1/4-SHPED	0.0543	>0.0167
CED vs 1/4-SHMED	0.1237	—
1/4-SHPED vs 1/4-SHMED	0.8474	—

6.8   Answer Accuracy

Figure 7(a) shows the number of correct answers and the number of incorrect answers for T1 in a stacked bar chart. The correct answer rate of 1/4-SHMED is the highest. However, independence between the representation methods was not recognized from the chi-square test. We defined the score for T2 as the Jaccard coefficient between the set of adjacent nodes and the set of answered nodes, i.e., it is 1 when the two sets completely match, and 0 when there is no common element. Figure 7(b) shows the distribution of scores according to each representation method for T2 in a boxplot. From the Shapiro-Wilk test ($\alpha = 0.05$), T2 did not followed a normal distribution. Therefore, we performed multiple tests using the Friedman and Holm methods. Table 3 shows the test results for the scores T2. As seen above, regarding the accuracy of answers, no significant difference was observed between the representation methods. Therefore, H2 is not supported.

Table 3. Test result of score of T2

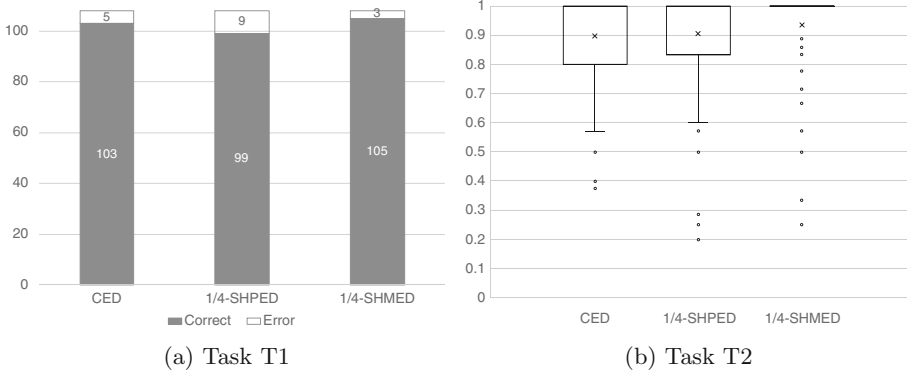| Comparison	Test result (p value)	Significance level
CED vs 1/4-SHPED	0.5862	—
CED vs 1/4-SHMED	0.1489	—
1/4-SHPED vs 1/4-SHMED	0.07817	>0.0167

(a) Task T1   (b) Task T2

Fig. 7. Correct answer rate

6.9   Qualitative Feedback

We asked the participants for opinions on visual representations using question-naires. The following comments were obtained on 1/4-SHMED.

- Positive opinions
    - Morphing made it easy to confirm the exact adjacency.
    - It can be judged whether two nodes are adjacent by observing the morphing of two stabs works simultaneously.
- Negative opinions
    - It is messy and difficult to see. My eyes are strained.
    - The stubs change too fast. The time for stubs to connect is too short.

Positive opinions indicate that morphing contributes to reading graphs. In contrast, from the negative opinions, it appears that visual clutter was not always resolved. The following can be considered as the main reasons. The first is that the morphing speed is too fast. In the implementation used for the experiment, to shorten the overall morphing time, the morphing speed was determined based on the tracking speed of human eyes; however, this appears to be too fast. The second is that there were a large number of stubs applying morphing. In the graph used in the experiment, out of the 144 edges, the average number of non-morphing edges is 24.5. Given that approximately 120 edges repeated morphing, the entire graph is considered to have caused visual clutter.

7   Concluding Remarks

We proposed morphing edge drawing (MED) which is time-varying partial edge drawing (PED) and showed the formalization of MED. We also developed a scheduling scheme for morphing such that dynamic stubs do not cause new crossings. We compared three visual representations, CED, 1/4-SHPED, and 1/4-SHMED, via a user study, and showed that 1/4-SHMED is better than 1/4-SHPED in terms of graph reading time. Thus, MED can function as a counter-measure against the time to read a graph by PED. In the future, it is important

to investigate eye-friendly morphing that causes less strain and has improved scheduling.

References

1. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999). https://doi.org/10.1126/science.286.5439.509
2. Becker, R.A., Eick, S.G., Wilks, A.R.: Visualizing network data. IEEE Trans. Vis. Comput. Graph. **1**(1), 16–28 (1995). https://doi.org/10.1109/2945.468391
3. Binucci, C., Liotta, G., Montecchiani, F., Tappini, A.: Partial edge drawing: homogeneity is more important than crossings and ink. In: 2016 7th International Conference on Information, Intelligence, Systems Applications (IISA), pp. 1–6, July 2016. https://doi.org/10.1109/IISA.2016.7785427
4. Blaas, J., Botha, C., Grundy, E., Jones, M., Laramee, R., Post, F.: Smooth graphs for visual exploration of higher-order state transitions. IEEE Trans. Vis. Comput. Graph. **15**(6), 969–976 (2009). https://doi.org/10.1109/TVCG.2009.181
5. Bruckdorfer, T., Kaufmann, M.: Mad at edge crossings? Break the edges!. In: Kranakis, E., Krizanc, D., Luccio, F. (eds.) FUN 2012. LNCS, vol. 7288, pp. 40–50. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30347-0_7
6. Bruckdorfer, T., Kaufmann, M., Leibßle, S.: PED user study. In: Di Giacomo, E., Lubiw, A. (eds.) GD 2015. LNCS, vol. 9411, pp. 551–553. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27261-0_47
7. Burch, M.: A user study on judging the target node in partial link drawings. In: 2017 21st International Conference Information Visualisation (IV), pp. 199–204, July 2017. https://doi.org/10.1109/iV.2017.43
8. Burch, M., Vehlow, C., Konevtsova, N., Weiskopf, D.: Evaluating partially drawn links for directed graph edges. In: van Kreveld, M., Speckmann, B. (eds.) GD 2011. LNCS, vol. 7034, pp. 226–237. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-25878-7_22
9. Collins, C., Viégas, F.B., Wattenberg, M.: Parallel tag clouds to explore and analyze faceted text corpora. In: 2009 IEEE Symposium on Visual Analytics Science and Technology, pp. 91–98, October 2009. https://doi.org/10.1109/VAST.2009.5333443
10. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. Softw.: Pract. Exp. **21**(11), 1129–1164 (1991). https://doi.org/10.1002/spe.4380211102
11. Holten, D., Isenberg, P., van Wijk, J.J., Fekete, J.: An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs. In: 2011 IEEE Pacific Visualization Symposium, pp. 195–202, March 2011. https://doi.org/10.1109/PACIFICVIS.2011.5742390
12. Robinson, D.A.: The mechanics of human smooth pursuit eye movement. J. Physiol. **180**(3), 569–591 (1965). https://doi.org/10.1113/jphysiol.1965.sp007718
13. Romat, H., Appert, C., Bach, B., Henry-Riche, N., Pietriga, E.: Animated edge textures in node-link diagrams: a design space and initial evaluation. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, pp. 187:1–187:13. ACM, New York (2018). https://doi.org/10.1145/3173574.3173761
14. Schmauder, H., Burch, M., Weiskopf, D.: Visualizing dynamic weighted digraphs with partial links. In: Proceedings of 6th International Conference on Information Visualization Theory and Applications (IVAPP), pp. 123–130 (2015)