

# グラフのメンタルモデルを保存する モーフィング手法の開発と評価

山縣 開登<sup>a)</sup> 三末 和男<sup>b)</sup>

概要：動的グラフや大規模グラフの可視化において、視覚的表現がグラフの特性に適するように、ノードリンク図と隣接行列図を切り替えて表示する手法が存在する。このような手法ではしばしば、表現の切り替え前後でグラフ構造の対応づけが難しくなる、メンタルモデル維持の問題が生じる。そこで本研究では、ノードリンク図と隣接行列図の切り替えを補間するモーフィングに着目し、モーフィングがメンタルモデル維持に有効かをどうかを検証した。評価実験の結果、「隣接行列図のみを見た」場合より、「ノードリンク図から隣接行列図にモーフィングした表現」を見た場合の方が、「最短パスの発見」タスクの回答時間が短いことがわかり、メンタルモデル維持にモーフィングが有効である可能性が示された。

## 1. はじめに

グラフの可視化では、ノードリンク図と呼ばれる視覚的表現が用いられることが一般的であるが、ノードリンク図にはサイズや密度が増大すると可読性が大きく低下するという欠点がある。

Henry らは大規模グラフの可視化において、大局的な読み取りが得意なノードリンク図と、局所的な読み取りが得意な隣接行列図を組み合わせたハイブリッド表現、NodeTrix を提案した。NodeTrix では、ツールの機能として、選択範囲の表現を局所的に切り替えるインタラクションを実装している [1]。また、Burch は、典型的な動的グラフの可視化ではグラフの特徴が時間経過によって変化するにも関わらず、視覚的表現は最初に選択された表現から不変であることを批判し、動的グラフの各時点におけるグラフの特徴に応じた適切な視覚的表現を選択し、自動で切り替える手法を提案した [2]。これらの手法の課題として、表現を切り替えた際、その前後のグラフの対応関係を追跡できなくなるというメンタルモデル（メンタルマップ）保存の問題が生じることがあげられた。

メンタルモデルが保存されず崩壊すると、切り替え後の表現から再度、一からメンタルモデルを構築し直す必要があり、これは閲覧者に対して高い認知負荷をかけてしまうため望ましくない。表現を切り替えてもメンタルモデルを維持し、閲覧者の認知的負荷を減らす事が、本研究の目的

である。目的の達成に向け、視覚的表現を切り替える際にノードやエッジがどこに移動したかがわかるようなアニメーション、すなわちモーフィングを作成し、被験者実験によってメンタルモデル維持に効果があるのかを測定した。

## 2. 関連研究

異なる視覚的表現を切り替えて表示する研究は、Henry ら [1] や Burch [2] が行なっている。Henry らは大局的な読み取りに強いノードリンク図と、局所的な読み取りに強い隣接行列図を組み合わせた NodeTrix という新たな視覚的表現、および選択した部分の表現を局所的に切り替え・操作することができるインタラクションを含んだツールを提案した。ツール内では、ノードリンク図から隣接行列図に切り替える際に補間アニメーションが実装されている。このアニメーションは、メンタルモデル保存を目的としているが、アニメーションの効果について、評価はされていない。Burch らは、動的グラフ可視化においてグラフのサイズ・密度といった特性がグラフ構造の変化に伴って変化することに関わらず、視覚的表現は一貫して変化しないことを問題視し、グラフの特性を数学的に計算し、各時点でのグラフの特性に応じて視覚的表現を自動的に切り替える手法を提案した。Burch らは提案手法の課題として、表現を切り替える際にグラフ構造のどの部分に変化したのか追跡できなくなるという「メンタルマップ保存の問題」を挙げた。

メンタルマップ・メンタルモデル保存に関する研究・調査は、前述の Henry ら及び Archambault ら [3], [4], Bartram ら [5], 森 [6], が行なっている。Archambault らは、動的グラフの可視化においてメンタルマップ保存が与える影響

<sup>1</sup> 筑波大学

University of Tsukuba

a) yamagata@vislab.cs.tsukuba.ac.jp

b) misue@vislab.cs.tsukuba.ac.jp

について調査した [3], [4]. Bartram らは, 情報可視化分野ではその定義が曖昧なメンタルマップ・メンタルモデルについて議論し, 定義や用語, キーコンセプトについてまとめた. 森は, 二部グラフ表現から隣接行列図へのモーフィングを作成した. また, ただ表現を切り替えた場合と, 3種類のモーフィングを挟んで切り替えた場合それぞれに対して, 1 から 3 個のエッジを追跡するタスクを実施し, 実行時間によってモーフィングがメンタルモデル保存に有効か検証した. 結果, 事後アンケートによって, モーフィングによりタスク実行における自信度が向上することがわかったが, タスク実行時間においてモーフィングの優位性を定量的に示すことはできていない.

Ghoniem ら [7] は, グラフのサイズと密度が, ノードリンク図, 隣接行列図の可読性に与える影響について調査し, サイズの小さなグラフではノードリンク図が有利だが, サイズの大きい, またな密度が大きいグラフでは隣接行列図が有利であると結論付けた. Lee ら [8] は, グラフ可視化システムの評価において, 視覚的表現が焦点を当てるグラフの要素と, 評価に用いられるタスクリストを整理した. M.Okoe ら [9] は, 大規模なグラフデータ可視化におけるノードリンク図と隣接行列図の性能について独自に設定した 14 種類のタスクによって比較し, 接続性・記憶性のタスクに関してはノードリンク図が有利, 共通隣接・エッジグループに関するタスクでは隣接行列図が有利であり, パス関連のタスクはどちらの表現でも困難であるが, ノードリンク図の方が有利であると結論付けた.

Behrisch ら [10] は, 行列表現によって可視化されるパターンを整理した上で, 様々な分野・目的に応じて利用される表形式データの並べ替えアルゴリズムを整理した.

### 3. メンタルモデル

Norman は, 認知科学の概念である「メンタルモデル」をインタラクションシステムやユーザビリティの分野に応用した. Norman は著書 [11] で, 概念モデルを「通常は極めて簡素化された, あるモノがどう動くかについての説明」, メンタルモデルを「人の頭の中であって, モノがどう動作するのかについてその人の理解を表す概念モデル」と説明している. メンタルモデルという言葉は, 一般的にはこのように「あるモノがどう動作するのか」ということに焦点が当てられていることが多いが, 情報可視化分野においては少々違った意味合いで使われ, 未だに情報可視化におけるメンタルモデルの定義は明確に定まっていないのが現状である.

また, 『ユーザビリティ ハンドブック』[12] では次のように説明されている. 「私たちは, 目の前に対象物がなくても頭の中にその対象物を思い浮かべて推論することができる. このとき頭の中に思い浮かべられているものが対象物のメンタルモデルである. 人は, 多くの場合, 外的な出

来事をこのような内的モデルに翻訳し, そのモデルを操作することによって様々な判断を下すことができる.」この本はユーザビリティの分野の書籍であり, 前段落の説明と矛盾していないが, より高次の定義がされており, 「対象物について頭の中に思い浮かべているもの」ということに焦点が当てられている. また, NII の湘南会議 No.127 では, 情報可視化分野におけるメンタルモデルの定義や概念について議論され, 「可視化表現から解釈したいくつかの内部表現をまとめた, 認知地図を理解する過程で生成されるもの」とレポート [5] にまとめられている. これらに基づき, 本研究ではグラフのメンタルモデルを「人間が視覚的表現を読んだ時にできるグラフ構造について解釈したモデル」と定義する. これは, 単に視覚的表現の絵図的なイメージではなく, 表現から読み取ったグラフ構造に関する, 閲覧者の理解を反映した抽象的なモデルである.

なお, 情報可視化分野では似た用語の「メンタルマップ」がしばしば用いられるが, メンタルマップは相対的な位置関係などに注目した概念であり, 湘南会議レポート [5] ではメンタルモデルよりも低次の, 内部表現の一種だと説明されている.

## 4. データおよび視覚的表現

### 4.1 データ

本研究で扱うグラフの定義を示す. グラフ  $G$  は  $n$  個のノードの集合  $V$  と, エッジの集合  $E$  の組で構成される. 1 エッジはスタートノードとターゲットノードの 2 つの頂点の組で表される. 本研究では無向グラフを扱うためスタートノードとターゲットノードの順序は影響せず, 組み合わせのみが重要となる. また, 輪の存在はなく, エッジに重みはないとする.

### 4.2 視覚的表現

本研究で扱う視覚的表現は, ノードリンク図と隣接行列図の 2 つである. 以下ではそれぞれの表現について特徴, 特性と本研究内での実装についてをまとめた後, 2 つの表現の差異を整理する.

#### 4.2.1 ノードリンク図

ノードリンク図は, 接続のあるノード同士を線で分けて結ぶことで接続関係を表す視覚的表現である. 特性としては, 直感的な読み取り性能が高く, グラフの可視化表現としては一般に馴染み深いという利点がある反面, グラフのサイズや密度の増大に伴いエッジの交差が生まれて視覚的に混雑し, 可読性が低下するというスケーラビリティの問題がある [7]. 実装に関して, ノードの配置 (レイアウト) は, 力学モデルを利用し計算している.

#### 4.2.2 隣接行列図

隣接行列図は, ノード  $i$  を行列の第  $i$  行と第  $i$  列に対応させ, 各ノードペアの隣接関係を行列の各セルで表す視覚的

表現である。有向グラフの可視化においては、行にスタートノードを、列にターゲットノードを対応させ、ノード a からノード b への隣接関係を (a, b) のセルで表すが、今回は無向グラフを扱うのでスタートノードとターゲットノードの区別はなく、ノード a とノード b 間の隣接関係を (a, b) 及び (b, a) のセルで表す。特性としては、一般への周知度は低いが、オクルージョン問題が発生しない構成 [7] から、グラフのサイズや密度が増大しても可読性が低下しづらいという利点がある。

実装においては、ノードリンク図でノードを表していた円を行・列のヘッダとして配置し、各ノードを行・列で表現する。エッジは、隣接関係のあるノードペアの対応するセルに、正方形を配置して表現する。また、エッジがどの行・列にあるのかをわかりやすくするため、ヘッダの円と、そのノードに接続しているエッジを表す正方形を線分で結んだ。隣接行列図では行・列内でノードどのような順番で並んでいるかが、クリークやクラスター、次数が高くハブ的な役割を果たしているノードの有無といった情報を視覚的なパターンとして可視化する上で非常に重要になる。このノードの並べ方に関する問題は、Behrisch ら [10] がまとめている。本研究においては、隣接行列図を描画した際にノードが適切な並び順になるよう、Fekete 氏が公開している reorder.js[13] という JavaScript ライブラリ内の重心法を利用した関数を利用し、グラフデータに前処理を施している。

## 5. モーフィングの設計

モーフィングの一般的な意味として、『コンピュータグラフィックス』では、「モーフィング (*morphing*) は、(中略)ある画像を別の画像になめらかに変化させる画像処理の手法である。」[14] と書かれている。本研究内では、モーフィングは「ある視覚的表現を異なる視覚的表現へと、徐々に変化させていく補間アニメーション」を指す。

本研究内で 2 つのモーフィング実装した。2 つのモーフィングは、後述の被験者実験において比較対象となる 4 つの表現のうち、V3, V4 に当たる。以下ではそれぞれを V3, V4 として説明する。なお、どちらのモーフィングにおいても、ノードはノードリンク図と隣接行列図で一貫して円で表されるが、エッジはノードリンク図では線分、隣接行列図では正方形で表される。以下では区別のため、ノードリンク図におけるエッジを表す要素を「エッジ (線分)」、隣接行列図におけるエッジを表す要素を「エッジ (正方形)」と表記する。

### 5.1 モーフィング 1(V3)

V3 は、NodeTrix[1] で実装されていたノードリンク図から隣接行列図へ変化する際のアニメーションを参考に実装したものである。V3 は、以下のような 3 段階のフェーズ

によって構成されている。

**フェーズ 1** 図 1 の 1 から 6 を参照

**ノード** 各ノードが隣接行列図での対角線上に整列するよう移動する。

**エッジ** 各エッジ (線分) の端点 (すなわちノードとの接点) は、各ノードに追従する。同時に、現時点での各エッジ (線分) の中点を折れ点とし、折れ点が隣接行列図における該当するセルの座標に移動する。折れ点の移動の際、エッジ (線分) は折れ点で折れ曲がり、端点から折れ点までの線分は、必要に応じて伸縮する。なお、エッジ (線分) は複製され、フェーズ 1 終了時点で対角線を軸に対称となっているように移動する。フェーズ 1 終了時には、各エッジ (線分) が中点で直角に折れ曲がっている状態になっている。

**フェーズ 2** 図 1 の 7 から 9 を参照

**エッジ** 各エッジ (線分) の折れ点に、対応するエッジ (正方形) が出現する。この際、正方形の重心は線分の折れ点と一致し、正方形が徐々に大きくなるように出現する。また同時に、隣接行列図の対角線を表す正方形も同様に描画される。

**フェーズ 3** 図 1 の 10 から 12 を参照

**ノード** 各ノードが複製され、隣接行列図の行と列のヘッダ位置にそれぞれ移動する。

**エッジ** 線分の端点が接している各ノードの移動先に移動する。

### 5.2 モーフィング 2(V4)

V4 は、以下のような 3 段階のフェーズによって構成されている。

**フェーズ 1** 図 2 の 1,2 を参照

**ノード** 各ノードが、横方向に分裂する

**フェーズ 2** 図 2 の 3 から 7 を参照

**ノード** 分裂したノードの左半分・右半分が、それぞれ対応する行・列のヘッダ位置に移動する

**エッジ** 各エッジ (線分) の端点は、分裂したノードを追従して移動する。同時に、現時点での各エッジ (線分) の中点を折れ点とし、折れ点が隣接行列図における対応するセルの座標に移動する。この際、エッジ (線分) は折れ点の位置で折れ曲がり、端点から折れ点までの線分は、必要に応じて伸縮する。この際、エッジ (線分) は複製される。フェーズ 1 の終了時には、各エッジ (線分) が折れ点で直角に折れ曲がっている状態になっている。

**フェーズ 3** 図 2 の 8 から 10 を参照

**エッジ** 各エッジ (線分) の折れ点に、対応するエッジ (正方形) が出現する。この際、正方形の重心は折れ点と一致し、正方形が徐々に大きくなるように出現する。また同時に、隣接行列図の対角線を表す正

方形も同様に描画される。

### 5.2.1 2つのモーフィングの特徴の比較

V3では、一度ノードが対角線上に並ぶことによって、複製され、対角線の上下の領域にそれぞれ描画された各エッジの対応がわかりやすい。また、V4では、ノードが直接行・列のヘッダ位置に移動するため、V3に比べノードやエッジの移動の仕方は単純である。

## 6. 評価実験

### 6.1 実験の設計

実験によって確かめたいことは、「メンタルモデルの維持ができていくかどうか」である。「メンタルモデルが維持できている」とは、すなわち「切り替え前の視覚的表現から読み取ったグラフ構造を頭の中に維持して、切り替え後の視覚的表現を見る際に利用できる」ということである。これを測定するため、タスクは「ノードリンク図からは読み取りやすいが、隣接行列図から読み取ることは難しい情報」に関するタスクとし、Leeら[8]、Okoeら[9]、Ghoniemら[7]を参考に、「2ノード間の最短パスの発見」とした。ただし、タスクで出題される課題には以下の2つの制限を設けた。

- (1) 課題ノード間の最短パスは、正答となるパスが唯一存在する。これは、正答パターンが何種類かあることによってタスクの難易度が下がり、グラフデータによって難易度に差が出るのを防ぐためである。
- (2) タスクの各問で、正答となる課題ノード間の最短パスのパス長は予め与えられているものとする。具体的には、パス長は2か3で固定した。これは、すでに最短パスを発見できているにも関わらず、より短いパスを発見しようとしてタスクの実行時間が大きくなってしまふことを防ぐためである。

### 6.2 提示した表現

比較する表現は以下の4種類である。

#### V1 隣接行列図のみ

#### V2 ノードリンク図が隣接行列図に切り替わる（モーフィングなし）

#### V3 ノードリンク図がモーフィング（M1）によって隣接行列図に切り替わる

#### V4 ノードリンク図がモーフィング（M2）によって隣接行列図に切り替わる

V3及びV4でのタスクの実行時間が、V1及びV2でのタスク実行時間よりも短ければ、モーフィングがメンタルモデル維持に効果的であると示せると考えた。

### 6.3 提示したデータ

1つの表現に対して、グラフサイズが20, 30, 40の場合、課題のパス長が2, 3の場合で合わせて6パターン、6

問のタスクについて回答してもらおう。これを4種類の表現で実施するので、計24問のタスクについて回答してもらった。同じデータを使いまわすとグラフデータを覚えてしまうおそれがあるため、24個のデータを用意した。データはOkoeら[9]の研究で利用された、258ノード1090エッジを持つ実在するグラフデータを基とし、そこから任意のノード数の誘導部分グラフをランダムに抜き出すことで作成した。またノードにラベルが振ってあると、最短パスにおける中継ノードのラベルを記憶することができてしまい、タスクが期待するようなトポロジーの読み取りタスクにならないと考え、ノードにラベルは振ってない。

### 6.4 実験手続き

実験参加者には最初に、以下の説明をした。

- (1) グラフ可視化の基本用語（ノード、エッジ、パス）
- (2) 隣接行列図の読み取り方
- (3) タスク
- (4) 隣接行列図上でパスを発見するための読み取り方
- (5) 4種類の表現およびモーフィングの動き方

グラフが無向グラフである場合に限るが、隣接行列図上でのパスの発見は、スタートノードの列を探索し、その列の中から1つのエッジを選びそのセルの行を探索、その行の中から1つのエッジを選びそのセルの列を探索、...と繰り返すことで、比較的簡単に行うことができる。被験者に依って実験以前からの隣接行列図の理解度が異なることから、タスクの実行時間に差が出ることを防止するため、全員にこのノウハウについての説明をした。以上の説明が終わった後、それぞれの表現に対して12ノードのグラフデータを2個用意し、パス長2, 3のタスクを実施し、練習してもらった。練習は最低でも4種類の表現で2問ずつ、計8問やってもらったが、表現・タスクの理解が深まるまで必要に応じて繰り返した。

なお、V1, V2, V3, V4の順序は、順序効果を打ち消すためにラテン方格を用いて被験者ごとに割り当てている。また、タスクはなるべく正確に、できるだけ早く行うという指示を被験者に出した。

各表現における1つのタスクの流れを説明する。どの表現においても共通して、

- (1) タスクがスタート、3秒間のカウントダウン
- (2) 課題となる2ノードが提示される
- (3) 隣接行列図上で、課題ノード間の最短パスを構成するエッジ（正方形）全てをマウスクリックによって選択する
- (4) エッジ（正方形）をパス長の数だけ選択したらキーボードのEnterキーを押して確定する
- (5) (1)に戻る

という流れで進んで行くが、図3を見るとわかるように、各表現では(1)から(3)にかけて、どのタイミングでノー

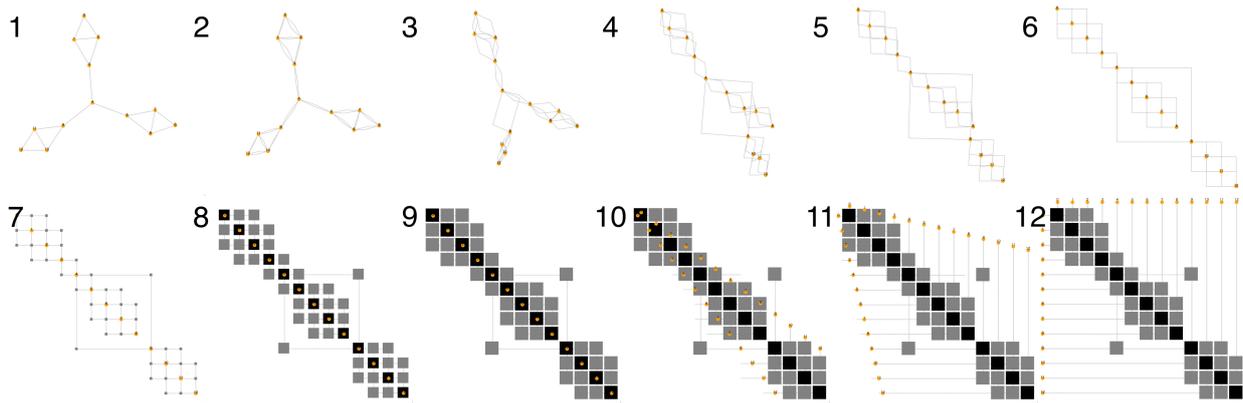


図 1 V3 のモーフィングの様子  
Fig. 1 Appearance of V3 morphing

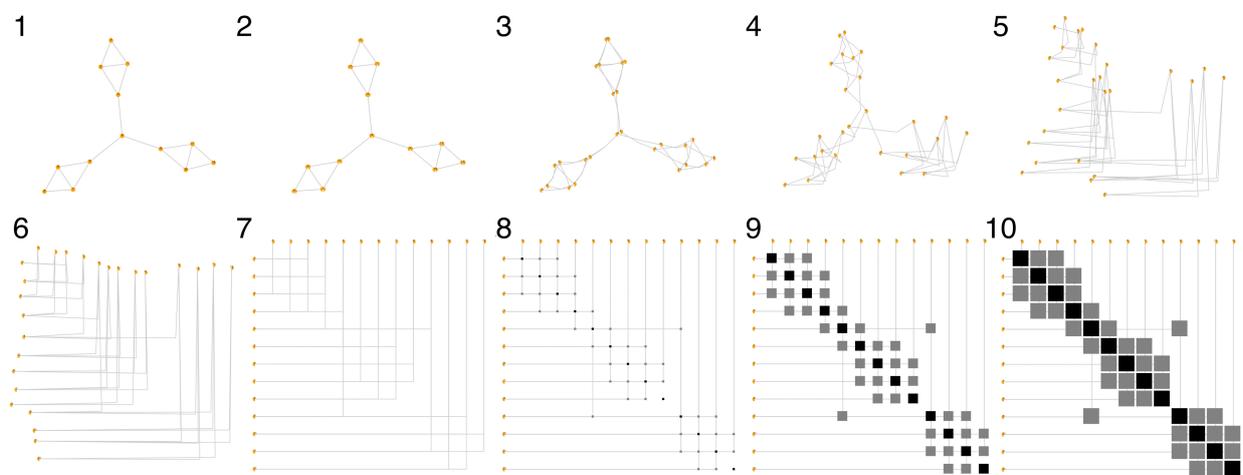


図 2 V4 のモーフィングの様子  
Fig. 2 Appearance of V4 morphing

ドリンク図 / 隣接行列図が表示されるのが異なる。

V1 では、(2) の段階ですでに隣接行列図が表示されている。V2, V3, V4 では、(2) の段階ではノードリンク図が表示されており、課題ノード提示の 3 秒後に、ノードリンク図が隣接行列図に切り替わる。またこの際、V2 は一瞬で切り替わるが、V3, V4 は 6 秒間のモーフィングを挟んで切り替わる。

### 6.5 実験参加者

実験参加者は全部で 21 人であり、全員が大学・大学院の学生である。元々 20 人の被験者を集めていたが、1 人が睡眠不足による集中力の低下を報告しており、タスクの遂行時間も全被験者の平均より大幅に長くなってしまった場合が散見されたので、外れ値として除外し、1 人を後から追加した。外れ値として除外した 1 人以外の 20 人について表 1 で報告する。

### 6.6 実験環境

各視覚的表現、モーフィング、及びツールは HTML +

表 1 実験参加者の構成

性別	男性	15	名
	女性	5	名
年齢	最大値	25	歳
	最小値	20	歳
	平均値	22.6	歳
	中央値	22.5	歳
	最頻値	22	歳
隣接行列図の認知度	全く知らなかった	7	名
	表現の存在程度は知っていた	10	名
	表現をよく知っていた	3	名

JavaScript で作成し、Google Chrome 上で動作させた。コンピュータは、MacBook Pro (Late 2016) 13 インチモデルを用いた。

### 6.7 結果と分析

以下では、アンケート結果、タスクの正答率、実行時間、回答時間についての結果を示す。

図 3 は、各表現において、タスク開始より何秒経過した



図 3 表現毎のタイムライン

表 2 正答数の集計結果 n:ノード数

表現		パス長: 2			パス長: 3			総合(延べ)
		n:20	n:30	n:40	n:20	n:30	n:40	
V1	正答数(人)	20	20	19	19	20	20	117
	正答率(%)	100	100	95	95	95	100	97.5
V2	正答数(人)	20	20	20	19	19	20	118
	正答率(%)	100	100	100	95	95	100	98.3
V3	正答数(人)	20	20	20	20	19	19	118
	正答率(%)	100	100	100	100	95	95	98.3
V4	正答数(人)	18	19	20	19	20	20	116
	正答率(%)	90	95	100	95	100	100	96.7

時点でどの表現が描画されているか、課題ノードがハイライトされるタイミングなどを示したタイムラインである。図3より、各表現によって、課題ノードがハイライトされてから隣接行列図が表示されて回答可能になるまでの時間に差がある。

以下では区別のため、課題ノードがハイライトされてから被験者が回答を完了するまでの時間を実行時間、隣接行列図が表示されて回答が可能になってから回答を完了するまでの時間を回答時間と呼ぶことにし、それぞれについて分析を行う。

### 6.7.1 正答率

各表現・タスクにおける正答率を表2に示す。表2より、どの表現においても正答率は95%を超えており、表現間の正答率に大きな差は見られない。タスクの性質として、自分が選んだエッジが解答のパスとして正しいかどうか、実際に辿ってみることで自分で確認することが可能であり、被験者には「なるべく正確に、(その中で)なるべく早く」という指示をしたための結果だと考えられる。誤答であった解答を詳しく確認してみると、誤答の原因は大きく2つに分けられ、

- (1) 課題ノードの列(行)を読み違えた
- (2) パス長3の場合に、スタートノードから自身へのループとなるパスを選択してしまった

ことに起因すると推測できる。また、パス長2のタスクでの誤答数が3であるのに対して、パス長3のタスクが6と多いのは、パス長2では1.が原因のミスしか発生し得ないが、パス長3のタスクではどちらのミスも発生し得るからだと考えられる。

表 3 各タスクの実行時間と回答時間の平均(秒)、及び標準偏差 n:ノード数

表現		パス長: 2			パス長: 3		
		n:20	n:30	n:40	n:20	n:30	n:40
V1	実行時間	11.41	11.75	18.13	21.65	69.94	38.18
	回答時間	11.41	11.75	18.13	21.65	69.94	38.18
	標準偏差	5.74	6.64	8.51	14.52	53.52	27.96
V2	実行時間	11.82	14.91	13.15	22.84	36.36	53.12
	回答時間	8.82	11.91	10.15	19.84	33.36	50.12
	標準偏差	2.40	5.47	3.96	7.23	21.18	29.15
V3	実行時間	18.58	19.06	20.15	31.16	38.68	48.88
	回答時間	9.58	10.06	11.15	22.16	29.68	39.88
	標準偏差	6.05	4.55	6.97	12.49	18.88	30.72
V4	実行時間	17.32	22.07	18.18	35.38	40.20	34.91
	回答時間	8.32	13.07	9.18	26.38	31.20	25.91
	標準偏差	3.62	8.90	3.67	18.88	17.87	13.22

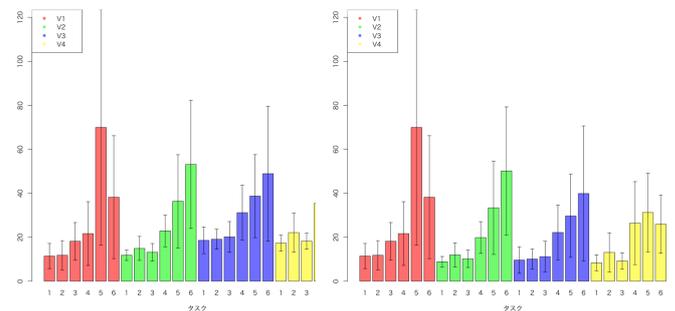


図 4 平均実行時間

図 5 平均回答時間

### 6.7.2 実行時間と回答時間

タスクの各問において、実行時間は課題ノードがハイライトされてから実験参加者がEnterキーを押すまでの時間である。つまり、V2, V3, V4においては、ノードリンク図が表示されている時間や、モーフィングにかかる時間も含まれている。回答時間は、隣接行列図が表示されてから実験参加者がEnterキーを押すまでの時間である。つまり、V3 = V4 > V2 > V1の順に、回答時間には含まれないが、可視化表現を観察できる時間が長い。

まず、各試行の平均実行時間と平均回答時間、標準偏差を表3及び図4, 5に示す。

図4は、各表現・タスクにおける平均実行時間を棒グラフで、標準偏差をエラーバーで示している。各表現を色で分け、左から赤:V1, 緑:V2, 青:V3, 黄:V4で表している。各表現毎に6本ある棒グラフが、1から6までのタスクでの平均実行時間を表しており、1,2,3がパス長:2のタスク、4,5,6がパス長:3のタスクである。図5は、各表現における平均回答時間と標準偏差を表しており、見方は図4と同じである。

図4, 5より、どの表現においてもパス長2のタスクよりパス長3のタスクの方が、実行時間、回答時間が長くなっており、パス長2のタスクとパス長3のタスクの難易度に大きな差があることが読み取れる。実行時間に関しては、パス長2のタスクに関して、V1・V2に比べ、V3・V4の実行時間は長くなっており、これはパス長2のタスクが簡

表 4 実行時間に関する分散分析

要因	p 値
表現	0.436
タスクのパス長	<2e-16
表現 × タスクのパス長	0.115

表 5 回答時間に関する分散分析

要因	p 値
表現	0.00138
タスクのパス長	<2e-16
表現 × タスクのパス長	0.11479

表 6 表現による回答時間の差の多重比較結果 (全体)

表現	V1 - V2	V1 - V3	V1 - V4	V2 - V3	V2 - V4	V3 - V4
p 値	0.167	0.034	0.008	0.914	0.673	0.965

単であったため、V3・V4のアニメーションに要する6秒間が差となって現れた結果であると考えられる。パス長3のタスクについては、V1・V2に比べV3・V4の実行時間が全体的に短い。回答時間に関しては、パス長2のタスクでは全ての表現の間で大きな差は見られず、パス長3のタスクに関しては実行時間で述べたものと同様である。

これらの事項を確かめるために分析を行った。まず実行時間、回答時間に関して、要因1を視覚的表現の種類、要因2をタスクのパス長として分散分析を行った結果を表4, 5に示す。また、有意水準は全て5%とした。

表4, 5より、タスクのパス長による実行時間、回答時間には有意差が見られた。また、表現の種類による実行時間の差には有意差が見られなかったが、回答時間には有意差が見られた。タスクのパス長と表現の種類の交互作用については、有意差が見られなかった。また、表現とパス長を要因として多重比較を行ったところ、表6のような結果となった。表6より、V1とV3, V1とV4間の回答時間に有意差があることがわかった。

### 6.7.3 タスクごとの体感的な難易度

本番タスク終了後に、各表現におけるパス長2のタスクと、パス長3のタスクについて、難易度はどれくらいに感じたかアンケートを取った。結果を図6に示す。図6より、パス長2のタスクでは、どの表現においても難しいと感じた人はいなかった。また、V1, V2ではやや簡単であると感じた人が多かったが、V3, V4ではとても簡単であると感じた人が多かった。つまり、パス長2のタスクでは、V1, V2に比べV3, V4の方が簡単にタスクを実行できると感じる人が多いようだ。パス長3のタスクでは、パス長2のタスクに比べて全体的に、難しいと感じた人が多い。またどの表現においても、とても簡単と感じた人はいない。V1では、とても難しいと感じた人が最も多く、より簡単に感じた人は右肩下がりに少なくなっている。V2では、とても難しいと感じた人からやや簡単と感じた人まで同程度いるが、やや簡単と感じた人が最も多い。V3では、

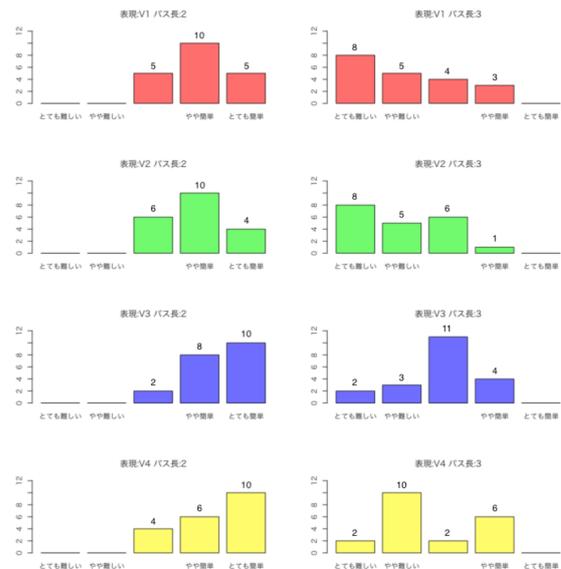


図 6 タスクの体感的な難易度

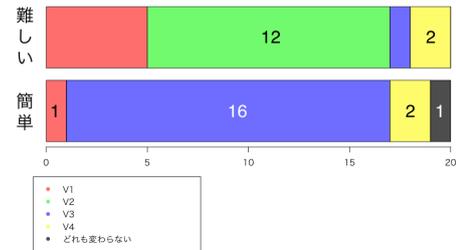


図 7 最もタスクを実行しやすかった/しづらかった表現

他の表現に比べ難しいと感じた人が圧倒的に少なく、どちらでもないかやや簡単だと感じた人が多い。V4では、やや難しいと感じた人が最も多く、次にやや簡単と感じた人が多いが、全体的に見ると難しいと感じた人が多い。

また、最もタスクを実行しやすかった表現と、最もタスクを実行しづらかった表現についてもアンケートを取った。結果を図7に示す。

図7より、最もタスク遂行が簡単だと感じた表現はV3であった。反対に、最もタスク遂行が難しいと感じた表現はV2で、次にV1だった。

### 6.7.4 事後アンケート: 自由記述

本番タスク後のアンケートで、被験者に感じたことなどを自由に記述してもらった。以下で、複数の被験者が記述したことを抜粋して紹介する。

- タスク・表現に慣れたのが本番タスク後半になってしまった。
- ノード数が40になると、アニメーションがあってもエッジを見失ってしまった。
- ノードリンク図でパスを読み取れていない状態でモーフィングが始まってしまう場合があり、その時はただ隣接行列図を読んだ場合と変わらないと感じた。

- グラフのデータによる難易度の差があった。
- 隣接行列図では、下の方の行、右の方の列は読み取りづらかった。

## 6.8 考察と議論

本実験に関して、「なるべく正確に、なるべく早く」と指示をしたこと、選択したエッジが正解しているか、選択が終わった段階で自分で確認ができるタスクの性質から、正答率に大きな差は出ず、回答時間に差がでた。アンケートで、最もタスクが遂行しづらい表現として V2 が多く挙げられたのは、ノードリンク図からグラフのメンタルモデルを生成したにも関わらず、瞬時に表現が切り替わったことでメンタルモデルが崩壊し、混乱が生じた結果であると考えられる。最もタスクを遂行しやすかった表現は、V3 が多く挙げられた。この質問では、V3 と V4 を合わせると 90 % の人がモーフィングありの表現を選んでおり、モーフィングがメンタルモデル保存に効果がある可能性が定性的に示された。実行時間に関しては、どの表現の間にも有意差は見られなかった。これは、モーフィングによる効果だが、モーフィング時間のオーバーヘッドと相殺してしまうからだと考えられる。一方回答時間に関しては、V1 と V3、V1 と V4 の間に有意差があったが、最もやりづらい表現として上がっていた V2 との間には有意差がなく、V1 との有意差がモーフィングの効果であると断言はできない。

また、事後アンケートの自由記述で寄せられた意見より、議論する。グラフデータによる難易度の差異を感じたとの報告があり、表現 × タスク毎に固定のグラフデータを用いたことから、グラフ構造によって実行時間・回答時間に差異がでたことも否定できない。タスク・表現に慣れたのが実験後半になってしまったとの報告から、より練習時間を増やすことで、モーフィングをさらに有効に利用できるようになった可能性がある。ノードリンク図でパスを読み取れず、ただ隣接行列図を見るのと変わらない場合があったとの報告から、V2・V3・V4 でのノードリンク図の観察時間を全て 3 秒間に統一したことに問題があった可能性がある。ノード数が 40 になると、モーフィングがあっても利用できなかったという報告から、メンタルモデル維持に貢献するモーフィングのスケラビリティとして、グラフのノード数が 30 から 40 の間に、限界があると考えられる。

## 7. 結論

本研究では、ノードリンク図と隣接行列図を切り替える際にメンタルモデル保存を目的とするモーフィングを作成し、モーフィングがメンタルモデル保存に有効かどうかを被験者実験によって検証した。

多くの被験者はモーフィングがある表現を好んだことがわかり、回答時間に関して、特にある程度難しいタスクでは、モーフィングありの表現の方が、隣接行列図のみの表

現より有利であることが示された。ただし、ノードリンク図と隣接行列図がただ切り替わる表現と、モーフィングがある表現の間に有意差は見られなかった。

回答時間の被験者によるばらつきが大きいこと、各タスクで固定されているグラフデータの難易度に差があることが、実験条件としての問題点だと考えられる。今後、改善してさらなる実験を行うことが、課題である。

## 参考文献

- [1] Nathalie Henry, Jean Daniel Fekete, and Michael J. McGuffin. NodeTrix: A hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1302–1309, 2007.
- [2] Michael Burch. Dynamic graph visualization with multiple visual metaphors. *Proceedings of the 8th International Symposium on Visual Information Communication and Interaction (VINCHI'15)*, pages 27–34, 2015.
- [3] D. Archambault, H. Purchase, and B. Pinaud. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):539–552, 2011.
- [4] Daniel Archambault and Helen C. Purchase. The “map” in the mental map: Experimental results in dynamic graph drawing. *International Journal of Human-Computer Studies*, 71(11):1044 – 1055, 2013.
- [5] Daniel Archambault, Karsten Klein, Kazuo Misue, in NII Shonan Meeting Report No.2018-12, Reimagining the Mental Map and Drawing Stability. pp.12-13, 2018.
- [6] 森このみ. グラフのメンタルモデルを維持するためのモーフィングの研究. 筑波大学情報学群情報メディア創成学類卒業論文, 2018.
- [7] Mohammad Ghoniem, Jean Daniel Fekete, and Philippe Castagliola. On the readability of graphs using node-link and matrix-based representations: A controlled experiment and statistical analysis. *Information Visualization*, 4(2):114–135, 1997.
- [8] Bongshin Lee, Catherine Plaisant, Cynthia Sims, Jean-Daniel Fekete, Nathalie Henry, and Cynthia Sims Parr. Task taxonomy for graph visualization. *BELIV'06: Proceedings of the 2006 AVI workshop on BEyond time and errors*, pages 1–5, 2006.
- [9] M. Okoe and R. Jianu and S. Kobourov. Node-Link or Adjacency Matrices: Old Question, New Insights. *IEEE Transactions on Visualization and Computer Graphics*, 25(10):2940–2952, 2019.
- [10] Michael Behrisch, Benjamin Bach, Nathalie Henry Riche, Tobias Schreck, and Jean Daniel Fekete. Matrix Reordering Methods for Table and Network Visualization. *Computer Graphics Forum*, 35(3):693–716, jun 2016.
- [11] D.A.Norman. 誰のためのデザイン? 認知科学者のデザイン原論 増補・改訂版. 岡本明, 安村通晃, 伊賀聡一郎, 野島久雄訳. 新曜社, 2015.
- [12] 『ユーザビリティ ハンドブック』編集委員会, editor. ユーザビリティ ハンドブック. 共立出版株式会社, 2007.
- [13] J.D.Fekete. <https://github.com/jdfekete/reorder.js>.
- [14] コンピュータグラフィックス編集委員会. コンピュータグラフィックス. 財団法人 画像情報教育振興協会 (CG-ARTS 協会), 2004.