

# ネットワーク可視化のための focus+context を考慮した エッジバンドリング手法

秋山 桂一<sup>†</sup> 藤田 秀之<sup>†</sup> 大森 匡<sup>†</sup> 新谷 隆彦<sup>†</sup>

<sup>†</sup> 電気通信大学 大学院情報理工学研究科 〒182-8585 東京都調布市調布ヶ丘 1-5-1

E-mail: <sup>†</sup>a2031002@edu.cc.uec.ac.jp, <sup>††</sup>{fujita,omori,shintani}@is.uec.ac.jp

**あらまし** 大規模ネットワーク可視化の課題の一つに、ノードを点、エッジを直線で表現する基本的な手法で可視化すると、多数のエッジが乱雑に重なり合い、可視化結果を読み取ることが難しくなる問題がある。配置の似ているエッジを束ねて曲線として描画することにより、エッジの重なり合いを低減するエッジバンドリング (Edge bundling) は、大規模ネットワーク可視化における重要な手法のひとつである。ここで、ズームやパンの操作により、一部の領域を拡大して描画する可視化インタフェースは、大規模ネットワーク可視化において重要である。拡大描画に関連し、描画された領域 (focus) の周辺に対する位置付け (context) を提示することは重要とされている。そこで、本研究では、focus+context を考慮したエッジバンドリング手法として、画面内のエッジの接続傾向と、画面内外にまたがるエッジの接続傾向を一つの画面で区別して提示する手法を提案する。提案手法を実データに適用し、定量的評価を行い、有効性を確認した。

**キーワード** 情報可視化, ネットワーク可視化, エッジバンドリング

## 1 背景と目的

人やモノの移動に関する場所間の関係、電車やバスの駅と路線のつながり、電力の送電網など、ネットワーク構造をなすデータは日常的に存在しており、これらを活用するために可視化することは重要である。大規模データの可視化においては、データ数が描画する画面やキャンパスのピクセル数を凌駕することも珍しくない。可視化結果には、大量のデータが重なり合って描画される、ビジュアルクラッタという問題が生じデータの読み取りが困難となる課題がある。大規模ネットワークの可視化においても、大量のエッジの重なり合いによって生じる、ビジュアルクラッタは課題とされており [1], その低減を目的とする可視化手法が多数提案されている。中でも、エッジバンドリングは主要な手法である。ノードを点、エッジを線として表現するノードリンクダイアグラムにおいて、エッジバンドリングは配置と形状の似た複数のエッジ群を束ねて曲線として描画することで、ビジュアルクラッタを低減し、ネットワークの概観を提示する手法である。

国内移住ネットワークデータ<sup>1</sup>(市区町村をノード、5年間の移住者100名以上の市区町村間にエッジ)に対し、ノードリンクダイアグラムで可視化した結果と、代表的なエッジバンドリング手法を適用した結果を図1に示す。図1(a)では、エッジの重なり合いによってビジュアルクラッタが発生し、ネットワークの傾向の読み取りが困難になっているのに対し、図1(b)では、バンドリングされたことにより、ビジュアルクラッタが低減し、関東周辺から西方向、北方向に移住が多いこと、海岸線

付近を端点とするエッジが多い等の概観が読み取れる。

ズームやパン操作により、一部の領域を拡大して表示するインタフェースは、ネットワークの可視化においても有用である。しかし、エッジバンドリングの適用結果を拡大すると、図1(c)のように、再びビジュアルクラッタが出現する場合がある。

また、ネットワークを拡大すると画面内のノード同士を結ぶエッジと、画面内外にまたがるエッジ、画面外のノード同士を結ぶエッジの3種類のエッジが出現する。ここで、画面内のノード同士を結ぶエッジは、可視化の興味の対象 (focus) として重要である。加えて、画面内外にまたがるエッジは、可視化領域の周辺に対する位置付け (context) を示すのに有用である。既存手法の可視化結果の拡大では、これらのエッジ群はまとめてバンドリングされており、色分けしてあっても、重なり合って区別ができない、という場合もある。

そこで本研究では、拡大描画でビジュアルクラッタを低減しつつ、画面内のノード同士を結ぶエッジと、画面内外にまたがるエッジを区別して束ねるエッジバンドリング手法を提案する。

本研究の貢献は次のとおりである。

- focus+context を考慮した新しいエッジバンドリング手法を提案する
- 既存手法拡大時に発生するビジュアルクラッタを低減する
- ネットワーク拡大描画時に現れる、画面内のノード同士を結ぶエッジ、画面内外にまたがるエッジを別々に違う場所に束ねる
- 提案手法を実データに適用して定量的に評価することで有用性を確認する

1: 「平成 27 年国勢調査 移動人口の男女・年齢等集計 (人口の転出入状況) 全国結果」(総務省) [2] と「国土数値情報 (市区町村役場データ)」(国土交通省) [3] をもとに作成

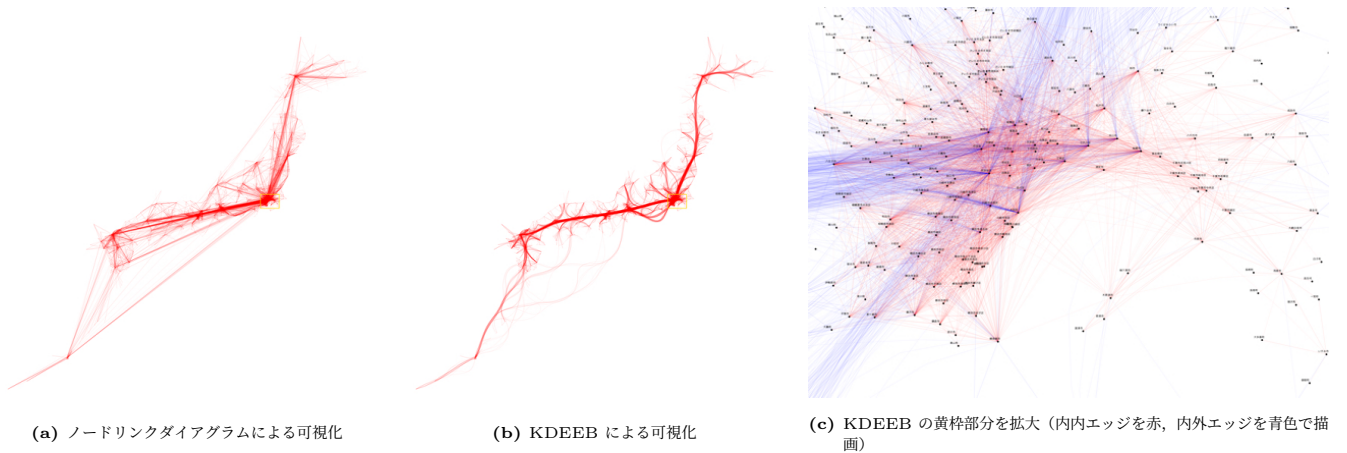


図 1: 国内移住ネットワークの基本手法による可視化と代表的なエッジバンドリング手法を適用した結果 (ノード数:1916 エッジ数:22468)

## 2 関連研究

本章では、本研究の関連研究として、近年のネットワーク可視化手法の研究動向と、ネットワーク可視化手法のひとつであるエッジバンドリングについてまとめ、本研究の位置付けを述べる。

### 2.1 ネットワーク可視化

古典的なネットワーク可視化は、グラフ理論の分野で、グラフ描画と呼ばれる技術として研究されてきた。主要な目的は、個々のノードや 1 本 1 本のエッジを、正確に読み取れるノードリンクダイアグラムの生成である。ノードリンクダイアグラムとは、ノードを点、エッジを線として描画する図であり、現在も一般的に用いられている。グラフ描画は、エッジを画面に水平または垂直に描画する、エッジを重ねない、重なるエッジを直交させるといった、複数の制約を満たすノード配置の最適化問題として取り組まれてきた [4]。

これに対し近年では、情報可視化の分野で、ノードリンクダイアグラムの新しい描画方法を含む、さまざまなネットワーク可視化手法が盛んに提案されている [4]。近年のネットワーク可視化手法の最大の特徴は、大規模なデータを対象としている点であり、主要な目的のひとつは、ビジュアルクラッタの低減である。個々のノードや、1 本 1 本のエッジを、正確に提示するというより、ノード群の分布の傾向や、多数のエッジによるノード群の接続傾向を提示することが、可視化上の要件となっている。

### 2.2 エッジバンドリング

エッジバンドリングは、近年の主要なネットワーク可視化手法のひとつであり、さまざまな手法が提案され、サーベイ論文にまとめられている [1] [5]。エッジバンドリングとは、ノードリンクダイアグラムにおいて、配置や形状の似たエッジ群を曲線として束ねたように描画することで、ビジュアルクラッタを低減する手法である。特徴として、ノードリンクダイアグラムを改善する手法の多くが、ノード配置を制御するのにに対し、エッジバンドリングは、エッジの描画方法を制御する。一般に、

ノードリンクダイアグラムでは、ノードを描画する面積よりも、エッジを描画する面積の方が大きいため、描画上の効果が期待できる。加えて、空間ネットワークと呼ばれる、各ノードが地理的な位置情報を持つネットワークにも適用可能である。空間ネットワークの可視化では、一般に、ノードの配置に所与の位置情報を用いるため、ノード配置ではなく、エッジ描画方法を制御する同手法は重要である。

初期のエッジバンドリング手法はネットワークデータの他、階層関係の情報 [6] や制御メッシュ [7] を必要とし、それらに基づいてエッジを描画することが多かった。一方、Holten らが提案した、ばね引力と静電気力に基づく、Force-Directed Edge Bundling(FDEB) [8] は追加情報を必要とせず、ネットワークデータ一般に適用できる点と、拡張が容易である点で画期的であった。同手法を拡張する手法として、エッジの方向や重みを考慮したバンドリング手法 [9]、エッジの属性を考慮したバンドリング手法 [10]、三次元ネットワークに対応したバンドリング手法 [11] など、さまざまな新しい性質を持たせたバンドリング手法が多数提案されている。しかし、Holten らが提案した FDEB はエッジの本数が増えると計算量が膨大になるという問題があり、近年では、エッジの輪郭を用いる手法 [12]、や FDEB を並列計算化した手法 [13] など、GPU を用いた並列計算によりバンドリングを行う手法が主流になりつつある。中でも、カーネル密度推定を用いる Kernel Density Estimation Edge Bundling(KDEEB) [14] は、GPU を使わない既存の手法に比べ、極めて高速で、アルゴリズムが単純で拡張しやすいことから、さまざまな関連手法が提案されている [15] [16] [17] [18]。本研究も、同手法を拡張して新しい機能を持たせた手法と分類できる。しかし、筆者らの知る限り、拡大描画に適応し、画面内のノード同士を結ぶエッジ群と、画面内外にまたがるエッジ群を分けて束ねるエッジバンドリング手法は存在しない。

筆者らは、先行研究において、ネットワークの拡大描画を考慮したエッジバンドリング手法 [19] を提案してきたが、本稿とは具体的な目的が異なる。本稿では、拡大したネットワークの注目領域 (focus) とその位置付け (context) の両者を可視化することを目的としており、より具体的には、画面内のノード同

士を結ぶエッジ群と、画面内外にまたがるエッジ群とを、画面内で重ならないよう束ねるという、新しい性質を持つエッジバンドリング手法を目的としている。

### 3 Kernel Density Estimation Edge Bundling

本論文では、既存手法である Kernel Density Estimation Edge Bundling(KDEEB) を拡張することで focus+context を考慮したエッジバンドリング手法を提案する。そこで、本章では KDEEB について説明する。

#### 3.1 概要

KDEEB はカーネル密度推定を利用し、エッジの密集度合いを表すヒートマップを作成し、そのヒートマップに基づいてエッジを曲線化し、バンドリングする手法である。ヒートマップの作成や、その後のエッジのバンドリングの処理において、一般的な描画 API を通して GPU を利用するアルゴリズムであるため、数あるエッジバンドリング手法の中でも、Image-based な手法に分類され、既存手法に比べ高速である点が特徴的である。

#### 3.2 手法

KDEEB の入力はノード集合  $V \subset \mathbb{R}^2$  とエッジ集合  $E \subset V \times V$  の組  $G(V, E)$ 、カーネル密度推定に用いる初期バンド幅  $h$ 、イテレーション回数  $I$ 、スムージング回数  $N_s$ 、バンド幅減少量  $\epsilon \in (0, 1]$  であり、出力はバンドリングされたネットワーク  $G(B)$  である。

KDEEB では、(1) エッジの分割、(2) 密度マップの作成、(3) 勾配の計算、(4) 分割点の移動、(5) エッジのスムージングの 5 つのステップからなり、(1) から (5) を指定回数繰り返すことによって最終的なバンドリングが実現される。

(1) のエッジの分割では、エッジ集合に含まれる全てのエッジ  $e$  上に複数の分割点  $s \in \mathbb{R}^2$  を設ける。このとき、あるエッジ上の隣接する分割点間の間隔  $\delta$  はネットワーク全体を囲う最小矩形の 1% 程度を KDEEB では提案している [14]。

(2) の密度マップの作成では、全てのエッジの全ての分割点でカーネル密度推定を行い、二次元のカーネル密度推定量の曲面、すなわち密度マップ  $\rho$  を作成する。任意の点  $p \in \mathbb{R}^2$  における密度  $\rho(p)$  は、カーネル関数  $K$  とカーネルの初期幅  $h$ 、イテレーション数  $i = \{1, 2, \dots, I\}$  を用いて、次のように計算する。

$$\rho(p) = \sum_{e \in E} \sum_{s \in e} K \left( \frac{p - s}{h \times \epsilon^i} \right) \quad (1)$$

ここで、カーネル関数  $K$  にはガウスカーネルや Epanechnikov カーネルを用いる。バンド幅  $h$  にバンド幅減少量  $\epsilon$  を乗算することで、徐々にバンド幅  $h$  が小さくなり、分割点が局所的に集中していくこととなる。

(3) の勾配の計算では、有限差分法で (2) で得られた密度マップから各分割点における勾配ベクトルを計算する。

(4) の分割点の移動では、(3) で得られた密度マップの勾配  $\nabla \rho$  に従い、全てのエッジの全ての分割点を動かす。イテレーション  $i$  における、ある分割点  $s$  の移動量を表すベクトルは

$$\frac{ds}{di} = \frac{h \epsilon^i \nabla \rho(i)}{\max(\|\nabla \rho(i)\|, t)} \quad (2)$$

で計算する。 $\nabla \rho$  を正規化することで、分割点の移動幅をカーネルのバンド幅に制限している。分母の  $t$  は、 $\nabla \rho$  の大きさが 0 の場合に、0 による除算が発生しないようにするためのものであり、 $t = 10^{-5}$  程度に設定する。

(4) のエッジのスムージングでは、分割点の移動した後、全てのエッジに対して任意の回数、ラプラスアンスムージングを行うことでエッジの平滑化をする。すなわち、 $i$  回目の繰り返し処理における、あるエッジの  $k$  番目のスムージング後の分割点  $s_{i,k}$  は、 $i - 1$  回目の隣接するスムージング前の分割点  $s_{i-1,k-1}, s_{i-1,k+1}$  を用いて次のように計算する。

$$s_{i,k} = \frac{s_{i-1,k-1} + s_{i-1,k+1}}{2} \quad (3)$$

この処理を数回繰り返すことで、スムージングを行う。

### 4 提案手法

本研究では、ネットワーク拡大描画時のエッジを、以下の 3 種類に分類する。

- 画面内のノード間を結ぶエッジ (内内エッジと呼ぶ)
- 画面内と画面外のノードを結ぶエッジ (内外エッジと呼ぶ)
- 画面外のノード間を結ぶエッジ (外外エッジと呼ぶ)

各エッジの分類は、ズームやスクロール操作等による、画面のビューポート更新に伴い変更される。

これらを踏まえて、本研究では、次の 2 つの目的を達成するための新しいバンドリング方法を検討する。

- 拡大時のビジュアルクラッタを低減すること
- 拡大時に現れる、内内エッジの束と内外エッジの束を別々に違う場所でバンドリングすること

一つ目の目的については、既存手法の密度マップの解像度に着目し、解像度を変更することで、ビジュアルクラッタの低減に取り組む。

二つ目の目的については、本研究では 2 つの方法を提案する。一つ目の方法は、基本的な方法で、内内エッジと内外エッジのそれぞれの密度マップを作成し、それぞれ別々にバンドリングする方法である。二つ目の方法は、内内エッジの密度マップと、内外エッジの密度マップを別々に計算し、その差を取った密度マップをバンドリングに用いる方法である。

#### 4.1 密度マップの高解像度化 (方法 1)

KDEEB では、ネットワークを描画するスクリーンの大きさと同じ解像度の密度マップを作成することを提案していた。しかし、ネットワークの拡大を考慮すると、拡大した領域のみで、スクリーンの大きさに等しい密度マップを作成しても、拡大した領域に含まれる分割点しか移動させることができない。すなわち、エッジのビューポートに含まれる部分のみがバンドリングされ、それ以外の部分はバンドリングされないことになる。この場合、内内エッジはバンドリングすることができるが、内

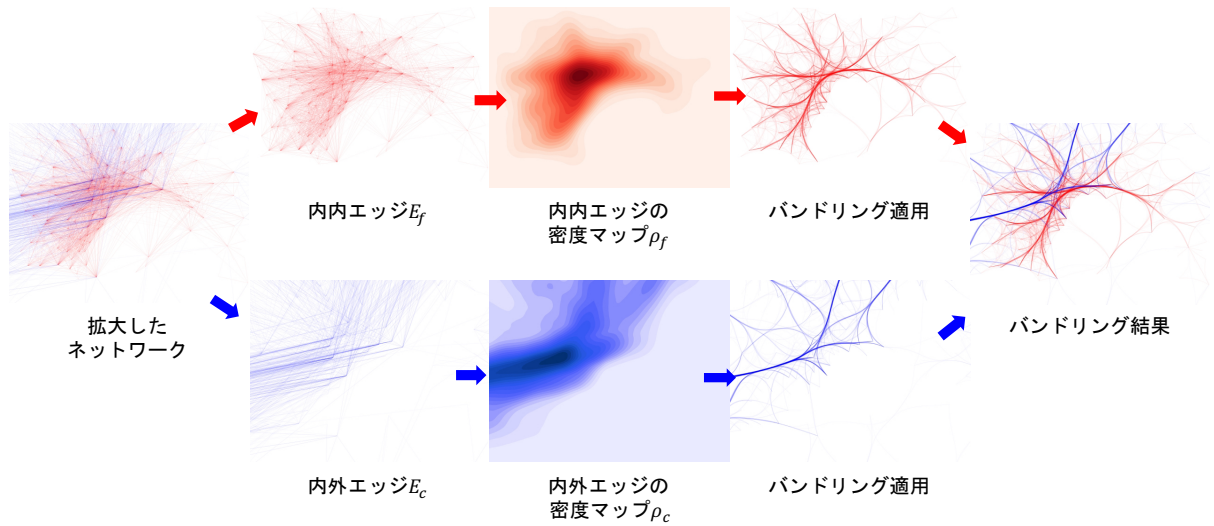


図 2: 別々バンドリング (方法 2-1) の手順の概要

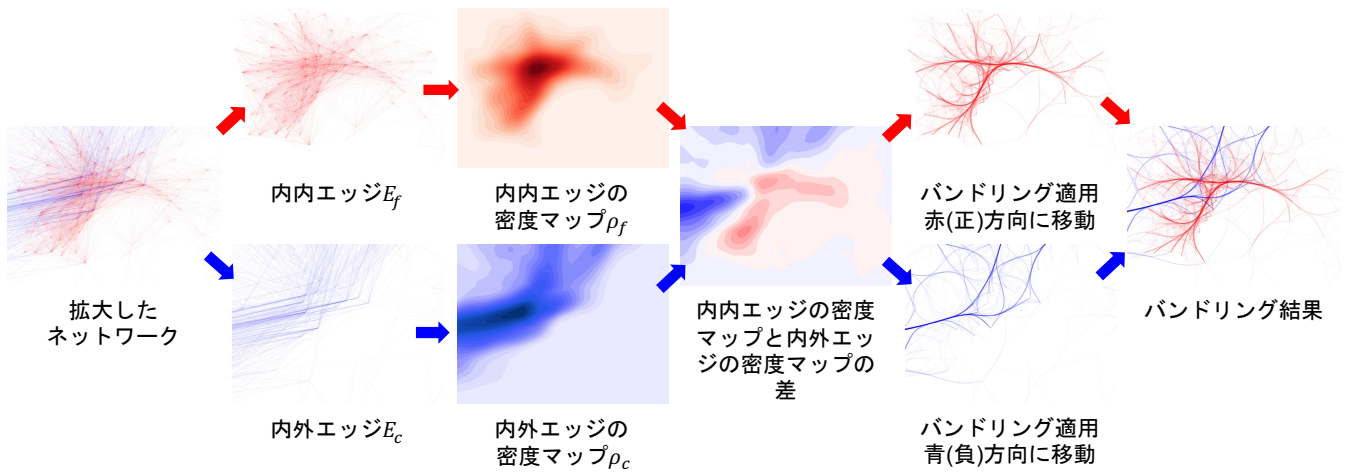


図 3: 別領域バンドリング (方法 2-2) の手順の概要

外エッジはビューポート外でバンドリングされなくなってしまう。内外エッジはビューポート外でバンドリングされない、ビューポート内で内外エッジによるビジュアルクラッタを十分に取り除けない。

一方で、スクリーンの大きさと同じ解像度で拡大した領域のみならず、ネットワーク全体の密度マップを作成すると、分割点の密集度合いに対して勾配を計算するための密度マップの解像度が低く、多くの分割点で勾配の方向が等しくなり、大雑把なバンドリングしかできないという問題がある。

そこで本研究では、密度マップの解像度をスクリーンの大きさよりも大きくして、密度マップを作成する。

#### 4.2 別々バンドリング (方法 2-1)

独立バンドリング (方法 2-1) は、内内エッジと内外エッジを分離してバンドリングする単純な方法であり、内内エッジと内外エッジを、それぞれ異なる密度マップを用いて独立にバンドリングし、結果を重ね合わせる方法である (図 2)。すなわち、内内エッジ群  $E_f$  と内外エッジ群  $E_c$  でそれぞれ、独立に内内エッジ群の密度マップ  $\rho_f$  と内外エッジ群の密度マップ  $\rho_c$  を作成し、内内エッジ群  $E_f$  を内内エッジ群の密度マップ  $\rho_f$  でバ

ンドリングし、内外エッジ群  $E_c$  を内外エッジ群の密度マップ  $\rho_c$  でバンドリングする。これらの結果を重ね合わせて描画することで、内内エッジと内外エッジを分離したバンドリング結果を実現する。

#### 4.3 別領域バンドリング (方法 2-2)

別領域バンドリング (方法 2-2) は、内内エッジと内外エッジの束が重ならないように、画面内の異なる領域でバンドリングするような密度マップを用いる方法である (図 3)。KDEEB ではエッジの分割点の密度が高い方へ分割点を移動させる。一方で分割点の密度の低い場所には分割点は移動せず、むしろ避けるようになる。そこで方法 2-2 では、方法 2-1 と同様に、内内エッジ群  $E_f$ 、内外エッジ群  $E_c$  でそれぞれ密度マップ  $\rho_f$ 、 $\rho_c$  を作成し、その差を取った差分密度マップの極大方向、極小方向に内内エッジ、内外エッジをそれぞれ動かすことで、内内エッジと内外エッジを、それぞれ画面内の異なる領域でバンドリングすることを狙う。差分密度マップ  $\rho_{sub}$  を次のように計算する。

$$\rho_{sub} = \frac{\rho_f}{\max(\rho_f)} - \frac{\rho_c}{\max(\rho_c)} \quad (4)$$

$\max(\rho)$  はそれぞれの密度マップの最大の値である。したがっ

て、 $\frac{\rho}{\max(\rho)} \in [0.0, 1.0]$  となるようにする。これは、内内エッジの密度マップと内外エッジの密度マップの密度推定量がどちらか一方に極端に偏っていても、意図せぬ密度マップの極値に分割点が集中しないようにするためである。

この密度マップ  $\rho_{sub}$  を用いて、内内エッジ群  $E_f$ 、内外エッジ群  $E_c$  の分割点  $s$  の移動ベクトルはそれぞれ、次のように計算する。

$$\frac{ds}{di} = \frac{h \epsilon^i \nabla \rho(i)}{\max(\|\nabla \rho(i)\|, t)} \times d \quad (5)$$

$$d = \begin{cases} 1 & (s \in e \in E_f) \\ -1 & (s \in e \in E_c) \end{cases} \quad (6)$$

すなわち、内内エッジの分割点ならば、差を取った密度マップ  $\rho_{sub}$  で推定される勾配  $\nabla \rho_{sub}$  の方向に、内外エッジならば、その逆方向に分割点を移動させる。

## 5 適用結果

本章では、既存手法および提案手法を国内移住ネットワークデータに適用して、結果を観察する。既存手法、提案手法は python3.6.8 で実装し、CentOS8.3, Intel Xeon E312xx (Sandy Bridge)6 コア 2.2GHz, メモリ 8GB 上で実験を行った。入力に用いたパラメータは既存手法では、初期バンド幅  $h = 20$ 、イテレーション回数  $I = 8$ 、スムージング回数  $N_s = 5$  回、バンド幅減少量  $\epsilon = 0.7$ 、分割点間の間隔  $\delta = 0.008$ 、密度マップの解像度を  $300 \times 300$  とした。提案手法の解像度の変更では、分割点間の間隔  $\delta = 0.0008$ 、密度マップの解像度を  $3200 \times 3200$  とした。これらのパラメータは Hurter らの研究 [14] を参考に経験的に決定した。

それぞれの方法で可視化した結果をを図 4 に示す。図 4 はそれぞれ、次の結果を示している。

- (a) 既存手法 KDEEB を適用した後、拡大した結果
- (b) 密度マップの解像度を変えた結果 (方法 1)
- (c) 解像度を変更し、内内エッジ、内外エッジをそれぞれバンドリングした結果 (方法 1+方法 2-1)
- (d) 解像度の変更し、内内エッジの密度マップと内外エッジの密度マップの差を用いてバンドリングした結果 (方法 1+方法 2-2)

まず、ビジュアルクラッタ低減の効果について観察する。既存手法を適用し、拡大した結果である、図 4(a) では、内外エッジは少しだけ、エッジが曲線で描かれているが、内内エッジについては、ほとんど直線で描かれており、以降の提案手法に比べ、ビジュアルクラッタ低減の効果は低いと言える。

提案手法の解像度を高くした結果である、図 4(b)、図 4(c)、図 4(d) では、図 4(a) に比べ、大きくビジュアルクラッタが低減しており、ビジュアルクラッタ低減の効果は高いと言える。中でも、図 4(b) は、図 4(c)、図 4(d) に比べ、内内エッジ、内外エッジを分けていない分、ビジュアルクラッタ低減の効果は高いように見える。

続いて、内内エッジと内外エッジの束の分離について観察

する。

図 4(a) では、バンドリングの効果が弱いことから、内内エッジと内外エッジそもそも束になっておらず、分離の性能について評価できない。

提案手法の解像度を高くした結果である、図 4(b) では、内内エッジの束と内外エッジの束が重なっており、内内エッジよりも手前に描画された内外エッジによって、内内エッジが隠れ、内内エッジの様子が読み取りにくくなっている。したがって、内内エッジ、内外エッジ分離の性能は低いと言える。

提案手法の解像度を高くし、内内エッジ、内外エッジをそれぞれバンドリングした結果である、図 4(c) では、例えば、図 4(b) と比べて、西に向かう内外エッジの束が内内エッジの束と別々の場所にまとまり、それぞれ読み取りやすくなったが、北東方向に向かう内外エッジの束は画面中心付近で内内エッジの束と重なってしまい、それぞれの様子が読み取りにくい。

密度マップに内内エッジの密度マップと内外エッジの密度マップの差を用いた結果である、図 4(d) では、内内エッジの束と内外エッジの束が違う場所にまとまり、図 4(b) に比べ、内内エッジの様子がより読み取りやすくなった。

## 6 評価

本章では、提案したエッジバンドリング手法に対して、ビジュアルクラッタ低減、内内エッジ、内外エッジ分離の性能を定量的に評価する。

### 6.1 ビジュアルクラッタ低減の性能の観点の評価

Saga らは、ビジュアルクラッタの低減を目的としてエッジバンドリングを適用したネットワーク可視化結果に対する、構造美学的な評価指標を複数提案している [20]。また筆者らは Saga らの評価指標を拡大描画した可視化結果の評価指標として拡張した [19]。本研究では、この拡張した指標を用い、ビジュアルクラッタ低減の性能を評価する。本章では、各指標について説明する。

#### 6.1.1 MELD'

エッジバンドリングではエッジを曲線化することにより、エッジを束ね、ビジュアルクラッタを低減するが、過度にエッジを曲線化すると、エッジの読み取りが難しくなる。MELD' (Mean Edge Length Difference) はエッジのビューポート内に描画されている部分がどの程度曲線化されたかを示す指標である。小さいほど、バンドリング前に比べ、曲線化されていないことを表し、良い結果といる。同指標はビューポート内のバンドリング前後のエッジ長の差の平均として、次のように定義される。

$$MELD' = \frac{1}{\|E\|} \sum_{e \in E} |L_V'(e) - L_V(e)| \quad (7)$$

ここで、 $L_V(e)$  はバンドリング前における、エッジ  $e$  のビューポート内に含まれる部分の長さ、 $L_V'(e)$  はバンドリング後における、エッジ  $e$  のビューポート内に含まれる部分の長さである。

#### 6.1.2 MOAD

エッジバンドリングは、曲線化したエッジ同士を重ねること



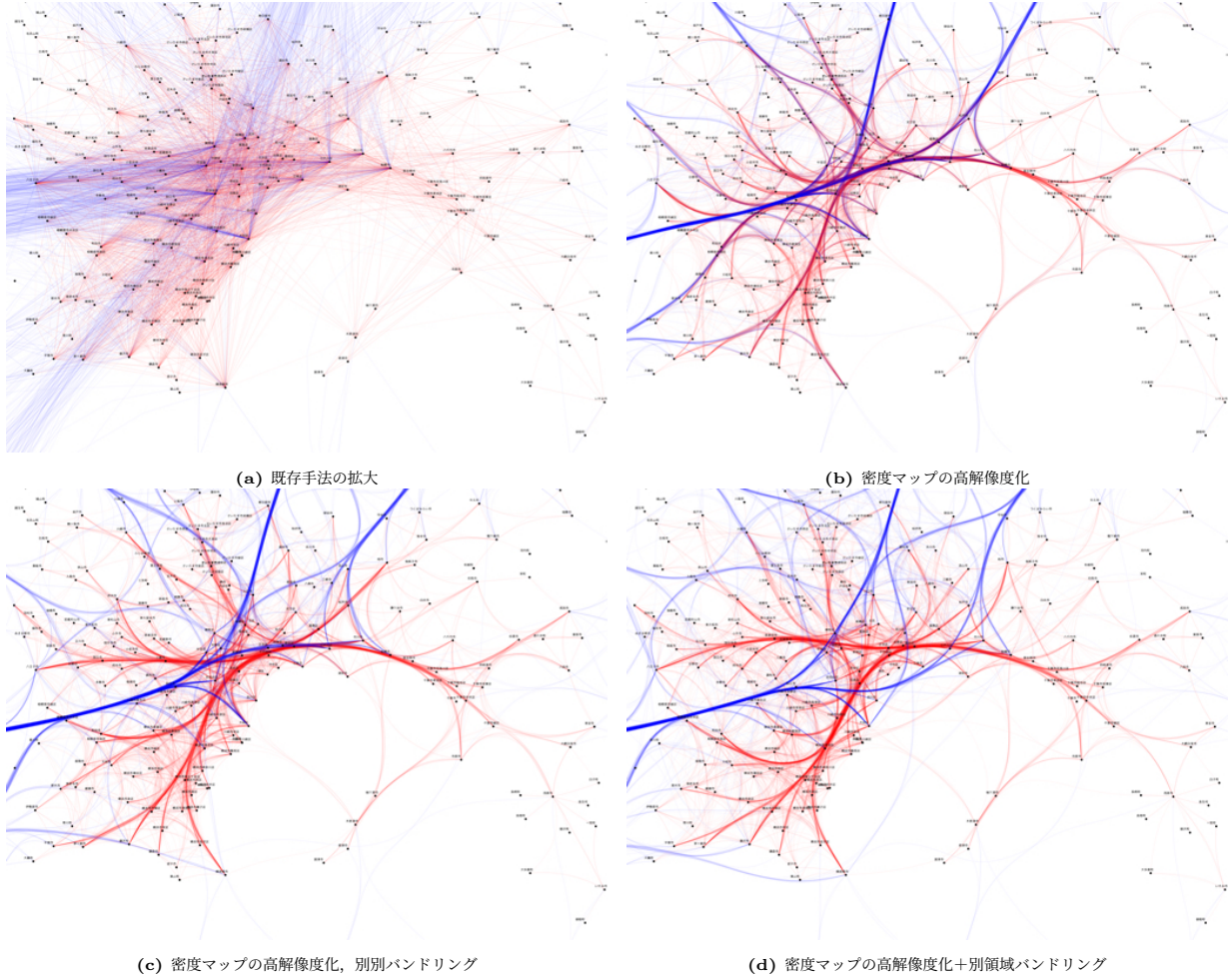


図 4: 既存手法, 提案手法の可視化結果 (内内エッジ:5775, 内外エッジ:1809, ノード:195)

で, 各エッジの描画面積を減少させることで, ビジュアルクラッタを低減する.  $MOA'$  (Mean Occupation Area') は拡大描画した領域内での, エッジが描画された領域の大きさを評価する指標である. 同指標はビューポート内のエッジを描画するピクセルが閾値を超えた, エリアと呼ばれる部分領域のピクセル集合の数の平均として, 次のように定義される.

$$MOA' = \left\| \left\{ a \mid a \in A_V, \left\| \bigcup_{e \in E} p(a, e) \right\| > \epsilon_o \right\} \right\| / \|A_V\| \quad (8)$$

ここで,  $\epsilon_o$  は閾値,  $p(a, e)$  はエリア  $a$  におけるエッジ  $e$  を描画するピクセルの集合である. エリア  $a$  は注目領域  $V$  が表示されている画面を格子状に分割して生成する正方形の集合  $A_V$  に含まれるピクセル集合の一つである.  $MOAD$  (Mean Occupation Area Difference) はバンドリングする前と後の同指標の差であり,  $MOAD$  が大きいほど, バンドリングによりエッジを描画する面積が小さくなったことを表す.

### 6.1.3 REDDD

エッジバンドリングは, エッジを束ねることで, ビューポートに散在していたエッジが複数箇所にまとめ, エッジを描画している部分とそうでない部分の疎密が明確化する.  $REDD$  (Relative Edge Density Distribution) はエリアごとのエッジを描画するピクセル数の疎密を評価する指標である. 同指標はビューポ-

ート内のエリアごとのエッジを描画するピクセル数の相対平均偏差として, 次のように定義される.

$$REDD = \frac{1}{\bar{r} \|A_V\|} \sum_{a \in A_V} |r(a) - \bar{r}| \quad (9)$$

$$r(a) = \frac{\left\| \bigcup_{e \in E} p(a, e) \right\|}{\|a\|} \quad (10)$$

$REDDD$  (Relative Edge Density Distribution Difference) はバンドリングする前と後の同指標の差であり,  $REDDD$  が大きいほど, バンドリングにより, エッジを描画するピクセルと, 描画しないピクセルの疎密が明確になったことを表す.

## 6.2 内内エッジ, 内外エッジ分離の性能の評価

本研究では, ビューポート内外の focus+context として, 内内エッジ群と内外エッジ群が区別できるように, それぞれ別々に, 違う場所にまとめるバンドリングする方法を提案した. この性能に関して, 内内エッジ群, 内外エッジ群を描画するピクセルの空間的な分離の度合いを評価する.

### 6.2.1 評価手法

内内エッジ, 内外エッジ分離の性能に関して, 内内エッジを描画するピクセル, 内外エッジを描画するピクセルは, それぞれ密集している方が良いが, 内内ピクセル, 内外ピクセルは互

表 1: 国内移住ネットワーク、関東周辺での適用の評価結果

	MELD'	MOAD	REDDD	CCP
(a) 既存手法	0.000143	0.0109	0.0299	0.0531
(b) 方法 1	0.000805	0.192	0.273	0.114
(c) 方法 1+方法 2-1	0.000683	0.175	0.255	-0.185
(d) 方法 1+方法 2-2	0.000707	0.128	0.175	-0.300

いに密集していない方が良い。そこで、本研究では、Saga らが提案した方法 [20] と同様に、可視化結果を格子状に分割して正方形の部分領域（エリア）を生成し、それぞれの正方形（エリア）における、内内エッジと内外エッジを描画するピクセル数の相関係数で評価する。

注目領域  $V$  の区切られたあるエリア  $a$  におけるエッジ  $e$  を描画するピクセルの集合を  $p(a, e)$  とし、あるエリア  $a$  の内内エッジ群  $E_f$ 、内外エッジ群  $E_c$  を描画するピクセル数  $P_{a,E_f}$ 、 $P_{a,E_c}$  はそれぞれ次のように表せる。

$$P_{a,E_f} = \left\| \bigcup_{e \in E_f} p(a, e) \right\| \quad P_{a,E_c} = \left\| \bigcup_{e \in E_c} p(a, e) \right\| \quad (11)$$

これを用いて、内内エッジのピクセル数と内外エッジのピクセル数の相関係数  $CCP$  (Correlation Coefficient of Pixel) は次のように表せる。

$$CCP = \frac{\sum_{a \in A_V} (P_{a,E_f} - \bar{P}_{a,E_f})(P_{a,E_c} - \bar{P}_{a,E_c})}{\sqrt{\sum_{a \in A_V} (P_{a,E_f} - \bar{P}_{a,E_f})^2} \sqrt{\sum_{a \in A_V} (P_{a,E_c} - \bar{P}_{a,E_c})^2}} \quad (12)$$

$CCP$  が  $-1.0$  に近いほど、内内エッジを描画するピクセルと内外エッジを描画するピクセルが同じ場所にまとまっていないことを表す。

### 6.3 評価結果と考察

図 4 に対する評価結果を表 1 に示す。図 4(a) から図 4(d) が、表 1 の (a) から (d) にそれぞれ対応する。

クラッタ低減の性能について比較すると、提案手法の (b)(c)(d) のいずれもが、既存手法 (a) と比較して MOAD, REDDD 共に大きく、性能が高いと言える。また、内内、内外を区別しないでバンドリングする提案手法 (b) が、内内、内外を区別してバンドリングする提案手法 (c)(d) よりも大きかった。エッジの曲率を抑える性能については、提案手法 (b)(c)(d) は既存手法 (a) に比べ、MELD が大きく、性能が低い。

内内の束、内外の束の分離の性能について比較すると、内内エッジ、内外エッジを区別した提案手法 (c)(d) は区別しなかった (a)(b) に比べ、 $CCP$  の値が  $-1.0$  に近く、より内内ピクセルと内外ピクセルが分離していると言える。また、単純に内内エッジと内外エッジを別々にバンドリングした提案手法 (c) に比べ、密度マップの差を用いた提案手法 (d) はより  $CCP$  の値が  $-1.0$  に近いので、最も内内ピクセル、内外ピクセルを分離できていると言える。

以上の結果から、密度マップの高解像度化はビジュアルクラッタ低減の効果が示された。また、内内エッジと内外エッジをそれぞれ分離して束ねることには、密度マップの差を用いることが効果的であることが示された。

## 7 実装

本研究の提案手法 (密度マップの高解像度化+別領域バンドリング) を、GPU を利用する処理として実装し、実行性能を確認する実装には WebGL2 を用いる。WebGL2 は 2022 年現在、Google Chrome や FireFox, safari など、一般的に使用されるモダンブラウザで特別な設定をすることなく、使用が可能である。また、PC だけでなく、スマートフォンやタブレットでも動作可能である。

本実装では、分割点の生成処理やエッジのスムージング処理は CPU 上で逐次処理し、バンドリング処理における密度マップ作成と全分割点の移動処理を GPU で実行する。

5 章で用いた国内移住ネットワークに加えて、米国航空路線ネットワーク [21]、米国移住ネットワーク [21] の一部の領域を拡大して提案手法を適用した結果を図 5 に、その計算時間を表 2 に示す。なお実行には MacBook Air(M1, 2020), Apple M1, メモリ 16GB, 8 コア GPU を用い、Google Chrome 上で行った。

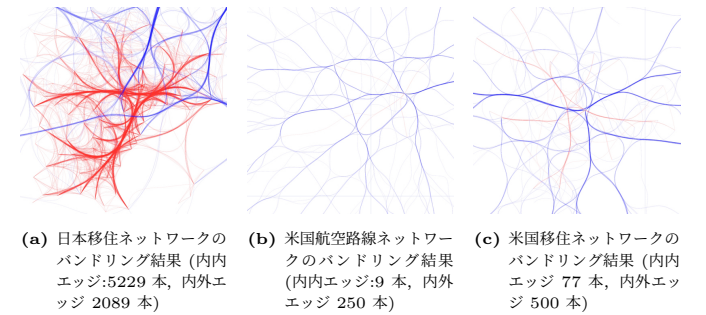


図 5: WebGL2 で実装した提案手法のバンドリング結果

表 2: 図 5 のバンドリングの実行時間

	合計エッジ本数	実行時間 (ミリ秒)
日本移住ネットワーク	7318	6614.9
米国航空路線ネットワーク	259	2264.6
米国移住ネットワーク	577	2639.4

表 2 を見ると、例えば、日本移住ネットワークは米国航空路線ネットワーク約 28 倍ほどのエッジがあるが、実行時間はおよそ 3 倍程度に収まっている。WebGL2 による並列処理のおかげでエッジ数が増加しても、十分にスケールしていることが確認できた。一方でインタラクティブなアプリケーションを想定すると、計算に 6 秒程度かかってしまうのは問題がある。今回、並列処理を行わなかった、分割点の生成処理やエッジのスムージング処理なども、Image-based な手法で行う方法を考える必要がある。

## 8 結 論

本論文では既存のエッジバンドリング手法適用結果を単純に拡大すると、再びビジュアルクラッタが発生する問題と、拡大した際に現れる、画面内のノード間を結ぶエッジ群、画面内外をまたがるエッジ群を区別してバンドリングするという課題に取り組んだ。既存手法の密度マップの解像度を変更し、それぞれのエッジ群で密度マップを作って差を取り、極大方向と極小方向へそれぞれエッジを曲げる新たなエッジバンドリング手法を提案した。実データに適用した結果、提案手法は拡大描画時にビューポート内でビジュアルクラッタを低減しつつ、内内エッジと内外エッジを分けてバンドリングすることに成功した。ビジュアルクラッタ低減の性能と、内内エッジ、内外エッジ分離の性能を定量的に評価した結果、既存手法と比較し、提案手法の有用性が示された。今後の課題としては、パラメータ調整の自動化、評価手法の体系化等が挙げられる。

## 文 献

- [1] Zhou, H., Xu, P., Yuan, X. and Qu, H.: Edge bundling in information visualization, *Tsinghua Science and Technology*, Vol. 18, No. 2, pp. 145–156 (2013).
- [2] 総務省: 平成 27 年国勢調査 移動人口の男女・年齢等集計 (人口の転出入状況) 全国結果, 総務省 (オンライン), [https://www.e-stat.go.jp/stat-search/files?page=1&layout=datalist&toukei=00200521&tstat=000001080615&cycle=0&tclass1=000001093875&tclass2=000001093876&stat\\_infid=000031518570&tclass3val=0](https://www.e-stat.go.jp/stat-search/files?page=1&layout=datalist&toukei=00200521&tstat=000001080615&cycle=0&tclass1=000001093875&tclass2=000001093876&stat_infid=000031518570&tclass3val=0).
- [3] 国土交通省: 国土数値情報 (市区町村役場データ), 国土交通省 (オンライン), <https://nlftp.mlit.go.jp/ksj/gml/datalist/KsjTmplt-P34.html#prefecture02>.
- [4] Herman, I., Melancon, G. and Marshall, M.: Graph visualization and navigation in information visualization: A survey, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 6, No. 1, pp. 24–43 (2000).
- [5] Lhuillier, A., Hurter, C. and Telea, A.: State of the Art in Edge and Trail Bundling Techniques, *Computer Graphics Forum*, Vol. 36, pp. 619–645 (2017).
- [6] Holten, D.: Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 12, No. 5, pp. 741–748 (2006).
- [7] Cui, W., Zhou, H., Qu, H., Wong, P. C. and Li, X.: Geometry-Based Edge Clustering for Graph Visualization, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 14, No. 6, pp. 1277–1284 (2008).
- [8] Holten, D. and Van Wijk, J. J.: Force-Directed Edge Bundling for Graph Visualization, *Computer Graphics Forum*, Vol. 28, No. 3, pp. 983–990 (2009).
- [9] Selassie, D., Heller, B. and Heer, J.: Divided Edge Bundling for Directional Network Data, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 17, No. 12, pp. 2354–2363 (2011).
- [10] 山下敬史, 佐賀亮介: RO-003 多属性グラフへのエッジバンドリングの適用 (O 分野: 情報システム, 査読付き論文), 情報科学技術フォーラム講演論文集, Vol. 14, No. 4, pp. 43–48 (2015).
- [11] Zielasko, D., Weyers, B., Hentschel, B. and Kuhlen, T.: Interactive 3D Force-Directed Edge Bundling, *Computer Graphics Forum*, Vol. 35, pp. 51–60 (2016).
- [12] Ersoy, O., Hurter, C., Paulovich, F., Cantareiro, G. and Telea, A.: Skeleton-Based Edge Bundling for Graph Visualization, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 17, No. 12, pp. 2364–2373 (2011).
- [13] Zhu, D., Wu, K., Guo, D. and Chen, Y.: Parallelized Force-Directed Edge Bundling on the GPU, *2012 11th International Symposium on Distributed Computing and Applications to Business, Engineering Science*, pp. 52–56 (2012).
- [14] Hurter, C., Ersoy, O. and Telea, A.: Graph Bundling by Kernel Density Estimation, *Computer Graphics Forum*, Vol. 31, No. 3pt1, pp. 865–874 (2012).
- [15] van der Zwan, M., Codreanu, V. and Telea, A.: CUBu: Universal Real-Time Bundling for Large Graphs, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 22, No. 12, pp. 2550–2563 (2016).
- [16] Lhuillier, A., Hurter, C. and Telea, A.: FFTEB: Edge bundling of huge graphs by the Fast Fourier Transform, *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 190–199 (2017).
- [17] Peysakhovich, V., Hurter, C. and Telea, A.: Attribute-driven edge bundling for general graphs with applications in trail analysis, *Proceedings of IEEE Pacific Visualization Symposium (PacificVis)*, IEEE, pp. 39–46 (2015).
- [18] Lyu, Y., Liu, X., Chen, H., Mangal, A., Liu, K., Chen, C. and Lim, B.: OD Morphing: Balancing Simplicity with Faithfulness for OD Bundling, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 26, No. 1, pp. 811–821 (2020).
- [19] 秋山桂一, 藤田秀之, 大森 匡, 新谷隆彦: ネットワーク可視化における拡大描画に適したエッジバンドリング手法, 情報処理学会論文誌, Vol. 63, No. 3, pp. 817–830 (2022).
- [20] Saga, R.: Quantitative Evaluation for Edge Bundling Based on Structural Aesthetics, *Proceedings of the Eurographics / IEEE VGTC Conference on Visualization: Posters, EuroVis '16*, Goslar, DEU, Eurographics Association, pp. 17–19 (2016).
- [21] Sugar, C.: d3.ForceBundle, GitHub (online), <https://github.com/upphiminn/d3.ForceBundle>.