

## SNSを俯瞰する都市型レイアウト形成システム

## City-like Layout System to Overlook a Social Networking Service

川村 真人†  
Masato Kawamura高井 昌彰‡  
Yoshiaki Takai

## 1 はじめに

近年 SNS の利用人口が急速に増大している。このようなソーシャルメディア人口の増大に伴って起こる問題の一つに、人々が持つ関係性の複雑化が挙げられる。複雑な関係性を視覚的に把握するための可視化手法は古くから研究されており、グラフ構造を力学シミュレーションによってレイアウトするばねモデル[1]はその典型的な一例である。しかし、このような可視化手法だけでは複雑化した SNS に見られるようなグラフ構造に対して不十分である。

一方、グラフ構造の可視化に関する問題に対して、関係性を現実世界の地勢的な構造に対応づけて可視化するアプローチがある。Topigraphy[2]はタグクラウド表現に地形図のメタファを導入することで、タグ間の関連性を視覚的に分かり易く表現する。HatenarMaps[3]はブログホスティングサービスの「はてなダイアリー」が形成するコミュニティを地図として可視化するものであり、その応用であるBlogopolis[4]はブログランキングサイトの TopHatenar が収集したデータから3D都市景観を自動生成する。このようなアプローチを SNS が持つソーシャルグラフに適用することで、視覚的に分かり易い可視化を実現できると考えられる。

そこで本稿では、現実世界の地勢的な構造である都市景観を利用して SNS のソーシャルグラフの可視化を行い、俯瞰による関係性の把握が可能な都市型レイアウト形成システムを提案し、その可視化の実行結果について述べる。

## 2 関連研究

本節では現実世界の地勢的な構造をアナロジーとして用い可視化に役立っている関連研究を挙げ、その特徴と有用性について述べることで本研究の位置づけを明らかにする。

## 2.1 Topigraphy – タグクラウドの可視化

タグクラウドはウェブサイト等で使用される表現手法の一つである。タグ群を雲のように表示し、参照数などのパラメータに応じてタグのフォントサイズや色を変化させることで利用者は具体的な数値を確かめることなくタグの性質を推し量ることができる。

しかし、一般的なタグクラウド表現は、一定数以上のタグを表示すると適切なタグの選択が困難になるという問題がある。これは、目的と関連の薄いタグを表示することで選択の妨げとなるためである。

このような問題に対し Topigraphy はタグクラウド表現に地形図のメタファを導入した(Fig.1)。関連性の高いタグを近くに配置することで目的と関連の薄いタグを除外し、参照数などのパラメータを地形の高度に見立てた等高線を用いて表現することで、適切なタグの選択を容易にする。このように、地形図を用いた表現はデータの関連性を一度に表す場合に効果的である。

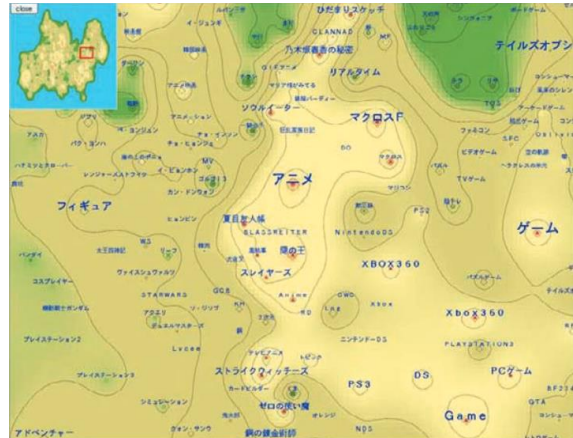


Fig.1 タグクラウドを地形図で表現する Topigraphy[2]

## 2.2 HatemarMaps – ブログコミュニティの可視化

ブログホスティングサービスの「はてなダイアリー」は、そのユーザーがコミュニティ・クラスターを形成しており、その独特のコミュニティは「村社会」を比喻して「はてな村」と呼ばれることがある。HatenarMaps では、「村」に例えるというアナロジーを用いて「はてな村」を可視化し、俯瞰的にコミュニティの動向を把握することができるウェブサービスである(Fig.2)。

HatenarMaps における「はてなダイアリー」の各ユーザーには獲得したブックマーク数に応じた領土が割り当てられ、その領土は関連するユーザーの領土が隣接するように配置される。これにより注目度の高い記事を持つユーザーが大きな領土を持つことで発見が容易になるとともに、関連性の強い記事が隣接する領土として配置されるため必要な情報を見つけ易くなる。Topigraphy と同様に、地図表現は関連性を一度に表す際に効果的である。

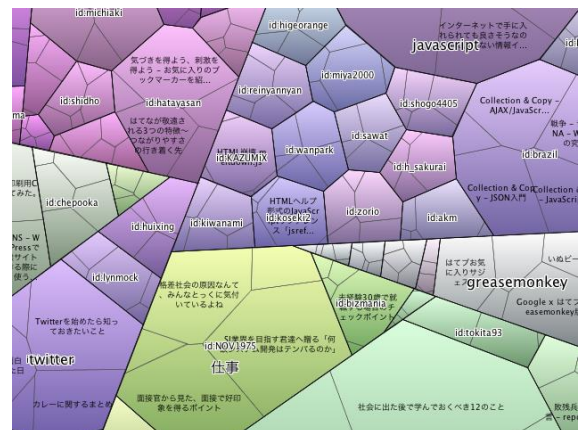


Fig.2 ブログコミュニティを可視化する HatemarMaps[3]

†北海道大学大学院情報科学研究科, Graduate school of Information Science and Technology, Hokkaido University

‡北海道大学情報基盤センター, Information Initiative Center, Hokkaido University

### 2.3 Blogopolis – 3D 都市景観を利用した可視化

HatenarMaps の応用である Blogopolis は、ブログランキングサイトの TopHatenar に登録された 20 万件超のブログ記事が持つ関係性を 3D 都市景観表現で可視化したものである (Fig.3).

3D 都市景観表現を用いることで、無数に存在するブログ記事から生成される複雑な構造を一望することができる。ブックマークエントリーをオフィスビルとして表現し、購読者数に応じてビルを高くすることで、エントリーの注目度を視覚的に分かり易く表現している。HatenarMaps と同様に、ユーザーの持つブックマーク数に応じて領土が割り当てられ、ブログ記事の持つタグの内容に応じてクラスタリングを行うため、必要な情報を見つけやすい。本研究は、この Blogopolis との共通点が多い。



Fig.3 ブログ記事から都市景観を生成する Blogopolis[4]

### 3 都市型レイアウト形成システムの概要

前節で述べたように、複雑な関係性に対して現実世界の地勢的構造を可視化に用いることで、視覚的に分かり易い表現を行える。特に、「はてな村」に挙げられるような強い社会性を持つ構造に対しては、地勢的構造の持つ地域性がクラスタの表現を行う際に効果的であると考えられる。そこで、本研究ではこのアプローチを、近年の SNS が持つ複雑な関係性をグラフ構造に落とし込んだソーシャルグラフに適用する。

詳細を示す前に、対象とする SNS について述べる。流行している SNS は多数存在し、またそれらが保持するデータの内容は異なるため、可視化の対象として最も適切なものを選ぶ必要がある。現在、日本で利用率の高い SNS は Twitter, Facebook, Line の三つである。可視化を行う際に必要な大量のデータを得るためには、SNS が保有する利用者数は重要なファクターであるが、これら三つの SNS における利用者数の差はあまり見られなかった。そこで、取得できるデータの性質について着目する。コミュニティの形成手法が閉鎖的な Facebook や Line に比べ、Twitter は公開されているデータが多く、データ取得用の API が豊富に用意されているという利点がある。ユーザーの性質やユーザー間の類似度を計算する際には、出来るだけ多くのデータが公開されていることが望ましい。以上の理由から、本研究のデータ取得対象として Twitter を選ぶ。

Twitter から取得できるデータは、前述した関連研究である Blogopolis で利用されているデータと同様の性質を持つ。例えば、Blogopolis におけるブログエントリーを Twitter のユ

ーザーと対応付けることができる。従って Blogopolis の手法と同様に、ユーザーの性質をオフィスビルといった建物に反映させ、これらを適切に配置することで、効果的な可視化を実現できると考えられる。

Blogopolis ではブログ記事に付与されているタグ情報を用いて K-means 法でクラスタリングを行い、得られたデンドログラムを入力とした重み付きボロノイ図の領域分割手法に従ってノードの各領域を定める。しかし、実際にこの手法を Twitter から得られるソーシャルグラフに適用すると、ブログ記事のタグに相当するものがないという問題が生じる。同様の手法を用いるためには、フォロー関係やツイート内容から類似度を計算する必要があるが、これを適切に用いることは難しい。

そこで、本研究では従来手法であるばねモデルを用いてノードの位置決定を行う。ばねモデルによりレイアウトされたグラフにおける「ノードが重ならない」という特徴は都市景観を生成する際に最も重要な性質であり、加えて、実装方法がシンプルであるという点においてメリットがある。また、ばねモデルは時間変化によって随時変形していくため、都市の時間発展を可視化できるという狙いもある。

以上の検討に基づき、本稿で提案する都市型レイアウト形成システムの概要を Fig.4 に示す。Twitter から取得されるデータにおいてユーザーをノード、ユーザー間のフォロー関係をエッジとしたグラフを作成し、それをばねモデルレイアウトで変形する。得られたノードとエッジの座標を用いて建物や道路を配置する (Fig.5)。このようにして 3D 都市景観を自動生成することで、Twitter のユーザーが持つ関係性を俯瞰的に把握することができる。

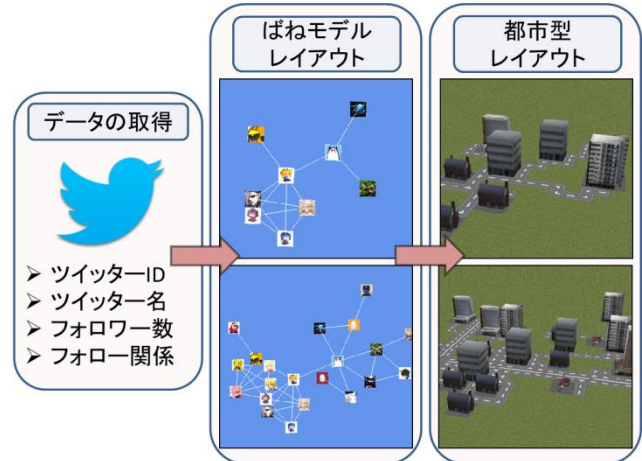


Fig.4 システムの概要

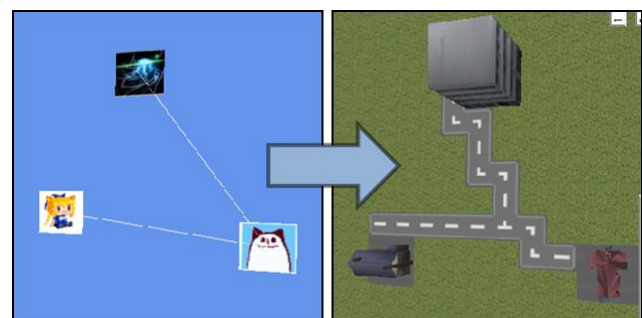


Fig.5 ノードとエッジの可視化方法

### 3.1 建物の配置

ばねモデルのレイアウトから得られたノードの配置を利用して建物の初期位置を決定する。ノードに対応する建物が配置される都市平面は、一定の大きさを有する格子状の区画に整理されているものとする。ばねモデルでのノード配置に基づき、同一区画への重複を排除しつつ、最も近い区画に建物を配置する(Fig.6)。

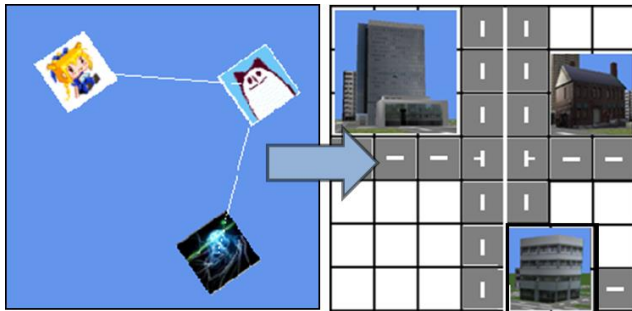


Fig.6 建物の格子配置

一般に、建物はその大きさに比例してその利用者数も増えていくものである。例えば、小さな一軒家などはその家族および知り合いが訪れる程度のものであるのに比べ、大規模マンションや複合商業ビルの利用者数はずっと多い。そこで本システムでは、区画に配置される建物の大きさを、対応するノード(Twitter ユーザ)のフォロワー数から定める(Fig.7)。



Fig.7 フォロワー数に応じて建物の規模を変化させる

### 3.2 道路の配置

前述したように、ソーシャルグラフのノード間を結ぶエッジは都市空間の道路で表現する。

格子状の都市空間では、道路は必ずしも建物間を最短距離で結ぶ直線ではなく、区画配置の制約を受けるのが自然である。そこで、建物間の接続性を満たしつつ、区画の制約条件を満たすような自然な道路配置を行う必要がある。

建物間の接続性を保つために、二つのノード間のエッジを格子平面で覆い尽くすように道路を初期配置する。この結果、不自然な多数の折れ曲がりや不要な分岐を持った道路が形成されるため、都市空間として適切と考えられるような整形を行う必要がある(Fig.8)。

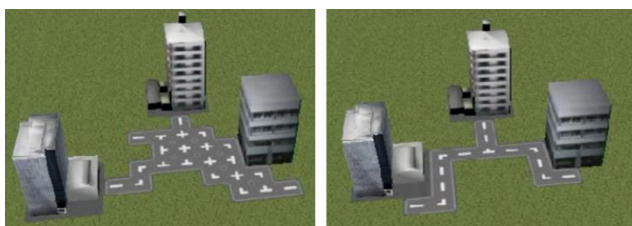


Fig.8 不適切な団子状道路(左)と整形された適切な道路(右)

道路の整形を行うために、道路を構成する各格子に対して接続数を定義する。接続数とは、ある道路マスに対するその上下左右にある道路マスの個数を指す。例えば、Fig.9の四角で囲まれている道路マスは下左右の3方向に繋がっているため、接続数は3となる。

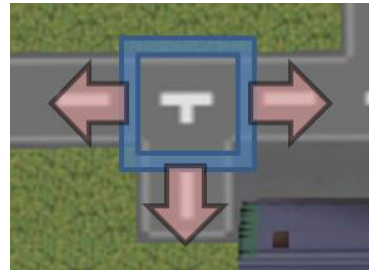


Fig.9 接続数3の道路マス

この接続数の総和に着目すると、道路の適切さが判別できる。例えば、Fig.10の左図は右図に比べて総接続数が少ないため、より適切な道路であると言える。

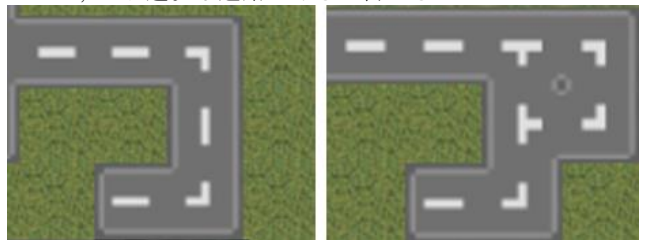


Fig.10 接続数10の道路(左)と接続数16の道路(右)

したがって、道路の整形処理は総接続数を評価値とした道路配置の探索問題とすることで実現できる。本システムでは、一定領域内の総接続数を評価値とした焼きなまし法により整形を行う。また探索を進める間に道路の加除を行うが、その際に道路が途切れてしまい接続性が失われた場合は元に戻す処理を加えている。このような探索処理にはある程度の計算コストを要するので、本システムでは対象となる道路の格子数が一定以上の場合のみ整形処理を行う。

### 3.3 幅の広い道路の配置

発展した都市空間には街の中心部分にメインストリートとなる幅の広い道路が存在するのが自然である。このようなメインストリートを都市発展のある段階で新たに構築するため、建物及び道路の配置パターンを一定時間ごとに評価する。

ある一定の大きさを持った領域に含まれる建物と道路の要素をそれぞれ重みづけして評価し、その領域の過密度(都市発展の度合い)を判定する。過密度が一定水準を超えた領域に渡って、幅の広いメインストリートを配置する(Fig.11)。重複を防ぐために、同じ向きのメインストリートは一定以上の距離を置いて配置する。



Fig.11 幅の広い道路の配置

#### 4 実行結果と考察

本システムを用いて 50 ノード程度のソーシャルグラフ (Twitter のユーザー 50 人を抽出しグラフ化したものに相当) の可視化を行った結果を Fig.12 に示す. このように都市型レイアウトは, Twitter から得られたデータを用いて 3D 都市空間を生成し, 建物の規模やアイコン表示といった表現を用いて俯瞰的に関係性を把握するものである. ばねモデルの変形やノード追加処理に応じて建物や道路が増えながら変化していくため, 都市が発展していく様子を眺めているかのような気分を味わうことができる. またカメラを切り替えることでウォークスルー視点になり, 街中を動き回することも可能である.



Fig.12 50 ノード程度のグラフの可視化結果

1 ステップにつきばねモデルによるグラフレイアウトの更新と前述の 3 つの配置処理を行い, 数ステップ毎にノードを追加することでソーシャルグラフを発展させる. 処理時間は 1 ステップ当たり 10 ノードで約 6ms, 50 ノードで約 72ms となった. Table.1 に示した所要時間について見てみると, 道路の配置処理に全体の約 83% の時間を要しており, この処理がボトルネックとなっていることが分かる. これは, 焼きなまし法の探索処理が大きなコストになっているためである. このように道路の整形手法については課題が残る. また, ノードの数に対して実行時間が  $O(n^2)$  に近いオーダーで増えていく傾向があるため, 規模が数百ノードになると計算量が膨大になるという問題がある.

Table.1 1 ステップの所要時間 (ノード数 50)

処理内容	所要時間
ばねモデルの更新	1ms
建物の更新	10ms
道路の配置・更新	60ms
メインストリートの配置・更新	1ms
計	72ms

Fig.13 に示すように, 都市の発展にともないメインストリートにより基盤の目状の構造が現れるという特徴が見られる. これは, メインストリートが過密度の高い座標に配置されるルールと, 同じ向きのメインストリート同士は重ならないように一定以上の距離を置いて配置されるルールによって発生する. 建物が密集して配置されるクラスタ付近は特にこの現象が起こり易く, ノード数が増加し二つの異なるクラスタが発生した場合は, それらを繋ぐメインストリートが配置されることも期待される.



Fig.13 基盤の目状の構造

また発展過程において, Fig.14 に示すように, ソーシャルグラフのクラスタの中心付近が都市空間の交差点として可視化されるという特徴が見られた. このように都市型レイアウトでは, メインストリートの交差点によってクラスタの存在や中心となるノードを視覚的に容易に把握することができる.

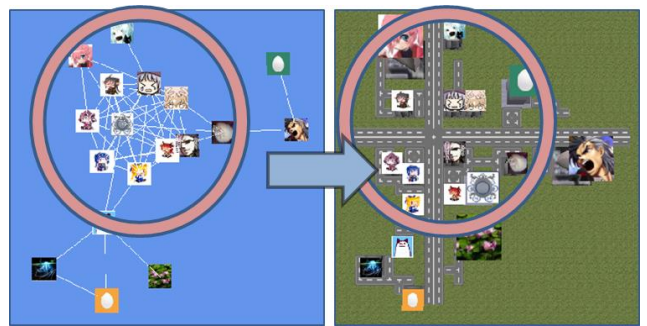


Fig.14 クラスタ中心に対応する都市空間の交差点

本システムでは開発言語に C# を, 3DCG 描画には XNA を利用した. また都市建物の 3D モデルの取得には Google 3D ギャラリー[5]を利用した. 本システムのハードスペックは CPU Intel Xeon 3.16GHz, メモリ 4.00GB である.

#### 5 まとめ

本稿では都市型レイアウトによる SNS ソーシャルグラフの可視化手法を提案し, システムの実装方法と可視化結果について述べた.

現在, 都市景観描画のクオリティ向上のため, 本システムの Unity への移植を計画している. また, ノード数増大に対するスケーラビリティの実現や, メッセージの発信内容の性質を建物や構造物の種類に反映させる可視化は今後の課題である.

#### 参考文献

- [1] P. Eades (1984) "A heuristic for graph drawing" Congressus Numerantium 42
- [2] Topigraphy <http://www.ntt.co.jp/journal/1009/files/jn201009019.pdf>
- [3] HatenaMaps <http://d.hatena.ne.jp/kaisch/20080609/1212980260>
- [4] Blogopolis <http://d.hatena.ne.jp/kaisch/20090827/1251406675>
- [5] 3D ギャラリー <http://sketchup.google.com/3dwarehouse/>