

ユーザ参加型動的ストリート画像フロー生成機構の設計

石塚 宏 紀^{†1,†2} 木 實 新 一^{†3,†2}
戸 辺 義 人^{†3,†2} 瀬 崎 薫^{†1,†2}

現在, Google Street View に代表される, 写真画像を用いて仮想的に街を散策できるサービスが広く利用されている. しかしながら, ある定められた時間内のみ撮影された画像を用いる従来のサービスでは, 季節, 時間帯, 個人の嗜好等, 条件を考慮して多様な風景を再構成することができない. 本論文では, こうした条件に応じて適切な風景を再構成できるサービスを実現するために, 一般の人々から集めた様々な街の写真を蓄積し, 道路に沿って必要な画像列を高速に検索できる動的ストリート画像フロー生成機構を提案する. 一般に, 多次元空間索引を用いれば, 場所とそれ以外の属性値を指定して対応する写真を高速に検索可能である. しかし, 従来の幾何学的な空間分割による索引手法では, 道路ネットワークのような実世界の構造に沿って生じる問合せ要求を円滑に処理することが難しい. そこで我々は, 道路ネットワークを考慮した索引手法である KDRN-Tree (KD-Tree with Road Network) を提案し, 街中に複数ユーザが撮影した多数の画像を実験データとし, 問合せ処理の評価を行った. その結果, KDRN-Tree の有効性と実用的なシステムへの適応可能性を確認した.

A Design of a Generating System of Dynamic Street-Based Image Flows utilizing Participatory Sensing Approach

HIROKI ISHIZUKA,^{†1,†2} SHINNICHI KONOMI,^{†3,†2}
YOSHITO TOBE^{†3,†2} and KAORU SEZAKI^{†1,†2}

We propose a dynamic mechanism for generating a series of relevant images along streets using user-submitted photos. Such a mechanism can provide users with personalized, contextually-relevant "street view" images. The proposed mechanism is based on a novel multi-dimensional index called KDRN-Tree (KD-Tree with Road Network), which can process a street-based retrieval query efficiently by considering a road network structure. We introduce KDRN-Tree, and describe its design and implementation in detail. The results of our experiments show that KDRN-Tree is more efficient than KDB-Tree, and that it is feasible to realize a dynamic mechanism for generating street-based image flows using KDRN-Tree.

1. はじめに

近年, GIS 技術の発展とインターネットの普及により, Google Maps^{?)} や Bing Maps^{?)} のようなオンラインマッピングシステムを用いて地図情報を気軽に利用できるようになった. 更に, 連続的な写真画像を用いた Google Street View^{?)} のようなサービスの出現により, 旅行前に目的地の外観や雰囲気を確認したり, 仮想的に街を「散策」することが可能となった. しかしながら, これらのシステムが提供する情報は非常に限られたものであり, 動的で多様な都市空間の豊かな表情を十分に伝えているとは言い難い. 例えば, Google Street View を提供するために, 特殊なカメラを搭載した車両が用いられているが, 撮影コストが高いこともあり, 細粒度・高頻度な撮影は現実的に困難である. そのため, 昼夜の景観, 季節や天候による雰囲気の違いを考慮した道案内システムのように場所の多様なイメージを考慮したサービスを提供することは難しい.

一方, カメラ機能, GPS 受信機, 加速度センサ, 地磁気センサ等を搭載する高機能携帯電話が急速に普及しつつあり, 環境メタ情報 (撮影時の位置情報, 方向情報, 天気等) 付きの写真が誰でも簡単に撮影可能となった. Panoramio^{?)} や Flickr^{?)} といったオンライン写真共有サービスでは, 環境メタ情報付き写真が続々と投稿されており, Panoramio には 2 千万枚程度の環境メタ情報付き写真^{?)} が蓄積され (2010 年 1 月現在), Flickr には既に約 1 億枚の環境メタ情報付き写真が蓄積されている (2010 年 1 月現在)^{?)}. 一般のユーザが撮影した写真を適切に活用すれば, 街を撮影するための専用車両に頼ることなく, ユーザから収集した豊富な写真画像を用いた価値あるサービスが提供可能である.

膨大なユーザ投稿写真群の有効活用には, 投稿写真群の中から状況に応じて必要な写真を高速に抽出する技術が重要となる. 例えば, 市街地において人々が移動する経路を考慮して, 様々な検索条件に合致する写真を高速に順次抽出できれば, 「歩行者ナビ」サービスの提供や地理空間情報の可視化を行う際に, 時空間的な接続性を保持した連続的な写真群を円滑に提示できる. 我々は, このような移動経路に基づく連続的な写真群をストリート画像フ

†1 東京大学 空間情報科学技術研究センター

Center for Spatial Information Science, the University of Tokyo

†2 科学技術振興機構 CREST

JST CREST

†3 東京電機大学 未来科学部

School of Science and Technology for Future Life, Tokyo Denki University

ローと定義する。

本論文にて我々は、ユーザ投稿写真を用いて、指定した時間帯、季節、気象条件等の複数条件に合致した動的ストリート画像フローを高速に生成する機構を提案する。既存データベース管理システムに蓄積されたユーザ投稿写真を各指定条件に対して逐次的に検索処理すればストリート画像フローを生成可能であるが、大量の写真を扱う場合、膨大な処理時間を要するため、円滑なストリート画像フローの生成は困難である。また、位置条件を含む複数の条件に合致した画像検索には、kd-tree²⁾に代表される多次元索引機構が利用されるが、道路ネットワーク構造を無視した処理を行うため、ストリート画像フロー生成の高速化に適していない。本論文の貢献は以下の2点の特徴を持つ、従来の多次元索引機構よりも高速にストリート画像フローを生成できる KDRN-Tree の提案である。

- ・道路ネットワーク配置を考慮した領域分割

2次元平面上の道路ネットワーク配置を考慮した戦略的な領域分割によって、同じ道路セグメント上の写真をできるだけ同一領域にて管理可能とする。これによって、ストリート画像フローの高速生成を可能にする。

- ・道路ネットワーク正規化による索引の効率化

効率の良い索引を生成するためには、多くの道路が管理領域の次元軸に対して水平・垂直でなければならない。そこで我々は、力学的なポテンシャルモデルに基づいて道路ネットワークを自動変形し、索引機構と親和性の高い道路ネットワーク配置に正規化する。

我々は、KDRN-Tree(KD-Tree with Road Network)を実際に開発し、秋葉原の市街地を想定した動的ストリート画像フロー生成のシミュレーション実験を行い、KDRN-Treeによる問い合わせ処理の評価を行った。さらに、我々は、秋葉原にて実際に約3500枚の写真を撮影し、複数環境におけるストリート画像フローを生成した。KDRN-Treeは、K-D-B-Treeよりも最大で40%以上高速に処理が可能である。評価の結果として、本論の提案であるKDRN-Treeの有用性と実用的なシステムへの適応可能性を確認した。さらに我々は、秋葉原にて実際に約3,500枚の写真を撮影し、複数の環境条件を指定して、ストリート画像フローを生成し、本論の提案であるKDRN-Treeの有用性と実用的なシステムへの適応可能性を確認した。

以下、第??章では、動的ストリート画像フロー生成機構における既存研究と研究課題について述べ、第??章にて提案手法であるKDRN-Treeの詳細を示す。第??章にてシステム

の実装、第??章にてシミュレーション実験によるKDRN-Treeの検索性能評価結果と実際の写真を用いた動的なストリート画像フローの生成例を示す。

2. ストリート画像フロー生成機構

ユーザフレンドリーな地理空間情報サービスを実現するために、移動経路に沿った一連の写真を利用するシステムの研究や実用化が進んでいる。本章では、いくつかの既存研究を紹介するとともに、本論文の研究対象であるストリート画像フロー生成機構の特徴と技術的な課題について述べる。

2.1 関連研究

移動経路に沿った一連の写真を利用するシステムを構築するための手法として、Image Based Rendering(IBR)³⁾⁴⁾が知られている。IBR技術とは、複数画像から任意の視点における画像を生成する技術であり、複数写真を利用して実世界に似せた仮想環境を構築する際に特に有用である。Snavelyら⁵⁾はFlickrから収集したユーザ投稿写真群に対して、画像処理による特徴点抽出を行うことでカメラ変数の推定を行い、空間上にマッピングして直感的なブラウジングができるビューアを作成した。さらにHileら⁶⁾は、Snavelyらが開発したストリートビューア⁷⁾に拡張現実感技術を組み合わせた観光客向けのナビゲーションシステムを提案している。

しかしながら、IBRシステムにおいては、特定の条件を満たすように同じ場所を異なる視点で撮影する必要がある。また、IBR技術の根幹である画像処理によるカメラ変数の推定⁸⁾には、同一環境で撮影された写真を利用する必要があり、複数のユーザが各自好きなときに撮影した写真を有効活用することは難しい。

これとは対照的に、本論文の研究対象であるストリート画像フロー生成機構は、検索条件を満たす写真を移動経路に沿って提示するものであり、様々なユーザが撮影した写真を幅広く活用することが可能である。すなわち、非常に滑らかな画像空間を生成するために厳しい制約を課して重い処理を行うよりも、高速に実行できる軽い処理を用いて様々なユーザが貢献したデータをできるだけ迅速に活用することを重視している。ストリート画像フローに対してSTAMP⁹⁾等の手法を適用すれば、重い処理を行うことなく移動経路に沿った複数の写真を効果的に提示し、仮想ウォークスルー等のアプリケーションを実現できる。また、写真に撮影方向や画角に関する情報が付加されていれば、柔軟な検索が可能なインタフェース¹⁰⁾を介してストリート画像フローを利用することも可能である。

2.2 研究課題

個人の嗜好やコンテキストに応じた動的な地理空間情報サービスを提供するためには、大量の写真から即座に条件を満たすストリート画像フローを生成できることが望ましい。しかしながら、移動経路に沿って一連の写真を利用するための画像処理法や提示法についてはこれまでに様々な研究が行われてきたものの、既存のシステムの多くは限られた数の写真を対象にしており、大量の写真から必要な写真を効率よく抽出して動的にストリート画像フローを生成することは困難である。

本提案における写真の検索は、複数の環境メタデータを用いた処理であり、高速化のために R-tree^{?)} や kd-tree などの多次元索引を用いることが考えられる。特に kd-tree は頻繁に追加されるデータの扱いに適していると考えられるが、高速にストリート画像フローを生成するためには道路ネットワークを考慮した空間分割を行う必要がある。具体的には、以下の二つの課題を解決しなければならない。

1. 領域分割における道路セグメントの保護

kd-tree ではノードに対応する管理領域を単純な方法で分割するため、連続した道路セグメント内のデータを細切れに分割してしまう。ストリート画像フロー生成においては、連続した道路セグメントのデータはまとめて扱うため、不要な分割が行われないように道路セグメントを保護する必要がある。

2. 道路ネットワーク情報の維持

実世界の道路配置は多種多様であり、索引構築が容易な形状をしているものもあれば、そうでないものもある。道路ネットワークのトポロジを保存しつつ、索引構築が最も容易な配置に変換する必要がある。

3. 提案手法 KDRN-Tree

第??章で述べた研究課題を解決するために、我々は、KDRN-Tree (KD-tree with Road Network) を提案する。KDRN-Tree は、K-D-B-Tree^{?)} に基づく実世界の道路構造を反映した多次元空間索引とする。連続した道路セグメントを保存して木を構成し、力学的なポテンシャルモデルを用いたネットワークの形状正規化を行って道路パターンが不規則な場合でも検索処理効率を高めることができるという特徴を有する。

図??~??に KDRN-Tree の領域分割とそれに対応する索引構造の例を示す。図??のように 2 次元地図面の道路にデータが分布している場合、KDRN-Tree では図??に示すよう

に、データ分布に応じた管理領域の分割を実行する。領域内にデータが 1 つもない状態から番号順にデータを挿入し、領域中のデータ数が一定値 P を超えた時点 (例では $P = 3$ に設定) で領域分割を行う。地図上で道路セグメントを 1 つも横断しない水平線または垂直線を分割線 (SL) として選択しつつ、各領域におけるデータの個数がバランスするよう領域を分割する。もし条件を満たす SL が見つからない場合は、位置情報の 2 次元地図面以外の軸 (例では照度軸) で分割を実行する。この手順で分割処理を繰り返し実行することで、最終的に図??に示すような木構造が形成される。実際には、内部ノードの分割処理や木の深さを調整処理を行う必要があるが、これらの詳細については第??章で説明する。

3.1 KDRN-Tree の構成

一般に KDRN-Tree の管理領域は写真の位置属性を表現する 2 次元地図と n 個の環境属性で構成される $k(= n + 2)$ 次元空間の超矩形であり、管理領域の重複はないものとする。木構造の高さは一定に調整され、最大 M 個の子ノードを管理する M 分木となる。KDRN-Tree は葉ノードと内部ノード、根ノードから構成される。これらを区別しない場合は単にノードと呼ぶ。葉ノード L の構成要素は以下のとおりである。

$$L = (parent, n_l, data[P], region)$$

ここで、 $parent$ は親ノードへのポインタ、 n_l は L のデータ数、 $data[P]$ はデータを指すポインタ P 個を格納するスロット、 P は最大データ格納数、 $region$ は L のすべてのデータ点を包括する k 次元空間の管理超矩形である。葉ノードは $P_{min} (\leq P/2)$ 以上のデータを持つ。 k 次元空間におけるデータ点の位置は、地図上のデータ取得位置とその他の属性値に基づいて決定される。内部ノード N の構成要素は以下のとおりである。

$$N = (parent, n_n, child[M], region)$$

ここで $parent$ は親ノードへのポインタ、 n_n は N の子ノード数、 $child[M]$ は子ノードを指すポインタ M 個を格納するスロット、 M は最大子ノード数、 $region$ は N のすべての子ノードの管理超矩形の総和と一致する管理超矩形である。 $parent$ の値が null の場合根ノードと呼ぶ。内部ノードは根ノードを除いて $M_{min} (\leq M/2)$ 以上の子ノードを持つ。 P_{min} と M_{min} の値は小さいほどデータ挿入が高速になるが、葉ノードや内部ノードが増加するため、速度とメモリコストのトレードオフの関係がある。

3.2 道路ネットワークを考慮した管理領域の分割

KDRN-Tree において、子ノードの管理領域は、親ノードの管理領域を超平面により分割することによって得られる。動的ストリート画像フロー生成において、同一道路セグメント上の写真をまとめて扱う必要があるため、管理領域は、道路ネットワーク構造に従って

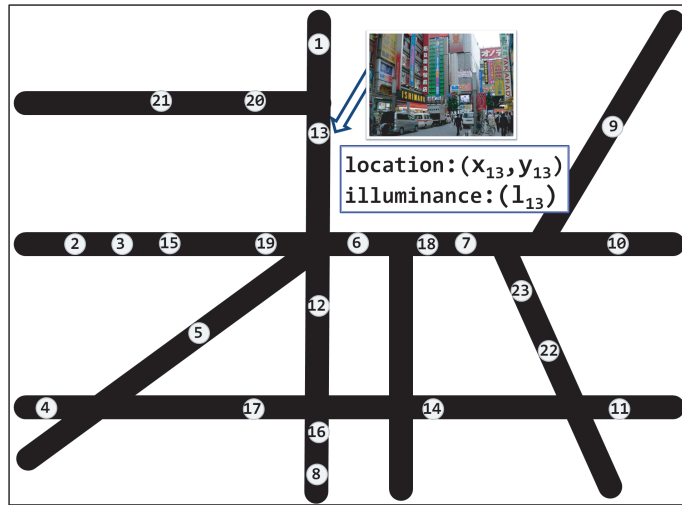


図 1 2次元地図上に分布する照度付き写真データ例
Fig. 1 Sample photo/illuminance points on a 2D Map

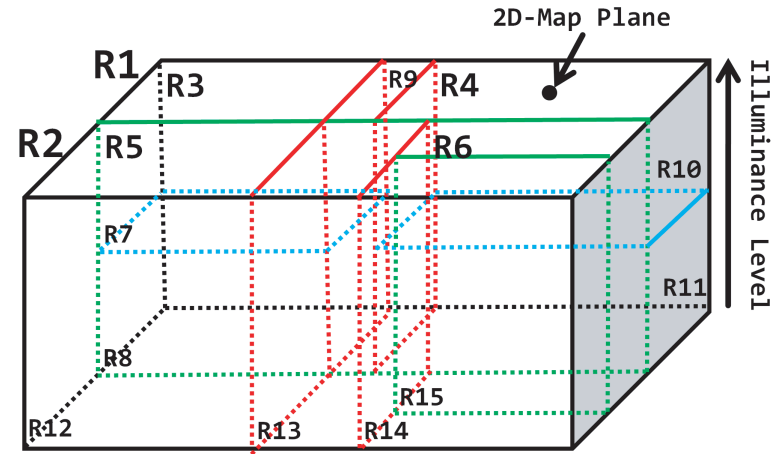


図 2 図??のデータ分布に基づいて分割された 3次元管理領域
Fig. 2 3D region for managing the data points in Fig.??.
The region is split based on the spatial pattern of the data points.

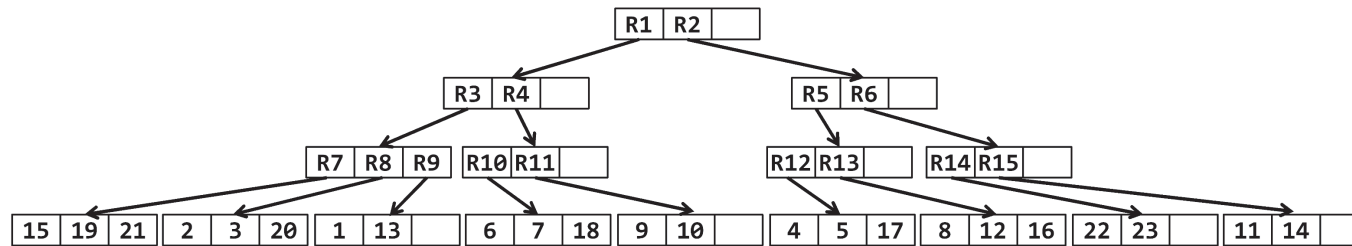


図 3 図??の分割に基づく KDRN-Tree の構造
Fig. 3 KDRN-Tree based on the splitting pattern in Fig.??

分割する．図??は，格子状の道路ネットワークにおける写真の撮影位置を点で示した例である．これらの写真には，撮影位置以外に照度センサ等により取得されたデータが環境メタ情報として付加されているものとする．この場合，図??に照度の軸を加えた 3次元空間領域が KDRN-Tree の管理領域となる．なお，以下では道路や歩道をパス，地図上の領域を区切る境界をエッジと定義する．図??において，水平および垂直方向の合計 6 個のパスが，9 か所で交差しており，すべてのパスの端点はエッジにより区切られる．エッジと交差

点によってこれら 6 個のパスは 33 個のパスセグメント（9 個の交差点領域を含む）に分割できる．すべてのパスセグメント（矩形領域）の境界線を垂直または水平方向にエッジまで延長し，この中で他のパスセグメントと 1 つも交差しないものを分割線 l_i とし，その集合を $SL = \{l_1, l_2, \dots, l_s\}$ とする．また，管理領域の地図をエッジからエッジに渡る垂直線および水平線で走査し，どのパスセグメントとも交差しない縦軸および横軸の「隙間」を自由区間 $w_j (1 \leq j \leq t)$ とし，その集合を $FI = \{w_1, w_2, \dots, w_t\}$ とする．例えば，図??から

をそれぞれ、全パスセグメントの中で斜め方向のパスセグメントが占める割合、斜め方向のパスセグメントの中で MBR が他のパスセグメントの MBR と重なるものの割合とすれば、索引親和性 $A = 1 - r_s r_o$ を最大化するようにパス形状を正規化すれば、効率の良い KDRN-Tree を構築することが可能となると考えられる。以下に示すように、 r_s を最小化するための 2 つの処理目的を設定し、索引親和性を向上させる。

- (1) 直線直交化 連結したパスの組が互いになす角がすべて $\pi/2$ の整数倍となる。
- (2) 水平垂直化 すべてのパスセグメントが管理超矩形の地図面の縦軸もしくは横軸と平行になる。

まず、直線直交化を達成するために、力学的なポテンシャルモデルに基づくパスの直交化および直線化を行う。具体的に、ノード間のパスセグメントをばねで置き換えればねめ込みモデル^{?)} と呼ばれる抽象モデル理論を用いてパスの形状変換を行う。ここで、以下のエネルギー関数を用いれば、道路長を初期値のまま保存し、道路間のなす角度を量子化し、道なりの道路を直線化できることが知られている^{?)}。

$$E = \frac{1}{2} \sum_{i \leftrightarrow j} k_{ij}^v (v_{ij} - u_{ij})^2 - \sum_{i \rightarrow j \rightarrow k} k_{ijk}^\theta \cos(\theta_{i \rightarrow j \rightarrow k} - \phi_{i \rightarrow j \rightarrow k}) - \sum_{i,j,k} k_{ijk}^\Theta \cos(\Theta_{i \rightarrow j \rightarrow k} - \pi) \quad (1)$$

$$\phi_{i \rightarrow j \rightarrow k} = \left[\frac{2\theta_{i \rightarrow j \rightarrow k}}{\pi} + 0.5 \right] \times \frac{\pi}{2} \quad (2)$$

エネルギー関数 (??) の第 1 項は、パスセグメントの長さが理想値のときに最小となる 2 次関数であり、 u_{ij} が理想長である。第 2 項は連結したパスセグメントのなす角が理想値のとき最小となり、理想角 $\phi_{i \rightarrow j \rightarrow k}$ は式 (??) を用いて設定する。なお、 $[x]$ は x を超えない最大の整数値を表す。第 3 項は連結したパスセグメントの直線化を示す。このエネルギー関数を用いた場合、各項が相反する効果を示すため、例えば、パスセグメントの長さを保存しようとする、角度が直角にならず、角度を直角にしようとするパスセグメントの長さが保存されないといった現象が起こる。したがって、エネルギー関数の各項の重み付けを示すばね係数の設定が重要であり、KDRN-Tree においては直線直交化を最優先してばね係数を決定する。本手法は、ネットワークのトポロジーを損なわない範囲で漸次的に直線直交化を行うものであるため局所解に陥る可能性があるが、実用上は厳密な最適解を求めることよ

り限られた時間内で許容可能な解を得ることが優先されるため、漸次的な直線直交化を行うことは妥当であると考えられる。

次に水平垂直化を達成するために、できるだけ多くのパスセグメントが管理超矩形の地図面の縦軸または横軸に対して厳密に平行になるようにネットワークの回転を行う。このための手順として、まずパスセグメントが平行になる比率 $1 - r_s$ が最大となるように座標を回転し、次に平行に近いパスセグメントが厳密に平行となるように座標の補正処理を行う。座標回転の角度は、ヒストグラム解析等を用いて決定することができる。

3.4 データの挿入と空間分割

KDRN-Tree を構築するためには、データを順次挿入し、正規化されたパス形状を用いて管理領域を分割していけばよい。データ挿入のアルゴリズムにおいては、木を走査し、挿入対象葉ノードを決定する *Insert*、葉ノードを分割する *SplitLeafNode*、内部ノードを分割する *SplitNode* の 3 つで構成される。以下でそれぞれのアルゴリズムの詳細を示す。

[Insert Algorithm] 環境メタ情報付き写真 d を KDRN-Tree に挿入する。

- INS1 データ項目の 2 次元地図上の位置とその他の n 個の属性値に基づき $k(= n + 2)$ 次元空間の点 p_d を求める。
- INS2 根ノードから木をたどり、 p_d を含む管理超矩形を持つ葉ノード L を求める。
- INS3 $n_l < P$ の場合、 L に d を挿入する。
- INS4 $n_l = P$ の場合、*Split Leaf Node Algorithm* を用いて L を分割する。

[Split Leaf Node Algorithm] 葉ノード L に $P + 1$ 個目のデータが挿入されたとき、 L を分割する。なお、 L の親ノード、分割線集合、自由区間集合をそれぞれ N , SL , FI とする。

- SLN1 SL から $l \in SL$ を通る分割超平面が葉ノード L の管理超矩形 *region* を $P/2$ もしくは $P/2 + 1$ のデータ点を持つよう分割するものを取り出し、これを均等分割線集合 $SL' (\subset SL)$ とする。
- SLN2 $SL' \neq \phi$ の場合、 SL' に含まれる任意の分割線に対応する分割超平面で *region* を 2 分する。SLN6 に進む。

- SLN3 $SL' = \phi$ の場合, FI から $w \in FI$ を通る任意の分割超平面が葉ノード L の *region* を $P/2$ もしくは $P/2+1$ のデータ点を持つよう分割するものを取り出し, 均等分割可能自由区間集合 $FI' (\subset FI)$ とする.
- SLN4 $FI' \neq \phi$ の場合, FI' の任意の自由区間に含まれる均等分割線に対応する分割超平面で *region* を 2 分する. SLN6 に進む.
- SLN5 $FI' = \phi$ の場合, 管理超矩形の k 個の軸のうち, $n (= k - 2)$ 個の属性軸の中から任意の属性軸 a を選択し, a に垂直に交わり, 葉ノード L の *region* を $P/2$ もしくは $P/2+1$ のデータ点を持つよう分割する (地図面と交わらない) 分割超平面で *region* を 2 分する. SLN6 に進む.
- SLN6 分割に用いた超平面と垂直に交わる軸の上で, 超平面と交差する点よりも値が小さい側の管理超矩形に対応するデータ点を L に割り当て, 新たな葉ノード L_{new} に他方の管理超矩形に対応するデータを割り当てる.
- SLN7 親ノード N の子ノード数を n_n とし, $n_n < M$ ならば L_{new} を L_i 直後のスロット $child[i+1]$ に挿入し, $child[i+1]$ 以降は一つずつ後ろにシフトする.
- SLN8 $n_n = M$ の場合, *Split Node Algorithm* を用いて N を分割する.
- [Split Node Algorithm] 内部ノード N に $M+1$ 個目のデータが挿入されたとき, N を分割する. なお, N の親ノードを N_p とする.
- SN1 N の子ノードを分つ分割超平面のうち, N の管理超矩形 *region* を完全に切断できるものをすべて求め, この中から N の子ノードを最も均等に近く分割するもの一つを選ぶ. 選ばれた分割超平面を c とする.

- SN2 c によって N を二つの管理超矩形に分割し, 子ノード群を二つのグループに分ける.
- SN3 もしどちらかのグループの子ノード数が M_{min} 以下ならば, 子ノードが少ないグループを一度削除してもう一方のグループの領域を拡大し, 削除されたデータを再挿入する. この操作によって, 子ノード数の調整を行う.
- SN4 分割に用いた超平面と垂直に交わる軸の上で, 超平面と交差する点よりも値が小さい側の管理超矩形に対応する子ノードを N に割り当て, 新たな葉ノード N_{new} に他方の管理超矩形に対応する子ノードを割り当てる.
- SN5 親ノード N_p の子ノード数が M より小さければ, N_{new} を N の直後のスロット $child[i+1]$ に挿入し, $child[i+1]$ 以降は一つずつ後ろにシフトする.
- SN6 N_p の子ノード数が M に等しい場合, *Split Node Algorithm* を用いて N_p を分割する.

3.5 データの削除

データが削除された場合の処理は, K-D-B Tree^{?)} のデータ削除アルゴリズムにしたがって処理を行う.

[Delete Algorithm] データ項目 d を KDRN-Tree から削除する.

- DL1 データ項目の 2 次元地図上の位置とその他の n 個の属性値に基づき $k (= n + 2)$ 次元空間の点 p_d を求める.
- DL2 根ノードから木をたどり, p_d を含む管理超矩形を持つ葉ノード L を求め, L から d を削除する.
- DL3 $n_i \leq P_{min}$ の場合, 隣接管理超矩形を拡大して葉ノードを統合する. もし, 葉ノードの統合によって内部ノードの子ノード数が M_{min} 以下になった場合, 内部ノードの統合を行う.

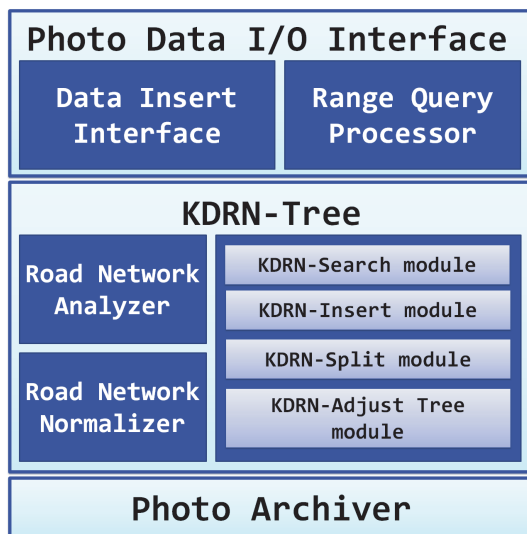


図 6 システム構成
Fig.6 System architecture

4. 実 装

提案手法に基づいて、我々は、KDRN-Tree を用いた動的ストリート画像フロー生成機構のプロトタイプを実装した。本プロトタイプは、Java SE Development Kit, Version 6 による約 7000 行のコードで開発され、大きく 3 つの機能で構成されている。本プロトタイプのシステム構成を図??に示し、以下で詳細を解説する。さらに、我々は、環境メタ情報付写真撮影システムを Android OS 搭載端末 HTC-03A と無線センサノード (Sun SPOT) を用いて実装した。

Photo Data I/O Interface

Data Insert Interface は、ユーザ投稿写真を受信し、写真データの実体を Photo Archiver に格納し、写真の環境メタ情報実体をキーとして実データへのポイントを KDRN-Insert module へ転送する。一方、Range Query Processor は、利用者からの多次元 Range Query を KDRN-Search module に転送し、該当する写真群へのポイントを得る。そのポイントを用いて写真データを Photo Archiver から取得し、利用者へと返答する。



図 7 きときと写真アップローダー
Fig.7 A System of KitoKito Photo Uploader

KDRN-Tree

core モジュール群として、4 つのモジュールを実装した。KDRN-Insert module と KDRN-Split module は、??章で解説したアルゴリズムが実装され、木の構築時に利用される。KDRN-Adjust-Tree module は、分割処理によって木の均衡が崩れた場合に呼び出される機能である。KDRN-Search module は、Range Query Processor によって呼び出され、木を与えられたキー範囲に基づいて下り、該当する写真へのポイントを返す。一方、Road Network Analyzer は、OpenStreetMap^{?)} などから道路ネットワーク情報を取得し、フォーマットを解析してパスとエッジを抽出する。さらに、Road Network Normalizer は、解析して得られたパスとエッジを基に KDRN-Tree と親和性の高いパス形状に正規化する。

Photo Archiver

投稿写真データの実体を格納するためのデータベースであり、Data Insert Interface から転送されたデータを格納しポイントを返す。また、Range Query Processor から転送されたポイントに対応する写真データの実体を返す。

4.1 環境メタ情報付写真撮影システム

本プロトタイプにおける環境メタ情報として、我々は、写真の撮影位置、撮影時間、撮影方

表 1 シミュレーション実験の条件
Table 1 Simulation setup

CPU	Intel Core 2 Duo 2.4GHz
主記憶	3.5GB RAM
挿入写真数	100 - 5000
環境メタ情報	緯度, 経度, 照度, 撮影角度
道路ネットワーク	東京都千代田区秋葉原地区の一部
実験領域のサイズ	500m × 500m
データ挿入分布	ランダム分布
照度範囲	0 - 1500(lx)
撮影角度範囲	0 - 360 °
最大子ノード数 M	10
最小子ノード数 $Mmin$	$M/2$
最大データ格納数 P	M
最小データ格納数 $Pmin$	$Mmin$
KDB-Tree の分割ルール	Cyclic splitting algorithm

角, 撮影時の照度データの 4 つとした。撮影位置, 撮影時間, 撮影方角は, 独自開発したソフトウェアであるときと写真アップローダーによって, 撮影時に HTC-03A 内部の GPS 受信機と電子コンパス等の情報を取得し, 自動的にサーバへと送信される。また, 同時に照度センサが搭載された無線センサノード (Sun SPOT) を用いて照度データを収集しサーバへと送信される。Photo Data I/O Interface の Data Insert Interface にて, 撮影時間における照度データをさらに環境メタ情報として写真に付加する。環境メタ情報付写真撮影システムにおける使用機器ときと写真アップローダーの撮影画面を図??に示す。

5. 実験による評価と考察

KDRN-Tree の有効性を検証するために, 我々は, 実際の道路ネットワークデータを準備してシミュレーション実験を行い, 他の手法と性能の比較を行った。更に, 市街地で携帯電話とセンサを用いた写真データを収集を行い, プロトタイプによる動的ストリート画像フロー検索を実行し, 動的ストリート画像フロー生成システムの実現可能性を検証した。

5.1 実験方法

KDRN-Tree の検索性能を評価するために, 既存の多次元索引の中でも特に定期的なデータ挿入に適した領域分割形バランス木構造であるとされる KDB-Tree との比較実験を行った。また, バス正規化の効果を吟味するためにバス正規化なしの KDRN-Tree との性能比較も行った。表??にシミュレーション実験条件を示す。また, 各木構造は主記憶上に作成す

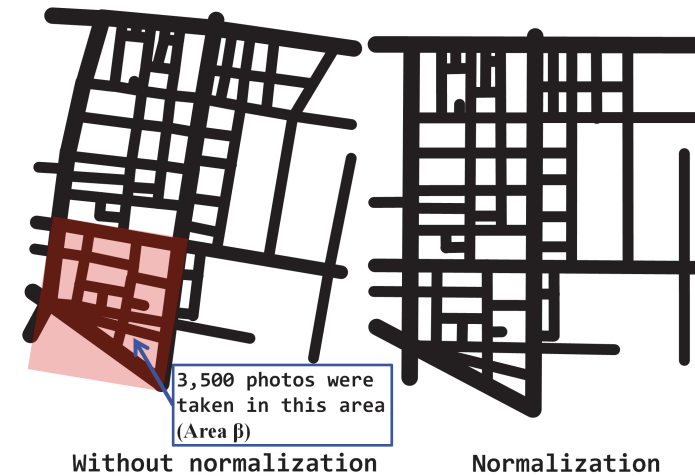


図 8 シミュレーション実験の道路ネットワーク
Fig.8 Road network used in our simulation

る。OpenStreetMap^{?)} の秋葉原駅周辺の道路ネットワークデータを利用して, 道路上にランダムに 100 ~ 5,000 枚の写真データを発生させた。実験に利用した道路ネットワークと正規化後の道路ネットワークを図??に示す。写真に付随する撮影角度および照度については, それぞれ 0 ~ 360° および 0 ~ 1500lx の範囲のランダムな値とした。検索クエリは, 各道路セグメントに対応した Range Query とした。なお, この Range Query における照度と撮影角度の値範囲は, 道路セグメント内のデータ検索時間を測定することが目的であるため全領域として指定した。各クエリを 10,000 回試行し, 検索時間の平均値を算出した。また, 道路パタンの違いによる性能の変化を分析するために, 実験道路ネットワークから 2 種類の部分マップを切り出し, 100 ~ 2,000 枚の写真データを同様の条件で発生させて, 同様に Range Query の検索時間の平均値を算出した。本論における検索時間は, 質問を発行してから該当する葉ノードが得られるまでに要する時間とする。

さらに, 我々は, 図??内のシミュレーションの実験領域の一部である Area β で実際に約 3,500 枚の写真データを撮影した。撮影写真は, 平日 4 日間に渡って 5 名のユーザが撮影した。これらの写真の撮影時間, 天候はさまざまであった。

5.2 結果

シミュレーション実験の結果を図??に示す。横軸は道路ネットワーク上に分布する写真の

枚数、縦軸は検索時間の平均値を表している。写真の枚数が増加するに伴って KDRN-Tree は KDB-Tree よりも検索速度の面で有利になっている。さらに図??における道路ネットワーク正規化前の KDRN-Tree(without normalization) よりも KDRN-Tree の方が検索速度が向上している。全体として KDRN-Tree は、KDB-Tree に対して、最大 43.23%の速度向上が得られ、KDRN-Tree(without normalization) に対して、最大 11.79%の速度向上が確認された。

部分マップを使った検索性能実験の結果を図??に示す。道路ネットワーク正規化による形状変化が少ない部分(マップ A)の結果である図??と正規化による形状変化が大きい部分(マップ B)の結果の図??からわかるように、マップ B の方が、KDRN-Tree(without normalization) に比べてより大きな速度向上がみられることがわかる。実際に、マップ A において、KDRN-Tree は、KDRN-Tree(without normalization) に対して最大 8.4%の速度向上が見られたのに比べて、部分マップ A において、KDRN-Tree は、KDRN-Tree(正規化なし) に対して最大 10.1%の速度向上が確認された。

更に、我々は、市街地で収集した 3,500 枚の写真のプロトタイプを入力し、実際に動的ストリート画像フローを生成して、本提案の実用的なシステムへの適応可能性を確認した。図??、図??は出力例であり、図??内の白矢印の示す道路セグメント、および方向を対象に、異なる照度で Range Query を処理した結果である。図??における照度範囲は、350xl 以上であり、図??における照度範囲は 350xl 未満である。これらの Range Query の検索結果として出力された画像の数に違いはあるが、同一の道であることは見て取れる。しかしながら、昼と夜では、人通りや照明の具合等で道の雰囲気が大きく変化している。このように昼と夜のストリート画像フローを動的に得られることが確認できた。

5.3 考察

KDB-Tree および KDRN-Tree は kd-tree に基づく多次元索引であり、複数属性を対象とする範囲検索を高速に処理することができる。ユーザが投稿した大量の写真データから動的にストリート画像フローを生成するためには、データ数が多い場合も指定道路セグメント上の検索条件を満たすデータを短時間で検索できる KDRN-Tree が有利である。人気のある観光地などでは同時に多数の検索要求を処理する必要性が生じるため、環境メタ情報付きの写真がネットワーク上で広く共有されるようになるにつれて KDRN-Tree の有用性が高まると考えている。

道路ネットワークの正規化により KDRN-Tree の処理効率を向上できることも、シミュレーション実験によって確認された。効率改善の大きさは、パスセグメント毎に異なり、正

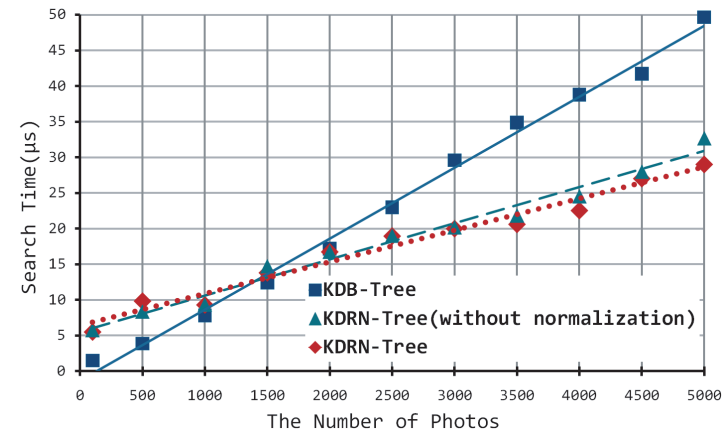


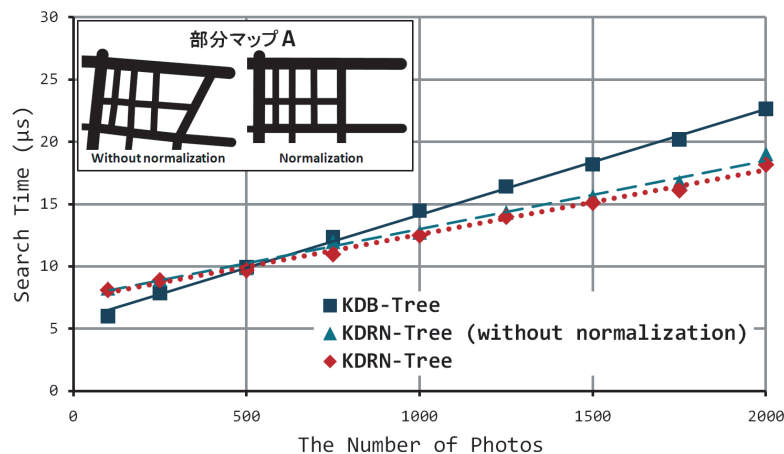
図 9 シミュレーション実験結果
Fig. 9 Simulation result

規化以前に索引と親和性の低い部分領域では、特に大きな効率改善が可能であると考えられる。もちろん、道路ネットワークのトポロジによっては、完全な正規化が不可能な場合がある。シミュレーション実験では、完全な正規化が不可能であっても正規化処理によって検索の効率化が可能であることを確かめることができた。

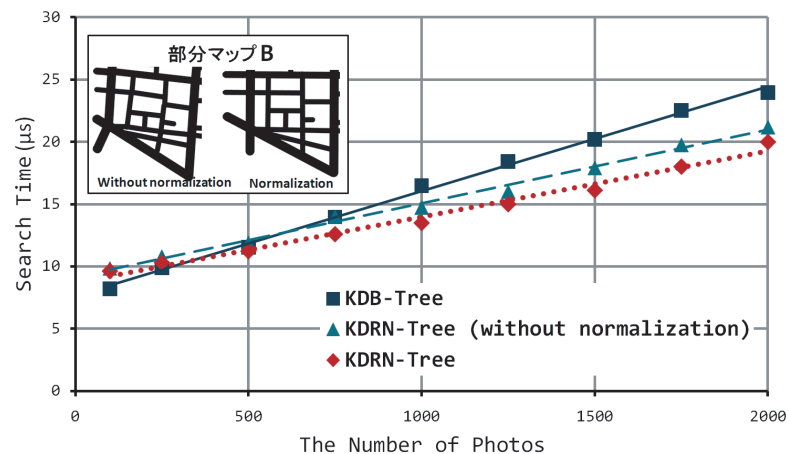
索引にデータを挿入する場合、KDRN-Tree ではまず地図面を通る分割超平面を探索するため、KDB-Tree よりも余分な処理の手間を要する。この余分な処理のコストは分割線集合 SL と自由区間集合 FI の要素数に依存する。これらの要素数が非常に多い場合は、検索処理効率とのトレードオフを考慮しつつ間引きして数を減らしてもよい。

6. むすび

本論文では、一般の人々から集めた様々な街の写真をデータベースに蓄積し、道路に沿って必要な画像列を円滑に検索できる動的ストリート画像フロー生成機構について議論した。この機構を実現するためには道路に沿った多次元データを高速に検索する必要があることを述べ、このような必要性を満たす従来の手法よりも優れた高速化手法である KDRN-Tree を提案した。また、評価実験を通して、検索処理における KDRN-Tree の有用性を示した。KDRN-Tree は、既存手法である K-D-B-Tree に対して最大 43.23%の検索速度向上を確認



(a) Simulation result in Submap A



(b) Simulation result in Submap B

図 10 道路ネットワーク正規化による検索性能への影響
Fig. 10 An effect of road network normalization

した．さらに，実環境実験を通して動的ストリート画像フロー生成機構の実用的なシステムへの適応可能性を確認した．本研究の今後の展開として，GPS トレース等による人流データを用いたパスセグメントの重み付けを行い，道路ネットワークの接続性も考慮した空間分割手法へと発展させていくことも考えて行く．

謝辞 本研究は JST-CREST プロジェクト「実世界検索に向けたネットワークセンシング基盤ソフトウェア OSOITE」の成果である．また，実験システムに協力頂いた西本 龍弘氏，福原 遼氏，福本 駿氏に感謝する．

参 考 文 献

- 1) Google Maps, <http://maps.google.com>, Mar.14, 2010.
- 2) Bing Maps, <http://www.bing.com/maps/>, Mar.14, 2010.
- 3) Google Maps Street View, <http://maps.google.com/help/maps/streetview>, Mar.14, 2010.
- 4) Panoramio, <http://www.panoramio.com>, Mar.14, 2010.
- 5) Flickr, <http://www.flickr.com>, Mar.14, 2010.
- 6) Panoramio's Blog, "New time January 20th, 2010 by Eduardo Manchon," <http://www.panoramio.com/blog/new-time/>, Mar.14, 2010
- 7) J.L. Bentley, "Multidimensional binary search trees used for associative searching," Commun. ACM, vol.18, pp.509-517, 1975.
- 8) S.E. Chen, "QuickTimeVR - An Image Based Approach to Visual Environment Navigation," Proc. ACM SIGGRAPH, pp.29-38, 1995.
- 9) S.J. Gortler, R. Grzeszczuk, R. Szeliski and M.F. Cohen, "The lumigraph," Proc. ACM SIGGRAPH, pp.43-54, 1996.
- 10) N. Snavely, S. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3D," Proc. ACM SIGGRAPH, pp.835-846, 2006.
- 11) H. Hile, R. Grzeszczuk, A. Liu, R. Vedantham, J. Kosecka, G. Borriello, "Landmark-Based Pedestrian Navigation with Enhanced Spatial Reasoning," Proc. of the 7th International Conference on Pervasive Computing, pp.59-76, May 2009.
- 12) N. Snavely, R. Garg, S. M. Seitz and R. Szeliski, "Finding Paths through the World's Photos," Proc. ACM SIGGRAPH, 2008.

- 13) D.G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. of Computer Vision, 60(2), pp.91–110, 2004.
- 14) 田中 浩也, 有川 正俊, 柴崎 亮介, "建築アーカイブ作成のための疑似3次元写真編集システム," 日本建築学会構造系論文集, no.564, pp.135–141, 2003.
- 15) 藤田 秀之, 有川 正俊, 岡村 耕二, "高精度な空間情報付き写真の3次元実空間マッピング," 信学論 (A), vol.J87-A, no.1, pp.120–131, 2004.
- 16) A. Guttman, "R-tree: A dynamic index structure for spatial searching," Proc. ACM SIGMOD, pp.49–59, 1984.
- 17) J.T. Robinson, "The K-D-B tree: a search structure for large multidimensional dynamic indexes," Proc. ACM SIGMOD, pp.10–18, 1981.
- 18) T. Kamada and S. Kawai, "An Algorithm for Drawing General Undirected Graphs," Information Processing Letters, 31, pp.7–15, 1989.
- 19) 丸山 貴志子, 谷崎 正明, 嶋田 茂, "デフォルメマップ生成のための道路形状正規化モデルとそのシステム評価," 信学論 (A), vol.J87-A, no.1, pp.108–119, 2004.
- 20) OpenStreetMap, <http://www.openstreetmap.org/>, Mar.20, 2010.



図 11 動的ストリート画像フロー生成例 1(照度範囲 : 350lx 以上)
Fig.11 Street-based Image flow generated by our prototype
(illuminance level : more than or equals to 350lx.)



図 12 動的ストリート画像フロー生成例 2(照度範囲 : 350lx 未満)
Fig.12 Street-based Image flow generated by our prototype
(illuminance level : less than 350lx.)