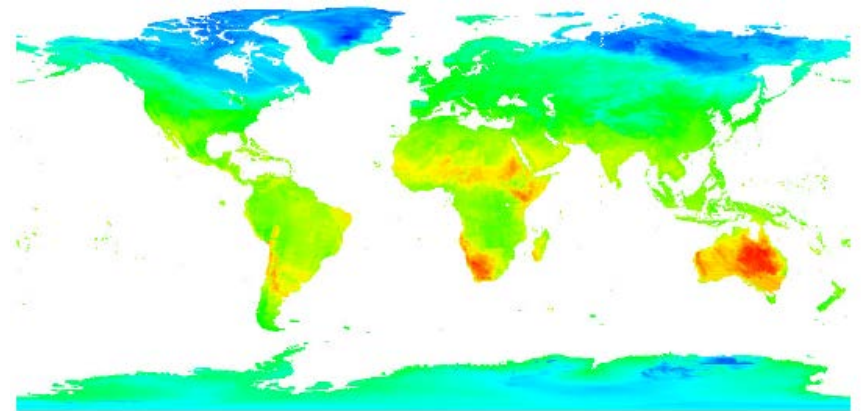


HadoopViz: A MapReduce Framework for Extensible Visualization of Big Spatial Data (5B-3)

- ▶ A. Eldawy, M.F. Mokbel, C. Jonathan (U. Minnesota)
- ▶ Hadoopを用いた空間データの可視化
- ▶ 特徴
 - ▶ 3フェーズのアプローチ: `partition-plot-merge`
 - ▶ **スムージング処理**への対応: 近傍のデータを一緒に処理する必要あり
 - ▶ **複数レベルの解像度**での出力にも対応 (zoom-in/out)
 - ▶ 可視化処理の抽象化
 - ▶ さまざまなデータ (散布図, 道路ネットワークなど) に対応
 - ▶ スケーラビリティ

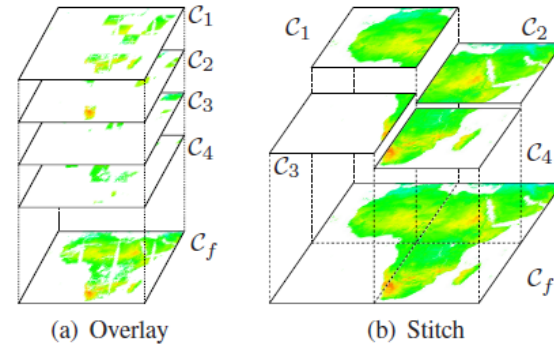
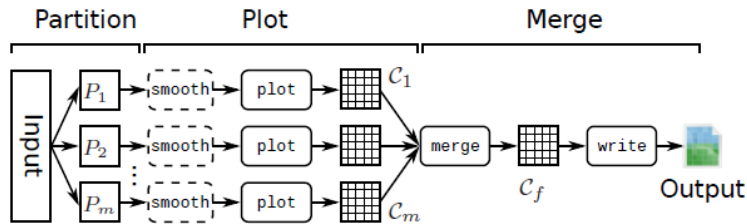


Some figures are incorporated from the original paper.

アイデア

▶ Partitioning-Plot-Merge処理

- ▶ 必要に応じてスムージングが入る
- ▶ 分割はoverlayではなくstichを想定

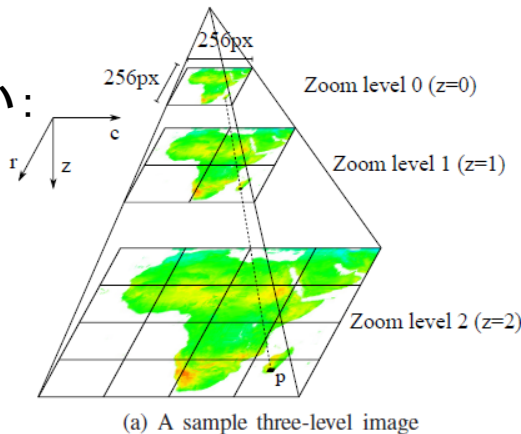


スムージングを入れるためには
stichでないと対応不可なため

▶ 複数レベル解像度への対応

複数レベルを一括して処理したい:

- ・単純な実装では、メモリを圧迫 (多くのタイルを保持)
- ・また、マージ処理のコスト大

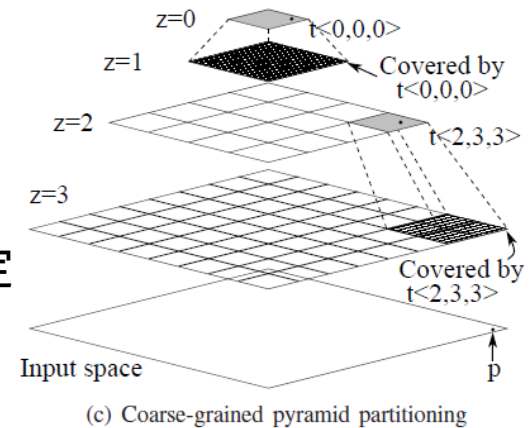


(a) A sample three-level image

解決策:
ピラミッド分割

[PODS'90]

- ・各ノードが担当するタイルを固定
- ・マージ処理が不要



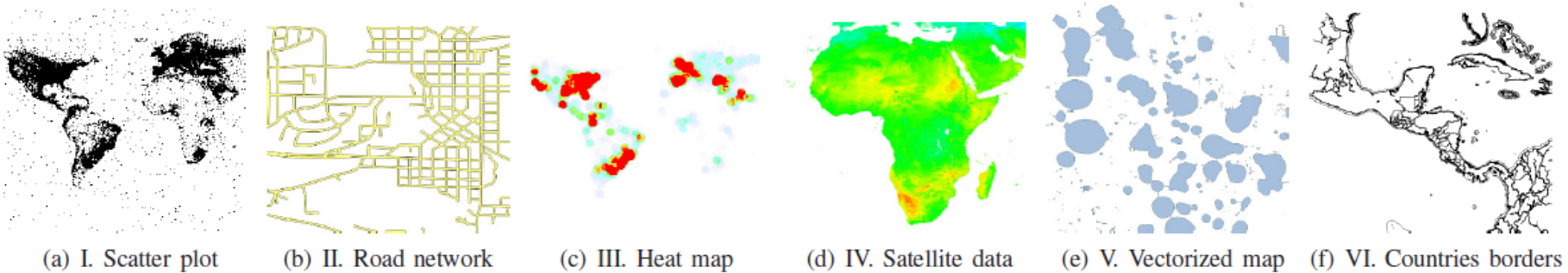
(c) Coarse-grained pyramid partitioning

アイデアと評価

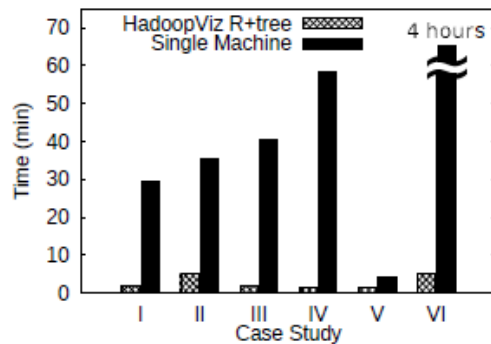
▶ 可視化の抽象化: 各処理をカスタマイズ可能

- ▶ スムージング処理, キャンバス生成処理, プロット処理, マージ処理, 出力処理

▶ 適用事例: それぞれについて実装

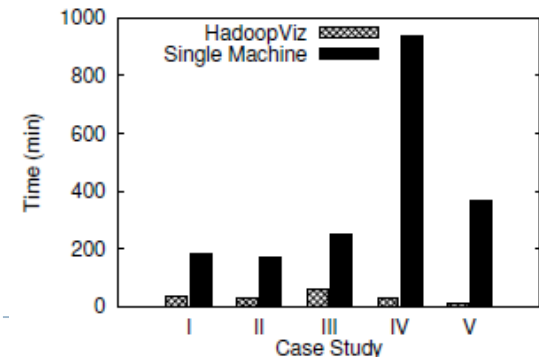


▶ 実験による評価: 各種パラメータに対する処理時間



単一解像度
の場合

複数解像度
の場合



(a) Case studies

(a) Case studies

担当: 石川(名大)

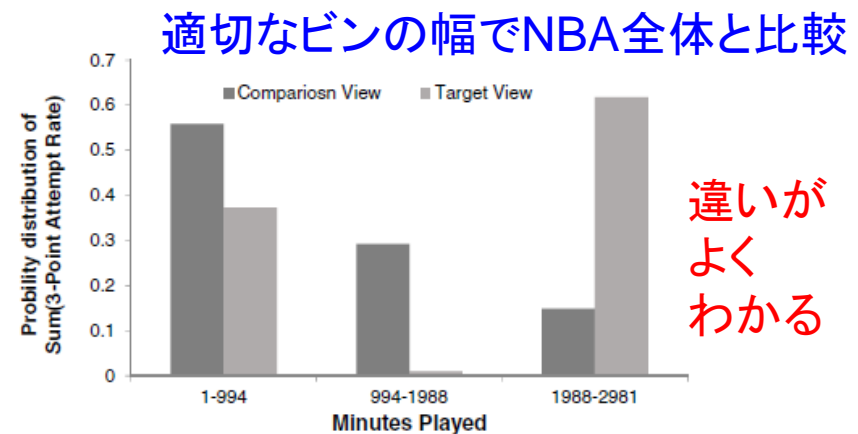
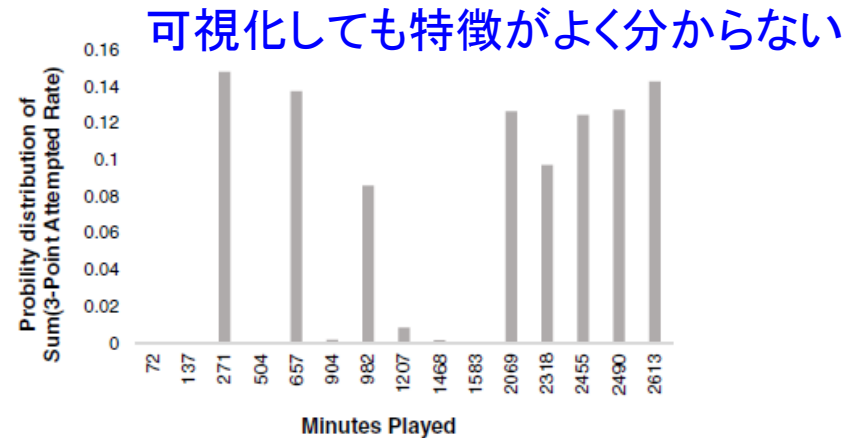
MuVE: Efficient Multi-Objective View Recommendation for Visual Data Exploration (6B-2)

▶ H. Ehsan, M.A. Sharaf, P.A. Chrisanthis (U. Queensland / U. Pittsburgh)

▶ 例: NBA選手のデータ

- ▶ 対象: Golden State Warriors (GSW)
- ▶ ヒストグラムを構築
- ▶ X軸: 選手が出場した時間
- ▶ Y軸: 3ポイントシュート達成率

```
SELECT MP, SUM(3PAr)
FROM Players
WHERE team = "GSW"
GROUP By MP
```



Some figures are incorporated from the original paper.

問題設定

- ▶ 基本となる問合せ: ユーザが指定

Q: SELECT * FROM DB WHERE T

- ▶ Tは述語の組合せ(例: team = “GSW”)

- ▶ 候補ビュー V_i : システムが列挙

V_i : SELECT A, F(M), FROM DB WHERE T
GROUP BY A
NUMBER OF BINS b

- ▶ どの属性 A でグループ化, ビンの数 b をいくつにするかで複数の候補が生成
- ▶ 集約関数 F(M) にも複数の候補あり

- ▶ 問題: 可視化結果として適切な
トップkのビューを推薦

- ▶ 3つの指標を考慮

- ▶ 興味深さ

- ▶ Interestingness

- ▶ 確率分布としてみたときの偏りを評価

- ▶ 有用性 (Usability)

- ▶ 相対的なビンの幅 b が小さい(きめ細かい)方がよい

- ▶ 正確性 (Accuracy)

- ▶ ヒストグラムによる近似が正確な方がよい

ビューの選択法

- ▶ ビューおよびビンの幅の**有用性**: 重み付け和として定義

$$U(V_{i,b}) = \alpha_D \times D(V_{i,b}) + \alpha_A \times A(V_{i,b}) + \alpha_S \times S(V_{i,b})$$

- ▶ 探索空間: 以下の組合せ

- ▶ グループピングに用いる属性と集約関数 (AVGなど)
- ▶ ビンの幅

- ▶ 目的: 有用性が高い**トップk個のビュー**を選択

- ▶ ビンの幅違いのみのビューが複数個入らないようにする

- ▶ 探索手法

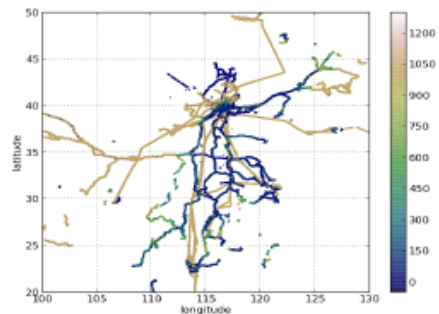
- ▶ 水平探索 (ビンの幅) と垂直探索 (ビュー定義) の組合せ
- ▶ 線形探索法, 局所探索法, MuVE法の比較
 - ▶ MuVE法: 枝刈りを工夫し, 不要なビュー・不要な有用性評価を省く

Visualization-Aware Sampling for Very Large Databases (6B-4)

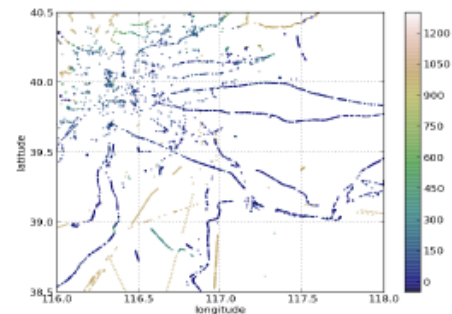
▶ Y. Park, M. Cafarella, B. Mozafari (U. Michigan)

▶ 結果のデモ

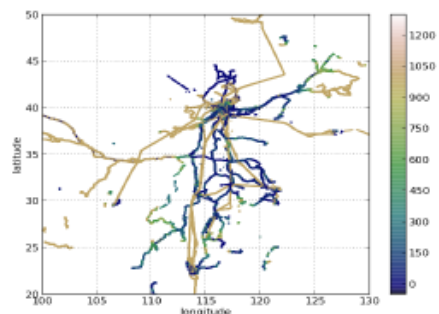
- ▶ 対象: GeoLifeデータセット
- ▶ 層化サンプリングとVAS(提案手法)の比較
- ▶ 全体的な表示にはあまり違いなし
- ▶ ズームインすると大幅に違う: VASでは重要な構造を保持
- ▶ どうサンプリングする？



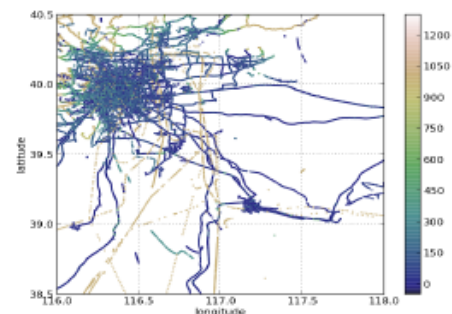
(a) Stratified Sampling (overview)



(b) Stratified Sampling (zoom-in)



(c) VAS (overview)



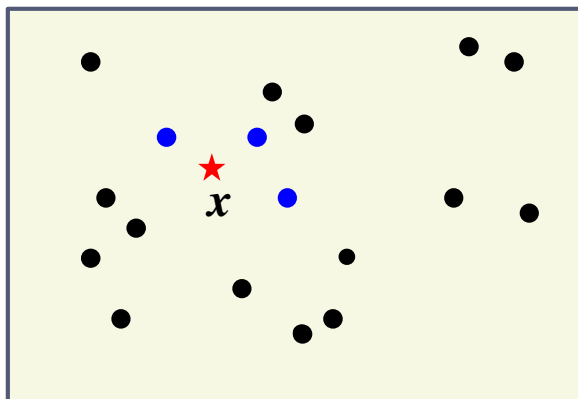
(d) VAS (zoom-in)

Some figures are incorporated from the original paper.

アイデア

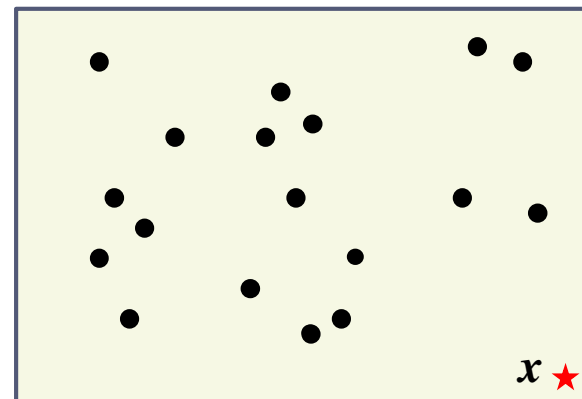
- ある位置 $x = (x, y)$ に着目したサンプリングの評価

- 例: $K = 3$ 個をサンプリング



← x の近傍の情報が保存されているとうれしい

x の周辺に何もなければどうでもよい→



- 点 x に対する評価を**損失関数**として表現: 小さい方がよい

$$\text{point-loss}(x) = \frac{1}{\sum_{s_i \in S} \kappa(x, s_i)}$$

κ は近傍に重み付けする関数: たとえば

$$\kappa(x, s_i) = \exp(-\|x - s_i\|^2 / \varepsilon^2)$$

$S = \{s_1, \dots, s_K\}$ はサンプル集合

サンプリング処理

▶ 最適化問題として定式化

- ▶ 問題の制約などを考慮して変換

$$\min \int \text{point-loss}(\mathbf{x}) d\mathbf{x} \rightarrow \min \int \sum_{s_i, s_j \in S} \kappa(\mathbf{x}, s_i) \kappa(\mathbf{x}, s_j) d\mathbf{x}$$

この積分が最小になる
K要素のサンプル集合Sを選ぶ

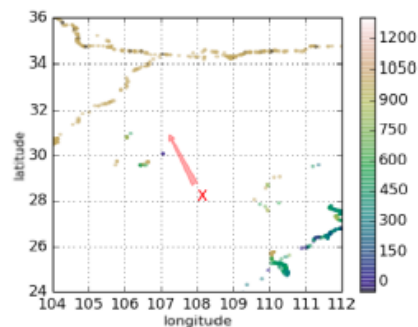


▶ 解法

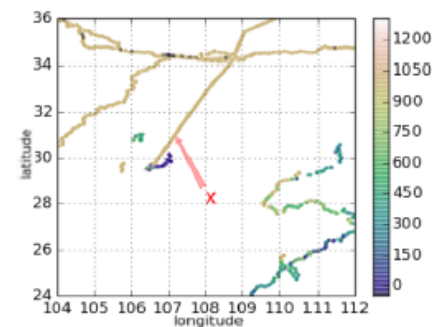
- ▶ 本問題はNP困難 → 近似解法を開発: 交換に基づく
- ▶ アルゴリズムの工夫: $O(NK)$ に

▶ 評価: 可視化の質と速度

- ▶ 質の評価ではクラウドソーシング
グを利用
- ▶ 指定された点の高度推定など



(a) Stratified Sampling



(b) VAS