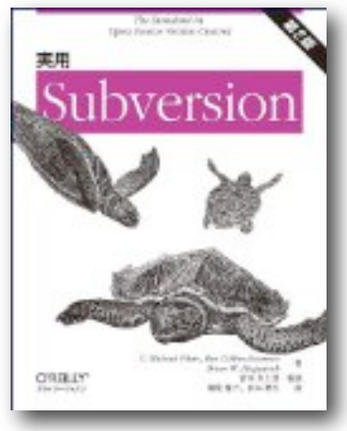
logopolisの裏側

浜本 階生

自己紹介

- 浜本 階生（はまもと かいせい）
- http://d.hatena.ne.jp/kaiseh/
- S2Swingのコミッタ
- 共訳『実用 Subversion 第2版』
  （オライリー・ジャパン）



テーマ

Blogopolisとは？

アルゴリズム

インフラ

Blogopolisとは？

動機

ブログ

↓

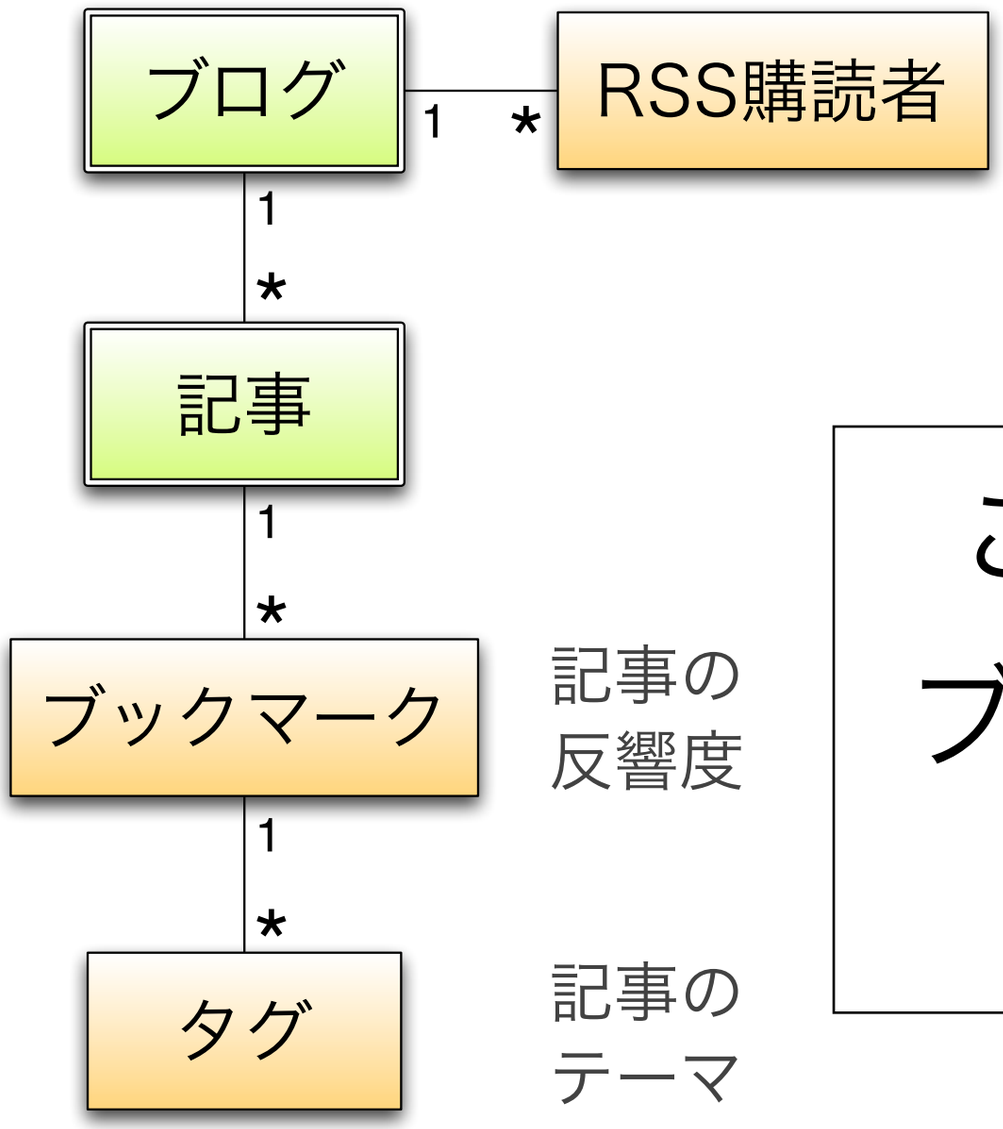
特定分野に的を定めて

PageRankでは得られない

新たな価値を提供する

情報検索システムを作りたい

ブログとソーシャルデータ



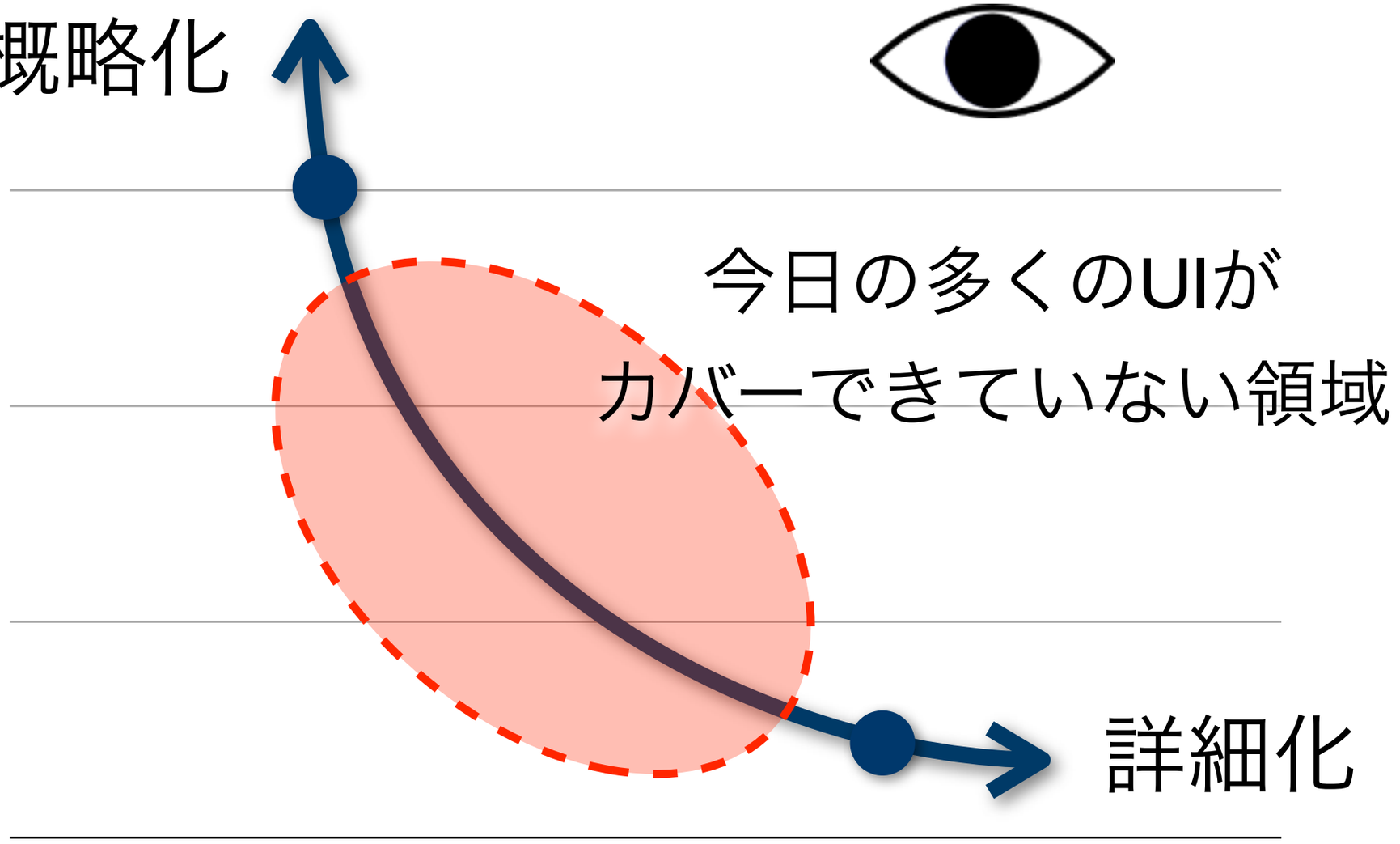
ブログの
オーソリティ

記事の
反響度

記事の
テーマ

この構造を活用した
ブログ検索システムが
存在しない

情報のトレードオフ



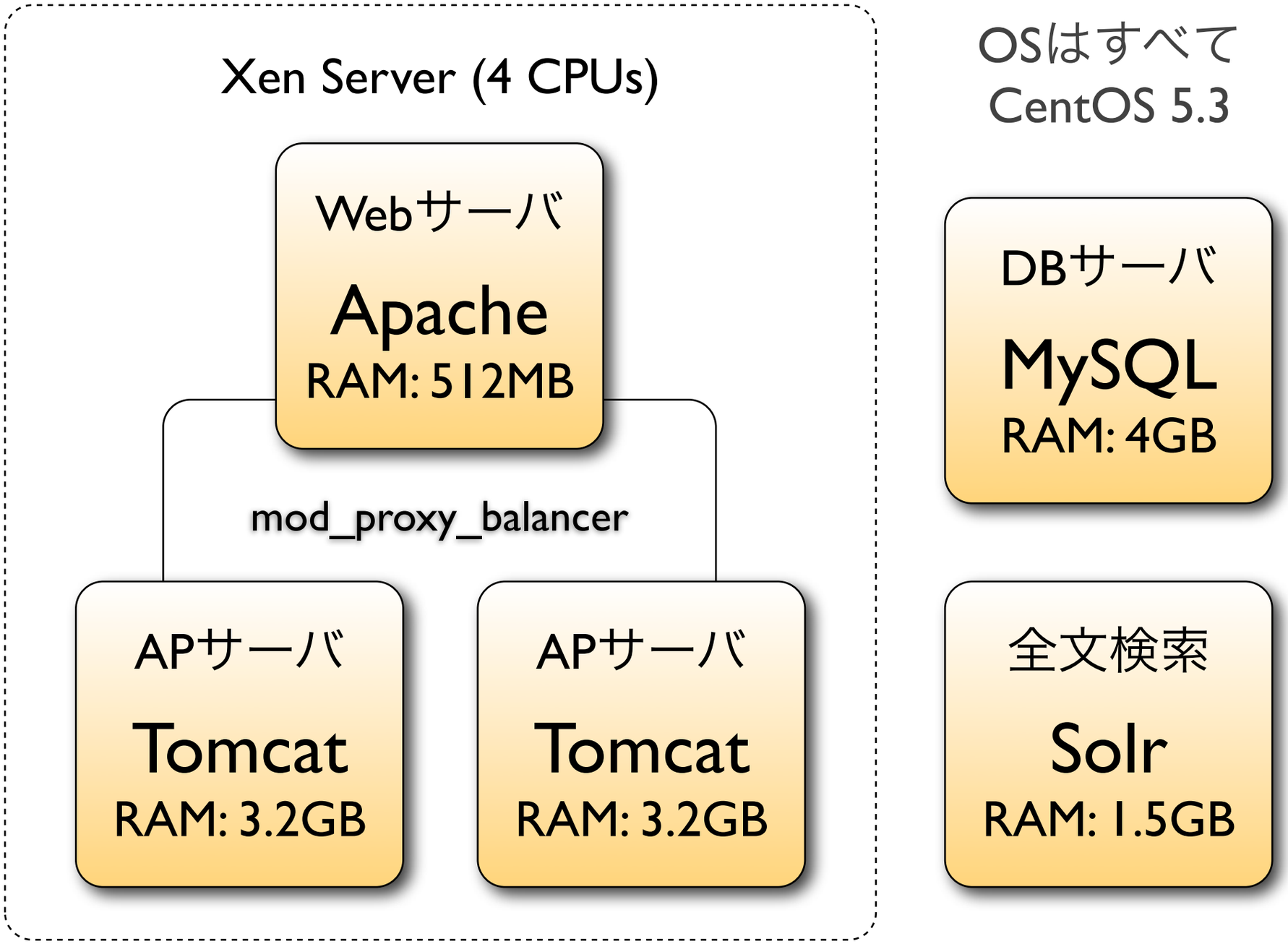
情報粒度の断絶を解く

「流れるようなUI」を作りたい

http://blogopolis.jp/

運用状況

サーバ構成


Xen Server (4 CPUs)
Webサーバ
Apache
RAM: 512MB
mod_proxy_balancer
APサーバ
Tomcat
RAM: 3.2GB
APサーバ
Tomcat
RAM: 3.2GB
OSはすべて
CentOS 5.3
DBサーバ
MySQL
RAM: 4GB
全文検索
Solr
RAM: 1.5GB

コード規模

	クラス数	ステップ数
Java *	359	30,000
AS3	73	6,000
計	432	36,000

＊TopHatenarを含む

2009/9/8 現在

データ規模

ブログ数	23万
記事数	30万
頂点数	430万
多角形数	105万
データベースサイズ	9GB (dump)

2009/9/8 現在

アルゴリズム

処理の流れ

1. ブログコンテンツとソーシャルデータの収集
2. ブックマークタグに基づくブログの
   クラスタリング
3. クラスタリング結果の2次元平面への
   マッピング

今日のトピック

クラスタリング

はてなブックマーク

ブックマークタグ

タグの集計

	IT	business	仕事	…
id:higayasuo	333	302	386	
id:fromdusktildawn	1301	68	2077	
id:kawango	634	266	18	

K-means法

- 非階層的なクラスタリング手法
- あらかじめクラスタの個数を決めて
  データを分類
- 高速に計算可能

階層的クラスタリング

- 位置の近いクラスタ同士を
  再帰的に統合
- クラスタ階層のツリー
  （デンドログラム）が生成される
- 計算量が大きい

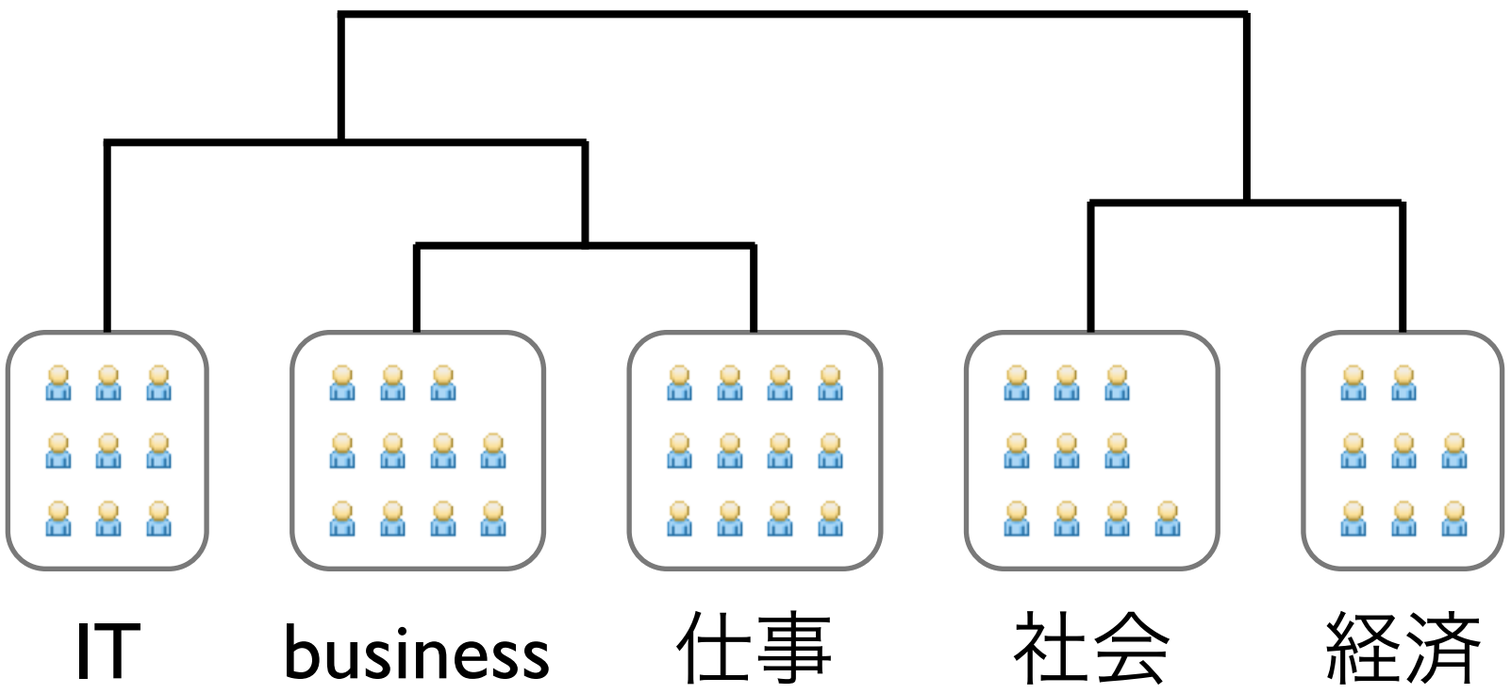
K-means法の適用

- 各ブログをデータ点とし、タグの出現頻度をその成分とする

		IT	business	仕事	...
p_0	id:higayasuo	333	302	386	
p_1	id:fromdusktildawn	1301	68	2077	
p_2	id:kawango	634	266	18	

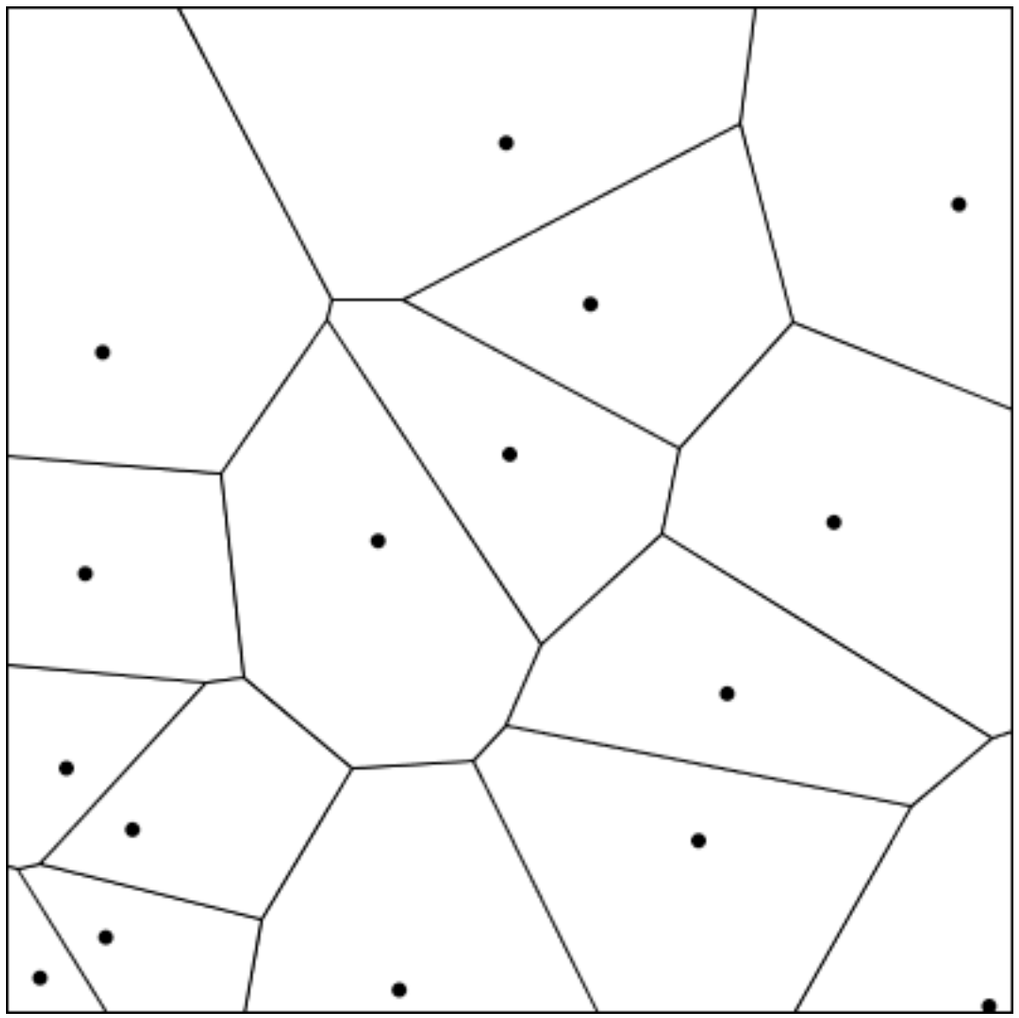
階層的クラスタリングの適用

- K-means法の実行結果として得られたブログクラスタの集合を入力とする

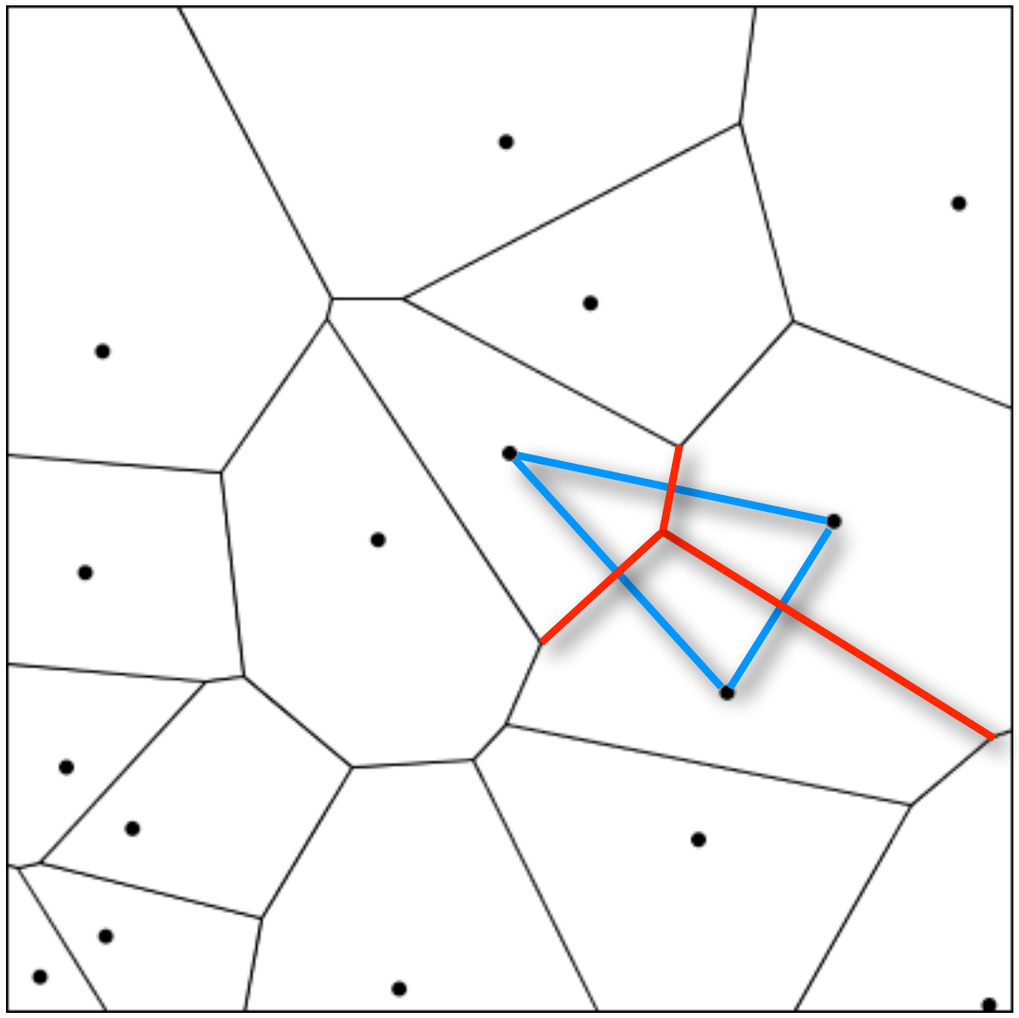


ボロノイ図

ボロノイ図



ボロノイ図の性質



ボロノイ辺は母点の

垂直二等分線

ボロノイ図の一般化

- 通常のボロノイ図

$$distance(p_i, q) = \sqrt{(x_i - x)^2 + (y_i - y)^2}$$

- Additively Weighted Voronoi Diagram

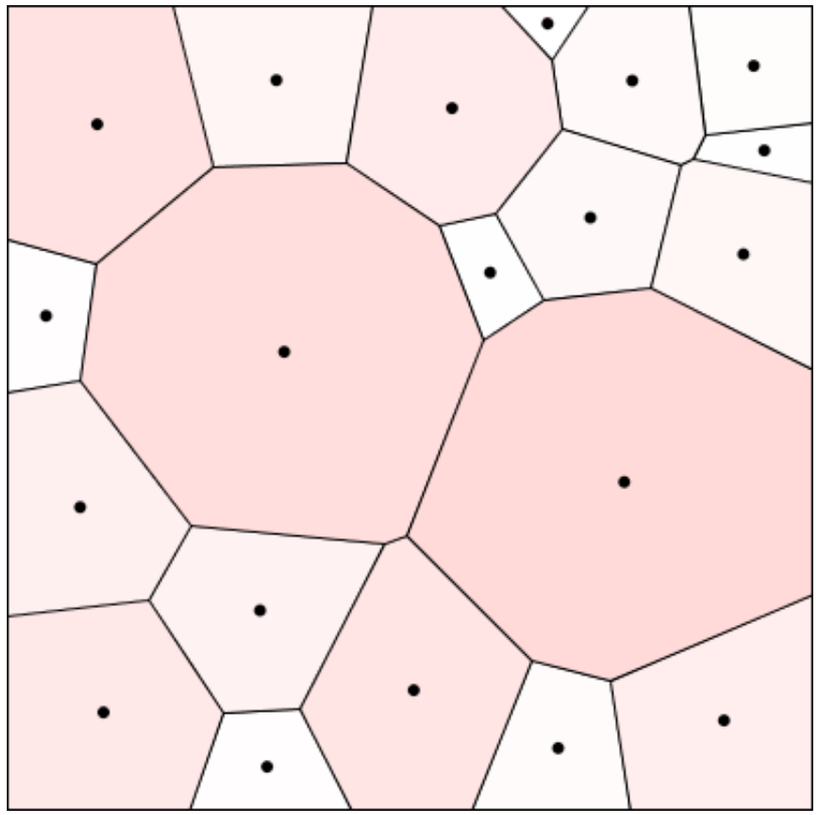
$$distance(p_i, w_i, q) = \sqrt{(x_i - x)^2 + (y_i - y)^2} - w_i$$

- Additively Weighted Power Voronoi Diagram

$$distance(p_i, w_i, q) = (x_i - x)^2 + (y_i - y)^2 - w_i$$

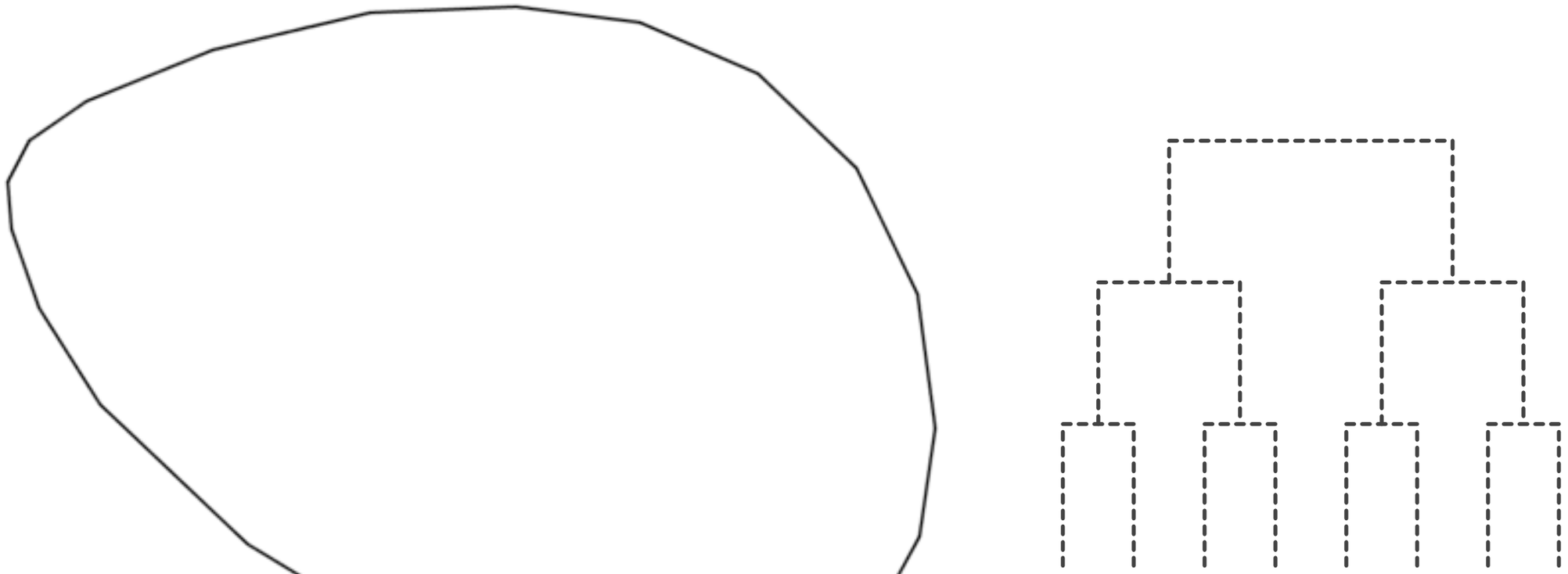
Centroidal Voronoi Tessellation

with Additively Weighted Power Voronoi Diagram

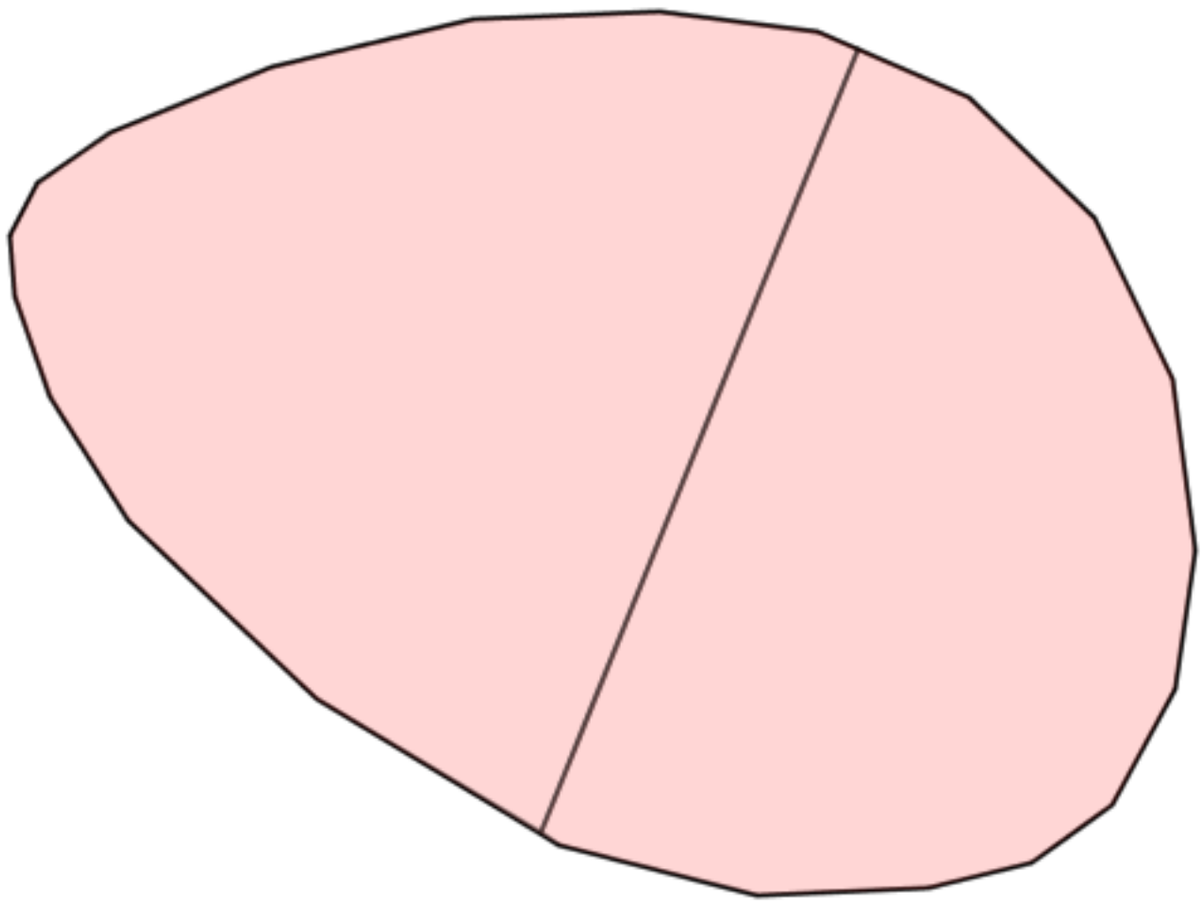


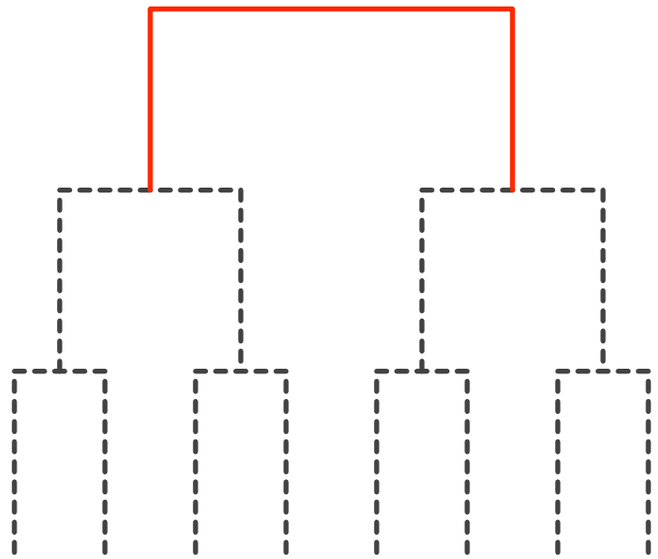
http://www.vimeo.com/6551478
にアップしました

Voronoi Treemap

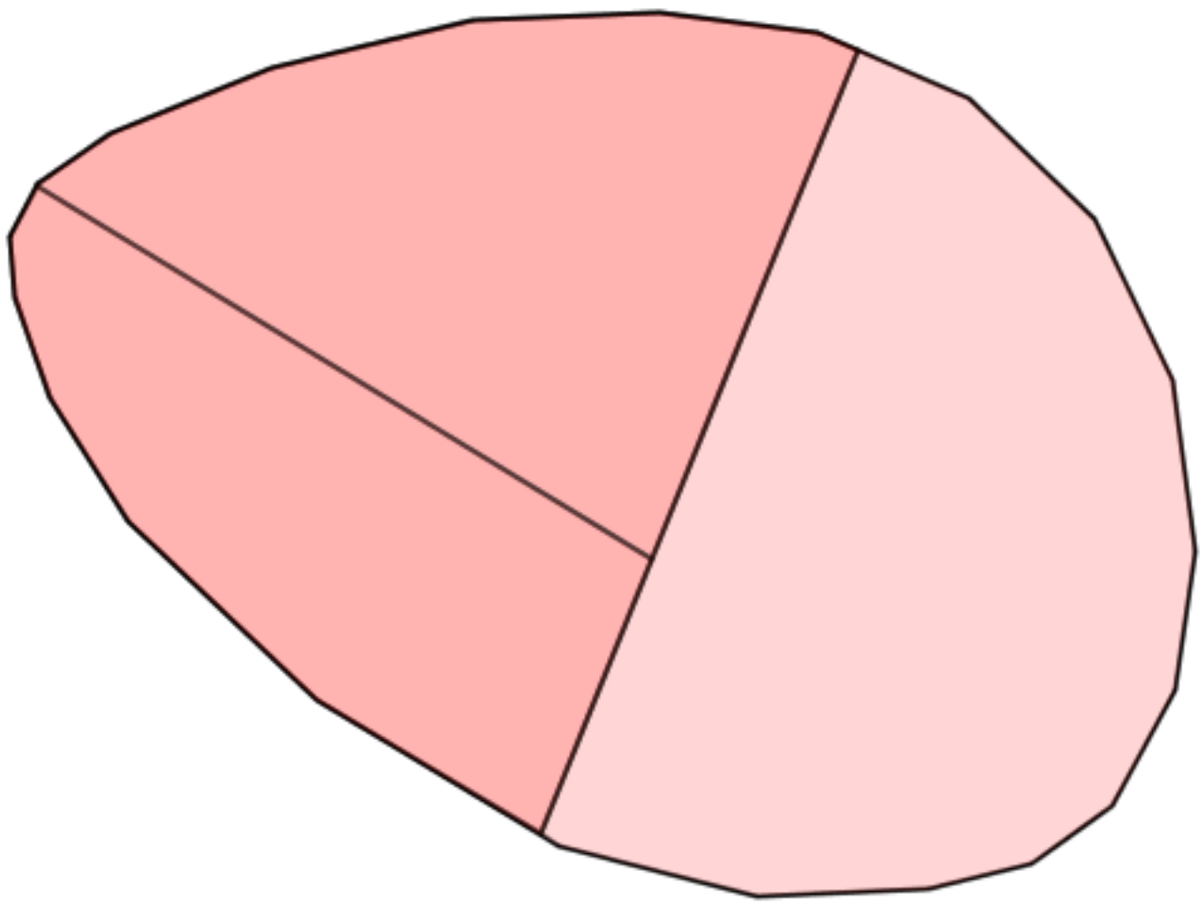


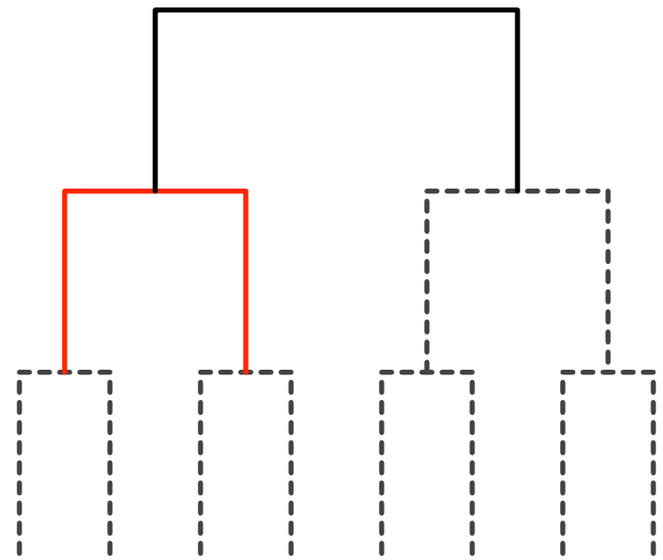
Voronoi Treemap



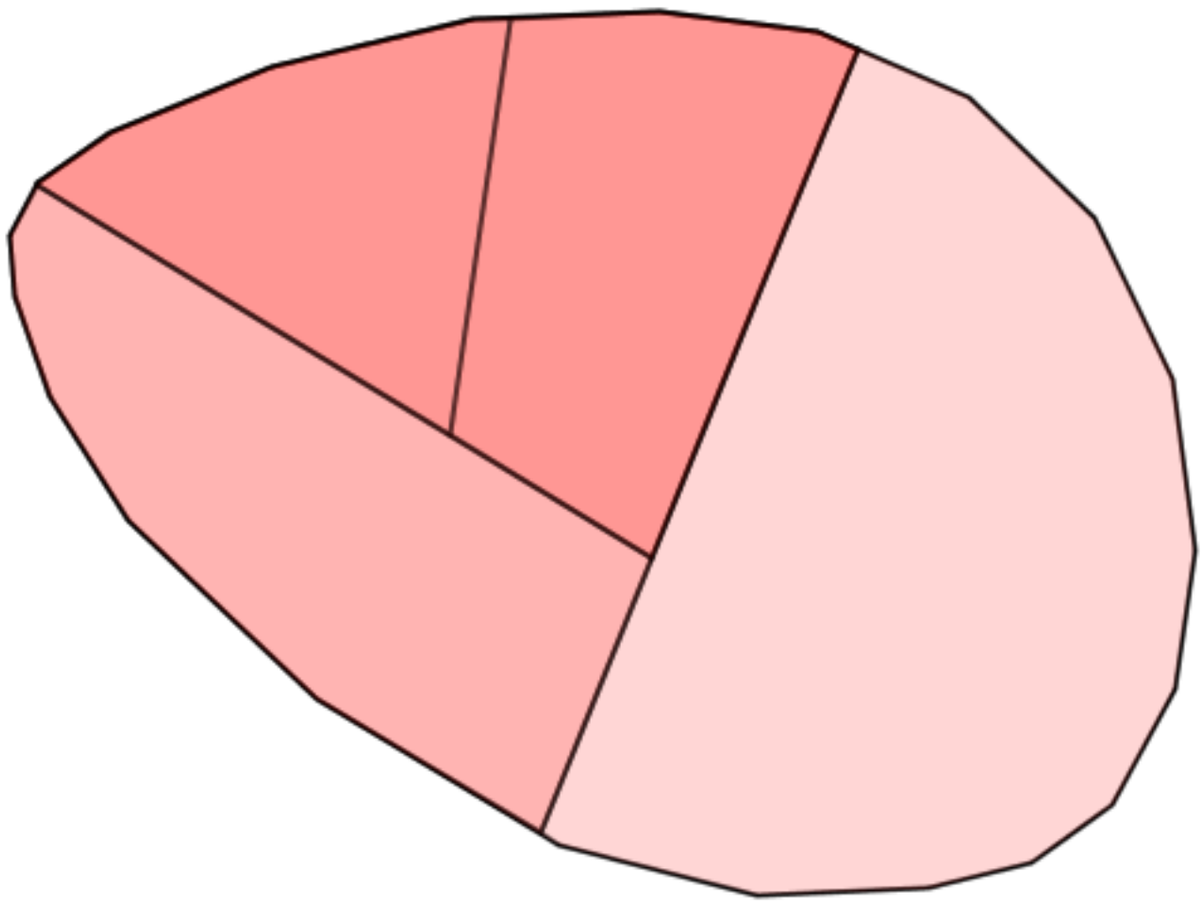


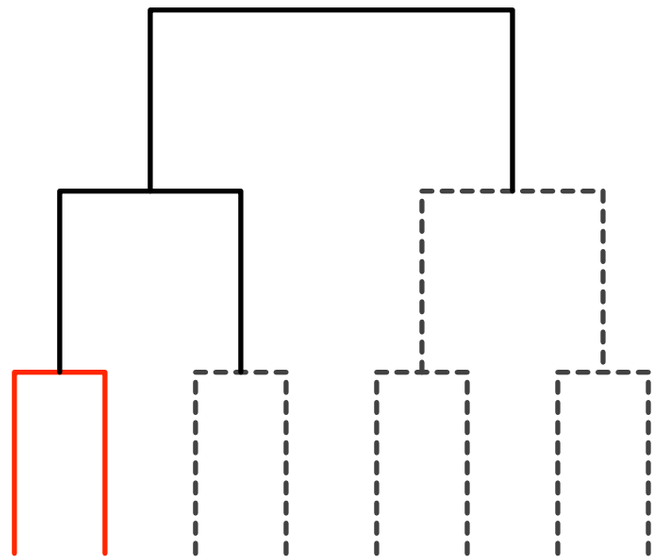
Voronoi Treemap



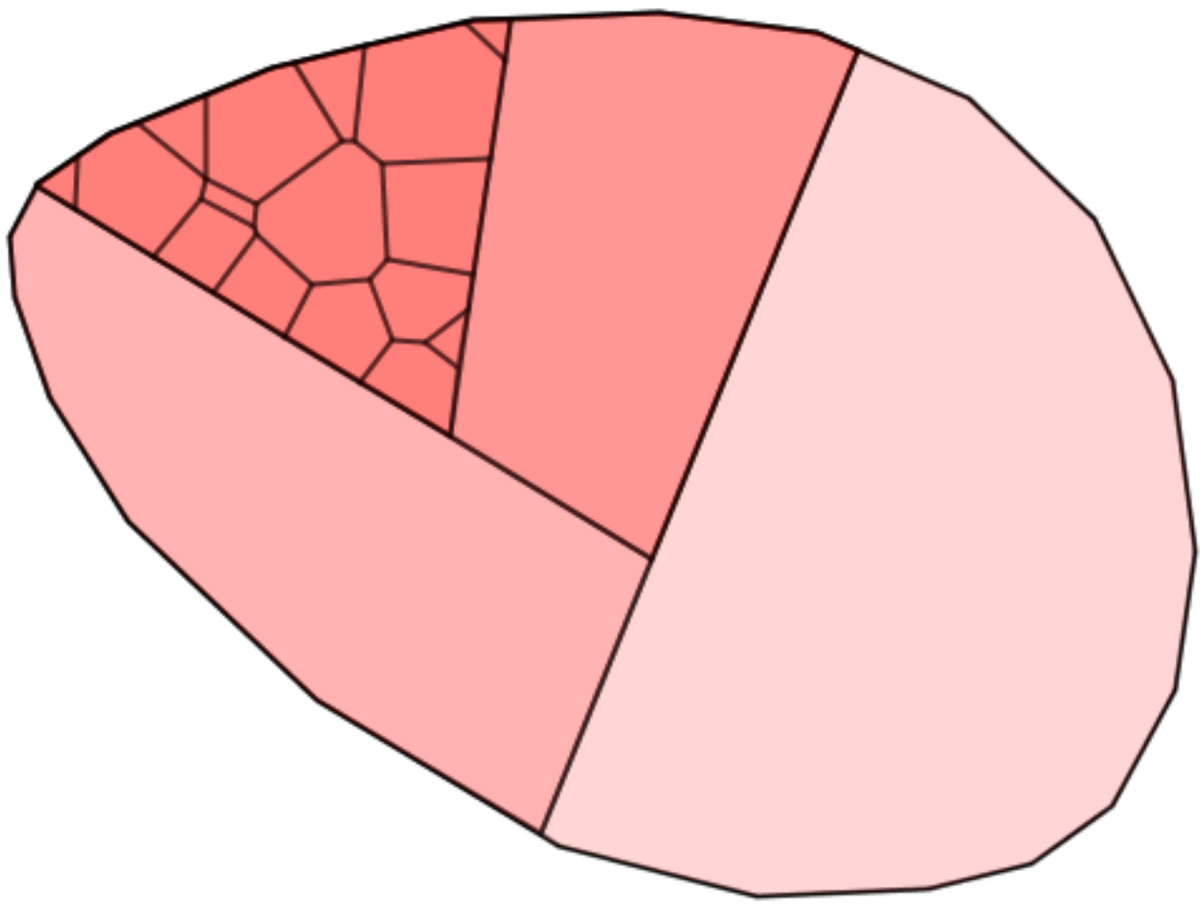


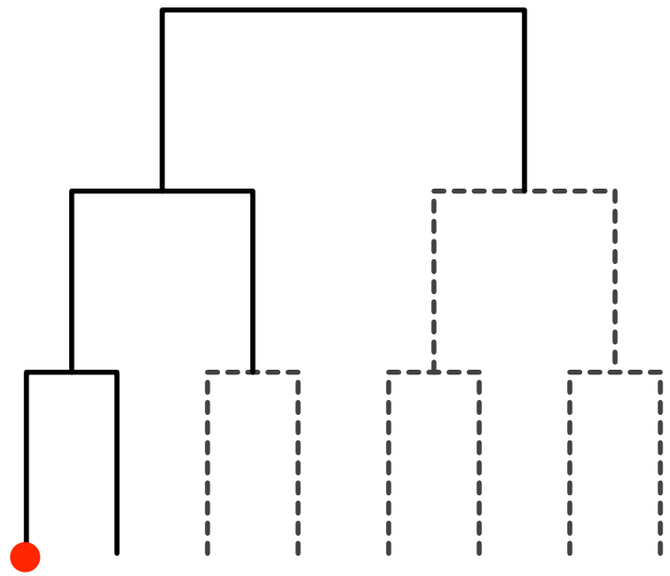
Voronoi Treemap



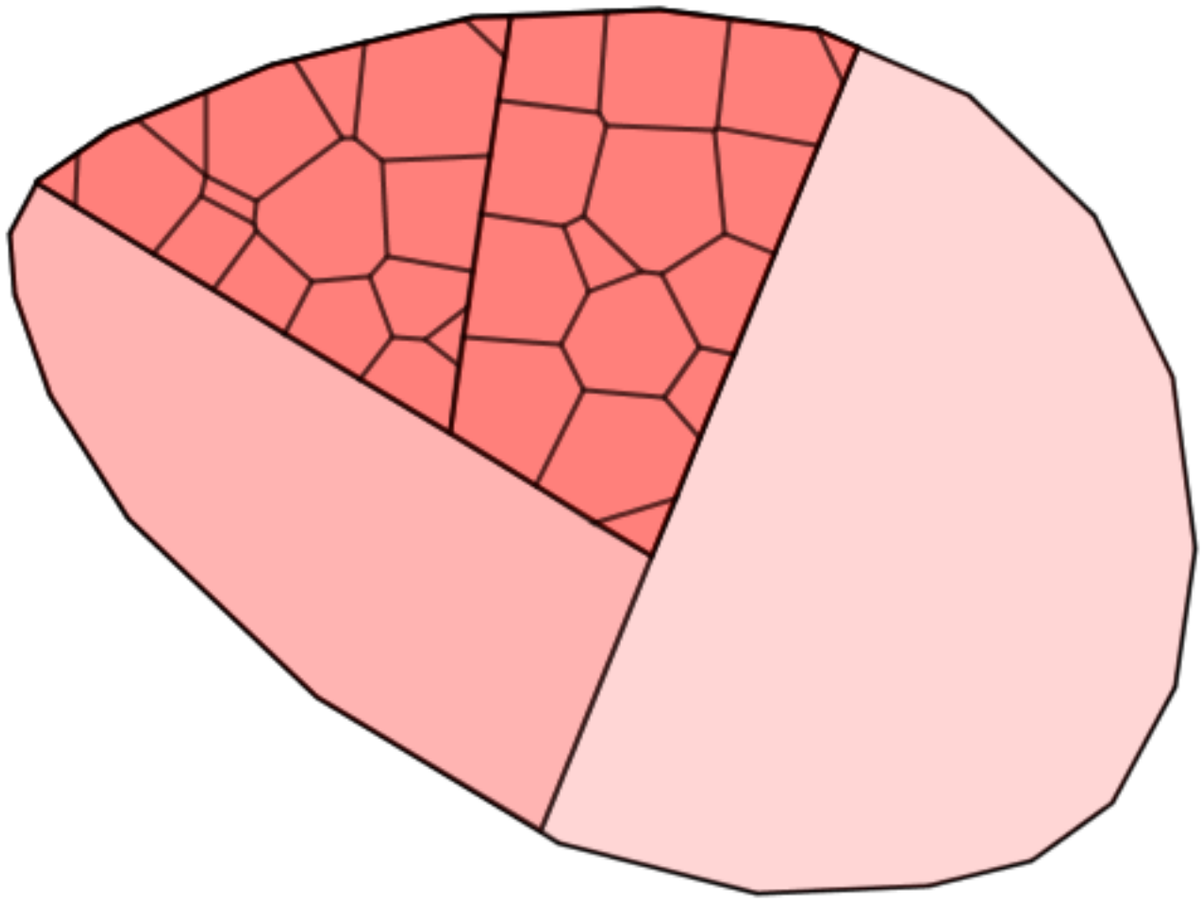


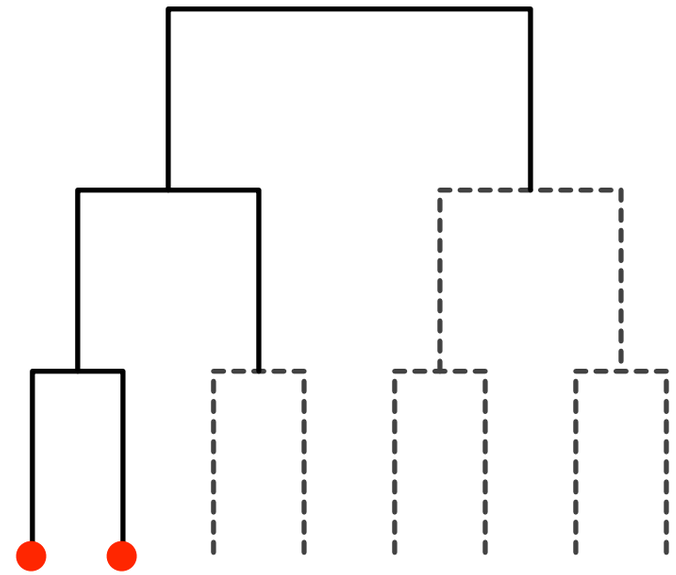
Voronoi Treemap





Voronoi Treemap





計算幾何

- 多角形の包含判定、面積計算、重心計算、切断など

- java.awt.geomだけでは対応できない
- サーバサイドとFlashの両方で必要
- 幾何ライブラリを自作

3D

Papervision3D

- AS3で書かれたFlash用3Dエンジン
- 高速に動作（要チューニング）
- QuadTreeレンダラをサポート
  - ポリゴンの前後関係を正確に描画可能

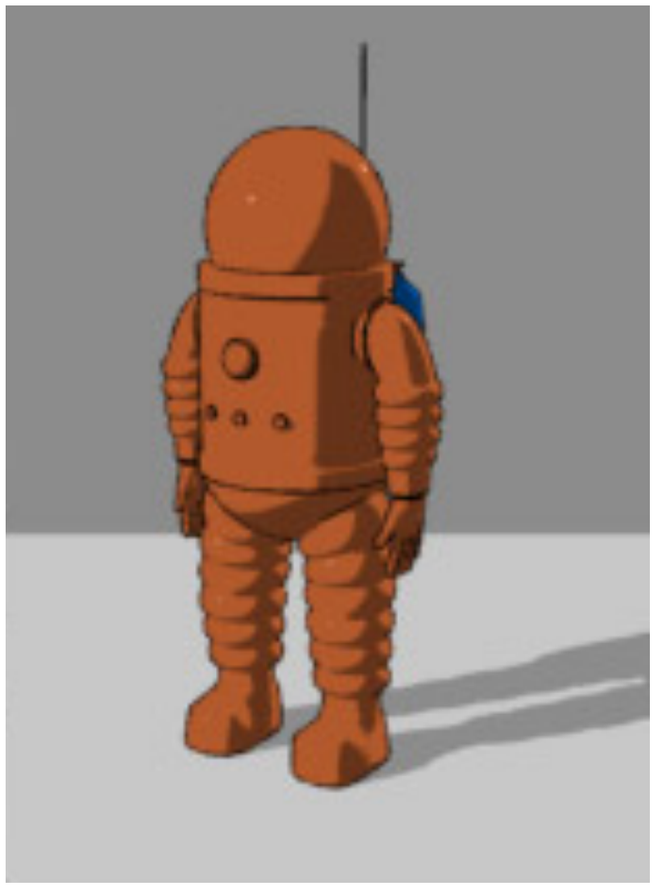
http://code.google.com/p/papervision3d/

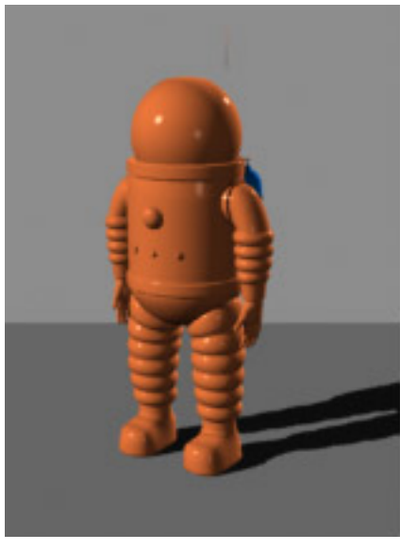
3D化にあたって

- 「いかにも3D」な無機質の雰囲気は避けたい
- ブログは人が書いているものだから「人の温もり」を出したい

トゥーンレンダリング

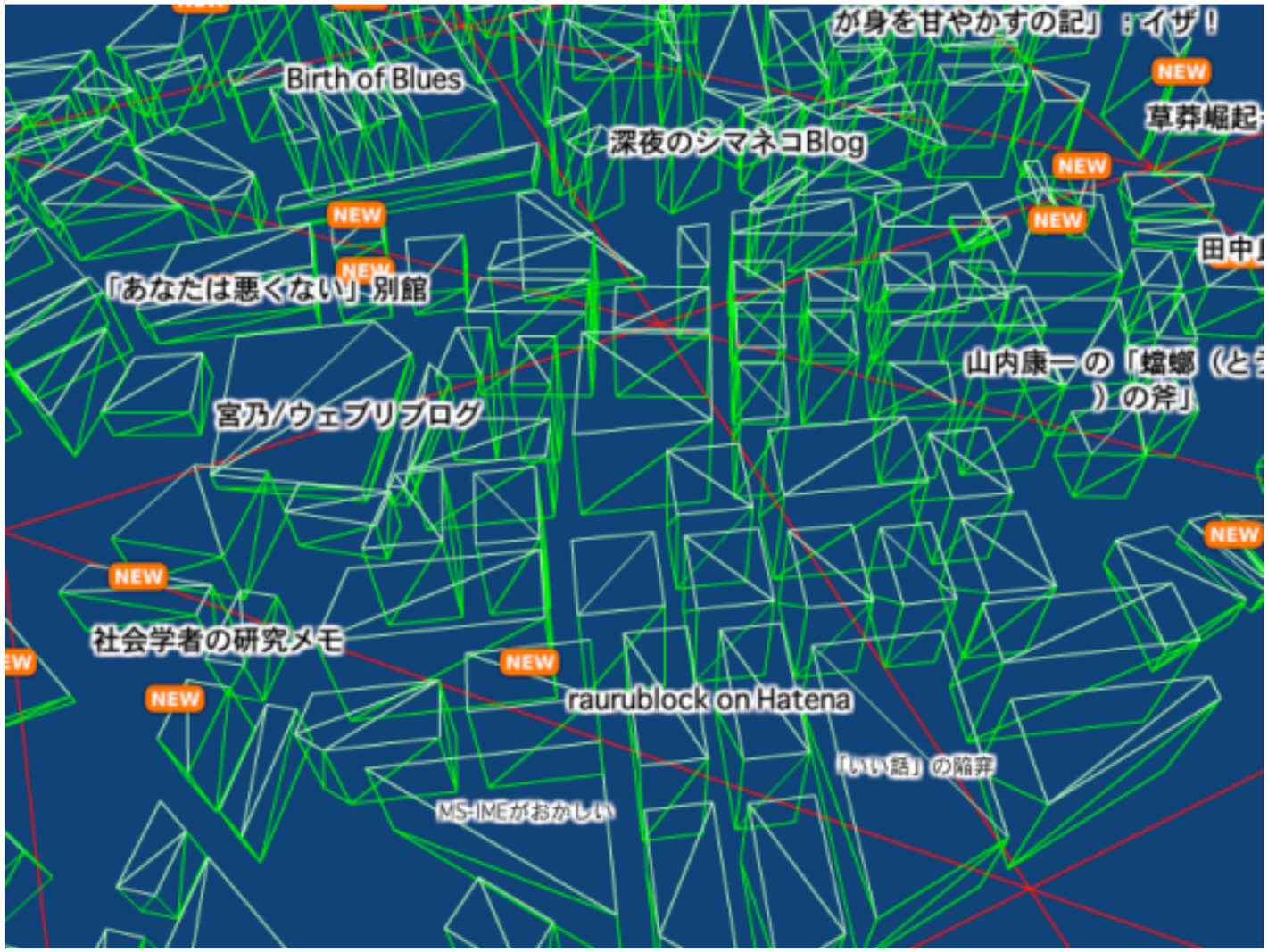
- 色調の単純化
- 輪郭線





（Wikipediaより）

シェーディング (I)

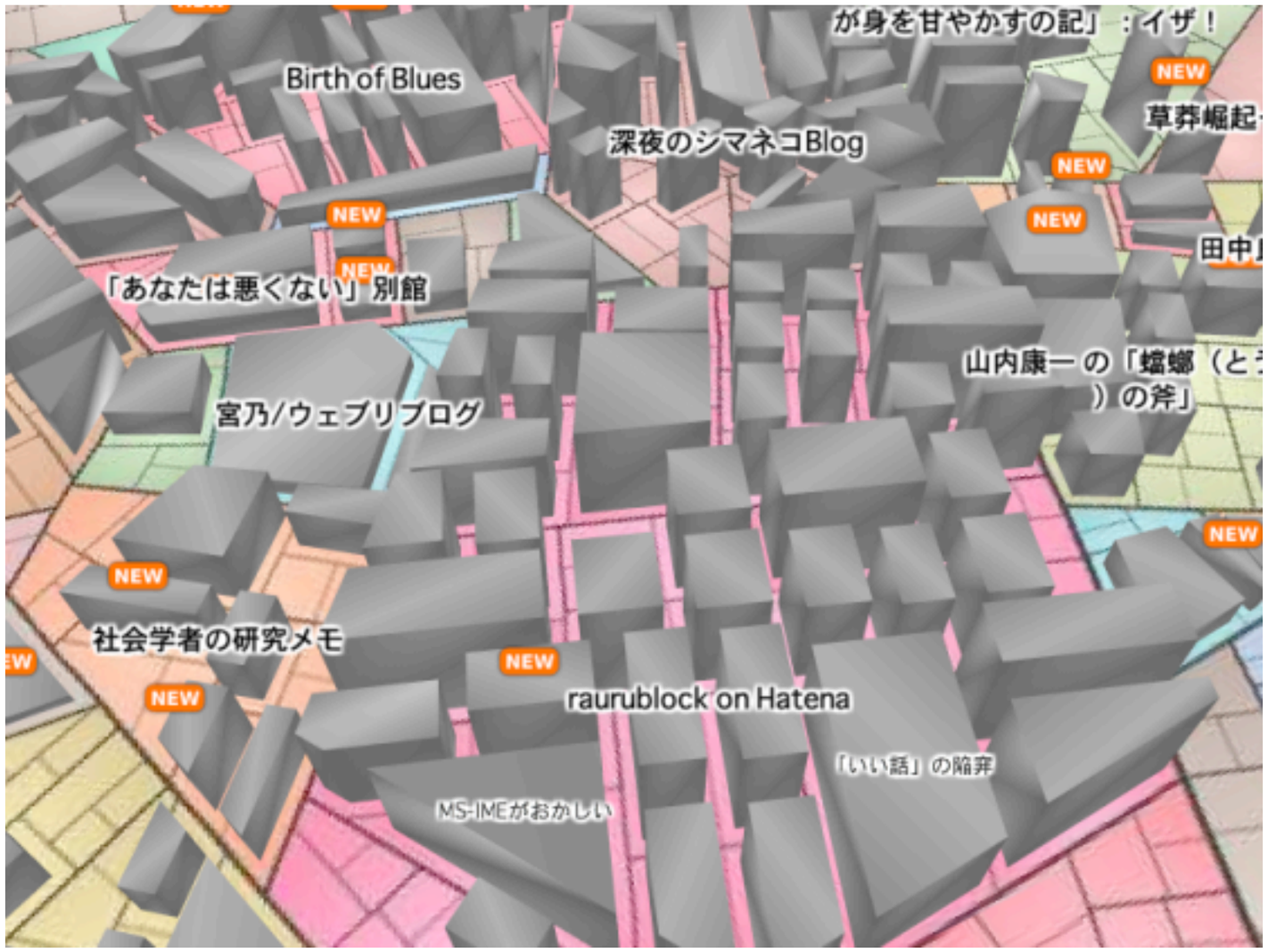


WireframeMaterial

シェーディング (2)

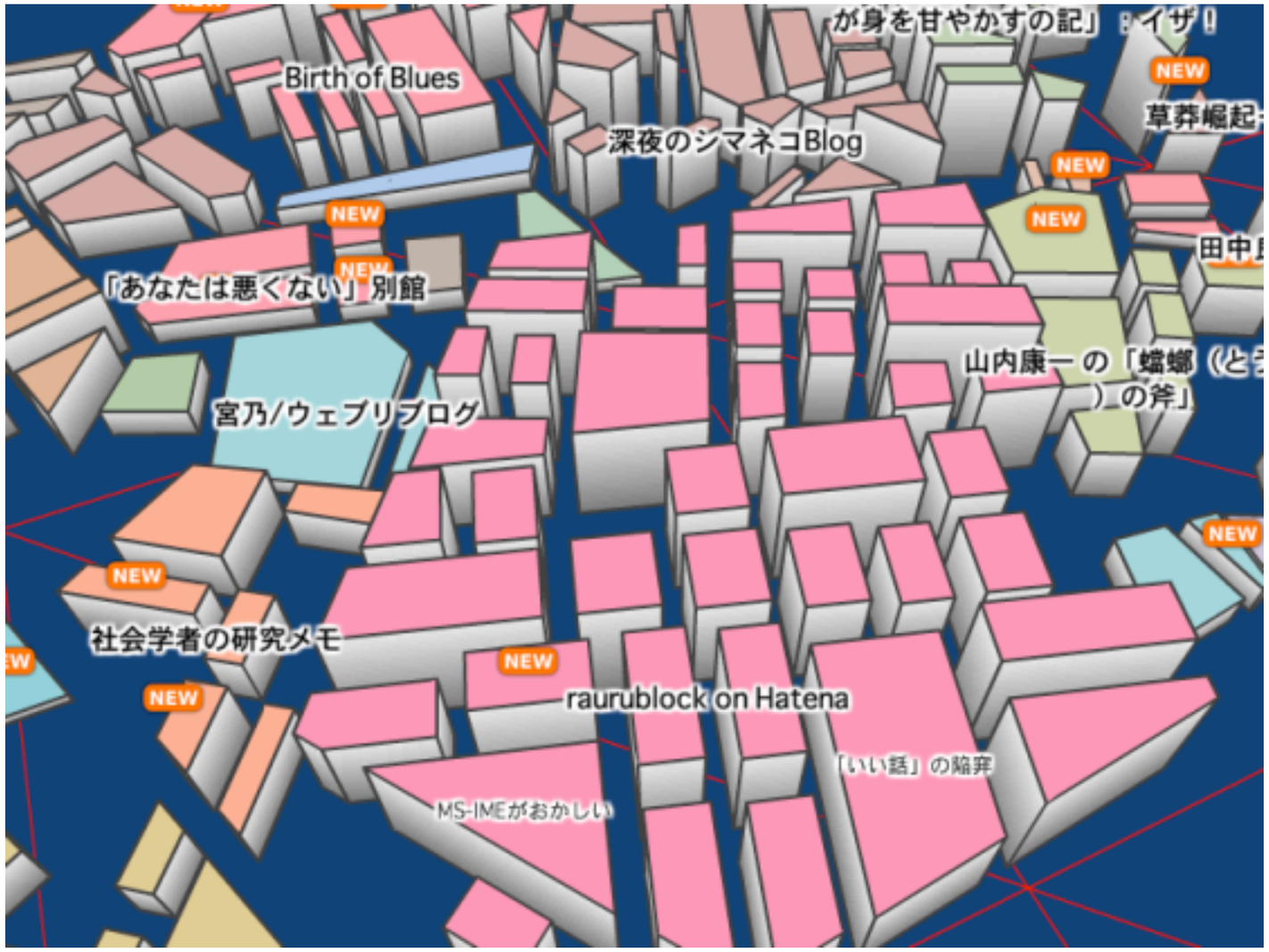
PhongMaterial

シェーディング (3)



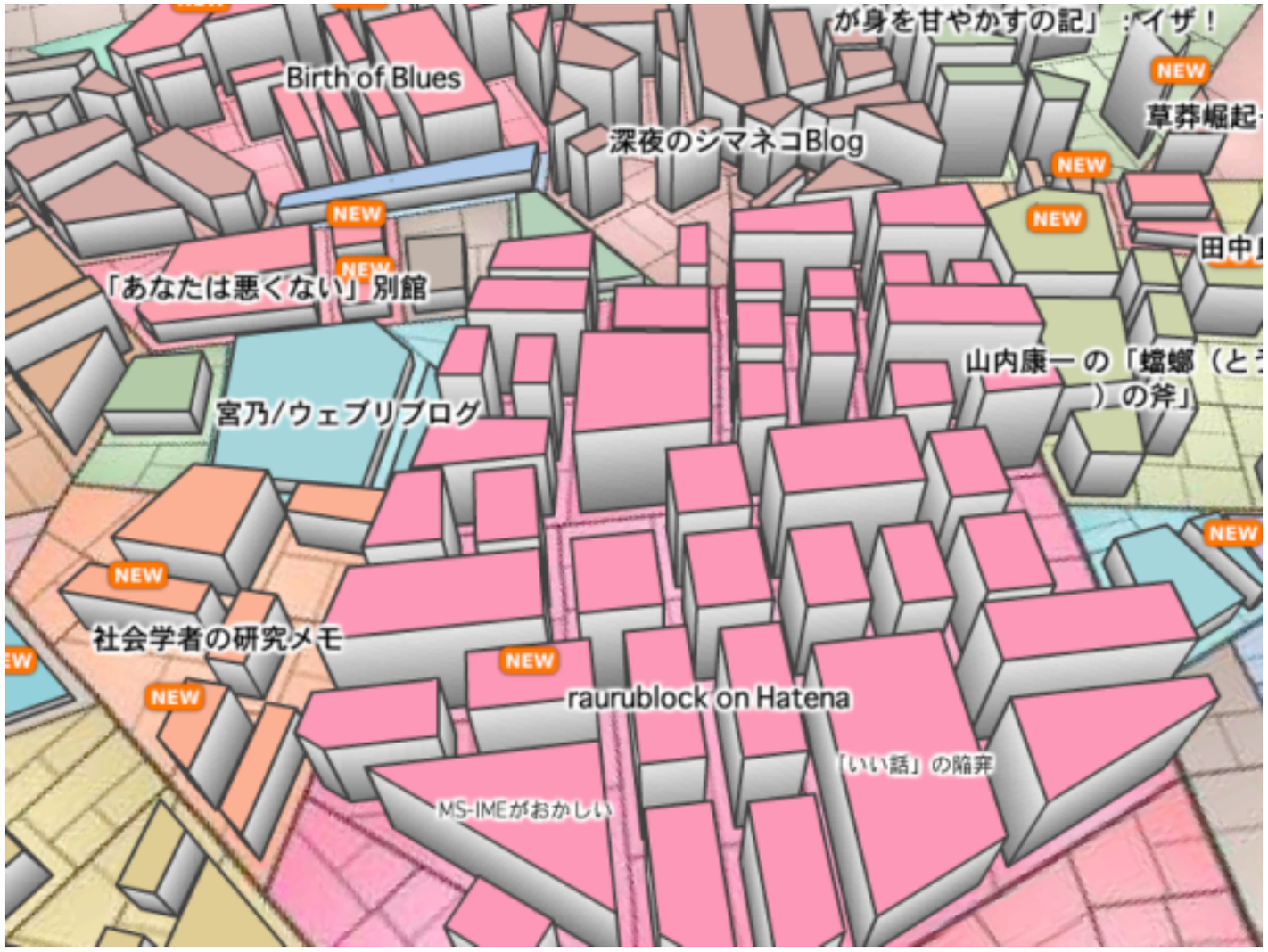
PhongMaterial + 地表

シェーディング (4)



自作マテリアル（トゥーンレンダリング風）

シェーディング (5)

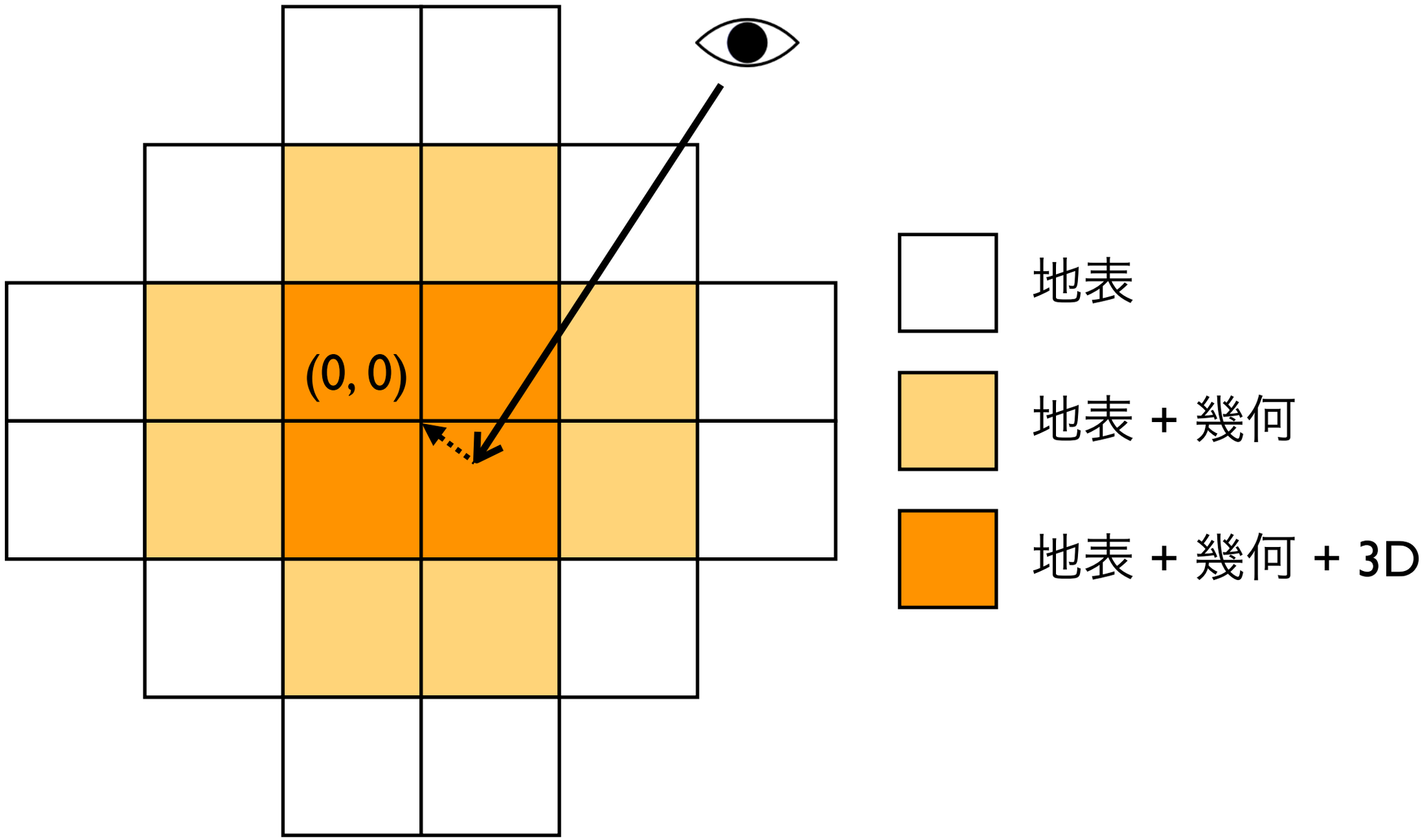


完成形

インフラ

データの扱い

グリッドシステム



ポリゴンインデックス

- 全ポリゴンのID, 外接長方形座標, 面積等をメモリに保持 → 100MB程度
- クエリ（座標範囲）に適合する全ポリゴンIDを返す
- クエリに応じて最適な二分探索を選択（座標ベース／面積ベース）

GNU Trove

- プリミティブコレクションライブラリ
  - ex. TIntArrayList, TIntHashSet, TIntDoubleHashMap

- オープンアドレス法
- あらゆる場面で大活躍

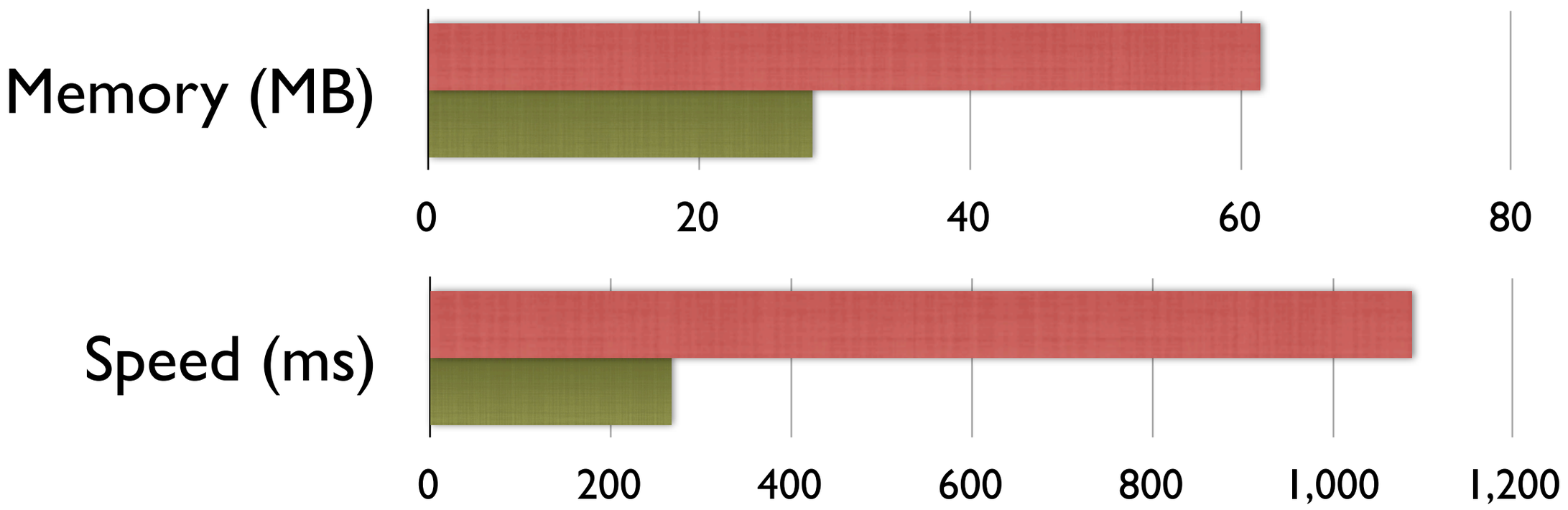
http://trove4j.sourceforge.net/

ベンチマーク

100万個のエントリーをput

java.util.HashMap<Integer, Integer>

gnu.trove.TIntIntHashMap



Ehcache

- 汎用キャッシュエンジン
- JCache (JSR 107)の実装としても利用可能
- Blogopolisでの動作状況
  - キャッシュヒット率： 66%程度
    （TopHatenarは90%程度）

http://ehcache.org/

Adobe Flex

S2Flex2の採用

S2Flex2

- AMF3を独自実装
- Flex 3以降でも使える
- JavaとAS3でPOJOを用意してServiceをコールするだけ！
- BlazeDSと違って設定不要

ByteArray (Flex)

- プリミティブ値や文字列をバイト列にシリアライズ／デシリアライズ
- さらにZLIB圧縮可能
- 座標値など、肥大化しやすいデータの送受信にフル活用

Webフレームワーク

Cubbyの採用

Cubby

- 軽量、簡潔、ハマりがない
- 多様なレスポンスを奇麗に書ける

標準装備	Forward, Redirect, JSON, etc
＋自作	Plain Text, Image, Permanent Redirect

jQuery Autocomplete

テクスチャ画像

SEO

ひがさんの言及

『Strutsにこだわりのない方なら、Cubbyはすごくお勧めです。Struts抜きで自由にWebフレームワークを作ってもいいといわれたら、Cubbyみたいなやつを作りたかった気がする。』

http://d.hatena.ne.jp/higayasuo/20080106/1199620097

データベース

S2JDBCの採用

S2JDBC

タイプセーフ

```
jdbcManager.from(RegionImage.class)
    .where("zoom = ? and x = ? and y = ?",
      zoom, x, y)
    .getSingleResult();
```

可変長引数

S2JDBC

- メリット
  - 「1回限り」のSQLを気軽に組める
- デメリット
  - SQLが散逸しやすい

全文検索

Apache Solrの採用

Apache Solr

- Javaの全文検索サーバ
- Apache Luceneがベース
- HTTPのAPI
  - 言語を問わず利用可能

http://lucene.apache.org/solr/

Solrの使用

- マルチコア（マルチスキーマ）を利用
  - ブログタイトル用スキーマ
  - 記事本文用スキーマ
- CJKAnalyzer（bi-gram）を使用
- URL検索用のトークナイザを自作
- 元データはMySQLに保存

稼働状況

- インデックスサイズ
  - ブログタイトル： 63MB
  - 記事本文： 1.6GB
- 応答時間： 数百ms以内

Solr雑感

- Luceneの単独使用に比べて
  圧倒的に簡単、便利！
- スキーマの設計によっては
  パフォーマンスが大きく低下するので
  注意が必要

まとめ

まとめ

- ソーシャルデータを活用したブログ検索システム
- 情報粒度の壁を打ち破るUIの追求
- S2Flex2, Cubby, S2JDBC, Apache Solr

今後の課題

- 現状は「検索」よりも「探索」に近い
  → 検索の実用性を向上
- 時系列やカンバセーションの視覚化
- コミュニティの醸成

最後に

Blogopolisの開発の支えとなっている

OSSとコミュニティの皆様に

心から感謝します

Blogopolis

http://blogopolis.jp/