地理的情報を考慮したウェブ画像検索方法の提案

和田　修平†　　井上　潮‡

†‡東京電機大学 工学部情報通信工学科 〒101-8457 東京都千代田区神田錦町 2-2

E-mail:　†05kc116@ed.cck.dendai.ac.jp,　‡inoue@c.dendai.ac.jp

あらまし　近年，ウェブ上には多数の画像が存在するようになり，それらを検索するための画像検索サービスも多く提供されるようになった．しかし，従来の画像検索システムにおいては，画像の持つ地理的情報はほとんど考慮されていない．本研究では，画像検索結果から画像が掲載されているウェブサイトの地理情報を抽出し，地図上に配置する画像検索システムを構築した．これによって，画像の地理的な位置関係を把握するなど，地理的情報を考慮した画像検索を行うことが出来るようになる．

キーワード　画像検索，情報抽出，地理情報

1. はじめに

近年，個人のホームページやブログに掲載されている写真画像を中心として非常に多くの画像がウェブ上に存在するようになった．そして，これら画像を検索するためのウェブ画像検索システムも種々のものが生まれている．

従来のウェブ画像検索システムは，画像の掲載されているウェブサイトから，画像の近傍のテキストを取得し，それを画像と関連付けることによって検索に利用しているものがほとんどであり，画像が掲載されているサイトの持つ地理的情報はほとんど利用されてこなかった．

そこで本研究では，ウェブ画像検索結果から画像が掲載されているウェブサイトの地理的情報を抽出し，地図上に配置して表示するという手法によって，画像の地理的な位置関係や地域間の画像分布の偏りを把握するなど，視覚的に地理的情報を考慮した検索を行うことの出来る画像検索システムを提案する．

2. 関連研究と既存システム

2.1. 関連研究

関連研究には，佐藤らによる「地図や画像を用いて回答できる質問応答システム」[1]がある．

これは，自然言語で与えられた質問文に対し，画像もしくは地図を用いて回答するというシステムであり，「～はどこですか」など場所に関する質問文が与えられた場合に，地図上にマーカを配置することで回答している．

このシステムでは，質問文から求められる地理的情報を地図上に表示することは可能であるものの，質問文に関連する画像を，ウェブ画像検索システムのように地図上に表示することなどは出来ない．

2.2. 既存システム

ウェブ画像検索結果を地図上に表示する既存システムとして，goo による「地図で画像検索」[2]がある．

これは，キーワードによる検索後，地図上に表示されたアイコンにカーソルを合わせることで，アイコンの位置に対応したウェブ画像検索結果が取得できるというシステムである（図１）．

その仕組みは，ウェブクローラーによりあらかじめウェブサイトを巡回し，画像のアドレスと掲載サイトのアドレス，画像に対応するキーワード，地理的情報から取得した緯度経度を関連付けてデータベースに保存しておき，地図上に配置するというものである．

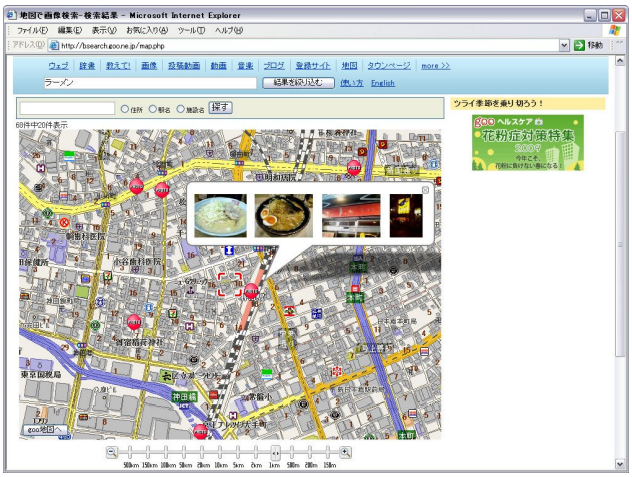

図１　「地図で画像検索」の画面

このシステムの問題点として，地図上のアイコンをクリックして画像を探す際に一覧のようなものがないために，対象となる画像が多くなると必要とする画像を探しにくいこと，検索順位に関係なく一様に画像が配置されるために，キーワードによる検索順位の高い画像，つまりユーザの要求への適応性が高い画像の判別が出来ないこと，データベースの肥大化により，デッドリンクや画像掲載サイトの更新などによるデータの不整合を防ぐためのメンテナンスが難しいことなどが挙げられる．

3. 本研究の目標と技術課題

本研究では，既存システムの問題を解決するため，検索の都度，ウェブ画像検索結果の上位から順に画像掲載サイトの文章の形態素解析を行い，地理的情報を取得して地図上に表示するという手法を取る．これによって，最新のウェブ画像検索結果の上位のみを，地理的情報を考慮して検索することが可能となる．

これを実現する際に問題となる，画像の掲載サイトの文書中に含まれている住所や施設名といった地理的情報の抽出をどのように行うかという課題は，奈良先端科学技術大学院大学によって開発されている形態素解析ツール Mecab[3]を用いることによって解決した．あらかじめ目的となる地理的情報の辞書を作成しておき，形態素解析を行うことによって，対象サイトの文章中から地理的情報を抽出する．

4. 実装

4.1. 開発環境およびシステム構成

実装したシステムにおける処理の流れを図２に示す．

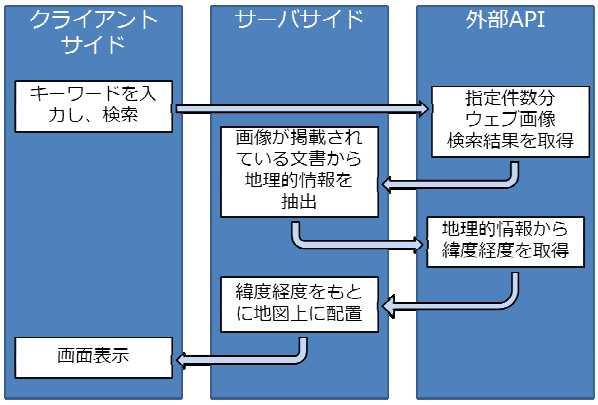

図２　処理の流れ

なお，ウェブ画像検索結果は Google AJAX Search API[4]を，地理的情報のジオコーディングおよび，検索結果の表示に用いる地図は Google Maps API[5]をそれぞれ使用した．

また，システムに使用したサーバの開発環境を表 1 に示す．

表１　開発環境

OS	Ubuntu8.04 Linux
Web サーバ	Apache 2.2.2
使用言語	PHP5.1.6
形態素解析エンジン	MeCab0.9.7

4.2. 形態素解析による地理情報抽出

地理的情報を地図上に配置するにあたって，ジオコーディングと呼ばれる地理的情報の緯度経度への変換作業が必要となる．このジオコーディングに関しては，本システムでは Google Maps API を利用した．

Google Maps API がジオコーディングを行える地理的情報には，大別すると施設名の文字列と住所の文字列の二つがある．ここでは形態素解析を利用したそれぞれの抽出方法について述べる．

4.2.1. 施設名の抽出

（１）施設名の辞書の作成

施設名は，その名称からジオコーディングにより緯度経度を取得できる建造物，ランドマーク等の名称を指す．具体的には以下の名称が挙げられる．

- 鉄道の駅の名称
- 学校や役所などの公共施設の名称
- 遊園地などの商業施設の名称
- 神社，仏閣，城郭などの建造物の名称
- 山，河川などの名称

このうち，鉄道の駅の名称と公共施設の名称については，国土交通省国土計画局で公開されている公共施設データ[6]を利用した．

Google Maps API によるジオコーディングにおいては，鉄道の駅の名称に関してはほぼ全てにおいて問題なく緯度経度が取得できるものの，公共施設の名称に関してはジオコーディングが行えず，緯度経度の取得できないものも多かった．

公共施設名の辞書に使用した，公共施設データの小分類のうち，Google Maps API によるジオコーディングの可否を表２にまとめた．

表２　Google Maps API による公共施設名のジオコーディング可否

おおむねジオコーディングが行えた	美術館，図書館，都道府県庁，役所，警察署，小中学校，高校，大学，幼稚園，病院，郵便局など
一部ジオコーディングが行えた	省庁，動植物園，裁判所など
ジオコーディング出来なかった	交番，派出所，福祉施設など

本システムでは，表２のうち，おおむねジオコーディングが行えた分類のみを，公共施設の名称として辞書化した．

また，この他のジオコーディングが可能である名称に関しては，百科事典サイト Wikipedia[7]より，「複合商業施設」，「日本の山一覧」といった項からそれぞれの

名称を抽出し，ジオコーディング可否の確認後に辞書に加えた．

（２）人名等の誤抽出の減少

施設名の抽出にあたっては，施設名の辞書と同時にMecab の標準辞書である IPA 辞書を用いて形態素解析を行った．これは，隣接する文字列の品詞 ID を考慮し，人名などを地名として誤抽出する割合を減らすためである．

具体的には，「氏」，「さん」といった人名の接尾にあたる品詞，「太郎」など人名にあたる品詞のひとつ前の品詞を人名と判断しやすくすることによって，地名として誤抽出しないようにしている．

また，「先生」，「社長」などの役職名，「株式会社」，「商事」といった組織名に関しても，そのひとつ前の品詞を地名として抽出してしまう事が多かったため，人名の末尾として新たに辞書に追加した．

4.2.2. 住所文字列の抽出

従来の住所抽出 API は，正規表現によって住所文字列を抽出しようとするものが一般的であり，誤抽出を避けるために都道府県名が省略されていたり，郡市町村名以下の地番が省略されていたりすると抽出できない場合が多い．

また，住所文字列の抽出に関しては，日本の住所文字列に様々な表記揺れが存在するため，施設名のように辞書を用いた形態素解析が行いにくいという問題もある．

都道府県名／郡市町村名／ 数字／丁目／数字／番地／数字／号

図３　日本の住所の書式

日本の住所は一般的に図３のように表されるが，数字が半角全角，および漢数字の場合があること，丁目や番地が，「－」や「の」などによって表される場合があることなどが表記揺れとして挙げられる．

また，都道府県名など住所の一部が省略されている，マンション名，階数名などジオコーディング時に障害となる不要な文字列が存在することがあるという問題もある．

これらの問題に際し，まず日本郵便のウェブサイト[8]で提供されている郵便番号データから，図３のうち郡市町村名の部分を抽出し，「住所文字列」辞書を作成した．都道府県名を辞書から除外しているのは，省略されていることが多く，また，住所文字列の抽出に際しては必要ないと判断したためである．

また同様に，半角全角および漢数字といった「数字」辞書，丁目，番地，ハイフンといった「セパレータ」辞書を作成した．

そして，これらの辞書を用いた形態素解析で，

① 住所文字列のあとに数字が 0 個以上続く
② ①のあとにセパレータを挟んで，数字が 0 個以上続く
③ ②のあとにセパレータを挟んで，数字が 0 個以上続く

という文字列を住所文字列として抽出するという手法をとった．具体的な処理の流れを図４に示す．

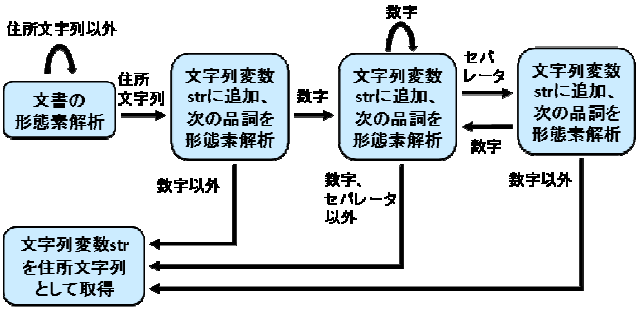

図４　住所文字列の取得処理

これにより，住所の一部が省略されていたり，表記揺れが存在している住所文字列であっても，正しく抽出することが可能となる．

また，住所を文字列として内部にデータを持つことにより，正規表現による抽出などデータを持たない場合に比べて様々な住所に対応しやすく，また誤抽出が少ないという利点がある．

なお，Google Maps API によるジオコーディングが行えない，京都府の「東入ル」といった特殊な住所については住所文字列の辞書から除外した．

4.2.3. 地理的情報の抽出の優先順位

本システムでは，ウェブ文書からの地理的情報の抽出にあたり，対象画像が指定されている img タグの近傍テキストから順に行単位で取得し，形態素解析を行う．これは，画像に関連する地理的情報は，その画像の周辺に現れやすいという考えに基づいている．

文書に対する地理的情報の抽出処理の流れを図５に示す．

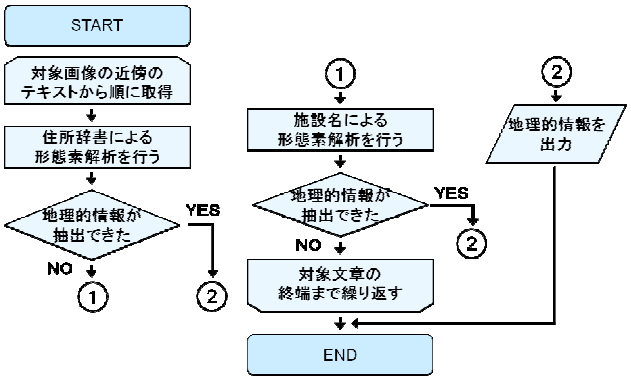

図５　地理的情報の抽出処理

図５に示すように，住所辞書による抽出を先に行い，抽出できなかった場合に施設名辞書による抽出を行うようにしている．これは，住所文字列の方がより正確な位置を指定できることが多いと考えられるためである．

4.3. 実行画面

本システムの実行画面として，「ラーメン」というキーワードによる検索を行い，地図上にウェブ画像検索結果の情報が配置された状態の画面を図６に示す．

図６　検索結果の表示画面

図６のように，ウェブ画像検索結果の上位より地理的情報が抽出できた画像の情報のみを地図上にマーカとして配置し，また地図右部にサムネイル画像を列挙した．

地図上のマーカもしくは縮小画像をクリックすることで地図上に図７のようにポップアップが表示され，画像や掲載元サイトの情報を得ることができる．

図７　画像の表示

なお，ウェブ画像検索に用いた Google AJAX Search API の仕様やシステムの応答速度を考慮し，一度に表示できるウェブ画像検索結果は８件ずつとなっている．ウェブ画像検索結果の上位８件目以降の情報を得たい場合は，画面右下の「次の８件を表示」リンクを利用する．

5. 評価

5.1. 地理的情報の抽出精度の検証

地理的情報の抽出精度の検証として，地名に関連すると考えられる「ラーメン」，「りんご」などの食べ物をキーワードとしたウェブ画像検索結果から正しい地理情報が抽出できているかを，40 種類のキーワードについて上位 16 件ずつ，計 640 件確認した．結果を表３に示す．

表３　地理情報の抽出精度

	件数	割合[%]
正しい地理情報を抽出した	457	71.4
誤った地理情報を抽出した	27	4.21
地理情報を抽出できなかった	156	24.3

表３より，７割程度の精度でウェブ画像検索結果から正しい地理的情報が抽出できていることがわかる．

また，地理的情報を抽出できなかったケースにおいては，対象サイトの文書中に地理的情報が含まれていないと考えられるため，それらを除外した抽出精度を考えると，94.4[%]の精度で正しい地理的情報を抽出できていることがわかる．

なお，ここでの誤った地理的情報とは人名など明らかに地理的情報でない情報のみを指し，抽出した地名が別の画像を指している文字列である場合など，適切であるとは言えない場合であっても，正しい地理的情

報としている．これは，画像に対する地理的情報が指している内容の適切さを形態素解析で判断するのは困難であると考えられるためである．

5.2. 抽出した地理的情報の内訳

5.1 節において正しい地理的情報を抽出した 457 件のうち，抽出した地理的情報が，施設名と住所のどちらであったかの内訳を表４に示す．

表４　抽出した地理的情報の内訳

	件数	割合[%]
施設名を抽出	370	81.0
住所を抽出	87	19.0

本システムは画像の近傍のテキストを優先的に取得しているので，画像の近傍には住所文字列よりも施設名の文字列がはるかに現れやすいという傾向があることが分かる．

5.3. システムの応答速度

（１）応答速度の分布

本システムは，外部サイトの文書の取得や形態素解析を,一度の検索につき８件ずつ行っていることから，検索から画面の表示までにある程度，応答待ちの時間を必要とする．

そこで，本システムの応答速度として，キーワードを入力，検索ボタンによってサーバにクエリを送信してから実際に地図上に情報が表示されるまでの時間を調査した．

ブラウザは Firefox3.0.5 を使用，また，応答時間の計測には Firefox のアドオンである Firebug1.3.2[9]を使用した．

5.1 節と同様に，地名に関連しそうな食べ物などのキーワードを１００件検索し，画面の表示までに掛かった時間の分布を表５および図８に示す．

表５　応答時間の分布

応答時間[s]	件数
0 ～ 2	0
2 ～ 4	0
4 ～ 6	23
6 ～ 8	21
8 ～ 10	37
10 ～ 12	9
12 ～ 14	4
14 ～ 16	3
16 ～	3

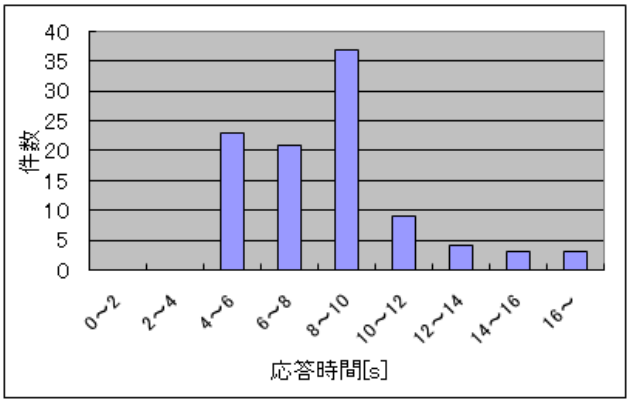

図８　応答時間の分布

表５および図８より，本システムの応答速度はおおむね４～１０秒の範囲内にあることがわかる．なお，応答速度の平均値は 8.1[s]であった．

（２）応答速度の内訳

本システムの応答時間は，大きく分けると次の５つの処理工程によって生じている．

- 対象 URL の読み込み
- 形態素解析による地理的情報の抽出
- ジオコーディング結果の問い合せ
- Google MAP の読み込み，表示
- 画像サムネイルの読み込み

そこで，システムに用いたソースコードを各工程ごとに分割し，5.3.（１）節と同条件の検索における，それぞれの工程の所要時間の内訳の平均を調べた．これを表６および図９に示す．

表６　応答時間の内訳

工程	応答時間[s]	割合[%]
対象 URL の読み込み	5.6	68.3
形態素解析による 地理的情報の抽出	0.4	4.8
ジオコーディング 結果の問い合わせ	0.8	9.76
Google MAP の 読み込み，表示	0.8	9.76
画像サムネイルの 読み込み	0.6	7.3

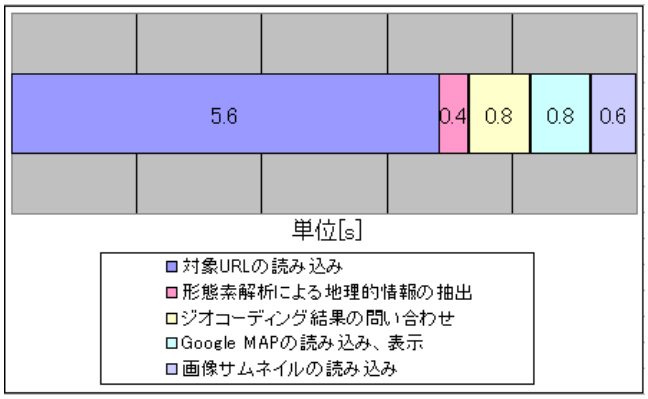

図９　応答時間の内訳

表６および図９より，本システムの表示に掛かる時間の７割近くは，画像が掲載されているサイトの文書の読み込みにあることがわかる．また，ジオコーディング結果の問い合わせの時間を含めると，本システムの応答時間の８割近くが外部サイトのレスポンス待ちにあることが分かる．

外部のウェブ文書の取得時間は，対象サイトのサーバの回線状況や文章量によって大きく左右されるため，これが図８における応答速度のばらつきの原因であると考えられる．

なお，画像サムネイルに関しては，Google AJAX Search API によって Google にキャッシュされている縮小画像を読み込んでいるため，ファイルサイズや回線状況の問題は存在せず，読み込み時間のばらつきにはほとんど影響しないと考えられる．

6. まとめ

本稿では，ウェブ画像検索および地図を組み合わせることにより，ウェブ画像検索の結果を地図上に表示するシステムを提示した．加えて，形態素解析を用いた施設名の抽出方法や，文字列処理と形態素解析による住所文字列の抽出方法についても提示した．

本稿で作成したシステムが正しい地理的情報を抽出できているかの評価を行った結果，約７割の精度でウェブ画像検索結果から適切な地理的情報の抽出が可能であることが分かった．また，ウェブ画像検索結果に地理的情報が含まれている場合に限れば，９割以上の精度で地理的情報の抽出が可能であることが分かった．

7. 今後の課題

（１）地理情報の抽出の優先順位

本システムでは，地理的情報の抽出に際し，対象画像から最も近くに現れた地理的情報を取得している．しかし，対象となるウェブサイトによっては title タグの情報を優先，つまり文書の先頭から優先的に取得したほうがいい場合や，文章の最後に地理的情報が含まれていることの多いウェブショッピングサイトなどの場合があるなど，必ずしも画像の近傍に，最も適切な地理的情報が存在しているとは限らない．

これらの問題を解決するためには，対象ウェブサイトの文書全体を取得後，形態素解析を行って抽出できた地理的情報をすべて配列に格納しておき，目的画像に対する地理的情報文字列の文字的な距離や，DOM 構造上の距離，文字列の長さ，文字列の出現回数，住￥文字列であるか施設名文字列であるかなどを考慮し，抽出できたそれぞれの地理的情報について適切な評価値を定めるという手段が考えられるが，システムへの負荷や応答速度との兼ね合いが難しいという問題を抱えている．

（２）地理情報の誤抽出の改善

本システムにおける地理的情報の誤抽出の多くは，人名や商品名といった単語を地名として抽出してしまうケースであった．

人名の場合，「さん」，「様」といった敬称が記述されていればほとんどの場合において誤抽出は防げるものの，敬称が省略されている場合，地名と人名とを形態素解析によって区別するのが難しいケースが多い．

解決のためには，人名の辞書及び地名の辞書において，適切なコスト値（単語の出現しやすさ）を定めるために，人名として現れることの多い単語の，地名としてのコスト値を増やすなど，調整を繰り返す必要がある．

また，誤抽出を発見次第，辞書ファイルに随時追加するなどの辞書のチューニング作業も不可欠である．

（３）ジオコーティング結果が複数ある場合の処理

日本には同名の地名が数多く存在しており，ジオコーディングの結果が複数になることがある．

例えば「神田駅」は東京，長崎，鹿児島の３箇所に存在するため，「神田駅」という情報だけでは地図上の位置を完全に特定することは出来ない．

本システムでは，ジオコーディング結果が複数存在する場合でもどの情報が適切であるかは特に判断せず，Google Maps API が１番目に返す情報を緯度経度として採用しており，ジオコーディングによって緯度経度を取得する際に，どの情報が正しいのかを判断していないという問題がある．

この問題に対応するためには，文書中に含まれる全ての地理的情報の緯度経度を取得し，その偏りから適切な地域を算出する方法が考えられるが，ジオコーディングによる待ち時間が長くなることや，地理的情報

の分布からの適切な算出が難しいという課題がある.

（４）システムの応答速度の向上

5.2（１）節で求めた本システムの画面表示までの時間の平均 8.1[s]は，使用中明らかに体感的に遅いと感じるものである.

これを改善するためには，あらかじめ様々なキーワードでの検索を行っておき,画像 URL と地理的情報を関連付けてサーバのデータベースにキャッシュしておく手段や，施設名の場合に地理的情報を記述してある辞書データに，地名と対応する緯度経度を付記しておき，形態素解析時に取り出すという手段，国土地理院のウェブサイトで公開されている数値情報を利用してジオコーディングをサーバ内で行う手段などが考えられる.

これらの手段の導入によりジオコーディングによる応答待ちの時間の短縮が期待できるが，反面，サーバで行う処理が増大し，負荷が生じるために複数ユーザが同時にシステムを利用した場合に応答時間への影響が考えられるという問題もある.

文　献

[1] 佐藤充, 森辰則, “画像や地図を用いて回答できる質疑応答システム”, 情報処理学会論文誌, Vol.2006, No.124, pp.113-120, November, 2006

[2] 地図で画像検索
http://bsearch.goo.ne.jp/maptop/

[3] Mecab
http://mecab.sourceforge.net/

[4] Google AJAX Search API
http://code.google.com/intl/ja/apis/ajaxsearch/

[5] Google Maps API
http://code.google.com/intl/ja/apis/maps/

[6] 国土交通省国土計画局 GIS ホームページ
http://www.mlit.go.jp/kokudokeikaku/gis/

[7] Wikipedia
http://ja.wikipedia.org/

[8] 郵便番号データダウンロード - 日本郵便
http://www.post.japanpost.jp/zipcode/download.html

[9] Firebug
http://getfirebug.com/