

花粉の飛散状況情報の提供方法に関する検討

指導教員 渡辺 恭人

学籍番号 0940199

阿部 真澄

提出日 2013年1月24日

目次

第1章 背景・目的

1-1 背景

1-2 目的

第2章 現状と問題点

2-1 現状と問題点

2-2 現状

2-3 問題点

第3章 解決方法の検討

3-1 アプローチ

3-2 既存のシステムとの比較

3-3 解決方法の検討

3-4 機能要件

3-5 求められる機能

第4章 設計

4-1 設計目標

4-2 システム構成

4-3 構成される機能

4-4 実装

第5章 評価

5-1 実装システムの動作確認

5-2 機能評価

5-3 関連システムとの比較評価

5-4 考察

第6章 まとめと評価

6-1 まとめ

6-2 今後の課題

参考文献

謝辞

第1章 背景と目的

1-1 背景

私たちの日常の中で花粉に悩まされていて外出するのが大変な人たちもいる。そして花粉は毎年の飛散量が違ってきます。花粉が飛んでいないからといって油断をしたりするとくしゃみが止まらなくなったり目が赤くなってしまったりすることがあります。

花粉の情報をスマートフォンなどで調べると花粉に対するアプリの情報などが表示されたりしている。その中でどのアプリが見やすのかなど困ってしまう人もいる。花粉はスギ花粉だけでなくハンノキ属やヒノキ科やブタクサ属やイネ科などさまざまな花粉があるのでアレルギーなどが多い人にとっては、1月から12月にかけて1年中近く花粉症に悩まされたりする人もいる。そこで、本研究では花粉の情報をいち早く知ることができる。

1-2 目的

本研究では、花粉症に悩まされている人たちに対して花粉の情報を提供することによって花粉に対する対応策をどのようにしたらいいのかわかるようになります。花粉症は突然発症するものもあればストレスを受けて免疫系が弱まりちょっとした刺激でも過剰に反応する場合があります。

また花粉症になっていないからと言って油断をしていると症状がひどくなります。症状が酷くなると、くしゃみや目のかゆみなどといった症状が起こります。そこで解消するには、花粉の情報をいち早く知ることによって花粉の飛散量などがどのくらい酷くなっているか、その場所に行くにはどのような服装や対策をして行けばいいのかなど分かりやすくすることによって花粉症の人も花粉症の症状がない人でも予防対策が出来るようになる。

そこで重要となってくるのが花粉情報の提供方法を検討し場所や時刻などを表示して花粉の情報がいつでもどこでも、その場所の花粉の情報を入手できるようになり、対策がしやすくなることを目指します。

第2章 現状と問題点

2-1 現状と問題点

まず初めに、日本人の4人に1人が花粉症患者だと言われています。花粉症は、花粉によって引き起こされるアレルギー疾患です。くしゃみ、鼻水、鼻づまり等のアレルギー性鼻炎や目のかゆみ、流涙などのアレルギー性結膜炎が最も多く見られます。まれには喘息やアトピー症状を併発するともある。花粉症は、地域差があり春先に見られるのはスギ花粉症と言われている。花粉はハンノキ属やヒノキ科やブタクサ属やイネ科の他にもあります。

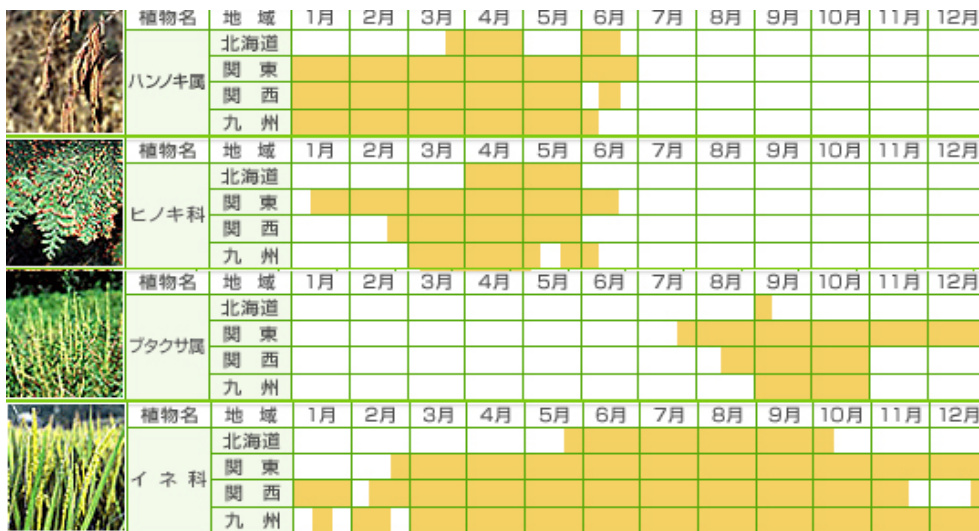


図1-2：「花粉症ナビ」出典：<http://www.kyowa-kirin.co.jp/kahun/calendar/>

花粉の情報は、気象庁や環境省などから提供されている。環境省の花粉情報サイト「図2-2」では環境省花粉観測システム「図3-2」（愛称：はなこさん）と言うものがある。



図2-2

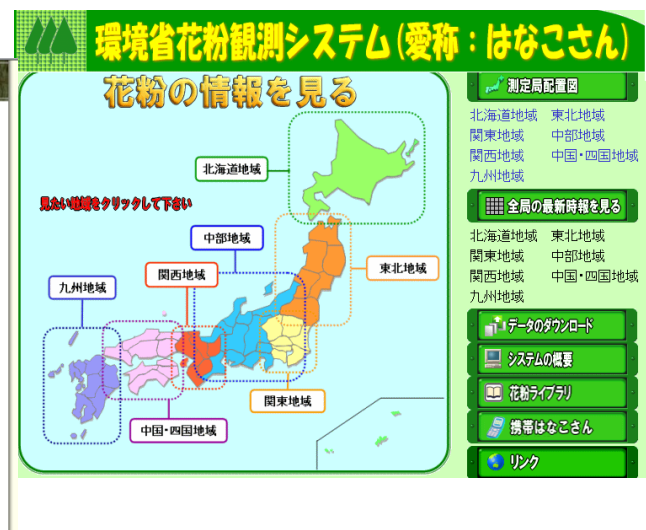


図2-3

スマートフォンのアプリでは iPhone では花粉チェッカー、Android では花粉ブラウザがありそしてドコモでは、ドコモ花粉ライブといった3つのアプリがあります。iPhone の花粉チェッカー（図4）の使い方は、起動時に位置情報を取得して自動的に地域設定を行います。花粉の状況を背景色で5段階表示をして木から飛び出す花粉の量でお知らせをします。花粉の飛散量が色で区別されるのでアプリを起動した時に赤色だったりした場合に外出をする時に花粉に対する予防対策がしやすくなります。



図 2-4 iPhone 「花粉チェッカー」 ©アスキー

Android の花粉ブラウザ（図5）使い方は、地域設定を行うと最寄の観測所で観測された花粉量、風速、風向きなどが表示されます。他の画面にすれば週間花粉量グラフなども表示できこのアプリが便利なところは、このツールにはウィジェットもついてくるのでアプリを起動しなくても花粉量をチェックできる。花粉の飛散量がグラフに表示をすることが出来るので花粉の時期は毎年の花粉飛散量が変わって来たりするので週間的にグラフをとって置くことによって花粉の時期がきたら花粉に対する対応策が立てやすくなります。



図 2-5 Android

「花粉ブラウザ」

ドコモのドコモ花粉ライブ（図6）の使い方は、地域設定を5箇所まで登録をすることができる。花粉アイコンがマイスポット用が変わるので、花粉量が一目でわかる。アラートメールを設定をすることによって自宅や会社、旅行先など登録した場所の花粉量に応じてアラートメールが届きます。花粉症が酷い人にとってはアラートメールのレベルを低く設定すれば花粉が酷くなる前に対策がしやすくなります。そして花粉に悩まされている人にとっては SNS に連帯しているので花粉に困っている友人にも花粉情報を共有できる。



図 2-6 ドコモ Android 「ドコモ花粉ライブ」

そして3つのアプリのメリットは、花粉の状況を一目でわかるようになるので花粉の飛散量が多いときに花粉に対する対応をいち早くすることができます。

iPhone の花粉チェッカーのメリットとしては、地域の設定をすることで今いる場所や気になる場所などの状況を知ることができる。また5段階の色分けによって花粉に対する予防対策がしやすい。Android の花粉ブラウザのメリットは、地域設定をした最寄の観測所で観測された花粉量、風速、風向きが表示されるので花粉の飛散量がわかりやすいしグラフなどで表示が出来る。なので、一週間でどれくらい花粉が増えたかなど知ることができる。またウィジェットもついてくるのでアプリを起動しなくても花粉の状況を見ることができます。ドコモのドコモ花粉ライブのメリットとしては、地域設定を5箇所まで登録することができ花粉のアイコンがマイスポット用が変わるので一目でわかる。アラートメールを設定することで花粉量が多い時にメールがくるので予防がしやすくなる。SNSにも連動しているので、友人にも花粉の情報を提供できる。

3つのアプリのデメリットとしては、スマートフォンのアプリは携帯端末での使用ができないのでどのアプリも使用できない。iPhone のアプリは Android で使用することができないので共通して使用できるアプリがない。ドコモのドコモ花粉ライブは SNS に連動していますが他の2つのアプリは SNS などに連動していないので友人などに共有などができません。また iPhone の花粉チェッカーのデメリットは花粉の飛散量が色分けされていますが、地図上の表示がないので周りの花粉の状況が分かりにくい。Android の花粉ブラウザのデメリットは、花粉の飛散量や風速などは分かりますが iPhone の花粉チェッカーのように色分けしドコモのドコモ花粉ライブのように地図上に表示されていないので分かりにくい。ドコモのドコモ花粉ライブのデメリットは、花粉の飛散量の設定を行うことでアラートメールを送ることが出来ますが時間帯まで設定をすることが出来ないので会社などで勤めている人にとっては、会議中などにメールが届いたりするおそれがあります。

2-2 現状

スマートフォンのアプリには花粉のアプリが iPhone では花粉チェッカー、Android では花粉ブラウザがあり、そしてドコモではドコモ花粉ライブといった3つのアプリがあります。この3つのアプリを比較したらこのようになりました。

	花粉飛散量	飛散場所	お知らせ機能	携帯での使用
花粉チェッカー	×	△	○	×
花粉ブラウザ	○	×	×	×
ドコモ花粉ライブ	△	○	○	×

図 2-7 既存アプリの比較

既存のアプリケーションでは、提供情報や表示方法は一長一短があり、どれも決定打に欠ける。またスマートフォンのアプリは iPhone と Android では使用できるアプリが異なってくるのでどちらのアプリを使用することができない。またスマートフォンのアプリは携帯端末での使用ができないので、携帯端末でツイッターなどを見ている人はドコモのドコモ花粉ライブのように SNS と連動しているアプリでの投稿は見る事が出来ますが投稿がない時には花粉の情報をすぐ知ることができません。

2-3 問題点

花粉症の人が花粉から防ぐために行っていることは、花粉のばく露を防ぐ方法と薬物による予防法がある。大量の花粉に触れることで体が花粉に対する抗体を生産する可能性が高くなります。スギ花粉に対する抗体をたくさん生産すると、何らかのきっかけでスギ花粉症を発症しやすくなり、軽症で花粉症であることに気がつかない人も花粉を鼻からたくさん吸い込んだり目に入ったりすると花粉症の症状が強くなります。よって花粉症に対する早めの準備が重要とされる。花粉症対策には、出来るだけ花粉が飛び始める 2 週間前から治療を始めることで症状を軽くし、症状がでる時期を短く最盛期に使用する薬の量を減らす初期療法というものがあります。花粉症の症状としては、くしゃみ、鼻水、鼻づまり等のアレルギー性鼻炎や目のかゆみや流涙などのアレルギー性結膜炎がもっとも多く見られます。まれに喘息やアトピー症状を併発することもあります。そこで一般的な花粉の対策としては、花粉情報に注意をして花粉が多い日の外出はなるべく避け、外出時はウールなどのコートを着ないようにし綿やポリエステルなどの化学繊維の物を着るようにしてマスクや眼鏡そして帽子などを着用する。また衣服やペッ

トなどに付着した花粉を室内に持ち込まないようにする。

そこで気象庁や環境省の提供データでは花粉に対する対策方法が掲載されていないために簡易的ではない。また花粉観測システム（はなこさん）ではデータだけが載っているのでもわかりにくい。また携帯電話やスマートフォンアプリについては、情報の確度や精度そして正確性に不安があり提供されている情報を見ても、対策が立てにくいし携帯電話ではスマートフォンのアプリを見ることは出来ないしスマートフォンのアプリでも iPhone や Android では使用できるアプリが異なってくるので共通して見られる物がない。

第3章 解決方法の検討

3-1 アプローチ

花粉に悩まされている人は、花粉の時期は出来るだけニュースを見るようにしたり、マスクをしたりして花粉に対する対策をとったりしていますが、旅行に行ったりする場合にはそういった準備をしたりすることがあるので、そういった情報をすぐに調べたい時に検索などをして対策ができるようになる。またスマートフォンのアプリなどを使用することによってニュースなどを見逃しがしたりしてしまった場合にすぐに花粉の情報をいち早く知ることができる。

3-2 既存のシステムとの比較

- ・ iPhone の花粉チェッカーは飛散量が分かりやすく色分けがされています。

花粉が多い時などにお知らせをすることが簡単にできます。

飛散場所などは地図上に表示することが出来ません。

- ・ Android の花粉ブラウザの花粉情報は最寄の観測所から提供されています。

飛散量などは分かりやすいが花粉の飛散量を見ただけではどのくらいで飛散量が多いのかがわかりにくい。

週間ごとにグラフにすることが出来るので花粉が週ごとにどのくらい増えたかなどがわかります。

- ・ ドコモのドコモ花粉ライブでは、飛散量がわかりやすくお知らせ機能の設定ができます。

花粉が多い時などに対策がしやすくなります。

これらの3つアプリでの提供情報や表示方法には一長一短がありどれも決定打に欠けます。

そこで今回のテーマとしては花粉の情報をいち早く知ることができ飛散量を色分けすることによって花粉に対する予防対策がしやすくなります。またお知らせ機能を付けることで花粉の飛散量が多い時に地図上で見なくてもしっかりと予防対策が出来ます。

	月別 情報	飛散 場所	時間別 情報	お知らせ 機能	飛散量	過去の データ
花粉チェッカー	○	△	×	○	○	×
花粉ブラウザ	△	△	×	×	○	△
花粉ライブ	○	○	×	○	○	×

図 3-8 花粉アプリの比較

3-3 解決方法の検討

解決方法としては、花粉の情報を地図上で色分けをして GPS と連動されること
によって今いる場所の情報を取得して色分けされているので花粉の状況が一目で
わかるようになる。そして地図上に洋服などのマークを表示することで花粉の飛
散量が多い時の対策方法がしやすくなります。またお知らせ機能を付けることで
家や宿泊先にいる場合に花粉の情報がすぐに分かるようになり時間設定をすれば
会社などで会議している時や朝早くにお知らせのメールなどがこなく出来ます。
そこで必要となってくるのが、環境省の花粉の飛散量情報を Web 上から自動取得
する手法を検討し最新の飛散量情報を蓄積しデータベースを構築し地域や日時を
指定して飛散量の情報取得 Web アプリケーションの設計と実装をして実装したア
プリケーションの評価をする。

3-4 機能要件

- 花粉の情報をどの場所からでも検索することができ花粉の飛散量を色分けす
ること
一般に仕事をしている人、通学している人など活動している人はさまざまな場
所に外出しており、そのいる場所ならどこでも情報を取得したいと考えます。
特に、日中飛散場所の一番多いところがわかるようにするために情報を取得す
る機能が必要となります。
- 情報更新など変化が起きた時に利用者に通知できること
利用者は常に携帯端末を見ているわけではないので、何らか花粉の状況が変化
したときに、自律的に利用者に情報提供することが期待されます。たとえば、

花粉の飛散量が多くなった場合にお知らせがあると予防対策がしやすくなります。

- 現在だけではなく過去のデータについての提供

花粉の飛散量は日々時々刻々と変化するものですが、季節的には毎年大きな変化があるとは限らない場合があります、過去のデータが参考になります。この情報を見ることが出来れば花粉の時期が来た時に過去の花粉の飛散量を見ることによって参考にすることができます。

3-5 求められる機能

- 花粉の飛散している場所の表示

指定した場所周辺の地図を表示します

- 飛散量の段階的に表示

非常に多い時は赤色にし、多い時は黄色にしたりやや多い時には緑色で表示したりまったく花粉がない場合には青色などで色分けをすることによって色を見るだけで花粉の飛散量が一目で分かるようになります

- 色分けをする場合の基準とその選択

飛散した花粉量は、1平方センチメートル当たりの花粉数です。その個数で基準を設定します。

文献【2】を参照して設定を行います。

- 非常に多い場合は50個以上
- 多い時は30から50個未満
- やや多い時は10から30個未満
- まったく花粉がない場合は10個未満

- お知らせ機能における飛散量の設定

花粉の飛散量の程度に応じて、危険度を通知する。自分の周辺状況が変化したら通知をする。

第4章 設計と実装

4-1 設計目標

設計の目標としては、解決方法で説明したように花粉の情報を地図上で色分けをして GPS と連動をさせて今いる場所の情報を取得して花粉の状況を一目でわかるようにする。そのためにすることは、洋服などのマークを地図上に表示をする。

またお知らせ機能の設定を行い花粉の飛散量が多くなった場合に花粉の飛散状況をメールなどで警告する機能をつける。お知らせ機能では時間帯の設定をすることによって自分の好きな時間帯や都合の悪い時間帯にお知らせのメールなどが届かないようにする。そこで重要になってくるのが、環境省の花粉の飛散量情報を Web 上から自動取得する手法を検討して最新の花粉飛散情報を蓄積しデータベースを構築して地域や日時の指定をすることで飛散量の情報取得 Web アプリケーションの設計をして実装を行います。

4-2 システム構成

システムの構成としては、このような地図に服やマスクそしてメガネなどのマークを表示することで花粉の飛散状況でどのような対策をすればいいのかが分かるようになります。

こうすることによって花粉の飛散量を色分けするだけでなくマークを表示することで花粉の飛散量の多い所に行く場合などにどのような対策をしていけば花粉症の症状があまり出ないようにになります。

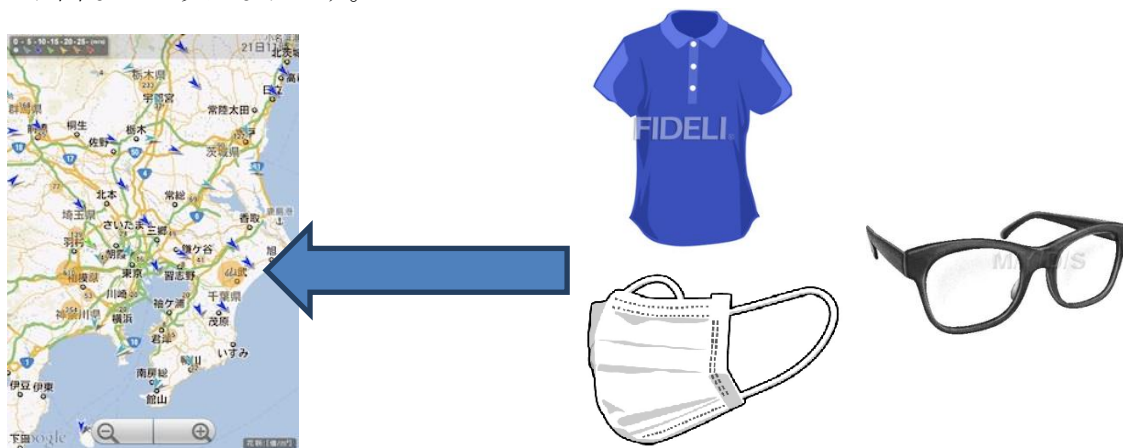


図 4-9 地図上で表示できるマーク

4-3 構成される機能

構成される機能

構成される機能としては、起動した時に地域設定を行います。

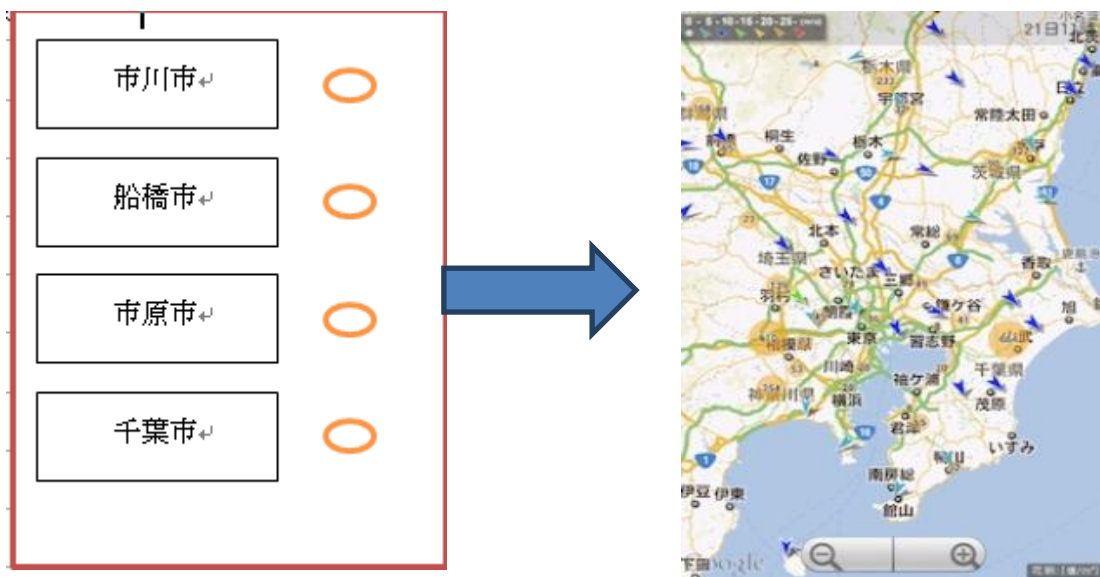


図 4-10 地域設定からのマップ表示

そして地図を表示して地図上に色分けをします。色分けは赤色だった場合には花粉の飛散状況がもっとひどい事を示します。また花粉の飛散状況がひどい場合にはお知らせ機能のメールを受信することでその日の花粉の状況がすぐに分かります。

花粉の飛散量が非常に多くなっているので外出をするさいは花粉に対する対策を十分にしてお出かけ下さい。
マスクやメガネが必要かも！！

図 4-11 お知らせメールの表示

このようなメールが来ることで花粉の対策をしないといけないといった心構えも出来るようになります。

そして花粉の状況などをツイッターに載せることで花粉症の人も花粉症の症状がまだ出ていない人でも花粉に対する対策や心構えが出来ます。



図 4-12 ツイッターでの表示

4-4 実装

本研究では、環境省花粉観測システムを利用する。本システムは毎年 3 月から 5 月頃までしかサービスされていないため、この過去データを利用して情報提供の仕組みを実現します。

(1) データファイルの準備

2012 年度データファイル (CSV 形式) をダウンロードしておき、ファイルから mysql のデータベースにデータを登録して利用します。

表 4-4-1 ダウンロードされたファイルとファイル名

- ファイル形式 : CSV	
• ファイル名	
- 北海道	hokkaido.csv
- 東北	tohoku.csv
- 関東	kanto.csv
- 中部	chubu.csv
- 関西	kansai.csv

- 中国・四国	chushi.csv
- 九州	kyushu.csv

(2) テーブルの設計

データを登録するテーブル (hanako) の設計を行います。テーブルの設計を表の 4-4-2 に示します。

表 4-4-2 テーブル hanako の設計

項目	フィールド名	型	備考
測定局コード	sokutei	int	
アメダス測定局コード	amedas	int	
年月日	ymd	timestamp	8桁年月日
時	hour	int	
測定局名	sokutei_name	varchar(50)	
測定局種別	sokutei_type	int	1:都市部、2:山間部、0:区分なし
都道府県コード	todofuken_num	int	01~47
都道府県名	todofuken_name	varchar(10)	
市区町村コード	citytown_num	int	5桁
市区町村名	citytown_name	varchar(40)	
花粉飛散数[個/m3]	hisan_num	int	
風向	wd	int	0:静穏、1:北北東、2:北東、3:東北東、4:東、5:東南東、6:南東、7:南南東、8:南、9:南南西、10:南西、11:西南西、12:西、13:西北西、14:北西、15:北北西、16:北
風速[m/s]	ws	float	
気温[°C]	temp	int	
降水量[mm]	rain	int	
レーダー降水量[mm]	radar_rain	int	
連番	id	int	ファイルからデータを読み込んでから、追加

設計したテーブルを実際に mysql 上に行います。テーブルの作成を表の 4-4-3 に示します。

表 4-4-3 テーブルの作成

```
mysql> create table hanako (sokutei int, amedas int, ymd date, hour
int,sokutei_name varchar(50), sokutei_type int, todofuken_num int,
todofuken_name varchar(10), citytown_num int, citytown_name varchar(40),
hisan_num int, wd int, ws float, temp int, rain int, radar_rain int);
Query OK, 0 rows affected (0.01 sec)
```

(3) データの登録

ファイルからデータベースに登録を行います。データベースに登録を表の4-4-4に示します。

- 北海道のデータファイル (hokkaido.csv) から読み込み
- 東北のデータファイル (tohoku.csv) から読み込み
- 関東のデータファイル (kanto.csv) から読み込み
- 中部のデータファイル (chubu.csv) から読み込み
- 関西のデータファイル (kansai.csv) から読み込み
- 中国・四国のデータファイル (chushi.csv) から読み込み
- 九州のデータファイル (kyushu.csv) から読み込み

表 4-4-4 ファイルからデータベースに登録

```
- load data local infile "/tmp/hokkaido.csv" into table hanako
fields terminated by ',';
- load data local infile "/tmp/tohoku.csv" into table tohoku
fields terminated by ',';
- load data local infile "/tmp/kanto.csv" into table kanto fields
terminated by ',';
- load data local infile "/tmp/chubu.csv" into table chubu fields
terminated by ',';
- load data local infile "/tmp/chubu.csv" into table kansai
fields terminated by ',';
```

```

- load data local infile "/tmp/chubu.csv" into table chushi
  fields terminated by ',';
- load data local infile "/tmp/chubu.csv" into table kyushu
  fields terminated by ',';

```

テーブル `hanako` の連番フィールドの設定を行います。設定の内容を表の 4-4-5 に示す。

表 4-4-5 連番フィールドの設定

- id フィールドの追加
 - `alter table hanako add id int;`
- id フィールドにインデックスを追加する
 - `alter table hanako add index ind1(id);`
- `renban` フィールドの属性を変更する
 - `alter table hanako modify id int auto_increment;`

`hanako` の連番フィールドの設定したテーブルの修正を行います。後の結果を表の 4-4-6 に示す。

表 4-4-6 修正後の `hanako` テーブル

```

mysql> show fields from hanako;
+-----+-----+-----+-----+-----+-----+
| Field          | Type      | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| sokutei        | int(11)   | YES  |     | NULL    |                |
| amedas         | int(11)   | YES  |     | NULL    |                |
|                |           |      |     |         |                |
| ymd            | date      | YES  |     | NULL    |                |
|                |           |      |     |         |                |
| hour           | int(11)   | YES  |     | NULL    |                |

```

sokutei_name	varchar(50)	YES		NULL	
sokutei_type	int(11)	YES		NULL	
todofuken_num	int(11)	YES		NULL	
todofuken_name	varchar(10)	YES		NULL	
citytown_num	int(11)	YES		NULL	
citytown_name	varchar(40)	YES		NULL	
hisan_num	int(11)	YES		NULL	
wd	int(11)	YES		NULL	
ws	float	YES		NULL	
temp	int(11)	YES		NULL	
rain	int(11)	YES		NULL	
radar_rain	int(11)	YES		NULL	
id	int(11)	NO	MUL	NULL	auto_increment
+-----+-----+-----+-----+-----+-----+					
17 rows in set (0.00 sec)					

(4) 特定位置に最も近い観測局を探す方法の検討

本研究では、利用者のいる場所の花粉飛散量を取得することを目指しています。今回利用した環境省花粉観測システムは全国 133 か所の観測局からデータを利用しています。そこで、利用者のいる場所の花粉飛散量を取得するには、最も近い観測局のデータを利用する必要があります。

したがって、利用者の位置情報と観測局の位置情報を比較して、最も近い観測局を特定した後、その観測局のデータを取得して提供します。

・観測局 133 カ所の緯度と経度を取得する

環境省花粉観測システムでは観測局の名称が登録されている。その名称から位置情報を取得します。このためインターネット上のサービスである NAPZAK の「住所から緯度経度を調べるツール」(<http://napzak.com/tool/gcode/>) を利用することで、である程度変換を行えます。地名から緯度・経度を取得出来なかったところは、手動で住所に変換して緯度・経度の取得を行います。

・作成データの一部はこのようになっています。

```
50110100,北海道立衛生研究所,43.08290909999999,141.33305840000003
50110200,北海道渡島合同庁舎,41.8198835,140.75322540000002
50120100,北海道上川合同庁舎,43.8073491,142.43932029999996
50120200,北海道十勝合同庁舎,42.9293963,143.2097235
50210100,青森市中央卸売市場,40.7858144,140.75359049999997
50220100,国立大学法人弘前大学,40.5800521,140.48219900000004
50310100,岩手県環境保健研究センター,39.6781298,141.13288510000007
50320100,大船渡地区合同庁舎,39.0654194,141.71462799999995
50410100,東北大学医学部,38.2716643,140.859056
```

観測局のテーブル設計 (sokutei) を行います。設計の内容を表の 4-4-6 に示す。

表 4-4-6 観測局のテーブル (sokutei) の設計

項目	フィールド名	型
観測局番号	sokutei	int
観測局名	sokutei_name	varchar(50)
緯度	lat	double

経度	lng	double
連番	renban	int

Mysql で sokutei テーブルの作成を行います。その結果を表の 4-4-7 に示す。

表 4-4-7 mysql でテーブルを作成

```
mysql> create table sokutei (sokutei int, sokutei_name varchar(50), lat double,
lng double);
Query OK, 0 rows affected (0.00 sec)
mysql> show fields from sokutei;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| sokutei        | int(11)       | YES  |     | NULL    |       |
| sokutei_name   | varchar(50)   | YES  |     | NULL    |       |
| lat            | double        | YES  |     | NULL    |       |
| lng            | double        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Sokutei テーブルにファイルからデータを挿入します。表の 4-4-8 に示します。

- ファイル名 : kansoku_loc.csv
- mysql での読み込みを行います。

表 4-4-8 mysql

```
- load data local infile "/tmp/kansoku_loc.csv" into table
sokutei fields terminated by ',';
```

Sokutei から緯度・経度を表示して観測局の登録データの確認を行います。表の 4-4-9 に示します。

表 4-4-9 登録データ

```
mysql> select * from sokutei where sokutei=50110100;
+-----+-----+-----+-----+
| sokutei | sokutei_name          | lat      | lng      |
+-----+-----+-----+-----+
| 50110100 | 北海道立衛生研究所    | 43.0829091 | 141.3330584 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

hanako と同じように sokutei の連番フィールドの設定を行います。表の 4-4-10 に示します。

表 4-4-10 連番フィールド (sokutei) の設定

- id フィールドの追加
 - alter table sokutei add id int;
- id フィールドにインデックスを追加する
 - alter table sokutei add index ind1(id);
- renban フィールドの属性を変更する
 - alter table sokutei modify id int auto_increment;

sokutei 連番フィールドを設定してから sokutei のテーブルの修正を行います。表の 4-4-11 に示します。

表 4-4-11 修正後の sokutei テーブル

```
mysql> alter table sokutei add id int;
Query OK, 133 rows affected (0.01 sec)
Records: 133 Duplicates: 0 Warnings: 0
mysql> alter table sokutei add index ind1(id);
Query OK, 133 rows affected (0.00 sec)
Records: 133 Duplicates: 0 Warnings: 0
```

```
mysql> alter table sokutei modify id int auto_increment;
Query OK, 133 rows affected (0.01 sec)
Records: 133  Duplicates: 0  Warnings: 0
mysql> show fields from sokutei;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| sokutei        | int(11)       | YES  |     | NULL    |                |
| sokutei_name   | varchar(50)   | YES  |     | NULL    |                |
| lat            | double        | YES  |     | NULL    |                |
| lng            | double        | YES  |     | NULL    |                |
| id             | int(11)       | NO   | MUL | NULL    | auto_increment |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

利用者に最も近い観測局を探す手順

- ・利用者の位置情報を（緯度・経度）を取得します。
- ・携帯電話の GPS 機能を利用します。
- ・観測局テーブルを検索して利用者の位置から距離が最も短い観測局を特定をします。
- ・その後、特定した観測局のデータを取得します。

ドコモの携帯から GPS 機能を使って位置情報を取得できるかどうかを fgps.php を用いて試験しました。このプログラムを表の 4-4-12 に示す。

表 4-4-12 テーブル名 : fgps.php のデータベース

```
<html>
<head>
<title>FOMA</title>
</head>
```



```
<body>
FOMA の位置取得<br>
<form method="get" action="fgps.php" lcs>
<input type="submit" name="submit" value="送信">
</form>
<body>
</html>
```

foma 受信

```
<html>
<head>
<title>FOMA</title>
</head>
<body>
FOMA の位置取得<br>
<?php
$a = $_GET["lat"];
$b = $_GET["lon"];
$c = $_SERVER["QUERY_STRING"];
print("$a<br>¥n");
print("$b<br>¥n");
print("$c<br>¥n");
?>
<body>
</html>
```

携帯で実行をした結果のこのように表示がされました。位置情報が取得できることを確認できました。

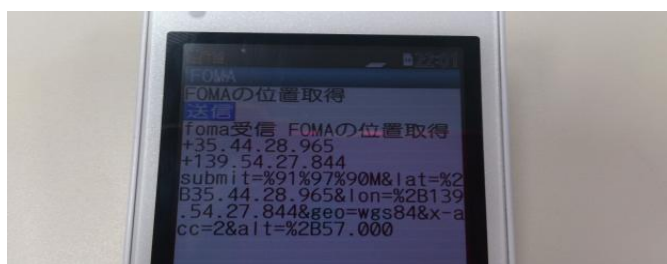


図 4-13 fgps を実行した画面

Mysql 上のテーブルに登録されている緯度・経度と特定位置の距離の計算

・参考にした URL : ueblog 「SQL で 2 点間の緯度経度から距離を測定する方法」

URL : <http://ueblog.natural-wave.com/2010/09/14/latitude-longitude-sql/>

例 1

```
- SELECT SQRT(POWER((35.742224 - lat) / 0.0111, 2) +
power((139.90770539999994 - lng) / 0.0091, 2)) AS distance FROM
sokutei ORDER BY distance
```

Mysql モニタ上で緯度・経度から実行結果を行います。表の 4-4-13 に示します。

表 4-4-13 千葉商科大学の緯度・経度を使用

```
mysql> SELECT sokutei, id , SQRT(POWER((35.742224 - lat) / 0.0111, 2) +
power((139.90770539999994 - lng) / 0.0091, 2)) AS distance FROM sokutei
ORDER BY distance;
```

```
+-----+-----+-----+
| sokutei | id | distance          |
+-----+-----+-----+
| 51200400 | 31 | 12.9826865629718 |
| 51210100 | 34 | 16.125444684314 |
(途中略)
| 54210100 | 120 | 1135.21393916978 |
```

```
+-----+-----+-----+
133 rows in set (0.01 sec)
mysql> select * from sokutei where id = 31;
```

```
+-----+-----+-----+-----+-----+
| sokutei | sokutei_name          | lat          | lng          | id |
+-----+-----+-----+-----+-----+
| 51200400 | 東洋眼科病院          | 35.8602464  | 139.975497  | 31 |
```

```
+-----+-----+-----+-----+-----+
```

1 row in set (0.00 sec)

(5) 最も近い観測局のデータから指定した日時のデータを抽出する。

- ・観測局の番号：sokutei を使用します。
- ・今日の日時と時刻を使って、今年の同時期から2ヶ月後のデータを取得します。
- ・日付例：2012-3-22
- ・時刻例：17
- ・SQL の条件式を作成して2ヶ月後のデータを取得します。表の4-4-14 で示します。

表 4-4-14 SQL の作成

```
- select hisan_num from hanako where ymd='2012-03-22' and  
hour=17 and sokutei=51200400;
```

hisn_test.php を使って緯度・経度を測定して最も近い観測局を検索します。表の4-4-15 で示します。

表 4-4-15 hisn_test.php

```
<!DOCTYPE html>  
<html lang="ja">  
<head>  
    <meta charset="UTF-8">  
    <title>観測局検索</title>  
</head>  
  
<body>  
  
<?php  
mb_language("uni");  
mb_internal_encoding("utf-8");  
mb_http_input("auto");
```

```

mb_http_output("utf-8");

$db = mysql_connect('localhost','b140000','hoge hoge');
mysql_query("SET NAMES utf8",$db);
mysql_select_db('b140000');

$lat = 35.7422248;
$lng = 139.907705399999994;

$sql = "SELECT sokutei, sokutei_name, id, lat, lng, SQRT(POWER(($lat - lat) /
0.0111, 2) + power(($lng - lng) / 0.0091, 2)) AS distance FROM sokutei ORDER
BY distance;";

$result = mysql_query($sql);
$rows = mysql_num_rows($result);
if($rows == 0){
    echo "<p>該当データ無し</p>";
    echo "<a href=menu.php>メニューに戻る</a>";
}
else {
    $row = mysql_fetch_array($result);
    echo "<p>";
    echo "ID: " . $row["id"];
    echo "<br />";
    echo "緯度: " . $row["lng"];
    echo "<br />";
    echo "経度: " . $row["lat"];
}

```

```

        echo "<br />";
        echo "観測局名: " . $row["sokutei_name"];
        echo "<br />";
        echo "観測局番号: " . $row["sokutei"];
        echo "</p>";

        $lng = $row["lng"];
        $lat = $row["lat"];
        $sokutei = $row["sokutei"];
        //$size = 220;
    }

//現在から 2 ヶ月後の昨年の日時を使用
$year=date("Y")-1;
$month=sprintf("%02d",date("n")+2); //2 桁表示
$day=date("d"); //2 桁表示
$hour=date("G");//24 時間表示

$sql = "select hisan_num from hanako where ymd='".$year . "-" . $month .
"." . $day . "'" and hour=" . $hour . " and sokutei=" . $sokutei . "'";

echo $sql . "<br>";

$result = mysql_query($sql);
$rows = mysql_num_rows($result);
if($rows == 0){
    echo "<p>該当データ無し</p>";
    echo "<a href=menu.php>メニューに戻る</a>";
}

```

```

}
else {
    $row = mysql_fetch_array($result);
    echo "<p>";
    echo $year . "年" . $month . "月" . $day . "日" . $hour . "時の飛散量: " .
$row["hisn_num"];
    echo "</p>";
}

echo "<a href='menu.php'>メニューに戻る</a>";

?>

</body>
</html>

```

千葉商科大学の緯度・経度を使って最も近い観測局を（hisn_test.php）を使って調べました。表の 4-4-16 に示します。

表 4-4-16 千葉商科大学から（hisn_test.php）を実行した結果

```

ID: 31
緯度: 139.975497
経度: 35.8602464
観測局名: 東洋眼科医院
観測局番号: 51200400
select hisn_num from hanako where ymd='2012-03-22' and hour=17 and
sokutei=51200400;
2012年03月22日17時の飛散量: 65
メニューに戻る

```

携帯電話の GPS から位置情報を取得して、位置情報から最短距離の観測局を特定して日付、時刻を指定して飛散量を取得し表示をします。

携帯電話との連携

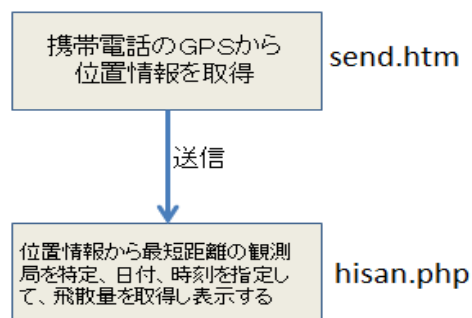


図 14-4 携帯電話との連携の仕組み

hisan.php を使って緯度・経度から花粉の飛散量の表示を行います。

位置情報を送信するための hisan.php の作成をします。表の 4-4-17 に示します。

表 4-4-17 hisan.php

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>観測局検索</title>
</head>
<body>
<?php
mb_language("uni");
```

```

mb_internal_encoding("utf-8");
mb_http_input("auto");
mb_http_output("utf-8");

/**
 * dms (度分秒単位) 表記を degree (度単位) 表記に変換
 *
 */

function convertDmsToDegree($dms)
{
    // 「.」ごとに分割
    list($a, $b, $c, $d) = explode('.', $dms);
    // 度分秒それぞれを 10 進法に直して加算
    $degree = $a + ($b / 60) + ($c / 60 / 60) + ($d / 60 / 60 / 100);
    return $degree;
}

$lat = convertDmsToDegree($_GET["lat"]);
{lng = convertDmsToDegree($_GET["lon"]);

$db = mysql_connect('localhost','b140000','hoge hoge');
mysql_query("SET NAMES utf8",$db);
mysql_select_db('b140000');

$sql= "SELECT sokutei, sokutei_name, id, lat, lng, SQRT(POWER(($lat - lat) /
0.0111, 2) + power(($lng - lng) / 0.0091, 2)) AS distance FROM sokutei ORDER
BY distance;";

```



```

$result = mysql_query($sql);
$rows = mysql_num_rows($result);
if($rows == 0){
    echo "<p>該当データ無し</p>";
    echo "<a href=menu.php>メニューに戻る</a>";
}
else {
    $row = mysql_fetch_array($result);
    echo "<p>";
    echo "ID: " . $row["id"];
    echo "<br />";
    echo "緯度: " . $row["lng"];
    echo "<br />";
    echo "経度: " . $row["lat"];
    echo "<br />";
    echo "観測局名: " . $row["sokutei_name"];
    echo "<br />";
    echo "観測局番号: " . $row["sokutei"];
    echo "</p>";

    $lng = $row["lng"];
    $lat = $row["lat"];
    $sokutei = $row["sokutei"];
    //$size = 220;
}

```

//現在から 2 ヶ月後の昨年の日時を使用

```

$year=date("Y")-1;
$month=sprintf("%02d",date("n")+2); //2 桁表示
$day=date("d"); //2 桁表示
$hour=date("G");//24 時間表示

$sql = "select hisan_num from hanako where ymd='" . $year . "-" . $month .
"-" . $day . "' and hour=" . $hour . " and sokutei=" . $sokutei . "'";

//echo $sql . "<br>";

$result = mysql_query($sql);
$rows = mysql_num_rows($result);
if($rows == 0){
    echo "<p>該当データ無し</p>";
    echo "<a href=menu.php>メニューに戻る</a>";
}
else {
    $row = mysql_fetch_array($result);
    echo "<p>";
    echo $year . "年" . $month . "月" . $day . "日" . $hour . "時の飛散量: " .
$row["hisn_num"];
    echo "</p>";
}

echo "<a href='menu.php'>メニューに戻る</a>";

?>

```

```
</body>
</html>
```

携帯で位置情報を送信し緯度・経度から最も近い観測局を測定するための send.htm を作成します。表の 4-4-18 に示します。

表 4-4-18 send.htm

```
<html>
<head>
<title>携帯位置情報送信</title>
</head>
<body>
携帯電話での位置送信<br>
<form method="get" action="hisan.php" lcs>
<input type="submit" name="submit" value="送信">
</form>
</body>
</html>
```

第5章 評価

5-1 実装システムの動作確認

携帯から send.htm を起動して位置情報の送信を行います。

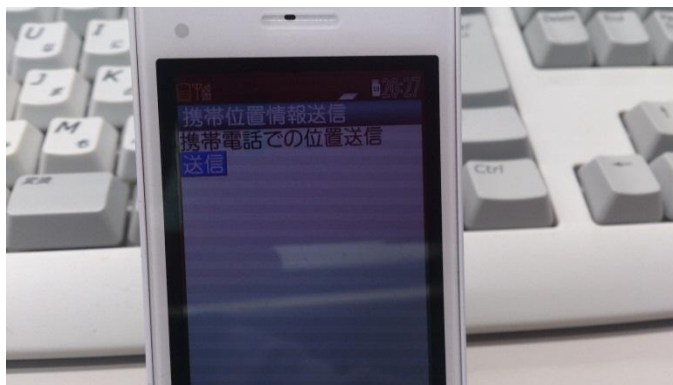


図 5-15 send.htm を実行した画面

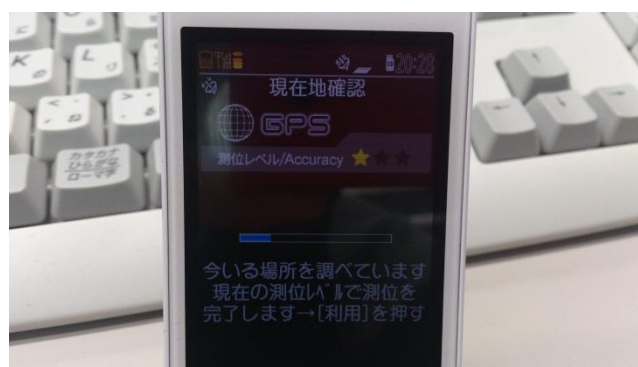


図 5-16 位置情報を送信している画面

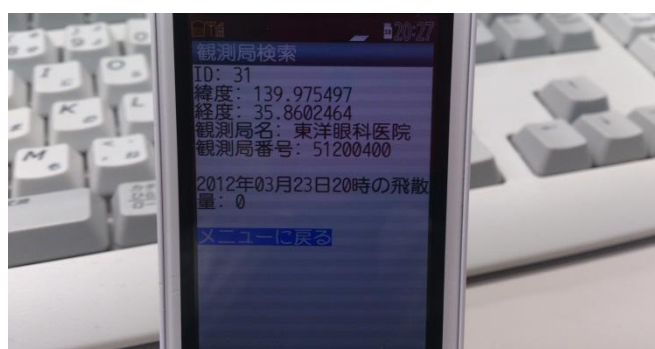


図 5-17 位置情報を取得した画面

大学で GPS を送信した結果一番近い所は、東洋眼科医院と表示がされました。

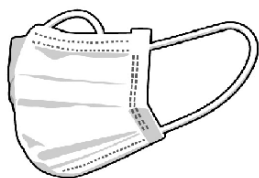
5-2 機能評価

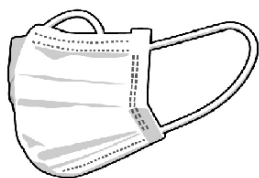

本研究の機能の評価は、花粉の飛散量を GPS の機能を使用することによって位置情報を送信して最も近い観測局から情報を得ることでその場所の飛散量を知ることが出来ました。しかしお知らせ機能の設定とマークを表示することが出来ませんでした。

5-3 関連システムとの比較評価

	飛散状況	位置情報	お知らせ機能	マーク
花粉チェッカー	○	△	△	×
花粉ブラウザ	△	△	×	×
花粉ライブ	○	○	○	×
本研究の内容	○	○	×	×

既存するシステムと本研究のシステムを比較してみた結果は花粉の飛散状況はどれも一様は分かるようになっていて位置情報がしっかりとしているのはドコモの花粉ライブが一番見やすくなっていてお知らせ機能もついているので花粉の飛散状況が酷くなった場合にはすぐに警告をしてくれるので使いやすくなっていました。本研究で考えたものでは、お知らせ機能の設定をすることで警告のメールが届くようにしてそのメールには花粉に対する対応策などを載せることによって出かける時などにどのような対策をすれば花粉症の症状が少しでも抑えることができるようになります。



そして地図上に花粉対策のマークを上図  のようなもの  を表示することによって花粉の飛散量が多い場所に行く場合にマークなどが表示

されていればお知らせ機能のメールを見てから地図上の飛散状況を見ることによってどのような対策をすれば効果的になるのかなどがわかりやすくなります。

5-4 考察

本研究では、GPS機能を使うことによって位置情報を取得して最も近い観測局から花粉の情報を取得することができ特定した観測局から日時の花粉飛散量を取得することが出来ました。しかし花粉の飛散量を地図上に表示をしてマークなども表示することが出来なかったため地図上に飛散量のマークを表示することによって花粉症の人たちが花粉に対する対策がしやすくなります。また花粉症の症状はいつ発症するかわからないので花粉症になっていない人でもこれを使っている人のツイッターなどを見たりすることで花粉の飛散量がどれだけ多いのかなどがわかったりするので花粉に対する心構えもできるようになります。

そしてお知らせ機能を付けることによって花粉の飛散量を地図上のマークなどを見なくても今の花粉の状況が分かるようになるので旅先や会社の出張などで他の県などに行った場合でもお知らせ機能を設定しておけば花粉症になりやすい人でも設定を低くしていれば花粉の対策をすぐに行うことが出来るようになるので花粉症の症状がもし出ても軽い症状くらいで抑えられるようになります。しかし本研究では花粉の飛散量は分かるようになりますが、花粉にはさまざまな種類があるので一般的にはスギ花粉で花粉症になる人がいます。それにスギ花粉だけでなくヒノキやブタクサなどといった花粉の種類もあるのでスギ花粉症ではないからと言って油断をしたりしていると花粉症の症状が出てきたりしてしまいます。そこで問題となってくるのが、花粉の飛散量だけでなく花粉の種類も分別が出来るようにする。花粉にはこのような種類がありました。

春に多く飛散する花粉は、スギ、ヒノキ、ネズ、シラカン、ハンノキ、オオバヤシャブシ、コナラ、リンゴなどがあります。



花粉の代表格であるスギ花粉です(出典:アレルギーコンパクトブック)

- ・スギ(2~4月) 北海道を除く全国
- ・ヒノキ(3~5月) 関東以南
- ・ネズ(4~5月) 北海道を除く全国
- ・シラカンバ(4~5月) 関東以北
- ・ハンノキ(1~6月) 全国
- ・オオバヤシャブシ(3~4月) 関東
- ・コナラ(4~5月) 全国
- ・リンゴ(4~5月) 主に東北

図 5-18 春花粉の種類

夏に多く飛散する花粉は、カモガヤ、オオアサガエリ、ハルガヤ、ホソムギ、スズメノカタビラ、スズメノテッポウなどがあります。



カモガヤ花粉です。(出典:アレルギーコンパクトブック)

- ・カモガヤ(4~7月) 全国
- ・オオアサガエリ(4~7月) 全国
- ・ハルガヤ(4~7月) 全国
- ・ホソムギ(4~7月) 全国
- ・スズメノカタビラ(3~5月) 全国
- ・スズメノテッポウ(3~5月) 全国

図 5-19 夏花粉の種類

秋に多く飛散する花粉は、ヨモギ、ブタクサ、オオブタクサ、カナムグラ、ヒメスイバ、ギシギシ、カラムシなどがあります。



ヨモギ花粉です。出典:アレルギーコンバクトブック

- ・ヨモギ(8~9月) 全国
- ・ブタクサ(8~10月) 全国
- ・オオブタクサ(8~10月) 北海道を除く全国
- ・カナムグラ(8~10月) 全国
- ・ヒメスイバ(5~6月) 全国
- ・ギシギシ(5~8月) 全国
- ・カラムシ(9~10月) 北海道を除く全国

図 5-20 秋花粉の種類

花粉は春先から秋にかけてさまざまな種類の花粉があるので花粉症の症状が出る人はこの中の花粉で症状が出てきたりするので1年近くも花粉症の症状に悩まされたりする人もいます。本研究では花粉の飛散量が分かることによってどこの地域の花粉がもっとも多いかなどが分かりますが、どの花粉が一番飛散しているのかまでは分からないで地図上の飛散状況を見てもスギ花粉で症状が出る人もいれ

ばスギ花粉ではない花粉で症状が出る人もいるので花粉の飛散量だけでなく花粉の種類まで表示することが出来れば花粉症の検査などを行っていただければ自分がどの花粉で花粉症の症状が出るのかなどが分かったりするので、そこで花粉の種類を表示することによってどの花粉が今一番ひどいのかなどが分かるようになるので、花粉の症状が出ても軽い症状で抑えられるようになると思います。

第6章 まとめと今後の課題

6-1 まとめ

本研究ではGPS機能を使うことによって位置情報を取得して花粉の飛散状況を最も近い観測局から花粉の飛散量を知ることが出来ます。今回実現は出来なかったのは飛散量を地図上で表示をすることで飛散状況が一目でわかるようにすることと地図上で花粉に対する対策マークを表示して花粉の飛散量が多い所に行く場合に対策がしやすくしたりすることが出来ます。もし花粉症の症状が出たとしても軽い症状で抑えることができます。そしてお知らせ機能を付けることで花粉の飛散状況を見なくても飛散状況が悪くなった場合にすぐにメールなどが届くことによって旅先や会社の出張などで出かけていても花粉の飛散状況が分かるのでマスクなどをつけて外出した方がいかなどがすぐに分かるようになります。

6-2 今後の課題

花粉の飛散状況はGPS機能を使うことで最も近い観測局から飛散量を知ることまでは出来たのでそこから地図上に花粉の飛散状況を表示して花粉対策のマークなどの表示をすることで花粉に対策がしやすくなります。花粉にはさまざまな種類があることがわかったので、花粉の飛散量だけでなく花粉の種類などを表示することによってスギ花粉で花粉症の症状が出ている人やスギ花粉以外で花粉の症状が出ている人もどの花粉が今一番飛散している花粉がどれなのかなどを詳しく表示することで、どの花粉で花粉症の症状が出ているのかなどが分かってくるのでどの花粉の時期にどのような対策をすればいいのかがわかるようになります。そのためには、花粉の種類をどのように表示をして見ただけでどの花粉の飛散状況が一番酷くてどの花粉が少ないのかなどを分かるようにすることで、花粉の症状が出る人はさまざまな花粉で花粉症の症状が出るのでどの花粉がどのように飛散しているのかなど表示していきます。

参考文献

【1】花粉症の原因・花粉カレンダー

<http://allabout.co.jp/gm/gc/302425/> 2013年1月7日

【2】環境省花粉情報サイト

<http://www.env.go.jp/chemi/anzen/kafun/> 2013年1月7日

(ライブラリ内の資料「花粉の飛散量の分別基準」)

【3】花粉チェッカー

<http://weekly.ascii.jp/elem/000/000/077/77162/> 2013年1月8日

【4】花粉ブラウザ

<https://play.google.com/store/apps/details?id=org.cuspy.android.kafun> 2013年
1月8日

【5】ドコモ花粉ライブ

<http://docomo-kafun-live.softonic.jp/android> 2013年1月8日

【6】花粉カレンダー

<http://www.kyowa-kirin.co.jp/kahun/calendar/> 2013年1月8日

【7】花粉の治療法

<http://allabout.co.jp/gm/gc/302432/> 2013年1月10日

【8】花粉の種類

<http://www1.ocn.ne.jp/~t-jibika/oldkafun/sub2.htm> 2013年1月10日

【9】NAPZAK 住所を入力して緯度・経度の変換を行います。

<http://napzak.com/tool/gcode/> 2013年1月22日

【11】 鼻の病気がわかる本 2007年8月31日 第2章

【12】 花粉症の最新克服法 2010年12月 第1章

謝辞

本研究を行うにあたり、ご指導いただいた渡辺恭人教授に大変感謝しております。論文の構成をはじめ、プログラミングや執筆に至るまで一からご指導して頂きました。本来の設計に近いものが出来ました。本研究のテーマ決定時からかなりの不安もありましたが、最後まで検討出来たのは渡辺教授のお陰です。本当にありがとうございました。

渡辺ゼミでは、プログラミングを基礎から教えていただき、思った通りに実行できた時のプログラミングの楽しさと、1字でも間違えたりして正しく動作しないプログラミングの難しさや悔しさなどを学ぶことが出来ました。

最後に、私の卒業研究に関わって下さった方全員にもう一度感謝を述べさせて頂き謝辞とさせていただきます。ありがとうございました。