

# GMS-5 衛星画像と気象記録の 統合データベース構築に関する研究

谷 誠、菊地 時夫

高知大学大学院理学研究科情報科学専攻  
780-8520 高知県高知市曙町 2-5-1  
{makotani,tkikuchi}@is.kochi-u.ac.jp

## 概要

高知大学菊地研究室では Web ページ気象情報頁において GMS-5 衛星画像を取り扱った情報を発信しており、そのデータ量はおよそ 200GB 以上に及んでいる。本論文では、GMS-5 衛星画像情報発信の手法として、雲画像と気象記録を利用した統合データベース構築を試みた。情報源には、気象情報頁に蓄積されている日本付近の雲画像と、日本気象協会発行の「気象」に毎月掲載の天気図日記を気象記録として利用した。統合データベースでの応用実験として、カルマン渦列等の天気事象が起こった日の気象記録から、特定の単語を選び検索をかけることで、同様の天気事象をもつ衛星画像が得られることを検証した。また、気象衛星画像検索データベースの拡張として、気象衛星画像間の類似性からの検索を試みた。ここでは、気象衛星画像の形状解析として雲域の輪郭線を追跡し、連続した座標の集合として抽出し、幾何データ型として気象衛星画像の撮影日時と共にデータベースに格納した。指定された日時の気象衛星画像の幾何データをデータベースから取得後、データベースに対し幾何演算を行うことで類似画像が得られることを目標に今後のアルゴリズム改良が必要である。

## 1.はじめに

気象情報頁からの衛星画像利用方法は、一時間毎に情報更新する「最新の雲画像」を取得するか、「保存書庫」として設けられている過去の衛星画像を取得するかである。ユーザーは衛星画像取得の際に、いつ、どこで、どの天気事象が起こっていたかを知った上で、任意の衛星画像を取得していると思われる。そこで、過去に起こった詳しい天気事象を知らなくても任意の衛星画像が取得できるようにすることを考えた。このため過去の気象記録を利用しデータベース化することで、気象用語や天候状態から該当する衛星画像を検索できる「雲画像と気象記録を利用した統合データベース」構築を試みた。データベースに登録する気象記録としては、日本気象協会発行の「気象」<sup>[1]</sup>に毎月掲載されていた天気図日記を用いた。システム構築には、データベース部に PostgreSQL<sup>[2]</sup>を、データベースと Web の連携を図るために PHP<sup>[3]</sup>と Apache を用

いた。

気象記録からの検索はデータベース化する情報量が少ないため(紙面により限られているため)、十分な結果を求めづらい場合がある。そこで気象衛星画像を直接データベース化し、利用する方法を検討した。しかし、画像データは非常に大きいのと、検索キーワードに相当するものがないことが問題である。一方、PostgreSQL で扱うデータ型には幾何データ型があるのでこの利用を考えた。幾何データ型を用いることで文字や数値だけのデータベースではなく、2次元空間オブジェクトをデータベースで扱うことができる。そこで、気象衛星画像から雲域を取り出し、幾何データ型としてデータベース化することを試みた。このデータベースを気象衛星画像検索データベースの拡張として画像間の類似性からの検索を行ったので、その結果について報告する。

## 2.GMS 衛星画像と気象記録によるデータベース構築

検索の情報源となる気象記録の格納には PostgreSQL を利用しデータベース化している。データベース化された情報に対し SQL による検索要求を Web 上から PostgreSQL へ発行する役目を Apache と PHP で担う。検索要求から結果取得までのおおまかな流れは、①クライアントから実行要求を受けた Apache は PHP スクリプトを読み込み、データベース検索 SQL を生成する、②PHP から PostgreSQL に接続を行い SQL の実行依頼を PostgreSQL に出す、③実行依頼を受けた PostgreSQL は SQL を実行しデータベースから結果を取得する、④実行要求に対する結果は PHP、Apache、クライアントへと渡される、ようになる。

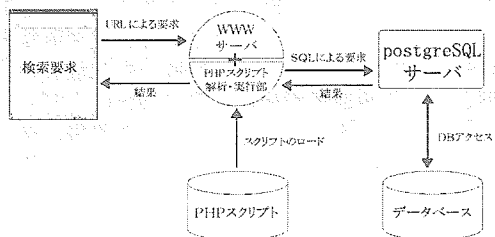


図1 統合データベース概念図

### 2-1.テーブル作成

検索方式は天気図日記文による全文検索ではなく、天気図日記よりキーワードを抽出し、キーワード検索とした。検索キーワードとなる要素には、降雨量などの数値で表されるものがあり、全文検索では数値を扱った比較演算子を使用できないからである。また、気象用語だけのテーブルを使用することで全文検索より検索速度が向上する。

構築する統合データベースにおいて、検索キーワードとなる語句は「気象」掲載の天気図日記中の単語や数値が基本となるため、主な検索キーワードを3種類(日付、降雨量などの天候状態、文章中の単語)とするテーブルの他、検索結果出力時における衛星画像 URL 表示用テーブルを以下のように作成した。

- ①日付や天気図日記文で構成するテーブル ew1
- ②降雨量などの天候状態検索用テーブル enum1

③天気図日記文中の単語検索用テーブル enum11

④衛星画像 URL 表示用テーブル picture  
作成したテーブルを図2と表1-1～表1-4に示す。

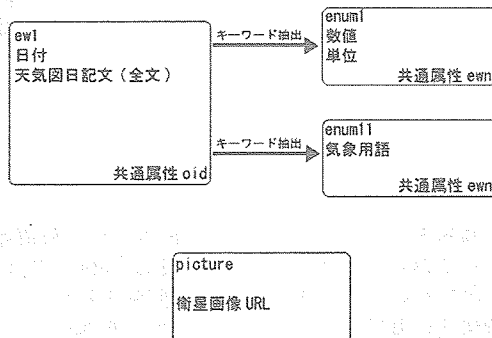


図2 テーブル関係図

OID	date	year	month	day	status
5291	2000/1/3	2000	1	3	天気図日記文
5311	2000/1/4	2000	1	4	天気図日記文

表1-1 ew1(日付、天気図日記文)

ewn	suuti	tanni
565291	18	°C
565311	9	°C
565311	3	°C

表1-2 enum1(天候状態検索用テーブル)

ewn	eword
565291	横浜
565291	前線

表1-3 enum11(単語検索用テーブル)

Day	hour	url
2000/1/3	1	http://...
2000/1/3	2	http://...
2000/1/3	3	http://...

表1-4 picture

### 2-2.データ入力

①全文データ入力(対象テーブル: ew1)

衛星画像が存在する期間を対象とし、1996年1月～2001年12月までの6年間を入力した。

## ②キーワード抽出入力(対象テーブル：enum11)

単語検索用テーブルに単語入力するには天気図日記文中の個々の形態素を解析、認識し検索単語になりうる語句を選択する必要がある。その形態素を解析する手段として形態素解析システム「茶筌」<sup>[5]</sup>を利用した。「茶筌」が利用するデフォルトの辞書では気象関係の語彙に分かち書きしがたい場合が生じるので「茶筌」に付属のユーティリティコマンドを用いて辞書ファイルに気象関係の語彙を追加した。日記の分かち書きはPHPスクリプトの中でモジュールを呼び出し、入力フォームから入力された文章に分かち書きを行いその結果をテーブルに格納した。そのテーブルを単語検索用テーブルとして扱うには不要な単語が多いため、気象に関係の無い単語、使用頻度の低い単語を除き必要な単語だけを単語検索用テーブルへ入力するための照合テーブルを作成した。

## ③数値データ抽出入力(対象テーブル：enum1)

天気図日記文に 100mm という数値データが含まれておりそれを分かち書きすると、[1][0][0][mm]と分けられる。そのため、数値データ抽出にはPHPスクリプトで数値を繋ぎ合わせて[100]の数値部分、[mm]の単位部分に分け抽出し、テーブル enum1 に入力した。

入力の結果、表 2 のデータ入力件数となった。テーブルの入力件数はそれぞれ 6 年間分の日数、mm・m/s などの天候状態の合計、入力された単語数の累計、衛星画像 URL の総数となっている。

テーブル	入力件数
ew1(日付、天気図日記表示)	2102
enum1(天候状態検索テーブル)	4280
enum11(単語検索テーブル)	36872
picture(衛星画像 URL 表示)	41490

照合テーブル	削除前	削除後
word 数	4746	1458

表 2 6 年間分の入力件数

## 2-3.検索実行の流れ

検索実行の流れは図 3 のようになる。入力フォームに検索キーワードを入力し検索を開始す

る。キーワードにあったテーブルで検索を行うために PHP スクリプトでテーブル毎の SQL を生成する。複数のテーブル間で検索を実行する場合には各検索 SQL を内部結合(INNER JOIN)し実行する。結果、テーブル毎の条件を満たすタプルを ew1 から取得する。内部結合(INNER JOIN)とは、テーブル結合時に各テーブル間に結合条件を与え、満たしていればそのテーブルのタプルを取り出すことができる。使用する 3 つのテーブル間での共通属性はテーブル ew1 の oid、enum1 の ewn、enum11 の ewn である。よって内部結合の結果、取得するタプルは各テーブルでの検索結果にあわせ、かつ ew1.oid=enum1.oid=enum11.oid の条件を満たすものとなる。その内部結合 SQL 文実行によって取得した ew1 のタプルから日付(date)と天気図日記を取り出し、ew1.date=picture.day となる picture の衛星画像 URL を取得し Web 上に表示する。

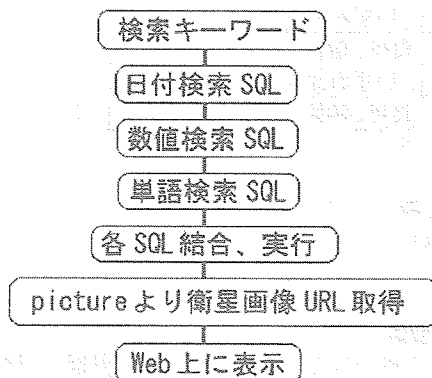


図 3 検索実行の流れ

## 2-4.インデックス検索による検索速度向上

データベース性能を向上させるための一般的な方法としてインデックスがある。インデックスの有無で特定行の検索速度が変わるため、表 1 のテーブルに対し、データベース性能向上の為に B-tree インデックスを作成し検索に使用した。

インデックスは各テーブルにおいて検索キーワードとなる列、検索対象となる列に作成した。検索速度性能評価の結果(表 3-1)、全ての検索に

において検索速度の向上効果が表れた。

表 3-1 の検索時間は picture を除いたテーブルでの評価である。picture テーブルは検索結果表示時に衛星画像 URL 取得のために検索され検索のアクションは日数分繰り返される。検索結果から 10 日間のデータが得られたら、衛星画像 URL 取得に 10 回繰り返し処理を行う。表 3-2 に 1 ヶ月間(31 日間)の URL を取得する作業、つまり 31 回 URL 取得を繰り返した場合の 1 回(1 日分)あたりの平均検索時間を示す。

picture テーブルは 1 回の検索結果で得られる最大のタプル数は 0 時~23 時までの 24 件である。picture テーブルのタプル数は 41490 存在し、他のテーブルよりタプル数が多いので picture テーブルでの検索は、インデックス作成の効果が高くなっている。

キーワード数	組み合わせ	検索時間 (index 無)	検索時間 (index 有)
1	日付、数値、単語 いずれか 1 つ	28.85	11.45
2	日付、数値、単語 いずれか 2 つ	192.74	14.72
3	日付、数値、単語	339.92	31.16

表 3-1 キーワード数別平均検索時間(msec)

index 無	index 有
93.83	0.27

表 3-2 picture における平均検索時間(msec)

### 3. 応用実験

データベースでの検索方法は、天気図日記から分かち書きで取得された単語、天候状態、日付で行う。そこである特定の天気事象が起こった時の天気図日記から特定の単語を選び、その単語で検索すれば同じ天気事象を持つ衛星画像が得られることが期待されるので、これを検証する。

#### 3-1. 検証 1. カルマン渦列

検証する天気事象として図 4 に見られるカルマン渦列をとりあげた。

カルマン渦列とは一様な流れの中に円柱を流れに直角に置くと円柱の下流側に互いに反対方向に回転する渦が交互に周期的に発生し下流へ

と流され、その結果、互い違いに規則的に並んだ渦の列が下流側に見られるようになる流体現象で、毎年のように冬の寒気の吹き出しがある頃、済州島の風下に見られることで知られている。

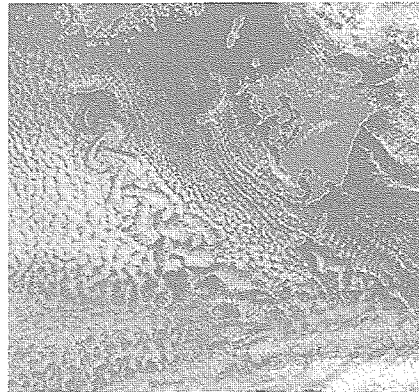


図 4 2000 年 2 月 21 日のカルマン渦列

図 4 は 2000 年 2 月 21 日の衛星画像であり、当日の天気図日記は表 4-1 の通りである。この文章中から検索単語として「冬型」と「低気圧」を選び検索を実行した結果、47 件のデータが取得できた。47 件中 14 件には衛星画像が存在せず、実質 33 件中 9 件にカルマン渦列と見られる天気事象が確認でき、さらに 9 件中 4 件に共通して天気図日記に記されている事柄が「冬型の気圧配置で低気圧が東北、又は北海道を東へ抜ける(遠ざかる)」であると判明した。

冬型強まる。低気圧は東北の東へ抜ける。冬型気圧配置で、上空に新たな寒気。輪島 21 時上空約 5,200m は -36.3℃。日本海側は所々のアメダス積雪で日降雪量 30cm を超える。静岡タンポポが開花

表 4-1 2000 年 2 月 21 日の天気図日記

検索単語	低気圧	冬型	低気圧+冬型
HIT 数	537	189	47

表 4-2 検索単語別 HIT 数

カルマン渦列が見られる日		
1997/1/25	1997/11/18/	1998/1/5
1998/2/6	1998/12/8	1999/12/26
2001/2/2	2000/2/21	2001/2/14

表 4-3 47 件中 9 件にカルマン渦列が見られる日

#### 4. 衛星画像から雲域抽出によるデータベース拡張

次に本データベースの拡張として、画像の特徴を解析しデータベース化することで類似画像を検索できるシステム構築を試みる。そのためにGMS衛星画像特徴抽出の要素として雲域を抽出する。そして抽出した雲域は多角形の幾何データ型としてデータベースに挿入する。類似画像検索時には幾何データ型に対し幾何演算を行った。取り扱う衛星画像は<http://weather.is.kochi-u.ac.jp/sat/gms.farecast.IR1/>以下に保存されている1996年9月4日～2001年12月31日までの画像である。また、画像ファイル形式はPGM(Portable Gray Map)である。

##### 4-1. 画像2値化

衛星画像に分布する雲域を抽出するにあたり、元画像のサイズは640×480であるが160×120の4分の1に縮小して処理を行うことにした。画像サイズ縮小後、画像を雲が存在しない空白部分(白)と雲が存在する部分(黒)に2値化することで余分な情報を削除する。PGMファイルはグレースケール画像なので2値化するにあたっての閾値は各画素値(0～255)を全画素数分集計し、その平均値とした。2値化後の画像は雲域が黒色領域、空白部分が白色領域になる(図5-1)。

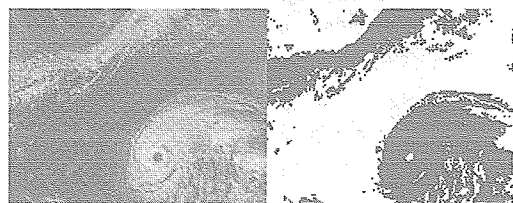


図5-1 元画像 → 二値画像

##### 4-2.8 近傍収縮膨張による図形変換処理

画像の2値化後(図5-1)、小さな雲域が多数存在しており雲域抽出の際に困難となるため、8近傍収縮膨張<sup>[4]</sup>による図形変換処理を行った。8近傍収縮膨張の処理と手順を次に示す。

まず注目画素をX0とし注目画素の近傍をX1～X8とする(図5-2)。2値画像よりすべての画素は0(黒)または255(白)で表されているので、X0を含む8近傍(X1～X8)すべての画素が雲域(黒)ならば注目画素X0の画素値を0(黒)に、雲域でない画素があればX0の画素値を255(白)にする

(収縮)。注目画素X0の画素値が0であればその8近傍(X1～X8)の画素値を全て0にする(膨張)。8近傍収縮膨張を図5-1(2値画像)に対して実行した結果、図5-3のようになった。

X1	X2	X3
X8	X0	X4
X7	X6	X5

図5-2 注目画素をX0とする8近傍モデル



図5-3 2値画像に8近傍収縮膨張した結果

##### 4-3.8 近傍輪郭追跡による輪郭抽出

図形変換処理によりさらに簡略化された衛星画像(図5-3)から雲域(黒領域)の輪郭追跡を行い、輪郭を座標の集合として抽出する。

8近傍での輪郭追跡アルゴリズムは次のようにした。対象の2値画像を順に走査し、輪郭追跡の開始画素x(黒領域)を探す。開始画素が見つければ、その点より8近傍を時計回りに追跡し次の画素(黒領域)をみつめる。そのための基本画素追跡順番(x0～x7)を図6-1としてx0からx7の順番で画素の追跡を行い各画素で次の画素(黒領域)が見つければx0～x7各画素に対応した追跡順番で次の画素追跡を行う(図6-2～図6-5)。この8近傍時計回りの追跡処理を繰り返し、輪郭抽出を行い処理前と処理後の区別のために追跡済みの黒領域の画素値を別の画素値に変換、最終的に開始画素xに戻れば輪郭追跡処理を終了する。

上記の一連の処理を全ての雲域に対して実行する。

x6	x7	x0
x5	x	x1
x4	x3	x2

図 6-1 基本画素追跡順番

6	7	0
5	x1	1
4	3	2

6	7	0
5	x2	1
4	3	2

図 6-2 追跡順番(x1 対応)、(x2 対応)

4	5	6
3	x3	7
2	1	0

4	5	6
3	x4	7
2	1	0

図 6-3 追跡順番(x3 対応)、(x4 対応)

2	3	4
1	x5	5
0	7	6

2	3	4
1	x6	5
0	7	6

図 6-4 追跡順番(x5 対応)、(x6 対応)

0	1	2
7	x7	3
6	5	4

0	1	2
7	x0	3
6	5	4

図 6-5 追跡順番(x7 対応)、(x0 対応)

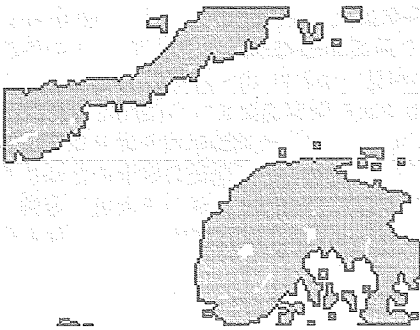


図 6-6 輪郭追跡処理後の画像(輪郭: 黒実線)

#### 4-4. データベース化

輪郭追跡処理後に抽出した座標の集合は衛星画像撮影日、時間と共にデータベースに挿入する。撮影日は日付データ型、時間は文字データ型、座標の集合は幾何データ型の polygon(多角形)として挿入する。幾何データ型は座標で扱い、2次元空間オブジェクトを表現することができ、多角形は座標点のセットで表現する。衛星画像は1時間毎に撮影されているので1日あたり最高24枚の画像がある。1画像あたりに抽出できる雲域の個数は20前後であり、1996年9月4日~2001年12月31日までの衛星画像から雲域抽出を行いデータベースに挿入した結果、588774件のデータになった。

#### 5. 類似画像検索方式

まず、類似画像検索において取り扱う、幾何データ型と幾何関数、演算子を述べる。

##### 幾何データ型

- ・座標点(point): 幾何データ型の基礎となる2次元構成要素  
書式: (x,y)
- ・矩形(box): 対角線上の角を表す一組の座標点(point)で表現  
書式: (x1,y1),(x2,y2)
- ・多角形(polygon): 座標点のセットで表現  
書式: (x1,y1),..., (xn,yn)

##### 幾何演算子

- ・&&: 重なるか否か  
使用法: box'((0,0),(1,1)) && box'((0,0),(2,2))'
- ・@: 含まれている  
使用法: point '(1,1)' @ circle '((0,0),2)'

##### 幾何関数

- ・height(polygon): 多角形の高さ  
使用法: height(polygon '((0,0),(1,1),(1,0))')
- ・length(polygon): 多角形の長さ  
使用法: length(polygon '((0,0),(1,1),(1,0))')
- ・width(polygon): 多角形の幅  
使用法: width(polygon '((0,0),(1,1),(1,0))')

##### 幾何型変換関数

- ・point(polygon): 多角形の中心  
使用法: point(polygon '((0,0),(1,1),(1,0))')

類似画像検索は特定の雲域を検索対象として選出し、その雲域の長さ、中心点や多角形の特徴を生かす。検索対象とした雲域に対して類似雲域とする条件を、①対象雲域に重なる。②対象雲域の長さに対して $\pm n$  誤差以内の長さを持つ。③対象雲域の雲幅に対して $\pm k$  誤差以内の幅を持つ。④対象雲域の中心点に対して $\pm m$  誤差以内の中心点を持つ。この4条件としAND検索を用いて、真となる類似雲域を検索する。次に4条件の詳細と命令SQLを示す。

①対象雲域の類似画像ならば、雲域で重なるとして幾何演算子の&&を用い、対象雲域と重なる雲域を検索する。

```
SQL=select 雲域 from cloud where 雲域 B &&
      雲域 A
```

②対象雲域の類似画像ならば、雲域の長さ  $z$  において類似するとして幾何関数 length を用い、対象雲域の長さ  $z \pm n$  の範囲内に存在する雲域を検索する。

```
SQL=select 雲域 from cloud where length(雲域
      B) > z-n and length(雲域 B) < z+n
```

③対象雲域の雲幅に対し $\pm k$  以内の幅を持つ対象雲域の類似画像ならば、雲域の幅  $w$  において類似するとして幾何関数 width を用い、対象雲域の幅  $w \pm k$  の範囲内に存在する雲域を検索する。

```
SQL=select 雲域 from cloud where width(雲域
      B) > w-k and width(雲域 B) < w+k
```

④対象雲域の類似画像ならば、雲域の中心点  $(x,y)$  において類似するとして幾何型変換関数 point(polygon)を用い、対象雲域の中心点  $(x \pm m, y \pm m)$  の線分を対角線とする矩形の範囲内に中心点が存在する雲域を検索する。

```
SQL=select 雲域 from cloud where point(雲域
      B) @ box '(x-m,y-m),(x+m,y+m)'
```

## 6.検索実行による結果

データベース化された雲域データから類似画像が検索できるかを検証する。2001年8月19日21時の衛星画像(図7-1)を検索対象として類似画像検索を実行する。

検索の条件は、

- ① 長さが400以上の雲域を対象雲域
- ② 検索雲域が対象雲域に重なる
- ③ 検索する雲域の長さの誤差を $\pm 50$
- ④ 検索する雲域の幅の誤差を $\pm 10$
- ⑤ 検索する雲域の中心点の誤差を $\pm 5$

とする。

①実行時の対象雲域は図7-2の雲域になる。この雲域の長さ、幅、中心点は次のとおりである。

長さ = 486.02869702974

幅 = 66

中心点 = (87,117)

以上の数値に対して③-⑤の誤差範囲でAND検索を実行する。その結果、32件のデータが取得できた(図7-3)。32件のデータを表5に撮影日とその枚数でまとめた。①で対象雲域とした雲域(図7-2)の類似画像検索結果としては、1997年7月25日、2000年7月7日、2000年8月19日(21時以外)、2001年8月20日の4日である。この4日間の衛星画像は対象雲域が台風であることから形状的に類似している。他の衛星画像は対象雲域に対して位置的には類似しているが、形状的には類似していない。

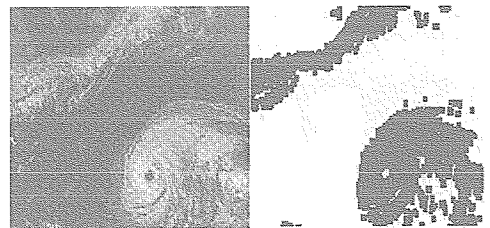


図7-1 元画像 → 2値画像 8近傍収縮膨張

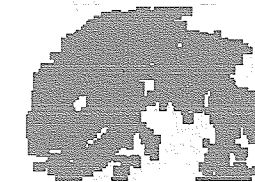


図7-2 長さ400以上で対象雲域になる雲域

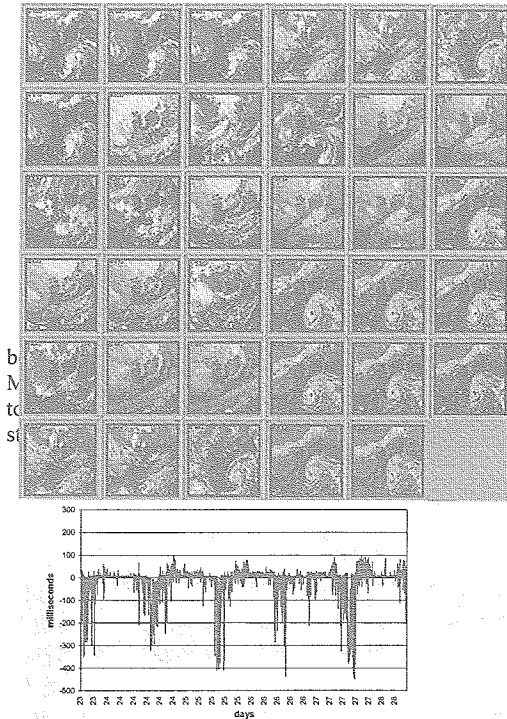


Figure 25. Time offset between mebius and Tokyo

Figure 25 shows the network traffic between Tokyo University NTP server and Mebius. Once again, it is similar to figures 23 and 24.

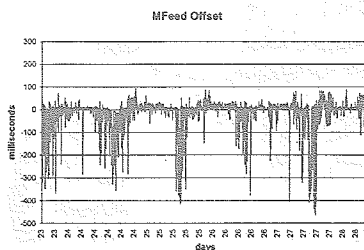


Figure 26. Time offset between mebius and mfeed

Figure 26 shows the network traffic between NTT's experimental public NTP server (MFEED) and Mebius. Again, it is similar to the previous graphs.

If network congestion occurs between Tokyo University and MFEED, for example, it could be seen comparing the tests made to produce figures 24 and 25. Unfortunately, at the time of the tests no network congestion was big enough to be clearly seen in the tests made.

### Final Conclusion

In this study it was demonstrated that a consumer JY time receiver can be used as a time source for WAN network evaluation purpose, adapting it with a PPS signal. However, the accuracy of the unit, even using PPS signal adaptation, is not enough to measure a local area network. Furthermore, the use of NTP is effective as a network performance evaluation tool. It has a good advantage upon the most used tools, like ping and traceroute, as it is able to show not only the network congestion, but also its direction.

### References

- NTP Official Home Page - <http://www.ntp.org>
- The Japan Standard Time Group - <http://jty.cri.go.jp>
- JY Time Receiver Unit Manufacturer  
<http://www.c-dex.co.jp>  
<http://www.tristate.ne.jp/rf-clock.htm>
- Experimental NTP servers - <http://www.jst.mfeed.ad.jp/>
- Programmable Chips FAQ - <http://www.optimagic.com/faq.html>
- RRD tool - <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/>
- Perl - <http://www.perl.org/>