

【論文】

被覆平面を用いた体系的な不可視領域の決定

飯 倉 善 和

Systematic Determination of Invisible Areas Using Covering Planes

Yoshikazu IIKURA

Abstract: Viewshed analysis based on digital elevation model is one of the most common operation in GIS. The viewshed or visible area is naturally defined using a sightline between a observation point and a target point, but its calculation is computationally intensive. In this paper, formal definitions on visibility are provided by using sightplane as well as sightline, and their equivalence is proved. To form the sightplane as a covering plane, three points including the observation point are selected. As these point can be selected in distance base order, redundant check of invisibility becomes evitable. On this basis, an efficient and simple algorithm is proposed and demonstrated for its superiority to the sightline algorithm.

keyword: 数値標高モデル (DEM), 視野域 (viewshed), 被覆平面 (sightplane),
走査順序 (scan order)

1. はじめに

数値標高モデルを利用してある地点（視点）から見通せる領域（視野域）を判定することは、地理情報システムに要求される基本的な機能の一つである（バーロー, 1990）。視野域を用いればアンテナなどの施設の立地や軍事作戦などの移動経路を合理的に決定することできる（Nagy, 1994）。また、起伏の激しい地表面に入射する電磁エネルギーには視野域からの放射エネルギーが含まれる。したがって、地表面での熱収支を厳密に計算する場合（Dozier and Frew, 1990）や、衛星画像から地表面の反射率を求める場合（Proy et al., 1989）において、視野域の同定は重要な問題となる。

これまで利用されてきた視野域の判定アルゴリズムのほとんどは、視点と対象点との間に視線を遮る

地表面が存在するかどうかという可視性の自然な定義（以後視線法とよぶ）から導かれてきた（Yoeli, 1985）。しかし、地表面の熱収支計算などへの適用を考えると、この考えでは計算量が膨大となるため、少数の地点で計算を行う（Proy et al., 1989）か近似的な取扱い（Dozier and Frew, 1990 ; Sandmeier and Itten, 1997）を行わざるを得なかった。

これに対して、Wang et al. (1996) は、視点における局所的な曲面（被覆面）を考えることにより、地点の一部をあらかじめ視線法による可視性の判定から排除できることに着目した。しかし、視点の特徴（頂点、穴点、平点など）により被覆面の形状を変えたり、視点ばかりでなく対象点における被覆面も考慮するなど発見的な取扱いが多く、アルゴリズムの内容は複雑である。したがって、視点の特徴の分類や被覆面のパラメータの計算などの前処理も考慮した場合、視線法だけのアルゴリズムにくらべてどれだけ計算時間が節約できるのかは明らかではない。

本論文では、視点を含む3点が構成する被覆平面の

飯倉：〒020-8551 盛岡市上田4-3-5

岩手大学工学部情報工学科

Dept. Computer and Information Science,

Faculty of Engineering, Iwate University

e-mail. iikura@cis.iwate-u.ac.jp

みを用いた視野域の判定アルゴリズム(被覆法)を提案する。不可視点を重複して判定することのないように被覆の順序を体系化することにより、アルゴリズムを明解かつ効率的なものとするができる。

第2節で、1次元の場合を例に視線法および被覆法の基本的な考えを説明するとともに、それらの2次元への拡張について検討する。第3節で2つの方法に基づく2次元の視野域判定の基本的なアルゴリズムを示す。また、考えられうる効率化の方法についても検討を行う。第4節では、実データに開発したプログラムを適用し、提案する被覆法の視線法に対する優位性を示す。

なお、利用目的に応じて視点と対象点の位置や格子点以外の標高の与え方が異なるという視野域の定義自体に曖昧さがあることが指摘されている(Fisher, 1993; Sorensen and David, 1993)。本論文では、第2節でおこなう簡明な定義のもとで厳密な視野域の性質の解明と明確なアルゴリズムの提案を行う。

2. 基本的な用語と概念

2.1. 1次元における視野域

1次元直線上に等間隔に置かれた地点(P_i)で標高(h_i)が図のように与えられているとする。簡単化のため、視点 P_0 から以下の定義で可視と判断される $0 < i < n$ の地点を視野域 V に含まれているとする。また、 i より小さい番号の地点を便宜的に左側、大きい地点を右側と呼ぶことにする。

定義1: 地点 P_i が視点 P_0 から可視であるとは、 P_i の左側のすべての地点 P_j が次の不等式を満足することである。

$$h_j < h_0 + (h_i - h_0) * (x_j - x_0) / (x_i - x_0) \quad (1)$$

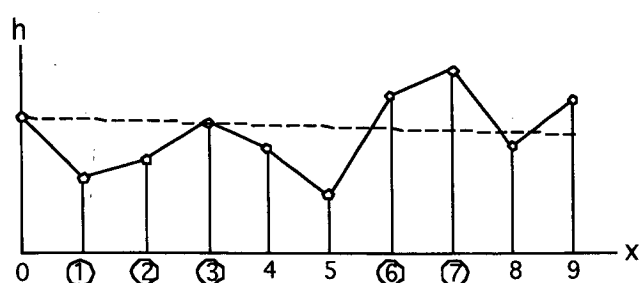


図1 視線を用いた可視域の判定

上式の右辺が視点から P_i への視線の方程式である。この定義に従って、可視と判定された地点を○で囲んだ。定義1の対偶から次の補題が成り立つ。

補題1: 地点 P_i が可視でないための必要十分条件は、 P_i の左側で以下の不等式を満たす P_j があることである。

$$h_j \geq h_0 + (h_i - h_0) * (x_j - x_0) / (x_i - x_0) \quad (2)$$

定義1に素直に従えば、 P_i の可視性の判定には、 $(i-1)$ 回の(1)式の計算が必要となるが、これらの判定のすべてが必ずしも必要では無いことが補題1からわかる。すなわち、(2)式より地点 P_i の可視性の判定を、 P_i に近い順に左側に向かって行えば、 P_i が不可視の場合には $j=1$ になる前に不可視を決定できる。図1では P_5 が不可視であることは P_5 への視線が P_4 によって遮られることでわかる。

可視判定の効率化を行う上で対象とする地点 P_i より左側の可視の地点の中で、最も P_i に近い可視地点 P_j (以下では P_i の影響点とよぶ)と地点 P_i の視線の関係が重要である。次の定理が成立する(後述する補題2と系1より証明できる)。

定理1: P_i が可視である必要十分条件は、影響点 P_j が(1)式を満たすことである。

例えば、 P_6 が可視であることは影響点 P_3 によって P_6 への視線が遮られないことで知ることができる。影響点を活用して、視点に近い地点から順番に可視性を判定することにより1回の走査ですべての点の可視性が判定できる。なお、 P_1 は定義により可視であり、 P_2 の影響点であることに注意されたい。

以上のような判定法を視線法とすれば、これとは発想の異なる判定法が考えられる。すなわち、見える点を考えるのではなく、見えなくなる点を考える方法であり、これを被覆法と呼ぶことにする。被覆法の原理は次の補題によって簡潔にあらわすことができる。

補題2: 地点 P_j より右側の地点 P_i で以下の不等式を満足するものは可視でない。

$$h_i \leq h_0 + (h_j - h_0) * (x_i - x_0) / (x_j - x_0) \quad (3)$$

(3) 式は (2) 式を変形することにより得られることから、補題2は補題1で判定の対象とした地点 P_i を右側に拡張したものであることがわかる。さらに、地点 P_j を影響点と考えれば補題2は定理1の必要条件の対偶を含んでいる。

また、定理1の十分条件の対偶である次の系は定義1から容易に導ける。

系1: 不可視な地点 P_i の影響点 P_j は (1) 式を満足しない。

以上の考察から、1次元の被覆法では、左から右に走査を行うだけですべての地点の可視性が判断できることがわかる。図1を例にとれば、定義により可視である P_1 (影響点) からの被覆を考えることにより、定理1より P_2 が可視であることがわかる。 P_3 も同様に可視であることがわかるが、補題2から P_3 の被覆を考えることにより P_4 、 P_5 が不可視であり、 P_6 が可視であることがわかる。

視線法でも左から右への走査ですべての地点の可視性が判断できるが、影響点を記憶しなければならないことが被覆法と異なる。

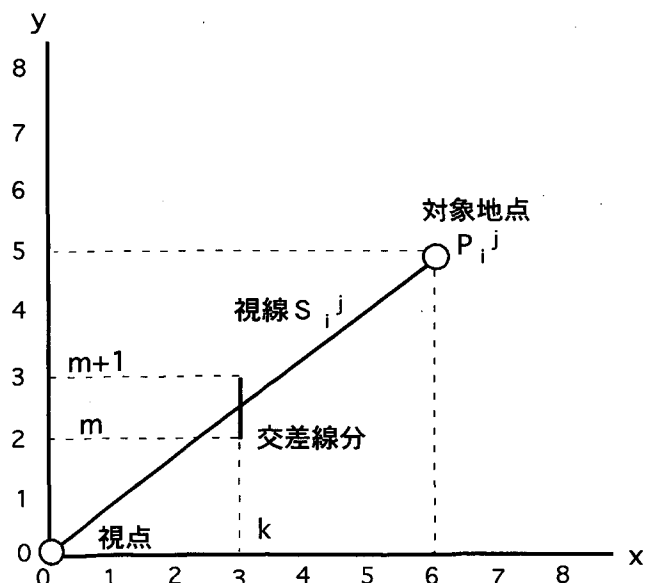


図2 2次元平面上の視線の投影

2.2. 2次元における視線法

2次元直交座標系 (x, y) 上に等間隔 (格子状) に置かれた格子点 P_i^j で標高 h_i^j が与えられているとする。簡単化のため、 P_0^0 を視点とする。また、地点 P_i^j から見て i より小さい番号の地点を左側、大きい地点を右側、 j より小さい番号の地点を下側、大きい地点を上側と呼ぶ。なお、本節では $i > j$ の領域を対象とするが、 $j > i$ の場合には i と j を交換して考えることにすれば一般性は失わない。また視線とは視点 P_0^0 と地点 P_i^j を結ぶ半直線のこととする。この時、視線上の標高 H_i^j は x の関数として以下のようにあらわされる。

$$H_i^j(x) = h_0^0 + (h_i^j - h_0^0) * x / i \quad (4)$$

地点 P_i^j への視線の xy 平面上への投影 S_i^j は、以下の直線の方程式であらわされる。

$$y = x * j / i \quad (5)$$

図2のように、視線は地点 P_i^j の左側で $x = k$ の直線と1ヶ所 $y' = k * j / i$ で交わる。この地点 (k, y') を交差点、 y' を切り捨てによって整数化した値を m とし、 $P_k^m P_{k+1}^{m+1}$ を交差線分と呼ぶ。また、交差線分の地表面の標高を以下の線形補間式で与える。

$$g_k^m(y) = h_k^m + (h_{k+1}^{m+1} - h_k^m) (y - m) \quad (6)$$

これらを用いて、可視を以下のように定義する。

定義2: 地点 P_i^j が可視であるとは、地点 P_i^j から左のすべての交差点の標高 $g_k^m(y)$ が以下を満足することである。

$$g_k^m(y') < H_i^j(x_k) \quad (7)$$

1次元における定義1と補題1と同様の関係が、定義2と次の補題3に成立する。

補題3: 地点 P_i^j が可視でない必要十分条件は交差点 P_k^m で以下の不等式をみたすものがあることである。

$$g_{km}(y') \geq H_i^j(x_k) \quad (8)$$

すなわち、すべての交差点のチェックは必要ではない。

1次元の視線法では、走査の方向に一意に定まる影響点を記憶することが計算の効率化のポイントであった。しかし、2次元の場合には視線が地点毎に変化することや、走査の方向（地点の選択の順序）が不連続になることが避けられない。したがって、影響点のような概念を成立させることは難しい。視線法で視野域を判定するには、各点毎に視線を遮る地表面（複数）の存在をチェックすることが避けられないと考えられる。

ただし、以下に示すような性質は視線法を効率化する上で有効である。

系2：地点 P_i^j の $x = x_{i-1}$ から x_k までの交差点が (7) 式を満足し、かつ P_k^m とその隣接点 P_k^{m+1} の双方が可視であるならば、 P_i^j は可視である。

2.3. 2次元における被覆法

被覆法を2次元に拡張するために、図3のような被覆平面を考える。

定義3：被覆平面 T_k^m とは視点 P_0^0 、地点 P_k^m 、および地点 P_k^{m+1} の3点が作る平面で、視線の投影 S_k^m と S_k^{m+1} との間を定義域とするものである。

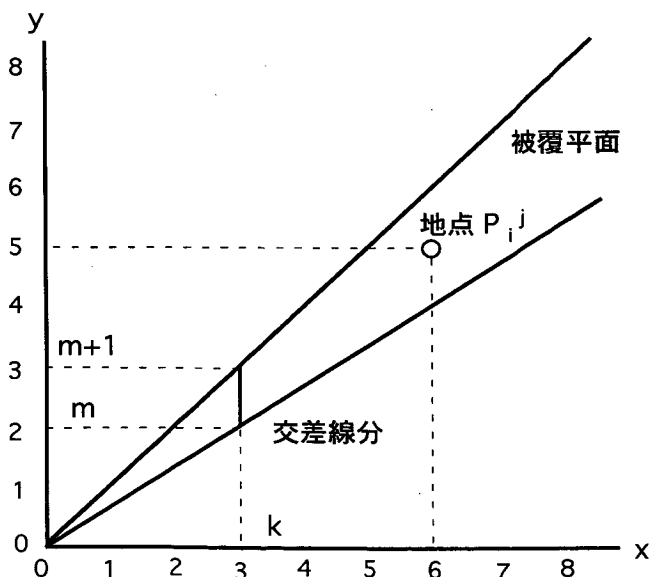


図3 被覆平面の定義域

3次元空間内で3点が定まれば、平面の方程式

$$T_k^m(x, y) = ax + by + c \quad (9)$$

は一意に定まる。係数 a, b, c は、以下のように計算される。

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} x_0 & y_0 & 1 \\ x_k & y_m & 1 \\ x_k & y_{m+1} & 1 \end{pmatrix}^{-1} \begin{pmatrix} h_{00} \\ h_{km} \\ h_{km+1} \end{pmatrix} \quad (10)$$

被覆平面を用いれば以下の性質が成立する。

補題4：被覆平面 T_k^m の定義域にある地点 P_i^j で以下の不等式を満足するものは、不可視である。

$$h_i^j \leq T_k^m(i, j) \quad (11)$$

証明：地点 P_i^j への視線 $H_i^j(x_k)$ および $x = k$ における交差線分を考える。被覆平面 T_k^m が交差線分上の標高 $g_k^m(y)$ を含むことに注意すれば、(11) 式から (8) 式が導ける。したがって、補題3より地点 P_i^j 視線は可視ではない。

<終わり>

視点に近い順にすべての交差線分の被覆平面を用いて、その定義域の地点の不可視の判定を (11) 式を用いて行う。この操作により、すべての不可視地点が判定できることは、補題3と補題4の証明から明らかである。しかし、この操作は不可視地点が重複して判定されるなど効率的でない。

処理の効率化のためには、被覆平面を適用する交差線分を次の定理で示すように体系的に選択する必要がある。

定理2：不可視な地点 P_i^j には、交差線分の中に P_k^m または P_k^{m+1} のいずれかが可視で、しかもその被覆平面 T_k^m が (11) 式を満たすものが必ず存在する。

証明：不可視な地点 P_i^j には、交差線分の中でその被覆平面が (11) 式を満たすものが少なくとも1個は必ず存在する。それらの交差線分のなかで

一番視点に近いものを P_k^m, P_k^{m+1} とする。これらが双方が不可視であるとするれば、 P_k^m または P_k^{m+1} の交差線分のなかで被覆平面が (11) 式を満たすものが必ず存在する。それらの交差線分の中の一つの被覆平面は地点 P_k^j に対しても (11) 式を満足する。これは、はじめに交差線分のなかで視点に一番近いものを選んだことに矛盾する。

＜終わり＞

この性質を用いて、被覆法を体系的に2次元に適用することにより視野域を決定できることがわかる。すなわち、視点に近い順に可視と判定された地点 P_k^m だけについて、被覆平面 T_k^m と T_k^{m-1} による不可視の判定を行えばよい。

処理をさらに効率化するために、被覆平面の定義域すべてについて不可視の判定が必要かどうかを考えてみよう。まず以下の性質に注意する。

系3：被覆平面 T_k^m による判定で $x = k'$ 上のすべての地点 P_k^m が可視ならば、その右側はそれらの地点の被覆平面 T_k^m だけにより、可視性を決定することができる。

ここで被覆平面 T_k^m 上での不可視の判定が、左から順に行うこととする。このとき、 $x = k'$ においてすべての地点 P_k^m 視点から順に可視であることは、可視になったそれらの地点からのすべての被覆平面が T_k^m より上にくることになる。したがって、 $x > k'$ で被覆平面 T_k^m により不可視と判定されるものは、必ず、 $x = k'$ の可視の地点の被覆平面により不可視

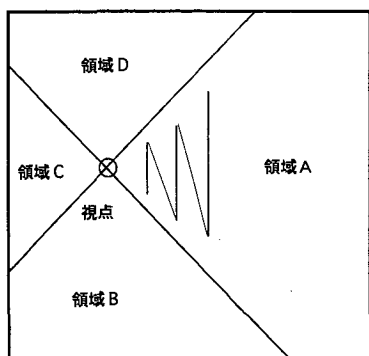


図4 領域の分割と走査の順序

と判定される。したがって、被覆平面 T_k^m による判定は必要では無くなる。

3. 視野域の計算アルゴリズム

2次元平面上の視野域を計算するアルゴリズムを、前節の理論に基づいて提案する。提案するアルゴリズムは、被覆法で系3で述べた性質を利用するものであるが、比較のため視線法と被覆法の基本アルゴリズム、および視線法で系2の性質を利用した方法についても紹介する。なお、アルゴリズムの実装には対話的な処理が可能で、豊富な表示解析機能を持ついわゆる第4世代言語である IDL (Interactive Data Language) を用いた (Starck et al. 1998)。

図4に対象とする視点と可視性を判定する領域を模式的に示した。前節では視点の右側の領域Aの上半分の可視性に関して検討を加えたが、領域A全体に拡張することは容易である。また、領域B、C、Dについては、反時計回りに画像を90度回転していけば、領域Aに適用するアルゴリズムを変更することなく使うことができ、見通しのよいプログラムが作成できる。IDLでは画像の90度の回転を効率的に行う関数 (rotate) が用意されている。

3.1. 視線法の基本アルゴリズム (視線法1)

視線法では視点 (point0) と対象地点 (point1) の間の交差点における地表面と視線の高さを定義2に基づいて順番にチェックする。これを実装したのが、以下のプログラムである。

```

1  pro sisen1,point0,point1
2  common pdata
3  xs = point0[0] + 1 & xe = point1[0] - 1
4  ys = point0[1] & ye = point1[1]
5  zs = dem[point0[0],point0[1]]
6  ze = dem[point1[0],point1[1]]
7  for i=xs,xe do begin
8      xx = double(i)
9      xa = zs + (ze-zs)*(xx-xs+1)/(xe-xs+2)
10     yy = ys + (ye-ys)*(xx-xs+1)/(xe-xs+2)
11     y1 = floor(yy) & y2 = y1 + 1
12     dy = yy - y1
13     z1 = dem[i,y1] & z2 = dem[i,y2]
14     zb = z1 + (z2-z1)*dy
15     if (zb ge za) then begin
16         dview[point1[0],point1[1]]=0

```

```

17   return
18   endif
19   endfor
20   dview[point1[0],point1[1]]=1
21   end

```

1行目は手続きの名前 (sisen 1) と仮引き数名が示されている。仮引き数は視点 point 0 と対象地点 point1 であり、2次元のベクターとして与えられる。2行目にはグローバルな変数 (common文) が宣言されている。これにはあらかじめメインのルーチンにおいて、数値標高モデル (dem) とその画像サイズ (imax, jmax) および計算した視野域を記録する配列 (dview) が定義されている。交差点の座標 (xx,yy) における地表面の標高 za と視線の標高 zb が15行目で比較されている。また、地表面の標高が高いとわかった時点で、計算が打切られる (補題3)。なお、このアルゴリズムでは、主プログラムで対象点を呼び出す順序は結果に影響を与えない。

3.2. 系2による視線法の効率化 (視線法2)

視線法では、交差線分の両端の画素が可視ならば計算を打切ることができる (系2)。この性質を利用するために、視線法1のプログラムの15行目以降のブロックを以下のように書き換えた。

```

15   if (zb ge za ) then begin
16       dview[point1[0],point1[1]]=0
17       return
18   endif else begin
19       if (dview[i,y1]*dview[i,y2]) then begin
20           dview[point1[0],point1[1]]=1
21           return
22       endif
23   endelse

```

このプログラムでは、19行で用いる交差線分の両端の画素の可視性が正しく与えられなければならない。したがって、主プログラムでは視点に近い順に対象点を呼び出す必要がある。図4に、対象点を呼び出す順序を模式的に示した。

3.3. 被覆法の基本アルゴリズム (被覆法1)

被覆法でも、視線法2と同様に対象地点の選択は視

点に近い順に行う必要がある。定理2に基づいた被覆法の基本プログラムの主プログラムを以下に示す。

```

1   pro main,ix,iy
2   common pdata
3   for k= 1,imax-ix do begin
4       kx = ix + k
5       ps = 0 > (iy - k)
6       pe = (jmax-1) < (iy + k - 1)
7       flag = 0
8       for ky=ps,pe do begin
9           if (dview[kx,ky]) then begin
10               hifuku2, [kx,ky], [kx,ky-1], [ix,iy], dview
11               if (flag) then
12                   hifuku2, [kx,ky-1], [kx,ky], [ix,iy], dview
13           endif else flag = 1
14       endfor
15   endfor
16 end

```

可視と判断された地点だけについて、被覆平面による右側の地点の可視性のチェックを行う (9行目)。重複しない可視の画素が続く場合、重複してチェックすることのないようにフラッグ (flag) をたててチェックしている。

被覆法のプログラム (hifuku 1) を以下に示す。

```

1   pro hifuku1,point1,point2,point0
2   common pdata
3   x0=point0[0] & y0=point0[1] & z0=dem[x0,y0]
4   x1=point1[0] & y1=point1[1] & z1=dem[x1,y1]
5   x2=point2[0] & y2=point2[1] & z2=dem[x2,y2]
6   aaa=[x0,y0,1.0],[x1,y1,1.0],[x2,y2,1.0]
7   bbb=invert(aaa)##[z0,z1,z2]
8   for ix=point1[0]+1,imax do begin
9       ys=ceil(y0+(y1-y0)*(ix-x0)/(x1-x0)) > 0
10      ye=floor(y0+(y2-y0)*(ix-x0)/(x2-x0)) < jmax
11      for iy=ys,ye do begin
12          za = dem[ix,iy]
13          zb = bbb[0]*ix + bbb[1]*iy+ bbb[2]
14          if (za le zb) then dview[ix,iy]=0
15      endfor
16  endfor
17  end

```

6～7行目は平面の方程式の係数 (10式) を求めるための行列の演算である。13行目のzbが被覆平面の標高値であり、対象とする地点の標高と14行目で比較されている。

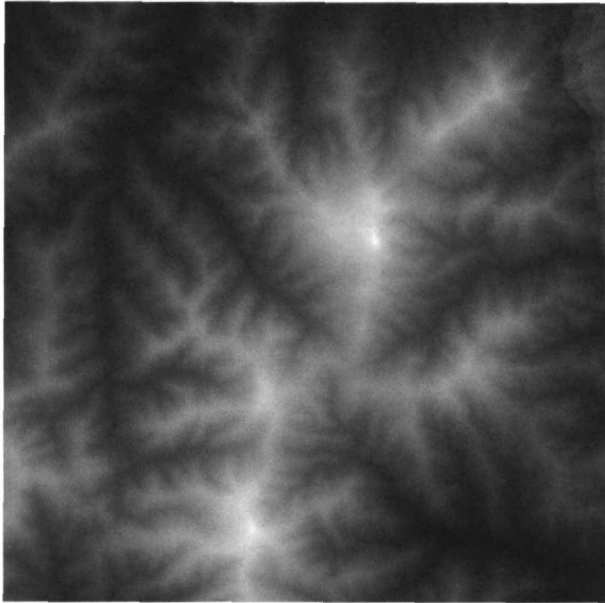


図5 対象とした数値標高モデル

3.4. 系3による被覆法の効率化（被覆法2）

被覆法を効率化するため、系3に基づく技法を基本プログラムの11行目以降に実装したものを以下に示す。

```

11  check = 1
12  for iy=ys, ye do begin
13      za = dem[ix, iy]
14      zb = bbb[0]*ix + bbb[1]*iy + bbb[2]
15      if(za le zb) then begin
16          dview[ix, iy]=0 & flag = 0
17      endif
18  endfor
19  if(check) then return

```

すべての画素が可視かどうかをフラッグ (check)

によりチェックしている以外の変更はない。

4. 数値例による検証

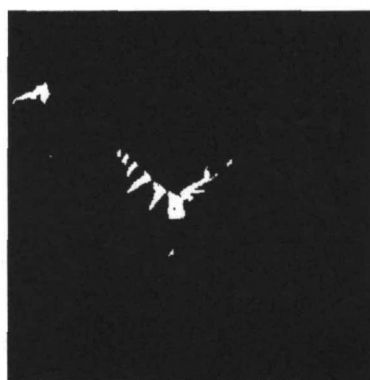
4.1. 対象としたデータ

国土地理院から日本全国をカバーした数値標高モデルが出版されている。このDEMの仕様は、座標系が等緯度経度、メッシュサイズが約50m、標高の量子化（2バイトの整数）が1mであり、2次メッシュ毎に分割されて一つのファイルに収められている。この中から岩手県全体のデータを連結し、ランドサットTMデータとの整合性の高いUTM座標系（メッシュサイズ30m）に変換した。

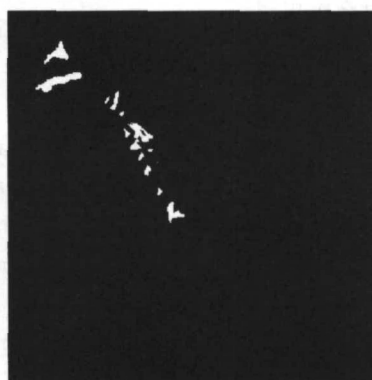
本論文では、図5に示すように岩手県の大川地区（約10km四方）を切り出して用いた。画像の大きさは301ライン、301ピクセルである。なお、この地域の標高の最高は1308m、最低は649mである。

4.2. 計算結果の例

対象データに4つの視野域の計算アルゴリズム（視線法1、視線法2、被覆法1、被覆法2）を適用した。視点をいろいろ変えて計算したが、ここでは最高点、最低点、中心点の例を示す。2次元平面上での可視の定義がどのアルゴリズムにも適用されているので、原理的には計算結果は同一になるはずである。数値的な計算誤差のために生じた非常に少数（1%以下）の例外を除いて、計算結果が同一になることを確認した。得られた視野域を図6に示す。標高の低い見通しの悪い地点と、標高の高い見通しの良



中心点
(150,150)



最低点
(104,200)



最高点
(184,182)

図6 得られた視野域

表 1 計算速度の比較 (秒)

視点の位置	視線法 1	視線法 2	被覆法 1	被覆法 2
中心点	34.1	33.1	21.0	3.5
最低点	25.8	24.5	18.8	3.7
最高点	88.1	42.1	76.7	6.0

い地点での視野域の違いが示されている。

4. 3. 計算速度の比較

各方法の計算速度を示したのが表 1 である。計算には Power Macintosh (7300/180) をスタンドアロンで用いた。IDL 言語は Ver. 5 である。

基本アルゴリズムの比較でも、被覆法 1 が視線法 1 より若干高速である。被覆法では、さらに系 3 に基づく改良の効果が非常に大きく、提案する手法 (被覆法 2) により 10 倍程度の高速化が見られた。とくに、視野域が広い場合に効果が大きい。視線法でも系 2 に基づく改良が見られるが、提案する方法に較べればその効果は小さい。

5. おわりに

数値標高モデル (DEM) から、視野域を生成する明解で効率的なアルゴリズム (被覆法) を提案した。これまでのアルゴリズム (視線法) では対象までの視線を遮る地表面の存在をチェックしていたが、提案したアルゴリズムでは視点を含む 3 点からなる平面 (被覆平面) によっておおわれる部分を不可視領域と判定することにより最終的に視野域を判定する。被覆平面を視点に近い地点から順に構成することにより不可視領域の重複した設定をさけることができることを理論的に示した。

今後、視線法と被覆法の計算量の詳細な解析と、より一層の高速化の可能性 (並列処理を含む) について検討していきたい。また視野域の計算を地表面からの放射エネルギーの計算や衛星画像の地形効果補正へ利用していきたいと考えている。

付記

視線を全く用いない方法が Wang et al. (2000) によっても発表された。彼等は、影響点と同値の概念を 2 次元に拡張することにより、簡便でかつ高速なアルゴリズムを提案している。しかし、その拡張が厳密ではないため視野域の計算結果は視線法と同じにならない。事実、簡単な例で視野法と異なる不自然な結果が得られることを示すことができる。なお、この付記および参考文献は学術委員会の了承のもとに本論文の条件付き受理後 (1 月 14 日) に付け加えたものである。

参考文献

- P. A. バーロー著, 安仁屋政武・佐藤亮訳 (1990) 『地理情報システムの原理』, 古今書院, pp.47-66.
- Dozier J., Frew J. (1990) Rapid calculation of terrain parameters for radiation modeling from digital elevation data, *IEEE Trans. on Geoscience and Remote Sensing*, **28** (5), 963-969.
- Fisher P.F. (1993) Algorithm and implementation uncertainty in viewshed analysis, *Int. J. GIS*, **7**, 331-374.
- Nagy G. (1994) Terrain visibility, *Computers and Graphics*, **18**, 763-773.
- Proy C., Tanre D., Deshamps P.Y. (1989) Evaluation of topographic effects in remotely sensed data, *Remote Sensing of Environment*, **30**, 21-32.
- Sandmeier S., Itten K. I. (1997) A physically-based model to correct atmospheric and illumination effects in optical satellite data of rugged terrain, *IEEE Trans. on Geoscience and Remote Sensing*, **35** (3), 708-717.
- Sorensen A. P., David L. P. (1993) Two algorithms for determining partial visibility and reducing data structure induced error in viewshed analysis, *Photogrammetric Engineering and Remote Sensing*, **59**, 1149-1160.
- Starck J. L., Murtagh F., Bijaoui A. (1998) *Image Processing and Data Analysis*, Cambridge University Press.
- Wang J., Robinson G. J., White K. (1996) A fast solution to local viewshed computation using grid, *Photogrammetric Engineering and Remote Sensing*, **62**, 1157-1164.
- Wang J., Robinson G. J., White K. (2000) Generating viewsheds without using sightlines, *Photogrammetric Engineering and Remote Sensing*, **66**, 87-90.
- Yoeli P. (1985) The making of intervisibility maps with computer and plotter, *Cartographica*, **22**, 88-103.