再帰的に二等分割された直角二等辺三角形上の頂点数を求めるアルゴリズム

A Simple Algorithm for Counting Vertices on Recursively Bisected Right-Angled Isosceles Triangles

三河賢治†，正会員 長谷川誠††

Kenji Mikawa† and Makoto Hasegawa††

Abstract Image compression algorithms are evaluated on several performance measures, two of which are the total amount of compressed image data and image quality. In this paper, we deal with an image compression algorithm that divides an image domain into some smaller triangles. Although the triangulations will cause image quality to deteriorate in general, the relationship between the triangles is not trivial. We, therefore, propose an efficient algorithm for counting the number of the internal vertices in a triangulated domain, to investigate their relationship. It is believed that the solution will contribute to further work on improving the process of image compression.

キーワード：直角二等辺三角形の二等分割，画像処理，バックトラック技法，組合せアルゴリズム

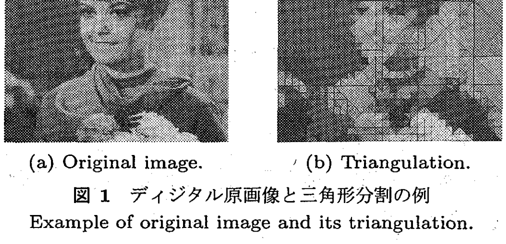
1. まえがき

近年，ディジタル画像データを幾つかの小領域に分割し，画像データを圧縮するアルゴリズム[2)4)6)]が活発に議論されている．本論文で扱う圧縮アルゴリズムは，始めに画像データを幾つかの直角二等辺三角形に分割し，次に各直角二等辺三角形を適当な閾値に基づいて逐次的に二等分割していく．各三角形の頂点には，原画像データの輝度値が与えられていて，頂点の輝度値から領域内部の画像データを近似する．このような圧縮のデータ構造は，二分木で実現される．二分木の内部節点数と葉数は，それぞれ画像データを分割した回数と分割された小三角形の数に対応する．ディジタル画像の三角形分割の構造を図1に示す．幾つかの小領域に分割するようなディジタル画像の圧縮は，画像の拡大縮小に強く，自然画像のベクトル化[1)4)]に代表されるように幾つかの応用が提案されている．

画像を圧縮するアルゴリズムは，原画像データと圧縮された画像データのデータ量や画質を比較して，総合的に評価される．具体的に述べると，原画像 R の大きさを幅 X，高さ Y，各画素の輝度値 $f(x,y)$，$0 \leqq f(x,y) < t$，圧縮された画像の各画素の輝度値 $g(x,y)$，$0 \leqq g(x,y) < t$，について，圧縮された画像の画質は，次式

$$PSNR = 10 \cdot \log\left(\frac{X \cdot Y \cdot t^2}{\sum_{(x,y)\in R}\left(f(x,y)-g(x,y)\right)^2}\right) \quad [\mathrm{dB}]$$

のように，すべての画素に対する平均2乗誤差の逆数により評価される．現在のところ，三角形の頂点数と画質との関係は明らかではなく，今後の解明が期待されている．そこで本論文は，三角形の頂点数と画質との関係を明らかにしていくための助走的な研究として，直角二等辺三角形を再帰的に二等分割した平面図形に対し，この平面図形の頂点数を効率よく求めるアルゴリズムを与える．任意の平面

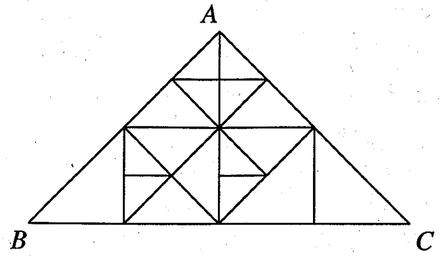


(a) Original image.　(b) Triangulation.

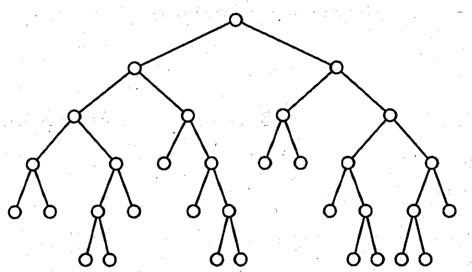
図1 ディジタル原画像と三角形分割の例
Example of original image and its triangulation.

2005年2月2日受付，2005年4月20日再受付，2005年5月27日採録

†新潟大学 総合情報処理センター
（〒950-2181 新潟市五十嵐二の町8050番地，TEL 025-262-7581）
††近畿大学 工学部
（〒739-2116 東広島市高屋うめの辺1番）
†Integrated Information Processing Center, Niigata University
(8050 Ikarashi-2-no-cho, Niigata-shi 950-2181, Japan)
††School of Engineering, Kinki University
(1 Takaya-Umenobe, Higashi-Hiroshima-shi 739-2116, Japan)

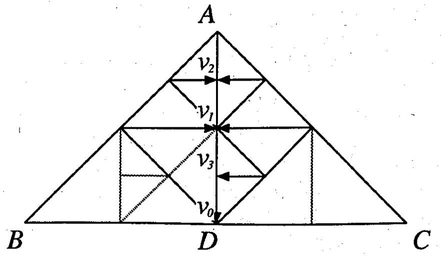


(a) Bisection triangulation.

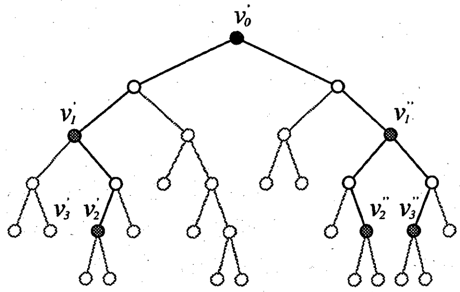


(b) Binary tree representation.

図 2　三角形の二等分割とその二分木表現
Bisection triangulation and its binary tree representation.



(a) Common vertices on bisection AD.



(b) Binary tree representation.

図 3　二等分線上の共有頂点と二分木表現
Common vertices on bisection and its binary tree representation.

図形の頂点の数を v，辺の数を m，面の数を f とすると，Euler の多面体定理から

$$v - m + f = 1$$

を満たすことが知られている．しかしながら，本論文で扱うような直角二等辺三角形の分割では，三角形を分割する箇所によって頂点数と辺数が変化するので，Euler の関係式を直接利用することはできない．本論文では，直角二等辺三角形の分割の構造に適したバックトラック手法により，三角形の分割回数 n に対して，$O(n)$ 時間で頂点数を求めるアルゴリズムを提案する．頂点の総当りによる従来のバックトラック手法では $O(n^2)$ 時間であった。

2. 準　備

三角形の分割と二分木との関係を次のように定義する．三角形の領域を二分木の節点 v に対応させ，三角形を分割する垂線について，長辺を底辺とする垂線の左領域を v の左の子，右領域を v の右の子とする．根のみの二分木は分割されていない三角形を表す．三角形を分割する回数と分割された領域数は，それぞれ二分木の内部節点と葉に対応する．このように二分木を構成すると，二分木の内部節点は，三角形を二等分する線分と一対一対応となり，二分木の内部節点が三角形を二等分する線分，内部節点に接続する左右の子が分割された領域とみなすことができる．三角形の分割とその二分木表現を**図 2** に示す．

三角形の分割と頂点数との関係は，次の二通りの場合に分けて考えることができる．一つ目は，三角形の領域を分割する毎に新しい頂点が一つ増える場合であり，二つ目は領域を分割してもある二等分線上で頂点を共有するために新しい頂点が増えない場合である．前者の三角形の分割回数（二分木の内部節点の数）を n，後者の共有する頂点数を c とすると，三角形の頂点数 v は，最も外側の三角形を構成する 3 頂点を含め，次式

$$v = n - c + 3$$

で求めることができる．三角形の分割回数 n は容易に計算できるので，共有する頂点数 c を求めるアルゴリズムが重要になる．三角形の分割の定義から，ある三角形が分割されると，次は分割された左右の小領域が再帰的に分割されていく．三角形を二等分する線分に注目すると，次の性質を得る．

[性質 2.1]　直角二等辺三角形を長辺に対し，垂直に二等分する線分は，他の再帰的に分割される直角二等辺三角形を二等分する線分とはならない．

3. 探索アルゴリズム

第 2 章で定義したように，二分木の節点を三角形の領域に対応させると，二分木の内部節点が分割された三角形を表し，葉が分割されていない三角形を表す．このように定義すると，二分木の内部節点は，分割された三角形の領域を表すと同時に，この三角形を二等分する線分と三角形の長辺との交点に対応することになる．この新しい対応関係と，交点が共有頂点であるかを判定する処理を加えれば，二分木の節点を探索するアルゴリズムが，そのまま共有頂点を探索するアルゴリズムとなる．

共有頂点を求めるアルゴリズムの考え方は次の通りである．性質 2.1 では，三角形を二等分する線分は，この線分によって分割された左右の小三角形を二等分する線分にならないことを示している．性質 2.1 を利用すると，三角形を二等分する線分上で，共有頂点となる頂点のみを計数す

```
procedure search(v_L, v_R : node)
begin
  if v_L <> false and v_R <> false then begin
    c := c + 1;
    V_L := leftchild(rightchild(v_L));
    V_R := rightchild(leftchild(v_R));
    search(V_L, V_R);
    V_L := rightchild(leftchild(v_L));
    V_R := leftchild(rightchild(v_R));
    search(V_L, V_R)
  end
end;
```

図 4 共有頂点探索
Common vertices search.

```
procedure traversal(v : node)
begin
  if check(v) <> false then begin
    v_L := leftchild(leftchild(v));
    v_R := rightchild(rightchild(v));
    search(v_L, v_R);
    traversal(leftchild(v));
    traversal(rightchild(v))
  end
end;
```

図 5 行きがけ順による二分木巡回
Binary tree traversal in preorder.

るアルゴリズムを構築し，分割された左右の小三角形に対して，このアルゴリズムを再帰的に適用していけば，漏れなく三角形のすべての共有頂点を計数することができる．

図 3 に示す $\triangle ABC$ と $\triangle ABC$ の二等分線 AD を用いてアルゴリズムを説明する．頂点 v_1，v_2 は AD 上で共有頂点であるが，v_3 は AD 上の頂点であるが共有頂点ではない．二分木上では，これらの共有頂点に対応する節点は，左右の部分木で内部節点として現れるが，共有頂点でない頂点は，左右の部分木のどちらか一方の節点が葉として現れる．AD によって分割された左右の領域から，AD 上で初めて出会う頂点は v_1 であり，次回以降の分割によって AD 上で出会う（または出会う可能性のある）頂点は v_2 と v_3 の二つである．v_1 を起点として，左右の領域から v_2 と v_3 に向かう分割の経路は，ちょうど M 字を描くような経路をたどる．これらの頂点の関係を二分木上で考えると，頂点 v_1 に対応する左部分木の節点 v_1' から右の子を経由して左の子 v_2' が頂点 v_2 に対応する節点であり，v_1' から左の子を経由して右の子 v_3' が v_3 に対応する節点である．一方，頂点 v_1 に対応する右部分木の節点 v_1'' から左の子を経由して右の子 v_2'' が v_2 に対応する節点であり，v_1'' から右の子を経由して左の子 v_3'' が v_3 に対応する節点である．AD 上で共有頂点となる頂点は，ちょうど左部分木と右部分木で対称な経路をたどることによって得られる．例えば，左右の部分木上で v_2（または v_3）に対応する節点の少なくとも一方が葉のとき，AD 上で共有しないので，v_2（または v_3）から探索を始める必要はない．図 3 では，右部分木上の v_3'' は内部節点だが，左部分木上の v_3' は葉なので，v_3 より後の共有頂点は存在しない．一方，左右部分木上で v_2 に対応する節点 v_2' と v_2'' はどちらも内部節点なので，v_2 を共有頂点として数え上げる．同様に，v_2 と v_3 を起点として，左右の領域から AD 上の共有頂点を探索する経路も M 字を描くような経路をたどる．

上記のように，AD 上の初めての共有頂点 v_1 を起点として，AD 上のすべての共有頂点を探索するアルゴリズムは，**図 4** で示すような再帰的な手続きで構成することができる．図中の二つの関数 leftchild(v) と rightchild(v) は，二分木の節点 v を引数として，それぞれ v の左の子，v の右の子を返し，v が葉または存在しないときは false を返す．手続き search の引数について，v_L は分割された三角形の左領域から共有頂点となる頂点に対応する二分木上の節点，v_R は右領域から共有頂点となる頂点に対応する二分木上の節点である．

一方，AD 上で初めて共有頂点となる v_1 は，二分木上では，$\triangle ABC$ の領域を表す根 v_0' から左の子を経由して，左の子 v_1' が左領域から共有頂点となる v_1 に対応し，v_0' の右の子を経由して右の子 v_1'' が右領域から共有頂点となる v_1 に対応する．以降，search(v_1', v_1'') を呼び出し，AD 上のすべての共有頂点を探索すればよい．この手続きを，すべての三角形の領域に対して再帰的に繰り返すと，三角形内部のすべての共有頂点が求まる．したがって，三角形の領域を二等分する線分上で，初めて共有頂点となる頂点を探索するアルゴリズムは**図 5** に示す手続きとなる．図中の関数 check(v) は，二分木上の節点 v を引数として，v が葉または存在しないときに false を返す．

次に，手続き traversal(v) の計算量を求める．二分木上の節点を行きがけ順にたどり，各節点 v で手続き search(v_L, v_R) を呼び出し，v に対応する三角形の領域を二等分する線分上のすべての共有頂点を探索する．このとき，線分上の共有頂点は他の三角形を二等分する線分上の共有頂点とはならないので，手続き traversal(v) と search(v_L, v_R) で探索される節点の個数は三角形の分割回数 n に比例する．したがって，手続き traversal(v) の計算量は $O(n)$ 時間である．

4. むすび

本論文では，与えられた二分木に対して，共有頂点数を $O(n)$ 時間で求めるアルゴリズムを提案した．このアルゴリズムの対象は，直角二等辺三角形に分割するような手法に限らず，ディジタル画像を逐次四等分割するような手法[1,2]にも対応する．

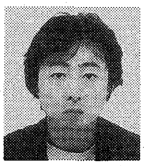
一方で，三角形の分割と同時に共有頂点数を求める方法については論じていない．共有頂点の探索は，ある部分木の根から対称な経路を探索することによって実現する．三角形の分割時に共有頂点を計数しようとすると，分割のある時点で対象になっている節点が，どの先祖（どの三角形

を二等分する線分）上で共有頂点となりうるかを判定する方法が著者の知る限り自明ではなく，対象の節点からある先祖までの経路に対称な経路を対象の節点から特定することが困難であると予想される．今後の課題である．

〔文　献〕

1) 長谷川誠，三河賢治："デジタル濃淡画像の三角平面パッチによる SVG 形式への変換"，映情学誌，**57**，10，pp.1337–1341（Oct. 2003）
2) 長谷川誠，山崎一生："可変ブロック分割と双 1 次曲面パッチ近似による画像データの圧縮"，信学論（D-II），**J84-D-II**，7，pp.1399–1408（Jul. 2001）
3) I. Semba："Generation of All the Balanced Parenthesis Strings in Lexicographical Order", Inf. Process. Lett., **12**, 4, pp.188–192（1981）
4) 李相善，田中弘美："アダプティブメッシュを用いた画像の並列領域分割"，信学論（D-II），**J82-D-II**，7，pp.1171–1179（Jul. 1999）
5) F. Ruskey, T. C. Hu："Generating Binary Trees Lexicographically", SIAM J. Comput., **6**, 4, pp.745–758（1977）
6) 若月大輔，石井郁夫，高橋章，今井博英，牧野秀夫："VR オブジェクトの局部的な形状詳細度制御のためのマルチスケールパッチ生成法"，信学論（D-II），**J86-D-II**，5，pp.697–705（May 2003）

みかわ　けんじ
三河　賢治　1995 年，茨城大学工学部卒業．2001 年，同大学院理工学研究科博士後期課程修了．2001 年，同大学非常勤研究員．2002 年，同講師．2003 年，新潟大学助手．博士（工学）．主として組合せアルゴリズムの設計と解析に関する研究に従事．



はせがわ　まこと
長谷川　誠　1991 年，新潟大学工学部卒業．1994 年，同大学院自然科学研究科修士課程修了．1994 年，富士写真フイルム（株）入社．1998 年，新潟大学講師．2005 年，近畿大学助教授．博士（工学）．主として画像データ圧縮に関する研究に従事．正会員．