

3次元空間における経路探索の一手法

藤村真生, 小堀研一, 久津輪敏郎
大阪工業大学

2次元平面の自動配線で用いられるLeeの手法を3次元に拡張し, 3次元空間内における経路探索を実現する. この時Octreeモデルを導入してボクセル数と処理時間を抑える. しかしLeeの手法は大きさの異なるボクセルに対応していない. そこで各ボクセルに経路の通過点の情報を付加する. またOctreeの分割レベルを効果的に利用することにより, 様々な太さを持った経路の自動探索に対して拡張を行った. 発表では幾つかの実験結果を提示し, 手法の有効性を示す.

An Automatic Routing Method in a Three Dimensional Space

Masao Fujimura, Ken-ichi Kobori, Toshiro Kutsuwa
Osaka Institute of Technology

This paper describes an automatic wiring method in a three dimensional space. In general, we can expand two dimensional Lee' algorithm into three dimensions in order to realize automatic wiring. However, the following three problems occur when this algorithm is simply expanded. The first problem is that the data of the space lattice increases. The second problem is that two or more wirings can not be done to one space lattice. The last is that this method can not deal with wiring bundled thick wires. To avoid the first problem, we introduce octree spatial partitioning model. Addition of information of number, by which automatic wiring can be realized at each space lattice, makes multiple wiring possible. Moreover, we can realize wiring bundled thick wires by using division level of octree data structure. Several experimental results show that the method is effective for the wiring design in a three dimensional space .

1. はじめに

電子機器内部など、3次元空間での部品間の配線は従来、設計者が設計図をもとに行ってきた。この作業は熟練を要するため、労力とコストに問題があった。そこで本稿では、コンピュータによる自動的な経路探索の手法を提案する。

2次元平面内における経路探索の手法として、ピクセルにラベリングを施して処理を行うLeeの手法¹⁾がよく知られている。これはプリント基板の配線に一般に用いられている手法であるが、処理をピクセルからボクセルに拡張することによって、容易に3次元空間に応用することが考えられる。しかし、現実にはボクセル数が膨大になり、処理速度の面で実用的ではない。

そこでボクセル数を削減できるOctreeモデル²⁾を用いることが考えられる。このモデルは空間全体を表す立方体を再帰的に8分割し、空間を階層構造で表現する。ところが、Leeの手法をOctreeモデルに単純に適用すると、大小の立方体が混在することから経路長の増加と経路の冗長性が大きくなるという2つの問題を生じる。

本稿ではOctreeモデルのオクタントに新たな情報を付加³⁾し、これらの問題を解決する。またLeeの手法では考慮されていない、束配線を自動的に行うための手法についても考察する。

提案手法の有効性を検証するため、Leeの手法をVoxelモデルおよびOctreeモデルに単純に拡張した経路探索の手法との比較実験を行った。その実験結果と考察についても報告する。

2. 経路探索の手順

経路探索の手順の流れを図1に示す。流れ図中に薄い網掛けで示した処理は、各経路ごとに繰り返し実行される。また濃い網掛けで示した処理は、太さの異なる経路を探索する場合に繰り返し実行される。それ以外の処理は1度だけ実行される。次にこれらの処理について詳しく述べる。

2. 1 配置と分割

図2を用いて経路を1本だけ決定する手順について説明する。なお、実際は3次元であるが、理

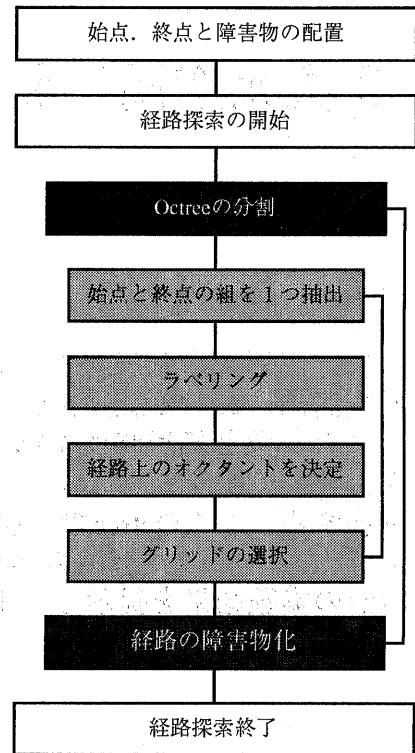


図1 全体の処理の流れ

解しやすいように2次元で説明を行う。

図2 a)において黒い領域は障害物である。最初にこれらの障害物と、経路の始点Iと終点Eを配置する。そして空間にOctreeモデルの分割⁴⁾を施す。分割により同図a)に正方形で示した可変サイズのオクタントが生成される。この時オクタントは障害物と干渉する領域と干渉しない領域に分類される。ここで、Iを含むオクタントのような最小のオクタントはOctree分割の木の深さが最も深いので、最大分割レベルのオクタントと呼ぶ。同図は最大分割レベル3の例である。

2. 2 ラベリング

図2 a)においてまず、IからEまでの各オクタントに対してラベリングを行う。この処理はIから始まり、Iを中心とする半径Dの同心円を描きながら広がってEまで到達する。ここでラベルは、最大分割レベルのオクタントの辺の長さを単

位長とする I から各オクタントまでの距離である。各オクタントの大きさは最大分割レベルのオクタントとの辺の長さの比によって定義する。

ラベリングの手順を次に示す。

初期設定：I をラベル 0 とする。また処理のループのための変数 D を 0 に設定する。

処理 1：以下の 3 つの条件をすべて満たすオクタント群を抽出する。

- ・ラベル D のオクタントの 6 近傍に存在する。
- ・ラベリングされていない。
- ・障害物の領域でない。

処理 2：抽出されたオクタント群からオクタントを 1 つずつ選択する。選択されたオクタント i の大きさが S_i の時、オクタントのラベル D_i を

$$D_i = S_i + D \quad (1)$$

とする。これは I から i までの距離である。

処理 3：処理 1 の条件を満たすオクタントが存在しない場合、又は処理 1 のオクタント群をすべて選択し終われば、D を +1 して処理 1 に戻る。

処理 4：選択されているオクタントが E ならば処理を終了する。含まない場合は処理 2 に戻る。

ラベリング結果は同図 b のようになる。

2. 3 経路上のオクタントの決定

ラベリングされたオクタント群を探索し、経路となるオクタントを決定する。

初期設定：E のオクタントを選択する。

処理 1：選択されているオクタントの周囲 2 6 近傍に存在するオクタント群を抽出する。

処理 2：抽出されたオクタント群から、最小のラベルが書き込まれたオクタントを選択し、経路の通過するオクタントとして決定する。

処理 3：選択されたオクタントが I ならば処理を終了する。含まない場合は処理 1 に戻る。

以上の処理は、E から I まで順に最短経路となるオクタントを辿りながら進む。図 2 b のラベリング結果からは同図 c に網掛けで示したオクタントが決定される。

2. 4 グリッドの選択

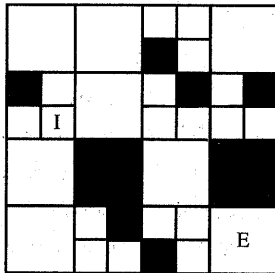
2. 4. 1 グリッドの無い場合の問題点

Lee の手法による経路探索では、経路となるボクセルの決定により処理を終了する。従って Octree モデルを使用した場合には、次のような 2 つの問題を生じる。

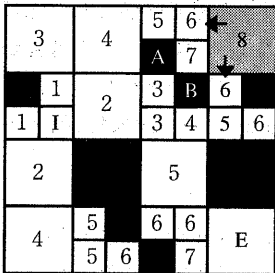
問題点 1：大きなオクタントを経路とする場合、そのオクタント全体が経路となる。従って 2 本以上の経路を通せない。

問題点 2：経路を線で表したとき、オクタント内部の経路が通過する点を、オクタントの中心座標に取らざるを得ない。従ってオクタント内部で経路の折れ曲がりが発生する場合があるため、経路長は実際よりも長くなる。

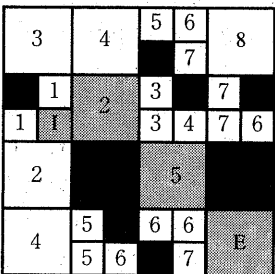
これらの問題を解決するため、オクタント内に通過可能な経路数、および各オクタントの 6 面において通過可能な点の設定を行なう。加えてオク



a. 配置と分割



b. ラベリング結果



c. 決定したオクタント

図 2 経路上のオクタントの決定

タント内の経路長を最短とする通過点を決定する。これらの手法について詳しく述べる。

2. 4. 2 オクタントの通過可能経路数

Octreeモデルの各々のオクタントに対して、経路が通過できる数 N_i を設定し、問題点1を解決する。 N_i の設定方法と、これを用いることにより複数の経路を通過させる方法について述べる。

ここで、最大分割レベルのオクタントに通過可能な経路の数を N とする。また大きさが S_i であるオクタントを通過する経路の、オクタント内部での平均長は S_i であるとする。この時、このオクタントに通過可能な経路の最大値は

$$N_i = 2 \cdot N \cdot S_i \quad (2)$$

とする。今回は N の値を1とした。実際の処理では、2. 1のOctree分割の際に各々のオクタントの大きさに応じて式(2)の計算を行ない、通過可能経路数の初期値として設定する。

2. 3の探索によって経路のオクタントが決定すれば、該当オクタントの通過可能経路数を-1し、通過可能経路数が0になればそのオクタントは経路の通れない領域、すなわち障害物となる。

2. 4. 3 グリッドの設定

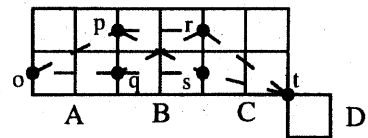
各オクタントの6面にグリッドを設定し、問題点2を解決する。1面当たりの通過可能経路数 E_i は、最大分割レベルのオクタントにおいて1面当たりに通過可能経路数 E を用いて

$$E_i = E \cdot S_i^2 \quad (3)$$

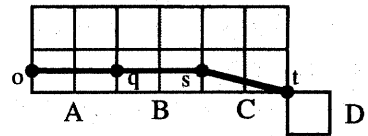
のように求められる。今回は E の値を1と設定した。これは2. 4. 2で N の値を1と設定した場合に対応している。オクタントの各面に、 x 、 y 、 z 各軸に平行な格子状にグリッドを配置する。経路はグリッドの中心を通ることとする。またすべてのグリッドにフラグを用意し、1度経路が通過したグリッドはフラグの値を変化させることとした。これによって同一グリッド内に複数の経路が設定されることを回避した。

2. 4. 4 グリッドの選択

2. 4. 3で設定したグリッドから、経路に対して適切なグリッドを選択することにより経路長を短縮することができる。ここでは2次元の図3



a. オクタントとグリッド



b. 決定した経路

図3 グリッドと経路の選択

を用い、この方法について説明する。

同図では隣り合った4つのオクタントA, B, C, Dが存在している。2. 3の探索結果から、AからDに向かう経路が決定されたとする。またA内のグリッド o は経路の通過する点としてすでに決定されているものとする。

本手法では、あるオクタント内の経路が通過するグリッドはそのオクタントを含んだ3個のオクタントから決定する。同図の例ではA内の経路が通るグリッドをA, B, Cのオクタントから決定する。

同図に灰色の線で区切られた領域が、グリッドのサイズであるとする。この時AからCまでのオクタントで通過可能なグリッドは p , q , r , s の4点である。従って経路は図3 aに示した

$$\begin{aligned} o-p-r, & \quad o-p-s, \\ o-q-r, & \quad o-q-s \end{aligned}$$

の4通りが存在する。これらの距離をすべて算出し、距離が最短になる経路を選択する。この例では $o-q-s$ が最短経路として選択される。従って、オクタントA内のグリッド q が経路の通過する点として決定される。

同様にして、B, C, DのオクタントからB内のグリッドを決定する。この時、CとDは面で接続せず、オクタントの稜線或いは頂点によって接続している。このような場合にはグリッドを面上に設定することが不可能なため、便宜的に稜線上または頂点に設定する。従って経路は

$q-r-t$, $q-s-t$

の2通りとなる。これらの距離を比較した結果経路 $q-s-t$ が選択され、B内の点 s が経路の通過する点として決定される。このような手順を繰り返し1本の経路を決定する。結果は図3 b) になる。

3. 太さの異なる経路への対応

これまでの手法では、経路は一定の太さと仮定していた。ここでは、最大分割レベルを経路の太さごとに変化させることによって太さの異なる経路の探索に対応する手法を提案する。

3. 1 処理の流れ

この手法の処理の流れは次のようになる。

処理1: 2. 1)の配置の際、経路の始点と終点の組に加え、経路の太さの情報も与える。この時、経路の太さを表現することが可能な最も浅い分割レベル L_i を各々の経路に設定する。これは、経路の太さと最大分割レベルのオクタントが持つ辺の長ささが一致する場合であり、太い経路ほど小さな値となる。また L を、すべての経路の中で最小の L_i の値に設定する。

処理2: Octreeモデルの分割を行う。この時の最大分割レベルは L とする。

処理3: L_i が L と同一である経路だけを決定する。この時の処理は前章までと同様の手法によって実行する。

処理4: 決定した経路を、障害物とする。この手順は次節で述べる。

処理5: L を $+1$ する。 L_i が L と同一の経路が存在すれば、処理2に戻る。存在しない場合は、すべての処理が終了したこととなる。

以上の手順を用いることにより、前章までの経路探索の手法に経路の障害物化と再分割の処理を加えるという単純な処理を加え、図1の濃い網掛けで示したOctree分割をループさせるのみで太さの異なる経路に対応することが可能となる。

3. 2 経路の障害物化

この手法では、Octreeの分割を行う際に処理の終了した太い経路を障害物として扱う必要があ

る。この方法について説明する。

処理の終了した経路を含んだあるオクタントを抽出し、最大分割レベルのオクタントの大きさを 1 とした時の経路の太さ T を求める。この時の T は実数である。経路の通過した2つのグリッド G_1 , G_2 を結ぶ線分を芯として、 T の幅を持つ6面体を生成する。この6面体を経路上の全オクタントに対して生成し、障害物として登録する。その後2. 1)で説明したOctreeモデルの分割を行い、経路が通過不可能な領域を設定する。これにより、すでに処理の終了した太い経路と、新しく探索される細い経路との交差を回避できる。

4. 実験と考察

これまで説明した手法の有効性を検証するための実験を行った。今回の実験で使用した計算機はSGI社のIndy (R4400SC, 150MHz) である。

4. 1 実験に用いたモデル

実験に用いたモデルは以下の通りに設定した。大小50個の障害物と、探索すべき始点終点の組を100組それぞれ任意に配置する。このような配置のモデルを3種類用意した。

空間の分割数は、一般的な工業製品の配線精度を考慮してVoxelモデルにおいて256x256x256を最大とし、分割数の変化による処理時間と経路長の変化を確認するため64x64x64までの3段階を設定した。またOctreeモデルでは、これらと同一の精度を表現可能な最大分割レベル6からレベル8までの3種類を設定した。

4. 2 単純な拡張手法との比較

以下の3種類の手法を比較する実験を行った。

1) Leeの手法を3次元に拡張し、Voxelモデルを用いた手法。

2) 1)と同様にOctreeモデルを用いた手法。

3) 2章で述べた提案手法。

比較した項目は3種類のモデルについて1本あたりの平均探索時間と平均総経路長である。空間の分割数を変化させた場合の変化についても計測し比較した。処理時間の比較を図4に、また総

経路長の比較を図5に示す。図5において経路長は最大分割レベルのオクタントの辺の長さを1とした値である。

図4の結果から、Octreeモデルを用いた場合に最も速く処理が行われている。Voxelモデルと比較して計算対象となるデータ数が少なく、モデルの拡張による効果が現れた結果となっている。ところが図5では、Octreeを用いた場合に総経路長が最も長くなっている。これは2章で述べた問題点2の経路の冗長性が現れた結果である。

提案手法では、処理時間については最短経路の探索処理のためOctreeを用いた手法に少し劣るものの、Voxelモデルと比較すると良好な結果を示している。経路長については、Octreeモデルにおいて発生していた問題が解決され、Voxelモデルと比較しても短くなることが実験結果から明らかとなった。またこれらの傾向が分割数を変化させても変わらないことから、提案手法は処理時間を短縮し、かつ経路の冗長性を抑えるのに有効な手法であることがわかった。

4. 3 太さの異なる経路の探索

太さの異なる経路を以下のように設定した。最大分割レベル6, 7および8の太さに相当する経路の数をそれぞれa, b, cとする。先の3種類のモデルそれぞれに対して、100本の経路の数をa : b : cに振り分ける。今回は33:33:34, 5:20:75, 5:5:90の3種類を用意し、平均を算出した。さらに、最大分割レベル8を設定して2章の手法により経路探索し、処理時間と総経路長を計測した。これを0:0:100として示した。

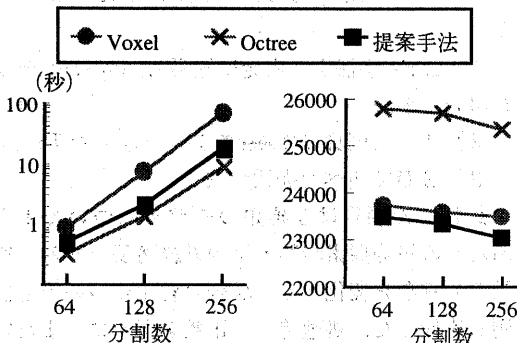


図4 処理時間の比較

図5 経路長の比較

表1 太さの異なる経路の探索

総経路長 (処理時間 単位: 秒)

	モデル1	モデル2	モデル3
33:33:34	24239(172)	23925(166)	23800(145)
5:20:75	23132(166)	23109(166)	23075(147)
5:5:90	23270(166)	23105(166)	23084(147)
0:0:100	23823(20)	23079(23)	23119(22)

以上の実験について、総経路長と処理時間の比較を表1に示した。太さの異なる経路探索の手法では、経路の太さに応じて空間分割の回数が増加し、多くの処理時間を要していることがわかる。総経路長については、細かい経路の数が多いほど短くなっていることがわかる。これは太い経路ほど高密度、高精細な探索が不利であることを示しており、現実の傾向と一致する。

5. おわりに

Leeの手法を拡張し、3次元空間において経路の冗長性を抑えた自動探索を実現する手法について提案した。また太さの異なる経路探索に対応する手法についても示した。実験においては、単純な拡張手法と比較して処理時間および経路長を共に抑えることが可能であることが示された。太さの異なる経路に対応した手法の実験結果でも提案した手法は有効であることがわかった。

しかし太さの異なる経路探索ではOctreeモデルの分割に多くの処理時間を要する。今後はこの分割方法を工夫して処理の高速化を図りたい。

参考文献

- 1) ブルーア, 林訳: デジタル計算機の自動設計, 産業図書, pp.313(1973)
- 2) F.Foley, A.Vandam, S.Feiner, and J.Hughes: Computer Graphics Principles and Practice, Addison-Wesley Publishing Company, pp.533(1990)
- 3) 田畑, 藤村, 小堀: 三次元における自動配線の研究, 情報処理学会第55回全国大会講演論文集(4), pp.301(1997)
- 4) 小堀, 石黒, 久津輪: 境界表現モデルからOctree表現への一変換手法, システム制御情報学会論文誌, Vol.8, No.3, pp.97(1995)