WINGED－EDGEの一考察

5H-3

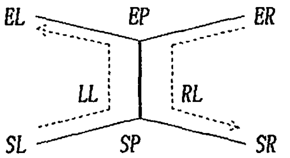
竹内一博, 殿岡光博, 永倉正浩
(株)富士通静岡エンジニアリング

1. はじめに

本報告では，従来のwinged-edgeの問題点を考察し，それらを解決するための拡張されたwinged-edge表現とその時の面の領域認識の方法について述べる．

2. 従来のwinged-edgeによるデータ構造と問題点

winged-edge構造では，各稜線（EDGE）を中心として，以下のデータを持つことで稜線及び，面の関係を規定し，立体を一意に決定している．具体的に図2.1に示す．



SP : Start Point
EP : End Point
SL : Start Left
SR : Start Right
EL : End Left
ER : End Right
LL : Left Loop
RL : Right Loop

図2.1 winged-edge構造

従来のwinged-edge構造では，以下の制限事項がある．

- $SL \neq SR \neq ER \neq RL \neq 0$ (0は該当稜線が存在しない)

従って，1枚の面の定義に以下の制限が発生する．

- 面の領域は，1つの外部LOOPと複数の内部LOOPから定義される．
- 1つのLOOPは必ず2本以上の稜線から構成される

この場合，図2.2のように曲面が複数に分割される．

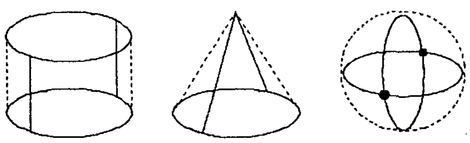


図2.2 従来のwinged-edgeの曲面を含む立体の定義例

この状態では，ユーザ側では1枚として認識される面が複数に分割されるため，ユーザ側と内部表現が一致しない．

3. 曲面を1枚と認識するためのwinged-edgeの拡張

前記の問題点の解決のために，我々は曲面を1枚の面として認識できるように，曲面の領域・稜線を以下のように考え，winged-edge構造の規約を定める．

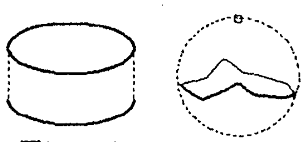
- 面の外郭は，一般に複数の外部LOOPによってその境界が定義される．
- LOOPは，1本の稜線によっても定義可能である．

【拡張winged-edgeのデータ規約】

a) $SR \neq SL \neq ER \neq EL \neq 0$
b) $SR = SL = ER = EL =$ 自分自身
c) ($SL \neq SR$ かつ $EL \neq ER$) かつ
　($SL = EL =$ 自分自身 または $SR = ER =$ 自分自身)
d) ($SL = SR =$ 自分自身) かつ ($EL \neq ER$)
e) ($EL = ER =$ 自分自身) かつ ($SL \neq SR$)

4. 面の認識

拡張winged-edgeでは，円筒面や球面など一般に外郭を定義するLOOPが一つでない．従って，直観的にいえば図4.1のように回転の軸を含むLOOP（定義は後述．以後オープンLOOPと呼ぶ）で挟まれる部分を面と認識する必要がある．



球面の極，円錐面の頂点は便宜的にオープンLOOPと考える．

図4.1 オープンLOOP

面上の点Pは図4.2のように，$P(u, v)$と定義出来る．

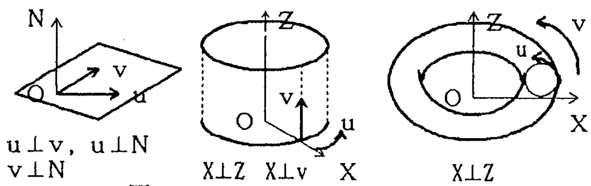


図4.2 面のパラメタ表現例

以下に面の分類と認識処理を行う上での概念を定義する．

Consideration of WINGED-EDGE Data Structure
Kazuhiro TAKEUCHI, Mitsuhiro TONOOKA, Masahiro NAGAKURA
FUJITSU SHIZUOKA ENGINEERING,Ltd.

【定義：周期的】
$P(u,v)=P(u+N\alpha,v)$ where Nは整数，αは周期定数
$P(u,v)=P(u,v+M\beta)$ where Mは整数，βは周期定数
を満たす時に，それぞれu，vについて周期的と呼ぶ．

【定義：オープンLOOP】
ループ順の点列　$P_i\ (u_i,v_i)\ i=1,n$　（n：整数）
を次の条件を満たすようにとる．

$|u_i-u_{i-1}|<\alpha/2$　αはu方向の周期定数
$|v_i-v_{i-1}|<\beta/2$　βはv方向の周期定数
$P_1\ (u_1,v_1)=P_n\ (u_n,v_n)$

この時，$u_1\neq u_n$ なら，u方向のオープンLOOPと呼ぶ．
同様に，$v_1\neq v_n$ なら，v方向のオープンLOOPと呼ぶ．

この時，全ての面は以下の３つのタイプに分類される．

【定義：面の分類】
- type１　uv両方向について周期的でない（例：平面）
- type２　uvのうち一方向のみ周期的（例：円筒面）
- type３　uv両方向とも周期的（例：トーラス面）

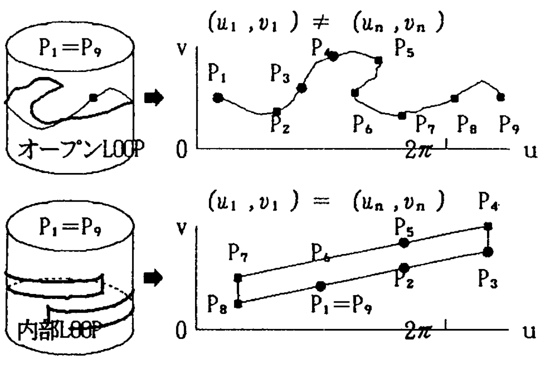


図4.3　オープンLOOPと内部LOOP例

【定義：２次元パラメタ空間の領域D_{uv}と基準線】
- αをu，βをvの周期とする時，領域D_{uv}を定義する．
$D_{uv}\equiv\{(u,v)\ |u\in[0,\alpha),v\in[0,\beta)\}$
- 仮想的な面の境界を定義する為に，基準線を定義する．
u方向が周期的な面では，曲線$u=0,u=\alpha$，
v方向が周期的な面では，曲線$v=0,v=\beta$を基準線と呼ぶ．

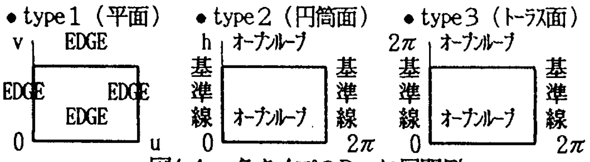


図4.4　各タイプのD_{uv}に展開例

また，内部LOOPは次の３つのcaseに分類できる．

- case１　内部LOOPが基準線に跨がっていない場合
- case２　基準線と内部LOOPで外郭を定義する場合
- case３　基準線と内部LOOPでD_{uv}が複数に分かれる場合

円筒面の例を図4.5 に示す．

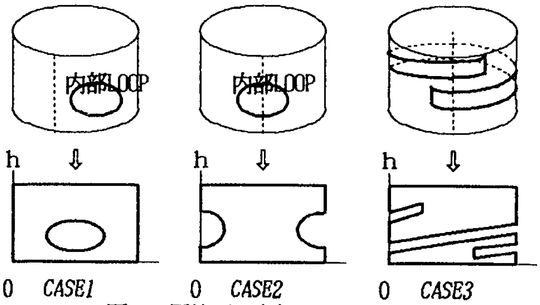


図4.5 円筒面の内部LOOPの分類

即ち，オープンLOOPを持つ面の外郭は，オープンLOOP，基準線，基準線に跨る内部LOOPにより定義される．従って，２次元パラメタ空間上での面の形状は，次のように定義される．

- 面の領域＝Σ〔面の外郭－Σcase１の内部ループ〕式4.1

なお，簡単の為にα・β定数，D_{uv}矩形として説明したが，α・βがそれぞれv・uの関数であっても良く，D_{uv}は矩形である必要はない．

以上，オープンLOOPを持つ面について説明してきたが，オープンLOOPを持たない面についても同様である．これにより，拡張winged-edgeで表現された面の認識が可能になる．

５．おわりに

今回の提案は，境界表現をとるソリッド表現において，面を自然に認識するための１つの方法論である．また，上記で示した面の考え方は，*half-edge* や〔３〕などのデータ構造においても有効である．

ユーザ／アプリケーションにとって自然な認識の単位が，計算機内部でも同じ単位として認識されることは重要である．

参考資料

〔１〕*B.G.Baumgart "A polyhedron representation for computer vision", AFIPS Conf.Proc., 44, pp589-596*
〔２〕*Kevin Weiler "Edge-Based Data Structures for Solid Modeling in Curved-Surface Environments" IEEE CG&A January 1985, pp21-40*
〔３〕*Y E Kalay "The hybrid edge : a topological data structure for vertically integrated geometric modeling" Computer Aided Design Vol21 No3 1989*