

非直角建物ポリゴンに基づく古代建物等の3Dモデルの自動生成

杉原 健一・沈 振江

Automatic generation of 3-D ancient building models based on non-orthogonal building polygons

Kenichi SUGIHARA and Zhen-jiang SHEN

Abstract: 3D models restoring ancient temples, pagodas and towns can be utilized in publicizing research findings from archaeological investigation and facilitating “Public Involvement” for urban planning. However, enormous time and labor has to be consumed to create these 3D models, using 3D modeling software such as 3ds Max or SketchUp. In order to automate the laborious steps, we are proposing the GIS and CG integrated system that automatically generates 3D building models from building polygons (building footprints) on a digital map. Most building polygons' edges meet at right angles (orthogonal polygon). In the digital map, however, not all building polygons are orthogonal. In either orthogonal or non-orthogonal polygons, we propose the new system for automatically generating 3D building models with general shaped roofs including hexagonal or octagonal-base pagoda by straight skeleton computation defined by a continuous shrinking process.

Keywords: 自動生成(Automatic generation), 3次元モデル(3D model), CG(Computer Graphics), 建物ポリゴン(building polygons), Straight Skelton, 非直角ポリゴン(non-orthogonal polygon)

1. はじめに

古の建物を復元する3Dモデルは歴史教育、考古学調査研究、古代建物の復元による観光まちづくり等の様々な分野で利活用が期待される重要な「情報基盤」である。例えは、津波対策のための「住居の高台移転の案」、あるいは「かつてあった古代の建物の復元」を3Dモデル化するには、それぞれ、プランナーが描く計画案の地図やコンサルタント企業が提出する発掘調査地図、古地図などの地図情報に基づき、主に3次元CG作成ソフトを用いて、膨大な手作業にて、街並みの3Dモデルを製作する。

この手作業を省力化するために、建物の3Dモデルを、製作ルールで自動生成する手続き型モデリング(Procedural modeling)が研究されている(Yoav I. H. Parish et al, 2001; Pascal Müller et al, 2006)。

Müllerらは、この手続き型モデリングにおいて、GISが蓄積・管理する電子地図から「建物境界線」を取り込んで、彼らの形状言語(shape vocabulary)において、基本形(basic shapes)に分類する。もし、それができない形状であれば、建物ポリゴンの押し出し処理(extruded footprint)とStraight Skeleton手法(O. Aichholzer et al, 1995)を用いて一般形状の屋根を生成している。しかし、彼らの論文において、電子地図上の建物ポリゴンへのStraight Skeletonの適用手法や適用結果などの図や記述はな

く、手法は明らかにされていない。

本研究では、明らかにされていない建物ポリゴンに対する、Straight Skeleton手法の適用、主に、「Splitイベント」が生じたときの屋根付き建物の3Dモデルを自動生成する手法を明らかにする。

筆者のこれまでの研究成果(Kenichi SUGIHARA, 2011)で、自動生成システムは、電子地図上の直角建物ポリゴン(orthogonal polygon)を長方形の集まりまで分割・分離して、これら長方形の上に、Box形状の建物本体や上から見て長方形の屋根を配置して、3Dモデリングを行った。しかし、全ての建物ポリゴンが直角ポリゴンとは限らない。本研究では、非直角建物ポリゴン(non-orthogonal)，例えば、正多角形の建物境界線に対して六角堂、八角堂などの多角堂の3Dモデルを、Straight Skeleton手法を用いて、自動生成する手法を提案する。

2. 本システムの構成と自動生成のプロセス

本研究における自動生成のシステム構成と古代建物の3Dモデルの自動生成のプロセスを図1に示す。古代建物の3Dモデル(図1右)の情報源は、コンサルタント企業が提出する発掘調査の成果や古地図などの地図情報をデジタル化した電子地図である。電子地図は、汎用GIS(ArcGISなど)によって、蓄積・管理される。電子地図上の建物ポリゴンは、GISのソフト部品(MapObjects)を用いてプログラム開発した「GISモジュール」にて、以下の「前処理」を行う。

杉原健一:〒503-8550 岐阜県大垣市北方町5丁目50番
岐阜経済大学 経営学部 情報メディア学科

Phone: 0584-77-3511

E-mail:sugihara@gifu-keizai.ac.jp

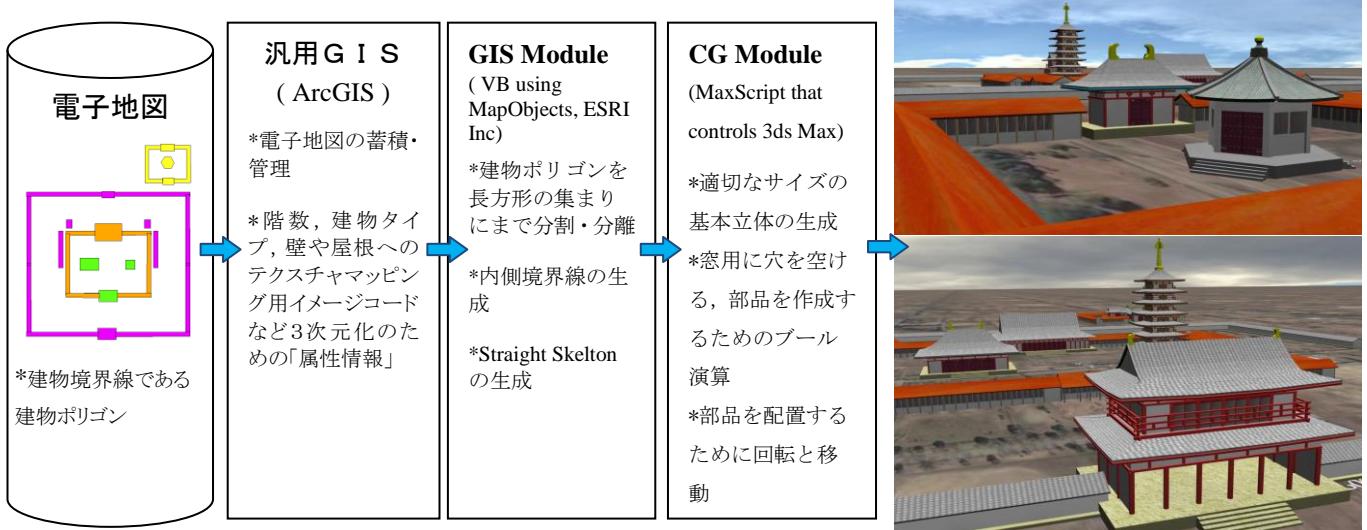


図-1 自動生成システムの構成と古代建物の3Dモデルの自動生成のプロセス

(1)直角ポリゴンを「長方形の集まり」にまで、分割・分離する。(2)建物ポリゴン上の不要な頂点をフィルタリングする。(3)建物境界線よりセットバックした所にある窓やドアを配置するため内側境界線を生成する。(4)建物ポリゴンに対してStraight Skeletonを生成する。

前処理したデータを、3次元CGソフト(3ds Max)をコントロールする「CGモジュール」(MaxScriptでプログラム開発)が読み込み、以下の処理を自動的に行い、3D建物モデルを自動生成する。

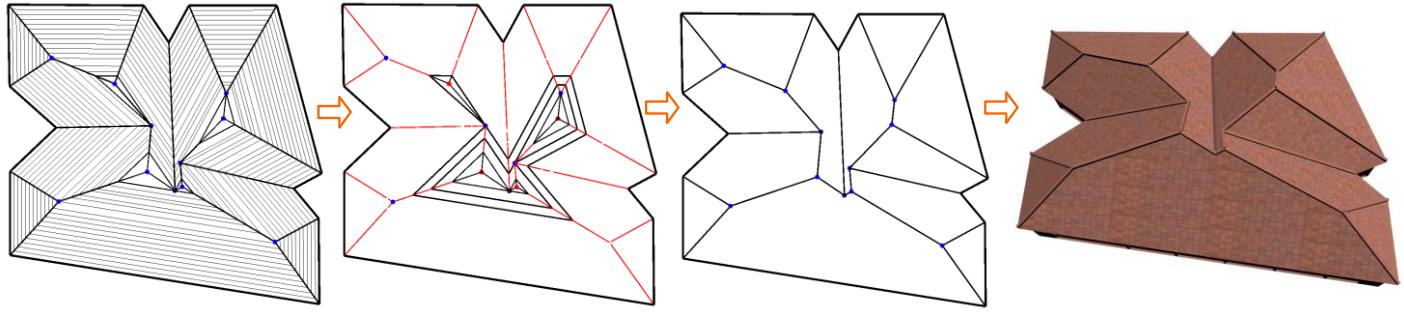
(1)屋根や建物本体、窓など建物の部品となる、適切な大きさの直方体、三角柱、多角柱などの基本立体(プリミティブ)を作成する。(2)これらの基本立体の間で、屋根や窓用に穴を開ける、または、部品を作成するためのブール演算を行う。(3)作成した部品を回転する。(4)正しい位置にそれらを配置する。(5)それらにテクスチャマッピングを施す。

このGISモジュールとCGモジュールでの処理は、本研究で開発したプログラムによって、全て自動的に処理される。

3. Straight Skeleton 手法の概要

Aichholzer ら(1995)によって提案された Straight Skeleton は、図2(a)に示すように、一般的な形状の Simple ポリゴンにおいて、ポリゴンの各辺がポリゴン内部方向に、平行に一定速度で縮小するとき、各頂点の軌跡を辿ることで得られる直線状の骨格(Straight Skeleton)として定義された。このStraight Skeleton は、図らずも、一般的な形状の建物ポリゴンに対して、図2(d)に示すように、一般的な形状の屋根を生成する手法として、適用できることが分かった(O. Aichholzer et al, 1995)。

筆者らが開発したシステムは、電子地図上の直角建物ポリゴンに対して3次元建物モデルを自動生成した。このStraight Skeleton 手法を用いることによって、建物ポリゴンが、直角ポリゴンか非直角ポリゴンかを問わず、屋根付き3次元建物モデルを自動生成することができる。但し、Aichholzer ら(1995)による Straight Skeleton 手法では述べられていない、主に直角ポリゴンの場合、想定しなければならない「同時発生イベント」、及び、縮小プロセスの最後のステージにおいて、考えなければならない「収束の形」があり、本



(a) 縮小ポリゴン群: Edge & Split イベントによるノード: 3角形が1点に収束することによるノード

(b) 縮小処理の結果、Split イベントが発生した時、分割されたポリゴンと3角形

(c) 縮小プロセスにおいて各頂点が辿る頂角の2等分線(angular bisector)の集まりとして生成される straight skeleton

(d) 自動生成された3次元建物モデル: 屋根板は Straight Skeleton によって分割された ‘interior monotone polygons’ に基づいて生成

図-2 Straight Skeleton 手法による縮小プロセスと自動生成された3次元建物モデル

研究では、これらのイベント及び収束の形を提案する。そして、あらゆる建物ポリゴンに対して一般的な屋根付き 3D モデルを自動生成するシステムを提案する。

Aichholzer らによると、図 2 (a) に示すように Straight Skeleton は、ポリゴン各辺が元の辺と平行になるような仕方で、一定速度でポリゴンの縮小処理を行うことで形成される。縮小するポリゴンの各頂点は、各頂点の頂角を 2 等分する線 (angular bisector) 上を移動する。この縮小プロセスは、次のイベントが生じるまで続く。

- (1) 辺消失イベント (Edge イベント) : 辺が縮小して消失する。消失辺の両側の辺が、以降は接することになる。
- (2) 分割イベント (Split イベント) : Reflex 頂点 (内角が 180 度以上の頂点) が辺に交差して、ポリゴンを分割する。分割されて 2 本になった辺と Reflex 頂点に付随する辺が、以降は接するという新たな隣接関係が生じる。

図 2 (a) では、Straight Skeleton 処理による等速度で縮小するポリゴン群、Edge イベントと Split イベントによるノード、3 角形が 1 点に収束することによるノードを表示している。図 2 (b) は、縮小プロセスを進める過程で、Split イベントが発生した時、分割されたポリゴンと 3 角形を表示している。図 2 (c) は、縮小プロセスにおいて各頂点が辺の頂角の 2 等分線 (angular bisector) の集まりとして生成される straight skeleton を表示している。Straight

Skeleton は、ポリゴン内部を n 個の「monotone polygons (単調変化ポリゴン)」に分割するという優れた性質を持つ (O. Aichholzer et al, 1995)。図 2 (d) は、自動生成された 3 次元建物モデルで、屋根を構成する各屋根板はこの「monotone polygons」に基づいて生成した。

本研究では、縮小処理では、上記の 2 つのイベントの他に、第 3 のイベントが生じることを提案する。この第 3 イベントは、図 3 上で示すように Reflex 頂点からの頂角 2 等分線が他の辺と交差して Split イベントが起こるが、分割されたポリゴンで、同時に Edge イベントが生じ、消失してしまうイベントである。この「第 3 のイベント (Edge & Split の同時イベント)」が起こる様子を図 3 に示す。

4. Straight Skeleton 手法の詳細とアルゴリズム

Straight Skeleton 手法を用いる縮小処理において、ポリゴンが、その各辺が、その元の辺と平行となるような仕方で、一定速度でポリゴン内部へ移動するとき、縮小処理前後の各辺の距離:「縮小距離 ($=d_{shri}$)」が縮小処理を一意に決める。縮小処理をし始めて、初めて、「Edge イベント」が生じる距離 ($=e_{d_{shri}}$) は以下の式で与えられる。

$$e_{d_{shri}} = \frac{L_i}{(\cot(0.5 * \theta_i) + \cot(0.5 * \theta_{i+1}))} \quad (1)$$

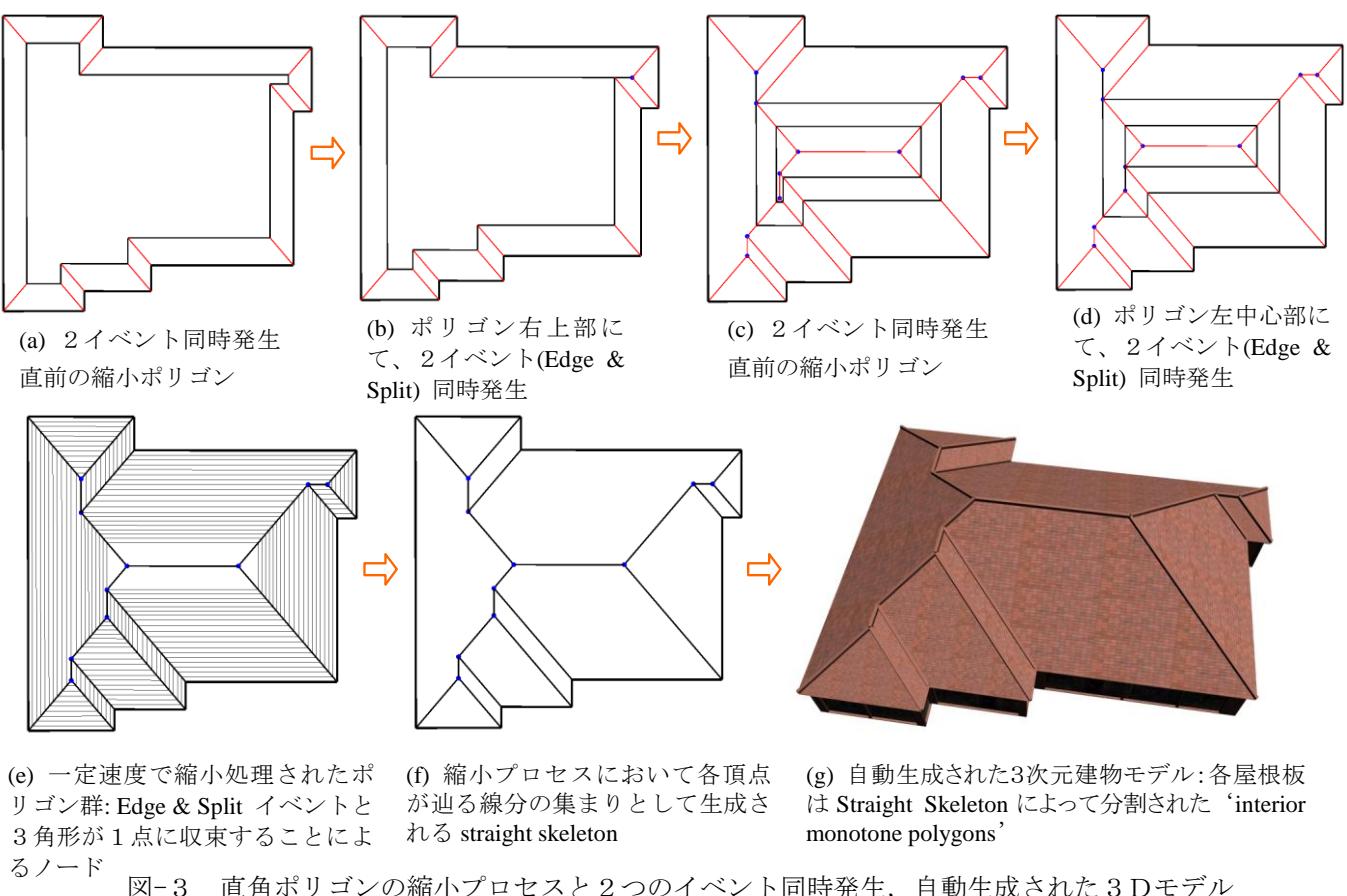


図-3 直角ポリゴンの縮小プロセスと 2 つのイベント同時発生、自動生成された 3 D モデル

ここで, L_i は辺長, θ_i , θ_{i+1} は辺両端の内角を表す. 但し, 辺両端の頂点の内角を 2 等分する線がポリゴン内部で交わるために, $0.5 * \theta_i + 0.5 * \theta_{i+1} < 180^\circ$ となる必要がある. この条件を満たす全ての辺について, d_{shri} を式(1)で求める. これらの d_{shri} の中で最小となる値が, 最初に Edge イベントが生じる縮小距離となり, その辺が最初に消失する辺となる. 本研究では, 「Split イベント」は, 縮小処理において最初に「Edge イベント」が生じるまでの「 $e_{\text{d}_{\text{shri}}}$ 」の間において, Split イベントが生じるかどうかを調べ, 生じた場合, その結果生じるノード(分割ノード)の位置を求め, ポリゴン分割することで Split イベント処理を行うこととする.

具体的には, 図 2 に示すように, Reflex 頂点を片方の頂点とする頂角の 2 等分線分(angular bisector)が, ポリゴンのどの辺と交差するかを調べる. 1 つの辺が, 複数の 2 等分線分と交差する可能性がある. その場合, 最も短い d_{shri} での 2 等分線分となる Reflex 頂点について, 分割ノードの位置を求める.

ポリゴンは, 図 2 に示すように, 複数回, Split イベントが生じて, ポリゴン分割が複数回行われ, 複数のポリゴンが生成される場合がある. 分割されたポリゴンはさらに分割される可能性もあるので, 図 4 のアルゴリズムに示すように, Split イベント処理ループでは, 全てのポリゴンについて, Split イベントが生じるかどうかを調べ, 全てのポリゴンが, さらに Split イベントが生じることがなければ, ループを抜け, 次に全ての分割されたポリゴンについて Edge イベント処理, そして, ポリゴン頂点数が 3 個なら, 3 頂点ポリゴン処理に移行するとした. 図 4 は擬似言語で書かれており, ポリゴンの縮小処理において, Split イベントか Edge イベントの各イベントが生じたとき, ノードの集まりまで縮小処理を収束させるためのアルゴリズムである. 図 4において, 最後の行の「 o_{edge} 」は, ノードを形成する少なくとも 3 つの, 縮小する前の元ポリゴンの辺(ここでは, original edge と呼ぶ)の情報である.

上記いずれかのイベントが生じ, ポリゴンはトポロジー的に変化し, ポリゴンは, 個数はそのまま 1 つか複数のポリゴンに分割される. 分割されたポリゴンが面積を持つかぎり, 縮小処理は続き, その頂点数を減らしていき, 2 点か 1 点のノードに縮小, 収束される. 全てのポリゴンが 2 点か 1 点のノードに縮小, 収束されれば, イベント処理は終了する. この場合, 本研究では, 図 3 (e) 中心部で示すように, 2 点, 即ち, 線分に収束する場合を提案する. これは, 縮小プロセスにおいて, 4 角形が, その向かい合う辺の $e_{\text{d}_{\text{shri}}}$ が, そのポリゴンにおいて最小で且つ等しい場合に「収束線分(a line of convergence)」に収束するとする.

