大規模 3 次元文化財モデルを用いた関連情報検索表示および関連付けシステム

An Associating, Retrieval, and Display System
for Archaeological Information using Huge 3D Models of Cultural Heritage

岡本 泰英[†], 大石 岳史^{††}, 正会員 池内 克史^{††}

Yasuhide Okamoto[†], Takeshi Oishi^{††} and Katsushi Ikeuchi^{††}

Abstract

Recent advancements in digital archiving technologies have enabled us to store greater amounts of priceless digital data related to cultural properties. However, accessing this information by using conventional database systems is too difficult for general users. To solve this, this paper proposes an associating, retrieval and display system for archeological information using huge 3D models of cultural heritage as interfaces to access information. To define target regions on 3D models, users can quickly select specific regions from huge models by using an interface that combines a splitting method by graph-cut and a lasso tool. In the display process, we achieved highly interactive rendering of huge 3D models by using an efficient 3D rendering algorithm on the basis of multi-resolution meshes.

キーワード: 大規模 3 次元モデル, ユーザインタフェース, データベース

1. ま え が き

近年のデジタルアーカイブ化技術の発達に伴い、文化財に関する考古学的、文化的に重要なデジタルデータが多く蓄積されるようになった。その情報の内容としては、テキストデータや、スケッチ、写真の画像データ、文化財の 3 次元モデルなど様々な種類の情報が含まれる。これらのデータは、調査活動などで専門家が用いる以外にも、観光や教育などの分野での再利用が可能であり、非常に価値が高い。

しかし、これらのデータは、一般のユーザがアクセスするのが難しいという問題を持つ。この種のデータを管理しているデータベースシステムの多くは、テキストベースで管理、検索を行うシステムであり、直感性が低く、その操作に慣れていなければ、ユーザが望む情報に効率よくアクセスすることは難しい。つまり、そのような情報へのアクセ

図 1 3 次元モデルの自由な視点・解像度でのシームレスな閲覧
Interactive viewing of huge 3D models from any view-points and with any resolution.

2009 年 10 月 23 日受付, 2010 年 8 月 19 日再受付, 2010 年 9 月 13 日採録

[†] 東京大学 生産技術研究所
(〒 153-8505 目黒区駒場 4-6-1, TEL03-5452-6242)

^{††} 東京大学大学院 情報学環
(〒 113-0033 文京区本郷 7-3-1, TEL03-5841-5938)

[†] Institute of Industrial Science, The University of Tokyo
(4-6-1, Komaba, Meguro-ku, Tokyo, 153-8505, Japan)

^{††} Interfaculty Initiative in Information Studies, The University of Tokyo
(7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan)

スには、対象物およびデータベースに関する高いレベルの知識が必要となり、専門家しか扱えないという問題がある。

図 2 3次元モデル上の特徴部分と付加情報との関連付けの利用。中段:情報を関連付けられた3次元モデル、上下段:関連付けられた情報

The usability by relating characteristic parts of 3D models and other information. The middle image: images of 3D model assigned with other information. The upper and bottom rows: attached information.

一方、文化遺産を3次元的にアーカイブ化する研究¹⁾²⁾が広く行われている。これらの研究では、形状取得に用いるレンジセンサやモデリング技術の発達により、非常に大規模かつ高精細な3次元文化財モデルの取得が可能となっている。得られた3次元モデルには、各々が意味を持つ特徴的な形状が含まれており、この形状に情報を関連付けることで、個々の情報を視覚的に管理することができる。

そこで、計測により取得した3次元モデルを、様々な情報に対するインタフェースとして利用することで、あらゆるデータを容易にかつ有効に活用するシステムを提案する。高精細な3次元モデルは、図1のように、全体から細部の形状に至るまで様々な縮尺レベルの形状を保持している。そのような階層的なデータに付加情報を記述できれば、直感的に様々なレベルの情報にアクセスすることが可能となる。

画像などの2次元データ、CADモデルなど単純な3次元モデルといった視覚的なデータを他の情報へのインタフェースとして用いることは、一般的になっている。例えば、Google Earth³⁾やVirtual Earth⁴⁾では、地図データ上の様々な地点に対しユーザが情報を付加し、アクセスすることが可能となっている。しかし、大規模な3次元モデルを対象とした場合、このようなシステムに比べそのデータ構造が複雑となり、その処理の負荷が非常に大きくなる。3次元モデルを対象としたものでは、Callieriらが大規模3次元モデル上に様々な情報を提示するシステム⁵⁾を提案している。しかし、これは情報の提示機能のみに留まり、モデル上に情報を関連付ける機能には言及していない。

これに対して本論文では、大規模な3次元文化財モデル上に、ユーザが様々な情報を自由に、かつ容易に関連付けることができるシステムを提案する。モデル上に情報を関

連付ける際、その対象としては、モデル上のある一点ではなく、図2に示すように、特徴的な形状の領域を指定することが望ましい。モデル中には、様々な縮尺レベルにおいて特徴的な形状が存在するため、点ではなく領域を用いることで、明確に対象を指定することが可能となる。

そのため、大規模3次元モデルから特徴形状の領域を効率的かつ正確に選択するために、本論文ではグラフカットを用いた直感的な選択ツールを提案する。また、3次元モデルと情報との関連付け手法、対象となる大規模な3次元モデルの階層的表現とその表示手法の実装について説明する。

本論文では2章にて、提案する3Dデータベースシステムについて説明する。特に部分的な形状に対し情報を与える際の、部分形状選択のインタフェースに関してその詳細な説明を行う。3章では、大規模な3次元形状モデルをインタラクティブに表示するための表示アルゴリズムについて解説する。4章では描画システムや形状選択インタフェースの評価を行い、5章にて結論を述べる。

2. 3次元形状と付加情報の関連付けシステム

本章では、3次元モデルに対する情報の関連付け操作において、特定の場所を容易に選択するためのツールを提案する。また、特定箇所選択や情報関連付けにおける、形状データ処理方法や情報閲覧の際の処理方法について説明する。

2.1 形状領域選択ツール

3次元モデル上の領域を指定する場合、従来の単純な選択方法では、正確な選択が難しいという問題がある。3次元領域の選択といっても、作業空間がモニター画面という2次元空間であるため、ユーザの意図する領域の選択には困難を伴う。また、複雑な形状部分を選択するのに、その境界をマウスカーソルでなぞる投げ縄ツールで行う場合には、高度な操作技術と多大な時間が必要となる。

そこで我々は、容易な操作によって3次元モデル上の特徴形状を選択できるインタフェースを以下において提案する。ここでは、グラフカットを用いることで、スクリーン上の描画画像から形状を選択する手法、およびその選択結果を3次元モデルに反映させる方法の詳細を説明する。また、このグラフカット選択手法をより直感的に用いるための、投げ縄を使ったGUIインタフェースについても説明を行う。

(1) グラフカットによる領域選択

グラフカット領域選択に用いるインタフェースで、ユーザの行う操作は、画像上で選択したい部分(前景)とそうでない部分(背景)の一部に、マウス操作でマーカを引くことである。このマーカ位置を初期状態として、グラフカットによって尤もらしい領域分割を求める(図3)。ここではこのグラフカットによる分割処理について説明を行う。

2次元画像の領域分割をグラフカットにより分割する手法はBoykovらによって提案されており⁶⁾、Lazy Snapping⁷⁾等においても用いられている。我々のシステムではこれら

図 3 Lazy Snapping による 3 次元モデルの領域選択 (白線が前景、黒線が背景を意味)

Segmentation of 3D regions by Lazy Snapping (The white marker means foreground, and the black marker means background) .

の手法を利用して、以下の処理を実装した。

我々のシステムで 3 次元形状領域の選択を行う場合、入力としてスクリーン上に描画された画像を用いる。この入力画像を Watershed⁸⁾ などの手法により予め小領域群に分割しておき、その各領域をノードとするグラフを生成する。このグラフにおいて、ユーザの引いたマーカーの下領域を前景・背景の初期ノードとし、以下のエネルギー関数 $E(X)$ が最小となるようなラベリング X を求めることで、画像を前景 (選択したい部分) と背景 (それ以外) に分ける。

$$E(X) = \sum_{i \in v} E_1(x_i) + \lambda \sum_{(i,j) \in e} E_2(x_i, x_j) \quad (1)$$

ここで $X = \{x_i | \text{前景なら } x_i = 1, \text{ 背景なら } x_i = 0\}$ とし、 v, e はグラフのノードおよびエッジ集合を表す。この式において、 E_1 は各ノードが前景または背景である場合の不自然さを表すエネルギーである。また、 E_2 は各隣接ノード間の不連続性を表すエネルギーである。 λ は適当な係数とする。

E_1 は各ノードの色や法線などのパラメータが、初期の前景及び背景集合のそれとどの程度異なるかにより評価する。具体的には、色値と法線ベクトルから求めるエネルギー E'_1 と経験的に与える重み係数 w から、次のように定義する。

$$E_1(x_i) = w_c \cdot E'^c_1(x_i) + w_n \cdot E'^n_1(x_i) \quad (2)$$

この式の E'_1 の定義では、まず初期状態が前景 ($i \in F$) もしくは背景 ($i \in B$) である場合、そのラベリング x_i が初期状態のそれと等しければ E'_1 は 0、異なれば ∞ とする。定義は以下になる。

$$\begin{cases} E'_1(x_i = 1) = 0 & E'_1(x_i = 0) = \infty & \forall i \in F \\ E'_1(x_i = 1) = \infty & E'_1(x_i = 0) = 0 & \forall i \in B \end{cases} \quad (3)$$

初期状態で前景でも背景でもないノード ($i \in U$) では、そのノードの色、法線を、初期前景・背景集合のそれと比較することで評価する。初期前景・背景の色、法線は K-means 法により 64 個のクラスターにまとめておき、前景のクラスター k における色、法線の平均を C_k^F, N_k^F 、同様に背景は C_k^B, N_k^B とし、これらを比較に用いる。ここでノード i と初期前景・背景集合との色の差は $D_c^S(i) = \min_k \|C_i - C_k^S\|_2$

とする。ここで S は F または B であり、 C_i はノード i の色の平均である。色の差は RGB 各成分の差のノルムで表す。また法線の差は $D_n^S(i) = \min_k |1 - N_i \cdot N_k^S|$ で表す。これらを用い、 E'_1 を次のように定義する。

$$\begin{cases} E'_1(x_i = 1) = \frac{d(i)^F}{d(i)^F + d(i)^B} & \forall i \in U \\ E'_1(x_i = 0) = \frac{d(i)^B}{d(i)^F + d(i)^B} & \forall i \in U \end{cases} \quad (4)$$

ここで、式中の各 $d(i)^S$ には、 E'^c_1 の計算では色の差 $D_c^S(i)$ を、また、 E'^n_1 の計算では法線の差 $D_n^S(i)$ を用いる。

次に、式 (1) の E_2 項について説明を行う。 E_2 は、前景と背景の境界となるノード間で、色、法線、深度値がどの程度異なるかによって評価する。 E_2 は次のように定義する。

$$E_2(x_i, x_j) = |x_i - x_j| \cdot \{w_c \cdot g(C_{ij}) + w_n \cdot g(N_{ij}) + w_d \cdot g(P_{ij})\} \quad (5)$$

このとき $g(a) = \frac{1}{1+a}$ であり、 C_{ij}, N_{ij}, P_{ij} は色、法線、深度の各ノード平均値の差とする。また、 w は経験的に与える重み係数である。

このエネルギー関数の最小化をグラフカットアルゴリズムにより求めることで、モデルの形状や色に基づいた、尤もらしい領域の選択が可能となる。

(2) 3 次元形状の領域抽出

前項で説明したスクリーン上での選択結果を 3 次元モデルに反映させることで、選択領域の特定を行う。3 次元モデルに対する分割処理は、Yuan らが 3 次元点群情報からグラフカットにより部分点群を抽出する手法⁹⁾を提案している。この手法では、3 次元モデルの各頂点をスクリーン空間に投影し、選択領域内にそれぞれの頂点が入るか否かを判定することで、モデル上の選択領域を特定する。

しかし、大規模なモデルのすべての頂点に対してこの処理を行うのは非効率である。また、表示中の画像が形状モデル自身によるオクルージョンなどを含む複雑なシーンであった場合、選択を意図していない隠れた形状部分が選択されてしまうなどの誤操作も起こり得る。

そこで、選択領域内にあるモデル上の部分領域を特定し、その部分の頂点にのみ上記の操作を適用する。本システムでは、3 次元モデルを階層構造に変換している (3 章にて後述)。この階層構造は、各ノードが分割済みの部分領域メッシュからなっており、モデルはこの部分メッシュ単位で表示される。そのため、表示中の部分メッシュ群を使って、選択領域の絞込みを行う。また、選択時の表示解像度に応じて選択精度を変えることができる。視点が近い時は、高解像度のデータを用いるため細かい形状が選択可能であり、逆に遠い時は粗い形状の選択が可能である。

3 次元モデルへの反映処理は、具体的には図 4 で示す手順で行う。グラフカットによって選択領域が確定した後、この選択領域内に含まれる部分形状群を絞り込む。描画の際には、スクリーンの各ピクセルがどの部分領域にあたるかを表す領域 ID 画像をオフスクリーンで描画しておき、こ

図 4 3D 形状データ選択操作 (a) スクリーン上での選択範囲決定, (b) 選択範囲内の部分形状領域群抽出, (c) 該当する部分領域群内の頂点から選択範囲内の頂点を選択
Selecting operation for 3D models (a) decision of selected area on the screen, (b) selecting partial patches in the selected area, (c) selecting vertices from those of selected patches.

図 5 階層構造に対する選択操作: 表示中の形状データ (枠内) だけでなく上下の階層のデータも選択する
Hierarchical selection: selecting not only displayed meshes (in the frame) but also upper and lower meshes.

れを用いることで選択領域内にある部分形状を容易に求め、絞り込むことが可能となる。背面となる領域は ID 画像上にも表示されないため、背面の誤選択を防ぐことができる。

こうして絞り込んだ部分領域上の頂点に、その選択箇所に割り振られた特定箇所番号を与えることで選択結果を記録する。一度の操作で選択しきれなかった場合は、視点を変更して図 4 の一連の操作を繰り返すことで、選択領域の補足、修正を行う。形状データは 3 章で後述するとおり、階層構造となっているため、図 5 で示すとおり、表示中の形状データの上下の階層の頂点に対しても選択結果を記録する。

(3) 領域選択 GUI

前景と背景にマーカーを引かせるインタフェースは、一般的な投げ縄ツールに比べ、少ない手間で選択が可能である。しかし同時に、自分の引いたマーカーからどのような選択結果が得られるか予想しにくいという欠点があり、望む結果を得るには操作を何度も行わなければならない場合がある。その一方で投げ縄ツールは、自分の操作が直接結果に反映

図 6 投げ縄グラフカット選択手法 (a) 太い投げ縄による境界の選択, (b) 投げ縄の内と外を前景・背景の初期ノードとする, (c) グラフカットにより選択領域を確定
Segmentation by Graph cut like lasso tool (a) selecting using bold lasso, (b) auto decision of initial nodes for foreground and background, (c) segmentation by graph cut.

されるため、直感性の点では優れている。そのため、ユーザ操作には投げ縄を用い、その内部の選択アルゴリズムにはグラフカット手法を用いるインタフェースを提案する。

投げ縄選択で難しいのが、境界を正確に辿る操作である。そのため、図 6 (a) のような大きさを持つ投げ縄線を用いることで、忠実に辿るのではなく、境界周辺を塗りつぶすことで境界を指定できるツールを提供する。このような太い線を使って境界付近を辿ることは非常に容易な作業である。

この投げ縄による領域選択を行った後、図 6 (b) で示されるように投げ縄線の内側に前景のマーカー、外側に背景のマーカーを初期状態として与え、グラフカットによる分割を行う。これにより、図 6 (c) で示されるような厳密な境界線による選択領域を決定する。この一連の処理により、ユーザが選びたい領域を、その境界が複雑な場合でも自動的に計算することができる。選択結果に過不足があった場合は、同様の操作または通常の投げ縄ツールによって、領域を付加したり、削除することで修正することができる。

2.2 情報の関連付けと表示

(1) 情報のモデルへの関連付け

前述のようなツールを用いて選択した形状データ上の特定箇所に対して、ユーザが自由に情報を関連付けできる機能が必要である。選択した特定箇所、関連付ける情報などは、図 7 で示すようなデータベースによって管理される。

領域を指定した場合、図 7 で示すように、特定箇所 ID (Part ID) とその領域の中心位置を、特定箇所テーブル (Identified Part Table) に登録する。同時にモデル上の選択領域内の頂点に対して、その特定箇所 ID を記録する。

ユーザは特定箇所を選択した後、そこに対して情報を関連付けることができる。与える情報としては、タイトル、概要などのテキストデータのほか、画像など様々な形式のファイルをサポートする。画像およびその他の形式のファ

図 7 3次元形状・付加情報関連付けに関するデータベース構造と関連付け情報の3次元モデルへの記録

Tables of database to associate information with 3D models, and the detail of associating procedure to 3D models.

イルの関連付けは、OSのファイルマネージャ上の該当ファイルのアイコンをドラッグアンドドロップすることで、容易に行うことができる。関連付けられたファイルは、データベースにアップロードされ、タイトルなどのメタデータと共に情報コンテンツテーブル（図7のInformation contents Table）に保存される。情報データの保存と同時に、情報と特定箇所との関連付け結果が関連付け情報テーブル（Assigned information Table）に記録される。

(2) 関連付け情報の表示

閲覧時には、関連付けられた情報は、図8で示すように、モデル上の該当部分とその領域の形でハイライト表示され、情報のタイトルや概要の記述された付箋と共に表示される。表示される情報は、表示中の3次元モデルの解像度によって変化する。画面に対して、関連付け領域の表示サイズが大きなものから優先的にハイライト表示する。これにより、図8の(a)の右で示されるように、拡大した時はより小さな領域に関連付けられた情報が表示されることとなる。

ユーザはこの表示された領域をクリックすることで、そこに関連付けられた情報に容易にアクセスできる。クリックされた領域のIDを用いて、データベースから情報が取得され、システム上のブラウザでHTML形式で表示される。ここでは、その領域に登録されたテキストや画像が表示されるほか、関連ファイルへのリンクも提示される。

これらの関連情報の編集もこのブラウザ上で行う。編集にはwiki形式のインタフェースを用い、テキストや画像・その他のファイルを自由に変更・また追加することができる。また、それら関連ファイルのアップロードもこのインタフェースによって可能である。変更されたデータの内容はデータベースに反映され、その変更履歴も共に保存される。

(a) interactive display with attached information

(b) browsing the detail of attached information

図 8 モデルから関連付けされた情報へのアクセス (a) モデルの閲覧画面。閲覧時の視点に基づき、情報が関連付けられた領域と情報を示す付箋が表示される。(b) 関連付けられた情報の詳細閲覧画面。

Access to related information from 3D models (a) interactive display of the 3D model with associated information in real time depending on the viewing scale. (b) browsing window of the detail of associated information.

3. 大規模3次元モデルの実時間描画システム

本章では、大規模な文化財3次元モデルをリアルタイムに表示するための効率的な表示手法について説明を行う。本システムにおいて、3次元モデルはユーザに情報を提示するインタフェースとして用いられるため、その表示は、ユーザの操作に従いリアルタイムに行われる必要がある。

大規模モデルのリアルタイムな表示手法は、これまでも多く提案されている。Level-of-Detail (LOD)¹⁰⁾は、解像度の異なる形状データを保持し表示には必要最小限のデータのみを用いる手法であり、現在広く用いられている。LODを発展させた手法としてProgressive Meshes¹¹⁾やAdaptive TetraPuzzles¹²⁾などが提案されている。また、LODをポイントレンダリングと組合せたQSplat¹³⁾やLayered Point Clouds¹⁴⁾も提案されている。ただし、これらの手法は表示のために特殊なデータ構造を用いるため、特徴領域の指定やアクセスなどの操作を実装することは難しい。

そこで、我々は表示以外に、これらの選択処理も効率的に行うために、形状の連続性をパラメータとした再帰的なグラフ分割によって、LODベースの階層構造を生成する。この階層構造はモデル形状に基づいて分割された部分形状から成っており、2章での選択処理にも利用できる。

図 9 LOD 階層構造の生成 左:再帰的に微小ボクセル群を 2 分割する, 右:最終的にできる分割ボクセル群

Construction of level-of-detail hierarchy left: recursive splitting of voxels by using graph-partitioning, right: results of split groups of voxels

3.1 データ構造の生成

多重解像度を持つ階層構造の生成のために, まず入力モデルの再帰的な分割を行う。ここで, 大規模な 3 次元モデルをポリゴン単位で分割すると, そのポリゴン数の多さから分割に要する処理時間と使用メモリー容量が増大してしまう。そこで, 本システムでは, 予め入力モデルをボクセル空間によって微小なメッシュ群へ分割し, この微小メッシュ単位でモデルの分割を行う。その後, 分割された各メッシュの持つポリゴンが, 一定数以下になるように簡略化する。これにより階層的なデータ構造を生成することができる。

(a) 微小メッシュ単位での再帰的分割

分割を行う前処理として, 図 9 で示すように, 入力モデルを適当な格子空間で分割し, モデルを含む各ボクセルをノード V とするようなグラフ $G(V, E)$ を生成する。この $G(V, E)$ をグラフ分割によって再帰的に分割する。ここで, エッジ E は隣接するボクセルとの間に張り, 隣接するすべてのボクセルがモデルを含まない場合には, モデルを含むボクセルで最近傍のものとの間にエッジを与える。

こうして生成されたグラフを再帰的に 2 分割することにより, 図 9 左で示されるような, 階層構造を得ることができる。分割の際に用いる各ノードの重み w_v^i には, そこに含まれるポリゴンの数 $|v_i|$ を与える。また, 各エッジには両ノードの平均法線方向の内積に比例するパラメータを重みとして与える ($w_e^{ij} = \alpha \cdot \hat{n}_i \cdot \hat{n}_j$)。グラフ分割の実装には高速な分割が可能な Metis ライブラリ¹⁵⁾を用いる。

(b) 各ノードメッシュの簡略化

再帰分割処理の後, 階層構造の各節ノードに簡略化されたメッシュを与える。この処理では, 節ノードの二つの子のメッシュを統合した後, ポリゴン数が一定数 (ここでは N_t とする) 以下となるようにメッシュの簡略化を行い, その節ノードのメッシュとする。この処理をボトムアップに行うことで, 階層構造の葉ノード以外のすべての節ノードにポリゴン数が N_t 以下のメッシュを与えることができる。

図 10 階層的形状モデルのデータ構造

Data structure for a hierarchical geometric model

簡略化は Garland らの手法¹⁶⁾により行う。この手法では, 簡略化のエッジ縮退で生じる新しい頂点位置を, エッジとその隣接面との距離の自乗和 (Quadric error metric, QEM) の最小化で求める。この QEM が小さなエッジから縮退を行うことで, 形状特徴への影響を抑えた簡略化が可能となる。この簡略化の際に用いた最大の QEM (最後に縮退させたエッジの QEM) を, 各簡略化済みメッシュの解像度を表すパラメータとして記録する。

この簡略化処理により, 根に近いほど解像度が粗くなる階層構造を生成できる。簡略化の際には, メッシュ間の境界部に接するエッジは保持する拘束をかけ, 解像度の異なるメッシュが混在しても不整合がおきないようにする¹²⁾。

(c) データ構造の生成

データ構造は, 図 10 で示すような三つの要素で構成する。ヘッダ部 (Header) には階層構造の骨格など, モデル全体に関するパラメータを記述する。ノード部 (Nodes) には各ノードの持つメッシュの中心位置, 法線範囲, 解像度, メッシュデータへのポインタなどを記述する。形状データ部 (Mesh data) には, 各ノードの持つメッシュデータ (頂点位置・接続情報) を深さ優先の順序で記述する。ヘッダ部, ノード情報部は階層構造探索などに必要な情報であり, これらのデータ容量は形状データ部に比較して非常に小さい。

3.2 描画

描画の初期動作として, 図 10 のヘッダ部およびノード部に記録されている階層構造や各ノードのパラメータの読み込みを行う。また, 形状データ部の先頭付近に記録された階層構造の上部の形状データに関しても予め読み込みを行う。

描画の際は, ヘッダ部に記述された階層構造を探索し, 適切な解像度のメッシュを選択する。メッシュの表示精度は, 簡略化の際の最大 QEM 値のスクリーン上での大きさ (QEM 値: q , メッシュと視点間の距離: r とすると $\frac{q}{r}$) で表し, この大きさが必要とする表示精度より小さければそのメッシュを表示し, 大きければ下の階層のメッシュを探索する。探索の際, そのメッシュのバウンディングボックスがスクリーン外にある場合や, 法線範囲が視線方向と交わらない場合には, それ以降の探索を省略する。

各ノードの形状データは描画時に動的に読み込まれる。読

図 11 ユーザテストでユーザに選択させる領域
The regions to be selected in the user's tasks.

込まれたデータは、キャッシュとしてメモリー上に確保され、次のアクセスでは高速な読み込みが可能となる。読み込みの際には描画されるノードのほか、次に呼び出される可能性の高い、その親・子ノードや隣接するノードも同時に読み込むことで、後の読み込み時間の短縮を図る。キャッシュ上のデータはLRU (Least Recently Used) 方式によってデータ置換管理を行う。キャッシュ済みデータ量がキャッシュサイズを超える場合には、最後の参照から最も時間が経過しているノードのデータから置換を行う¹²⁾。

4. 評価

提案システムの評価を行った。ここでは、形状選択ツールの有効性を調べるユーザテストとともに、表示における描画性能を評価を行った。

4.1 選択インタフェースの評価

領域選択手法をユーザテストによって時間と精度の評価を行った。被験者は年齢が20から40歳の8人で、このうち2人は3Dモデリングソフトなどに関して熟練しており、4人は中級レベル、2人は初級レベルである。この8人に、

- マウス操作による直接選択（投げ縄ツール）
- 前景・背景マーカによるグラフカット選択ツール
- 投げ縄式グラフカット選択ツール

の3種類のツールを用い、図11で示す3次元モデル上の10個の領域を選択するタスクを行わせた。なお実験中の習得効果による結果の変化を防ぐため、予め実験前にすべての被験者に各ツールでの練習タスクを10分間行わせた。また試行順序の違いによる結果の変化を防ぐため、実験の順序は、3名の被験者は投げ縄ツールでの選択を初めに行い、他の3名は前景・背景マーカ式グラフカット手法、残る2名は投げ縄式グラフカット手法を初めに行うこととした。

各ツールでの選択操作に要した時間の評価を行った。図12上にタスク領域ごとの平均作業時間および被験者ごとの各手法での平均作業時間を示す。投げ縄選択と前景・背景マーカによる選択手法を比べた場合、投げ縄で時間を特に要した領域（1,2,6,7）では、マーカ選択手法の作業時間が投げ縄の場合に比べ、約半分またはそれ以下という結果が

図 12 領域選択に要した時間の比較（上:領域毎, 下:ユーザ毎）
The result of consumed time for segmentation (upper:by regions, lower:by users) .

図 13 領域選択の誤差対消費時間
The result of errors to consumed time for segmentation.

得られたが、それ以外の領域に関しては、大きな差はみられなかった。一方、投げ縄式グラフカットの手法はすべての領域をおよそ半分の時間で選択できていることが分かった。

被験者ごとの結果（図12下）では、被験者1, 2が上級者、3, 4, 5, 6が中級者、6, 7が初級者という順に結果を示した。ほぼ全員が投げ縄選択よりマーカ選択手法の作業時間が少なく、投げ縄式グラフカット手法の場合は、さらに少ないという結果が得られた。ただし、上級者の被験者2は、マーカ手法よりも提案手法の方が選択時間が多く、初級者の被験者8は、投げ縄選択よりもマーカ手法の方が多かった。この結果から、上級者ほどマーカ手法の特徴をうまく掴んでいると言え、逆に言えば、マーカ手法は、やや高度な操作方法と言える。その点で直感的な操作である

表 1 3 次元モデルの生成時間と描画速度
The result of preprocessing time and rendering speed.

model	Input[faces]	Preprocess[sec]	Rendering[fps]
Mercede Church	1,148,939	340	32.8
Nara Buddha	3,109,824	818	24.7
Bayon Face	5,922,790	1641	13.8
Bayon Towers	9,343,426	3055	16.4

提案手法は、安定的に少ない時間での選択を実現できており、マーカ手法より有利であると言える。

次に各ツールでの作業時間と選択誤差の関係を評価した(図 13)。選択誤差は、正解となるタスク領域に対する、被験者が選り損なった領域と余分に選んだ領域の頂点数の割合で算出した。また、結果の時間・誤差は、領域ごとの難易度の差を考慮し、各領域での全試行の平均値を 100 として算出した。結果では、作業時間は投げ縄選択、マーカ選択手法、投げ縄式グラフカットの順で小さくなった。また選択誤差は投げ縄選択が最も大きく、マーカ選択と投げ縄式グラフカットはほぼ同じであった。

これらの結果から、提案した投げ縄式グラフカット手法が最も少ない作業時間・誤差で選択できる手法であるという結果が得られた。

4.2 描画性能の評価

サイズの異なる複数のモデルを用いて、描画性能の評価を行った。使用したシステムは CPU: Intel Pentium4 (3.2GHz), RAM: 2GB, GPU: ATI Radeon 9800XT, OS: Windows XP SP2 である。入力モデルは 100 万から 1 千万個の面数を持つ四つの 3 次元モデルを用いた。各入力モデルの面数、前処理時間、平均描画速度を表 1 に示す。

階層構造生成などの前処理に要した時間は Mercede Church (1M ポリゴン) で 340 秒であった。その他のモデルの場合も、入力モデルのサイズに比例した時間で処理が行われており、より大規模なデータに関しても現実的な時間でのデータ生成が可能である。

描画の速度に関しては Bayon Face (6M ポリゴン) の場合、平均 14fps であった。その他のモデルではこれより速い速度で描画がなされており、階層構造を用いた描画を行うことで、入力サイズが増大しても描画速度の低下はみられないことがわかる。この結果から、非常に大規模な 3 次元モデルであっても、それをインタフェースとして用いるには十分な速さで描画が行うことが可能といえる。

5. む す び

本論文では、大規模文化財 3 次元モデル上に様々な情報を容易に関連付けることができ、さらにモデルを閲覧しながら、それらの情報にアクセスできるシステムを提案した。

提案システム上では、3 次元モデル上の特定の領域に対して自由に情報を関連付け、後から 3 次元モデルを自由に閲覧しながら、それらの情報にアクセスすることが可能となっている。提案システムでは、各操作が簡単なマウス操

作によって可能であり、特にモデル上の領域の選択においては、グラフカットによる選択手法を投げ縄ツールと組合わせたインタフェースの提案を行った。また、大規模な 3 次元形状モデルを階層的な構造により保持することで、モデルの閲覧をリアルタイムに行えるシステムを提案した。ユーザテストにおいて、我々が提案した選択ツールはユーザにとって少ない作業時間および高い精度での領域選択が可能としていることがわかった。

今後の課題は、より自由にデータを提供するため、ネットワーク越しにある 3 次元モデルやデータも扱えるように拡張することである。特に、ウェブ上でこのようなシステムを実現することの有用性は、Google Earth などの人気をみても明らかであり、この実現のため、描画手法の更なる高速化とインタフェースの工夫を行いたいと考えている。

最後に本研究は、文部科学省プロジェクトの支援を受けて行われた。また、計測にご協力頂いた日本国政府アンコール遺跡救済チーム (JSA) の方々に感謝の意を表す。

(文 献)

- 1) K. Ikeuchi and D. Miyazaki, Digitally Archiving Cultural Objects, Springer (2008)
- 2) M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk, "The digital Michelangelo project: 3D scanning of large statues," Proceedings of ACM SIGGRAPH 2000, Computer Graphics, pp.131-144 (July 2000)
- 3) Google, "Google Earth," 2005. <http://earth.google.com>.
- 4) Microsoft Corporation, "Microsoft Virtual Earth," 2005. <http://www.microsoft.com/japan/virtualearth/>
- 5) M. Callieri, P. Cignoni, F. Ganovelli, G. Impoco, C. Montani, P. Pinci, F. Ponchio, and R. Scopigno, "Visualization and 3D data processing in David's restoration," IEEE Computer Graphics and Applications, 24, 2, pp.16-21 (2004)
- 6) Y. Boykov and M.P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images," Proceedings of International Conference on Computer Vision 2001, pp.105-112 (2001)
- 7) Y. Li, J. Sun, C.K. Tang, and H.Y. Shum, "Lazy snapping," ACM Transaction on Graphics (Proceedings of ACM SIGGRAPH 2004), 23, 3, pp.303-308 (2004)
- 8) L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," IEEE Transactions on Pattern Analysis and Machine Intelligence, 13, 6, pp.583-598 (1991)
- 9) X. Yuan, H. Xu, M.X. Nguyen, A. Shesh, and B. Chen, "Sketch-based segmentation of scanned outdoor environment models," Proceedings of the 2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling, pp.19-26 (August 2005)
- 10) D. Luebke, M. Reddy, J.D. Cohen, A. Varshney, B. Watson, and R. Huebner, Level of Detail for 3D Graphics, Morgan Kaufmann (2002)
- 11) H. Hoppe, "Progressive meshes," Proceedings of ACM SIGGRAPH 96, Computer Graphics, pp.99-108 (1996)
- 12) P. Cignoni, F. Ganovelli, E. Gobetti, F. Marton, F. Ponchio, and R. Scopigno, "Adaptive TetraPuzzles - efficient out-of-core construction and visualization of gigantic polygonal models," ACM Transaction on Graphics (Proceedings of ACM SIGGRAPH 2004), 23, 3, pp.796-803 (2004)
- 13) S. Rusinkiewicz and M. Levoy, "QSplat: A multiresolution point rendering system for large meshes," Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp.343-352 (July 2000)
- 14) E. Gobetti and F. Marton, "Layered point clouds: A simple and efficient multiresolution structure for distributing and rendering gigantic point-sampled models," Computers and Graphics, 28, 6, pp.815-826 (2004)
- 15) G. Karypis and V. Kumar, "Multilevel k-way partitioning scheme

for irregular graphs,” Journal of Parallel and Distributed Computing, 48, 1, pp.96–129 (1998)

- 16) M. Garland and P.S. Heckbert, “Surface simplification using quadric error metrics,” Proceedings of ACM SIGGRAPH 97, pp.209–216 (1997)

岡本 泰英 おかもと やすひで 1981 年生。2004 年東京大学理学部情報科学科卒業。2006 年同大学院情報理工学系研究科コンピュータ科学専攻修士課程修了。2010 年同大学院情報理工学系研究科コンピュータ科学専攻博士課程修了。2010 年より同大学生産技術研究所特任助教。博士（情報理工学）。大規模形状データの可視化、ユーザインタフェースに関する研究に従事。

大石 岳史 おおいし たけし 1999 年慶應義塾大学理工学部電気工学科卒業。2002 年東京大学大学院学際情報学府学際情報学専攻修士課程修了。2005 年東京大学大学院学際情報学府学際情報学専攻博士課程修了。同年東京大学生産技術研究所特任助手。2006 年特任助教を経て、2007 年より東京大学大学院情報学環特任講師。実物体の形状モデリングの研究に従事。博士（学際情報学）。

池内 克史 いけうち かつし 1949 年大阪府生まれ。東京大学大学院情報学環教授。工学博士。マサチューセッツ工科大学人工知能研究所、カーネギーメロン大学コンピュータ科学科、東京大学生産技術研究所などを経て、2000 年より現職。ロボットに目の機能を付加するロボットビジョンの研究を行ってきた。現在は、人間行動観察学習の理論を利用した無形文化財のロボットによる再現やアンコール遺跡などの有形文化財のデジタル化など「文化を科学する」研究に取り組んでいる。