

目次

第1章	はじめに	1
第2章	背景と目的	2
2.1	背景	2
2.2	関連研究	2
2.2.1	Edit-based	2
2.2.2	three-way merges	4
2.2.3	Differential Synchronization	4
2.2.4	CoMaya	5
2.2.5	WFE	5
2.3	3次元データの three-way merges への適用	7
2.4	目的	7
第3章	システム概要	8
第4章	提案手法	9
4.1	データ構造	9
4.2	固有 ID の付与	10
4.3	基本命令	10
4.4	クライアント接続時の処理	11
4.5	シャドウコピー	11
第5章	実験と考察	12
5.1	実験の目的	12
5.2	実験環境	12
5.3	実験方法	12
5.4	実験結果	12
5.5	考察	12
第6章	むすび	13
	謝辞	14

第1章 はじめに

近年, Google ドキュメント [hhhhh] や Microsoft Word Online[kkkkk] に代表されるテキスト編集を中心に, 同期編集システムの研究や開発が注目されている. 同期編集では, 同期する際に生じる編集の衝突を解消することが主な課題である. 編集の衝突を解消でき, 広く使われている手法として操作変換がある. CoMaya[lmlml] は操作変換を応用して 3D モデリングにおける同期を可能にした. しかし, [lmlml] の操作変換は, 面と頂点, オブジェクトと面などの依存関係がある要素に対する同期は不可能である. そのため CoMaya ではオブジェクト単位の同期に留まっている. より柔軟に同期編集をするには, 頂点の操作も含めた 3 次元データの同期する必要がある. Differential Synchronization[?] は, 操作変換とは別の同期手法であり, 元々はテキスト編集を対象に開発されたが, 依存関係の扱いが操作変換より容易になると予想される.

本研究では, Differential Synchronization の同期手法に基づいて, 3 次元データの依存関係を考慮した同期編集機構の手法を提案する. これにより, 複数人で同一の 3 次元データを同時に編集できる 3D モデリングソフトの開発を可能にする.

本システムはオブジェクトが複数の面から構成されるサーフェスモデルを扱う. Differential Synchronization では, 同期のため, 接続されたクライアントごとに, クライアントとサーバに, データのシャドウコピーを作る. クライアントとサーバで, 差分の計算とその適用を行い同期を行う.

本研究はデータを編集するため, クライアントのインターフェイスとして 4.3 節の基本命令を実装した. 実験のためにサーバと 3 つのクライアントを用意し, クライアントごとに任意の基本命令 50 件をランダムなタイミングで発行した. その後, 各クライアントで同期されたデータを比較した結果, データはすべて一致しており, 同期可能であることを確認できた.

本論文の構成は次の通りである. まず, 第 2 章で本研究の背景と目的について述べる. 次に, 第 3 章で本研究で提案するシステムの概要を述べる. そして, 第 4 章でシステムで利用する手法について述べる. 第 5 章で実験とその結果について述べ, 考察する. 最後に第 6 章で本研究のむすびとして今後の課題を述べる.

第2章 背景と目的

2.1 背景

近年, Google ドキュメントや Microsoft Word Online に代表されるテキスト編集を中心に, 同期編集システムの研究や開発が注目されている. 同期編集とは, 複数人で同一のデータを同時に編集できるシステムである. 複数人で行うため, 作業効率が上がることや, 他者が作っている部分を見ながら自分の作業している部分を修正できるメリットがある. また, 個々で作られたファイルをマージする手間も削減できる. 同期編集では, 複数のユーザから様々な編集が行われるため, あるユーザが最新ではないデータを編集する際に, 編集の衝突が起こる. 衝突が起こると各ユーザが扱うデータに矛盾が生じ, データの同期が失敗する可能性がある. データに矛盾が生じる例を図 2.1 に示す. 図 2.1 では, クライアント A の挿入命令をクライアント B が検知する前に, クライアント B が新たな挿入命令を送信している. よってお互いの命令を適用する際に, クライアント A が red を挿入した分の位置がずれてしまい, クライアント A とクライアント B の最終結果に矛盾が生じる. こういった衝突を解消することが同期編集の主な課題となっている. 衝突を解消するための同期機構にはいくつか選択肢があり, 後で変更することは困難なため, アプリケーション開発サイクルの早い段階で同期機構を選択する必要がある.

また, 3D モデリングにおける頂点や面などの, 3 次元データを扱う同期編集システムも研究や開発が行われている. 3 次元データを扱う際, 例えば, 面と頂点のデータ間にある依存関係を考慮しなければならない. 現在の研究では, オブジェクト単位の同期と, 依存関係のないデータのための同期編集に留まっている. より柔軟に同期編集をするには, 依存関係を考慮し, 頂点の操作も含めた 3 次元データを同期する必要がある.

2.2 関連研究

2.2.1 Edit-based

Edit-based は, ユーザの全ての操作を他のユーザに反映させる同期機構である. アルゴリズムとして操作変換 (Operation Transformation)[?] を扱う. 操作変換は, ある

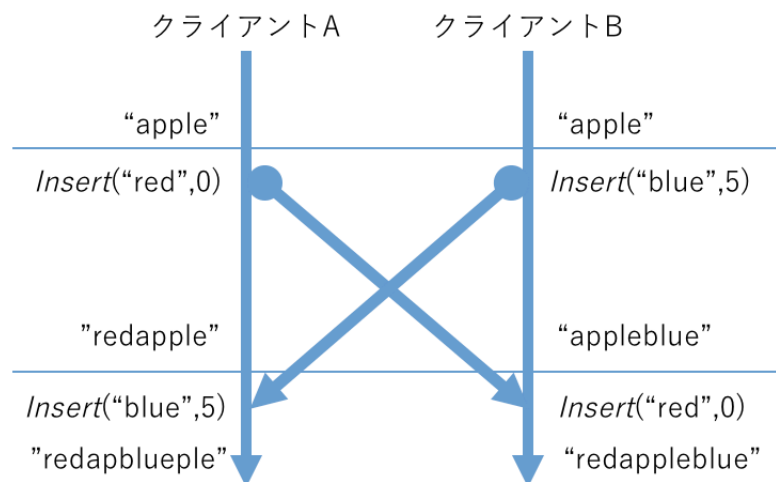


図 2.1 矛盾が生じる例

編集操作のデータを、以前に実行された編集操作の結果によって更新する仕組みである。これにより、編集の衝突によるデータの矛盾を防ぐことができる。図 2.2 は図 2.1 の矛盾を操作変換を用いて解消した例である。図 2.1 では、クライアント A が red を挿入した分の位置がずれるが、クライアント B の操作に対応していなかったが、操作変換によって図 2.2 にあるように red の文字数分、適用の際にクライアント B の命令のパラメータを更新する。これによりクライアント A とクライアント B の最終結果が一致する。

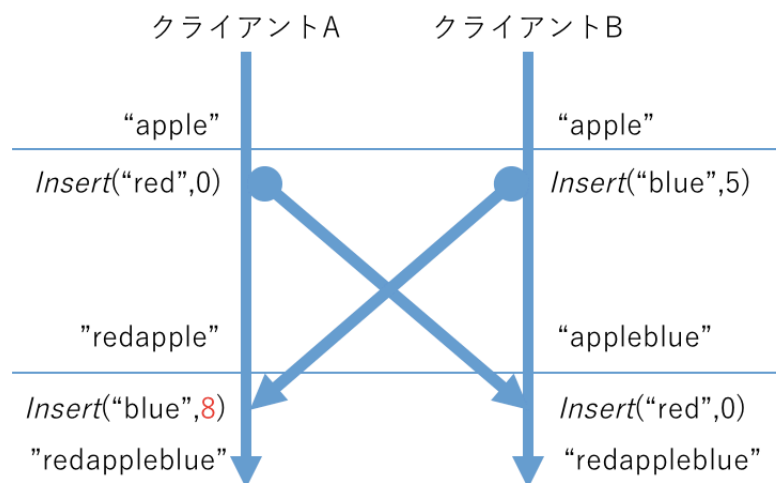


図 2.2 操作変換

2.2.2 three-way merges

three-way merges は3つのデータを、サーバでマージし最新のファイルを作る同期手法である。three-way merges の流れの概略図を図 2.3 に示す。図 2.3 のように、あるユーザの編集と、そのユーザ以外のユーザによる編集、さらにベースとなる元ファイルのデータを、サーバでマージし最新のファイルを作る。その最新のファイルをユーザに送信し同期を行う。ベースとなる元ファイルの存在により、ユーザによる最新ではないデータに対する編集が行われても、データの矛盾は生じない。

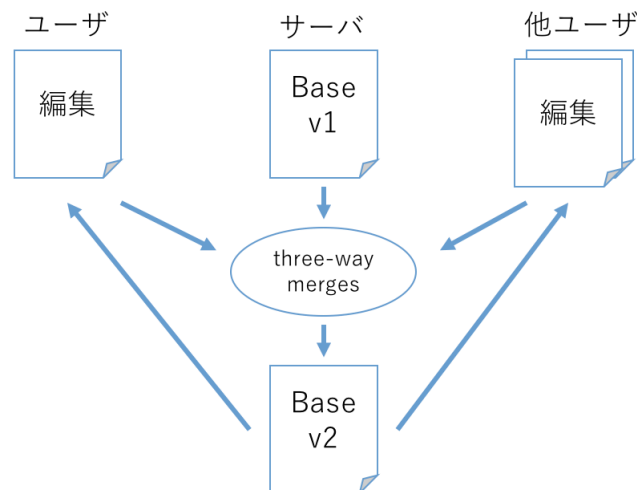


図 2.3 three-way merges

2.2.3 Differential Synchronization

Differential Synchronization は差分の計算とその適用によって同期を行う。図 2.4 は Differential Synchronization の Dual Shadow Method の同期の流れである。なお、図 2.4 中の各番号は処理の順番を表している。Dual Shadow Method は、クライアントデータとサーバデータの他に、クライアントとサーバに各データのシャドウコピーを用意する。処理の流れとしては、まずクライアントデータとそのシャドウの (1) 差分を計算しサーバに (2) 送信する。次に、クライアントでは (4) シャドウコピーを行い、サーバで (3) 適用処理を行う。サーバ側でも同様の手順で処理することで同期が可能となる。サーバデータは複数クライアントから利用され、クライアントデータと 2 つのシャドウは接続しているクライアントごとに用意される。2 つのシャドウコピーは各適用処理の後、必ず一致する仕組みになっている。編集が衝突し、データに矛盾が生まれた場合でも、それぞれ差を計算し適用することによって各データが同一のものに収束する。

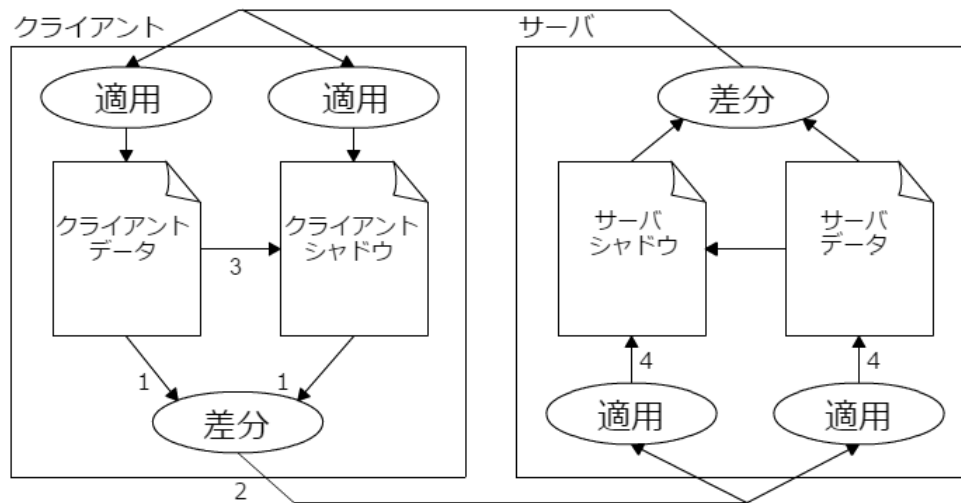


図 2.4 Differential Synchronization

2.2.4 CoMaya

CoMaya は 3D 統合ソフト Maya の API を利用し, 3D モデリングにおける同期編集を可能にしたものである. 図 2.5 は CoMaya のシステム図であり, 図中の (a) は Maya のインターフェース, (b) はオブジェクトと Maya の API の関係のグラフを表している. (c) は CoMaya によって拡張されたオブジェクトのデータと Maya の API を一次元に管理し, 操作変換を適用して同期を行う部分である. 一次元に管理するため, オブジェクトと面や, 面と頂点間の依存関係を解決できておらず, オブジェクトの位置やマテリアルの同期に留まっている.

2.2.5 WFE

WFE[?] は Web ページの同期編集システムである. ある時点でのサーバデータにおけるスナップショットと各ユーザによる編集データを用いる同期機構を採用している. Web ページの HTML における DOM 要素を編集単位として同期を行なった. WFE は, スナップショットがベースとなり元ファイルがあるので, three-way merges と同様に, ユーザによる最新ではないデータに対する編集が行われても, データの矛盾を起こさずユーザ間で同期を可能にした.

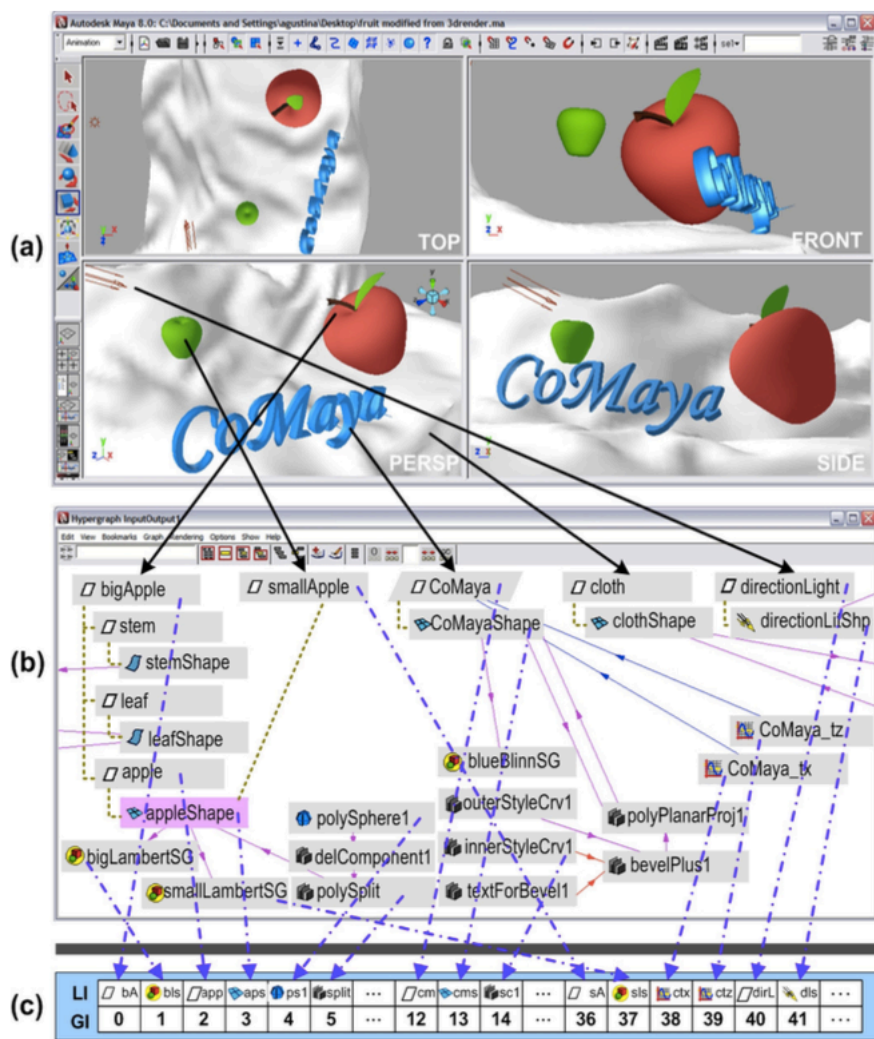


図 2.5 CoMaya のシステム図

2.3 3次元データの three-way merges への適用

WFE を参考に 3 次元データの three-way merges への適用を試みた。システムは、頂点編集が可能なクライアントと、編集のデータを管理するサーバプログラムによって構成した。ユーザの操作ごとにクライアントが編集のデータをサーバに送信し、サーバはその時点でのスナップショットと各ユーザの編集をマージした。クライアントは一定時間ごとにデータをサーバから受信し適用した。これらの手順よりクライアント間でデータの同期を行なった。その結果、各ユーザのデータは一致し同期可能であった。しかし、スナップショットをポーリングにより取得するため、頂点が増えデータの容量が増大する可能性がある 3 次元データでは対応しきれない。

2.4 目的

本研究では、依存関係の扱いが操作変換より容易になり、送受信の容量が少なくて済むと予想される Differential Synchronization の同期手法に基づいて、3 次元データの依存関係を考慮した同期編集機構の手法を提案する。これにより、複数人で同一の 3 次元データを同時に編集できる 3D モデリングソフトの開発を可能にする。

第3章 システム概要

本システムはオブジェクトが複数の面から構成されるサーフェスモデルを扱う。また、同期手法としてDSを採用し、サーバデータとそのシャドウコピーを管理するサーバプログラムによって構成される。第4章で用いる手法を用いて、依存関係があるデータを編集する場合でも矛盾が起きない同期を行う。

第4章 提案手法

4.1 データ構造

サーバのデータを従来のテキストと同様に一次元に管理すると、依存関係があった場合にデータの管理が困難となる。3Dモデリングで用いるエンティティごとにデータを作成し、そのデータモデル間で依存関係を持つことによって解決する。本システムでは3Dモデリングで用いるオブジェクト、面、頂点の3つのデータを定義し、新しくデータを作成する場合、データベースにデータを登録していく仕様にした。また、サーバデータとそのデータを各クライアントにコピーしたシャドウコピーを区別するために、シーンというデータを定義する。図4.1にデータモデル間の依存関係を表した図を示す。図4.1のようにシーンのデータはオブジェクトを、オブジェクトは面

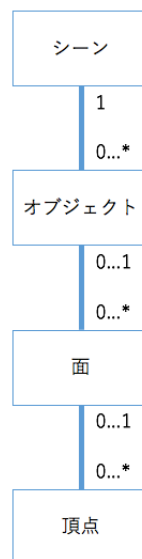


図 4.1 データモデル間の依存関係

を、面は頂点をそれぞれ子にもつ。また、子のみが親の参照先をもつ。

このデータ構造によって、子を削除した場合に親との依存関係も削除できる。親を新しく設定する場合、子のデータを複製しながらそれぞれに新しく設定する親を参

照先に設定する。子のデータを複製していくことによって、関係を削除した際も複製元のデータは残る。

4.2 固有 ID の付与

各データには、複数クライアント間で ID の衝突が起こるのを防ぐため、そのデータを作成したクライアント識別子を組み込んだ固有 ID を与える。本システムでは図 4.2 のように、クライアント識別子-データモデル識別子-インクリメント番号-ランダム文字列の順につなげた固有 ID を与えた。クライアント識別子は 1 から順に振られたユーザ ID を用いてクライアントが付与するように決め、初期データはシステムが作ったということで 0 を与えた。また、オブジェクトは 0、面は 1、頂点は 2 ということでデータモデル識別子を与えた。サーバでもクライアントでも作ったデータの数を記憶しておき、作るごとにインクリメントするインクリメント番号を固有 ID に組み込んだ。

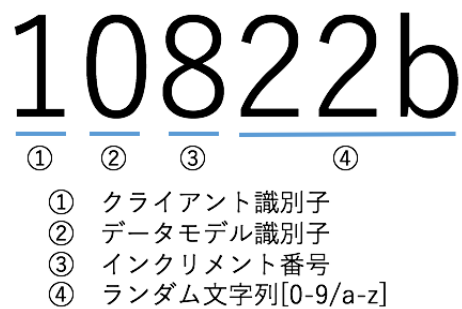


図 4.2 本システムの固有 ID

4.3 基本命令

オブジェクト、面、頂点の各データモデルに対して、作成、親への参照の追加、削除の 3 つの基本命令を実装した。これらの命令はシステムに対する最低限の基本命令であり、これらの命令を組み合わせることで、3D モデリングで使われる面の分割や押し出しなど、より高度な命令を実現できる。削除命令や親への参照の追加をする命令を適用する際、操作対象が他の命令で削除されていた場合は命令を無効とする。

4.4 クライアント接続時の処理

あるクライアントが初めて接続した場合、そのクライアントの識別子を持ったシーンのデータを作成する。このシーンのデータが、Differential Synchronizationにおいてそのクライアントのサーバシャドウになる。このあるクライアントが初めて接続した際、サーバデータとの差分を出すために、サーバシャドウはオブジェクトの情報を持たない。サーバデータとの差分を計算した後、シャドウコピーを行う。

4.5 シャドウコピー

本来、変数 $v1$ を変数 $v2$ にシャドウコピーするならば $v2 = v1$ のように代入することで実装する。しかし、本システムではデータをデータベースに登録して管理しており、単純な代入で処理できない。また、元のデータを削除し新しくシャドウコピーを作成する方法は、頻繁にシャドウコピーを行う DS では、データベースの DELETE 命令と CREATE 命令を頻発させることになり、処理においてボトルネックとなってしまう。そこでサーバデータとサーバシャドウの差分を取り、サーバシャドウに適用することによってシャドウコピーを作り解決する。

第5章 実験と考察

5.1 実験の目的

Differential Synchronization に基づき, 本研究での提案手法による同期機構で3次元データは同期可能であるかどうか確かめるため実験を行う.

5.2 実験環境

データを編集するため, クライアントのインターフェイスとして4.3節の基本命令と, サーバと同様に4.2節の固有IDを付与した. 本実験で利用したサーバの計算機環境を表??に示す. また3つのクライアントの計算機環境を表??, 表??, 表??に示す.

5.3 実験方法

実験のためにサーバと3つのクライアントを用意し, クライアントごとに任意の基本命令50件をランダムなタイミングで発行した.

5.4 実験結果

5.5 考察

第6章 むすび

一般的なサーフェスモデルの3Dモデリングソフトでは面の表と裏する仕様がほとんどである。現在の実装では、面を構成する頂点の順番を扱っておらず、面の法線方向を表現できていないため、3Dモデリングの同期機構としては不十分である。また、オブジェクトのマテリアル情報やテクスチャ座標情報も扱っていない。これらを実装することで、クライアントがデータから一意なオブジェクトを描画することができ、同期編集可能な3Dモデリングを実現できる。

謝辞

本研究を進めるあたり,終始御指導下さいました乃万司教授および山本邦雄助教に心から御礼申し上げます.そして,様々な点で研究をご援助下さいました松本隆二技術職員をはじめ知能情報メディア部門のみなさまに感謝の意を表します.最後に,常日頃よりご指導を頂き,研究を見守ってくださったのま研究室の皆様に心より感謝の念を表し,謝辞とさせていただきます.

参考文献

- [hhhhh] Google ドキュメント, https://www.google.com/intl/ja_jp/docs/about/, *google*.
- [kkkkk] Microsoft Word Online, <https://office.live.com/start/Word.aspx>, Microsoft.
- [lllll] Agustina, et al., “CoMaya: Incorporating Advanced Collaboration Capabilities into 3D Digital Media Design Tools,” *Proc. CSCW '08*, pp. 5–8, 2008.