
Twister: コンパクトなデータと高速表示を実現した3Dビューワー

Twister: A 3D Model Viewer for Rapid Display with Compressed Data

松岡 司* 香川 正明* 宮澤 明弘* 新垣 恒治* 坂本 健*
Tsukasa Matsuoka Masaaki Kagawa Akihiro Miyazawa Kohji Aragaki Tsuyoshi Sakamoto

安中 英邦*
Hidekuni Annaka

要 旨

大規模アセンブリモデルのネットワーク利用を可能にするため、コンパクトなデータサイズと高速表示を実現した3Dモデルビューワー、Twisterを開発した。Twisterは、CADシステムから取得した形状データと属性データから、表示用データを生成し、統合幾何テーブル方式を用いて圧縮した後、専用形式のTwisterファイルに書き出す。TwisterファイルはCADデータ交換の標準ファイルフォーマットであるIGESファイルに対して1%の圧縮率を達成し、ネットワーク転送時の負荷を大幅に低減する同時に、一般的なCADシステムに対し初期表示までの時間を約30分の1とした。Twisterの概要や技術、および機能を紹介し、他システムとの比較を行う。

ABSTRACT

We have developed a new 3D model viewer, Twister, which realizes compact data storage and rapid display for using highly complex assembly models in the network environment. Twister distills the data for displaying and evaluating model shapes from existing CAD files and converts them into native Twister files with our new data structure - Unified Geometry Table (UGT). UGT enables Twister to compress 3D data at a remarkably high ratio. As a result, a Twister file is about one hundredth the size of its corresponding IGES file and Twister is about 30 times faster than a typical CAD system in reading the file and displaying it.

* 画像システム事業本部 ソフトウェア研究所
Software Research Center, Image System Business Group

1. 背景

近年、コンピュータを利用した設計・製造において、3Dモデルの重要性はますます高くなっている。設計部署や製造部署においては、早くから3D CADシステムが導入されたこともあり、3Dモデルは設計・製造の効率化に寄与してきた。また、設計部署だけでなく、企画、デザイン、組立、保守などの製品に間接的にかかわる周辺部署においても、3Dモデルを共有し活用したいという要望が高くなっている。

しかし、3Dモデル利用の問題点として、以下が考えられる。

- ・ 巨大なモデルサイズ
- ・ 遅い表示速度
- ・ 難しい操作性
- ・ 高額なライセンス料金

実際の3次元設計においては、複雑で巨大な3次元モデルが要求され、その大きさは3D CADでは製品全体を表すフルアセンブリモデルで数百メガバイトから数ギガバイト程度のサイズになる。コスト低減と性能向上が進んでいる今日のハイエンドPCを利用しても、その巨大なデータサイズゆえにファイルを読み込んでから表示するまでの速度（初期表示速度）は数分から数十分の時間を要する。さらに、データ共有にはネットワークが不可欠であるが、一般的に普及している10Mbpsから100MbpsのWAN(Wide Area Network)やLAN(Local Area Network)には、3Dモデルのデータサイズは大きすぎて、初期表示速度のさらなる低下を招いている。

また、3Dモデルを利用するためには、3D CADを導入するのが一般的であるが、3D CADは形状入力を行う設計の専門家の道具であり、形状を見て確認することが主目的の設計部署以外の人々にとって3D CADシステムの操作はあまりに複雑であり、不要な部分が多くなる。また、その高機能性、多機能性ゆえに開発コストやサポートコストがライセンス料金、サポート料金に反映され、設計以外の多くの部署の人員に供給する経費コストは膨大なものとなる。

そこで、我々は、コンパクトな表示専用データ、高速な初期表示、表示機能と評価機能のみの簡単な操作系を実現した3DモデルビューワーTwisterを、シグマM推進グループ**

の依頼を受け、開発した。3D CADシステムのデータから表示用のポリゴンデータと計測に必要な曲線や曲面などの属性データだけを抽出し、専用の圧縮フォーマットのファイルに変換することにより約1%のコンパクトなデータサイズと約30倍の初期表示速度を実現した。また、形状の表示だけでなく、属性データも盛り込むことにより、計測などの形状評価作業もCADシステム上と同様に可能にした。

2. システム概要

Twisterは2つのプログラムViewerとCompilerから構成される(Fig.1)。Viewerはアセンブリを含む3Dモデルを高速に表示し、様々な表示方法やモデルの移動や計測、注記等の機能を提供し、3Dモデルの形状評価を行う。また、Compilerは、CADファイルを圧縮しながらTwister用の独自形式のファイル(Twisterファイル)に変換する。Compilerは、個々のアセンブリ、パーツ毎に独立したファイルを生成する。Twisterでは3Dモデルの形状生成や形状修正は行わずに、Compilerが生成したTwisterファイルをそのまま利用する。

3Dモデルをネットワークを介して共有する場合、そのサイズは重要な問題である。例えば、3D CADにより複写機やプリンタなどの製品全体をモデリングすると、ファイルサイズは数百メガバイトから、時には1ギガバイトを超える値になる。これだけ巨大なファイルでは、ネットワークでの共有は現実的ではない。Twisterでは、ViewerとCompilerという2つの構成にすることで、この問題を解決する。

Compilerは、CADファイルからアセンブリの構成情報や表示用データと計測時に必要な曲面情報等を抽出し、圧縮することで非常にコンパクトなファイルを生成する。これにより、Viewerは、ネットワークを介して3Dモデルのファイルを取得する際でも、初期表示速度を大幅に短縮できる。

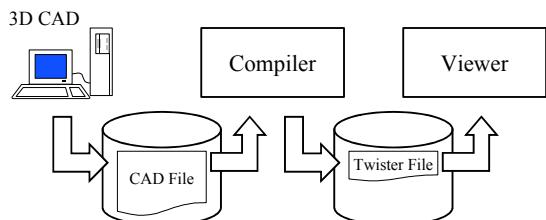


Fig.1 System Overview

Twisterを利用した3Dモデルの全体的な流れは次のように

** 画像システム事業本部 QAセンター 開発プロセス革新室 シグマM推進グループ

なる(Fig.2). 設計者はCADにて形状を入力後, 製品情報を管理するPDM(Product Data Management)システムに登録する. PDMが稼動するサーバー上では, Compilerにより自動的にCADファイルをTwisterファイルに変換し, サーバーに登録する. 3Dモデルを評価したいユーザーは, サーバーよりTwisterファイルを取得し, 表示する.

さらに, Viewerで表示しているモデルから, PDMの情報にアクセスし, 材質や仕入先, コストなどのモデルに関する様々な情報にアクセスすることができる. このように, Twisterは, CADとの利用だけでなく, PDMと連携して利用することで, Viewerを3Dモデルのポータルとすることができます.

設計中は, CADモデルの設計変更は頻繁に行われる. そのため, サーバーに登録するCADファイルは, アセンブリ, パート毎に独立したファイルとして構成し, Twisterファイルも, それぞれアセンブリ, パートに対応したファイルとして生成される. 設計変更では, 変更されたパートファイルやアセンブリファイルのみをCompilerにて変換するだけで済む.

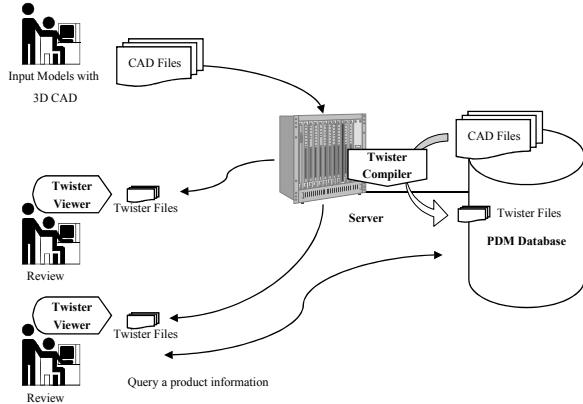


Fig.2 Flows of 3D Data

3. ファイル構成

TwisterファイルはViewerで表示するためのアセンブリファイルとパートファイルの集まりである. Compilerは3D CADシステムから, 個々のモデル別に情報を取得し, それぞれのファイルに格納する. アセンブリモデルについては, 子モデル構成, 配置行列, 表示色を取得・格納する. パートモデルについては, モデル形状, 曲線, 曲面やその他の属性を取得・格納する. アセンブリはFig.3に示すようなツリー構造を成しているが, そのツリー構造はアセンブリファイル

が階層直下の子モデルのファイル名を保存することにより, 実現されている.

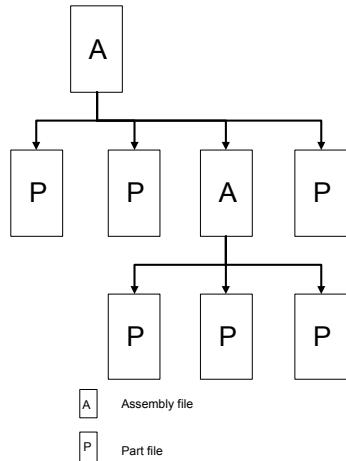


Fig.3 Assembly node(file) tree

3-1 アセンブリファイルのデータ構造

アセンブリモデルのデータ構造をFig.4(a)に示す. ヘッダと子モデル情報配列からなる. ヘッダには, アセンブリファイルであることを示すファイルフラグ, ノード名, 作成日付, バージョンがそれぞれ格納されている. 子モデル情報には, 子モデルがアセンブリかパートのいずれかを示すモデルのタイプ, ノード名, 配置行列, 表示色がそれぞれ格納されている. 配置行列は親のアセンブリの座標系内における子モデルの位置と姿勢を表す4行4列の行列である. 表示色にはOpenGL (標準的な3D表示ライブラリ) でシェーディング表示する際に用いられる環境光反射成分, 拡散反射成分, 鏡面反射成分, 透過率などのパラメータが収められている. 下位のノードに表示色の設定がなければ, 上位のノードの表示色が使われる. 上位下位の両方のノードに表示色の設定がある場合には, 下位のノードの表示色が優先される.

3-2 パーツモデルのデータ構造

パートモデルのデータ構造をFig.4(b)に示す. ヘッダ, UGT(Unified Geometry Table)と呼ばれる幾何テーブル群, 形状を表すモデルテーブル群, 面, 稜線, 寸法, 注記の各属性テーブル群から構成されている. ヘッダには, アセンブリファイルと同様のパートファイルであることを示すファイルフラグ, ノード名, 作成日付, バージョンに加えて, 境界箱

(形状を包み込む大きさの直方体), 表面積, 体積, 重心, 慣性モーメントの各属性がそれぞれ格納されている。

UGTには, 形状および属性の区別なく幾何要素が, 座標テーブル, ベクトルテーブル, スカラーテーブルに分けて格納されている。座標テーブルに関しては, さらに境界箱内部の高精度, 境界箱内部の低精度, 境界箱外部の座標に, ベクトルテーブルに関しては高精度, 低精度に分かれている。このように細分化することで, 後述する量子化処理において量子化ビット数を小さくしデータサイズをコンパクトにできる。

モデルデータとしては, 高LOD (Level Of Details:ポリゴンメッシュの詳しさを表す指標) のポリゴンメッシュ, 低LODのポリゴンメッシュ, ワイヤーフレームがあり, それぞれUGTの座標テーブルへのインデックス情報が格納されている。ポリゴンメッシュおよびワイヤーフレームは, CADシステムの曲面と曲線を, それぞれ多角形と線分により近似したものである。ポリゴンメッシュは近似の度合いにより, 高LOD, 低LODがある。インデックスの格納方式としては, ポリゴンメッシュについては三角形ストリップ形式¹⁾, ワイヤーフレームについてはラインストリップ形式¹⁾が用いられており, それぞれ, 三角形や線分間で頂点の共有を行い, よりコンパクトなデータサイズを実現するものである。

面, 棱線は, 3D CADシステム上の形状を構成する要素である。それぞれ, 属性として曲線, 曲面情報を保持している。Twisterファイルで保持する曲線, 曲面は, Viewerにおける計測機能で重視されるものに限っており, Fig.5に示す。寸法, 注記は, 3D CADシステム上の属性を構成する要素である。それぞれ, 表示するための座標, 文字情報などを保持している。

ポリゴンメッシュやワイヤーフレームを表現するのに用いられるインデックスなどのすべての整数型データは, ここまでで説明したファイル内の要素別に, その最大値を表現するだけのレコード長に切り詰めてTwisterファイルに格納する。また, 同一レコードや同一パターンの繰り返しが予測される箇所では, RLE(Run Length Encoding)圧縮を行う。多くの浮動小数点型データは, 4-2節で述べる方法により量子化されて格納される。

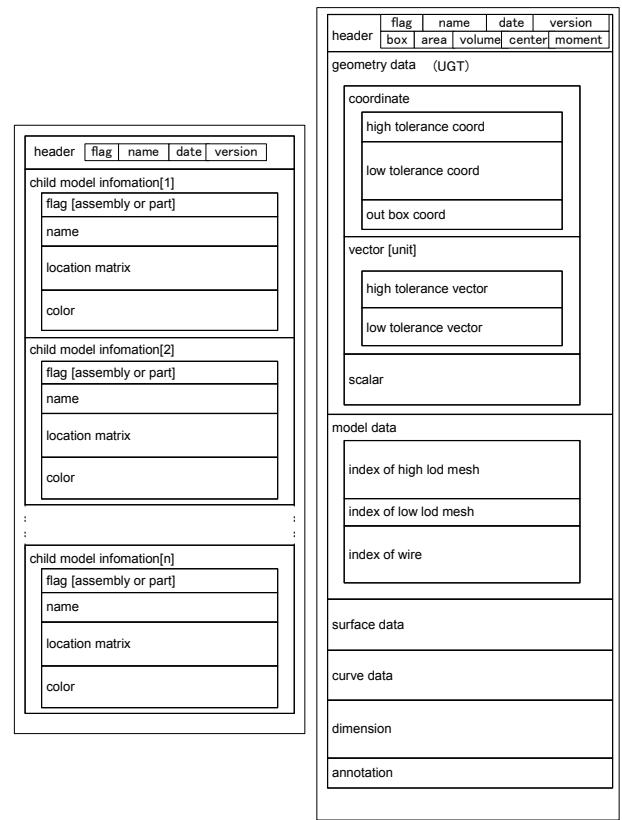


Fig.4 Data structure of Twister file

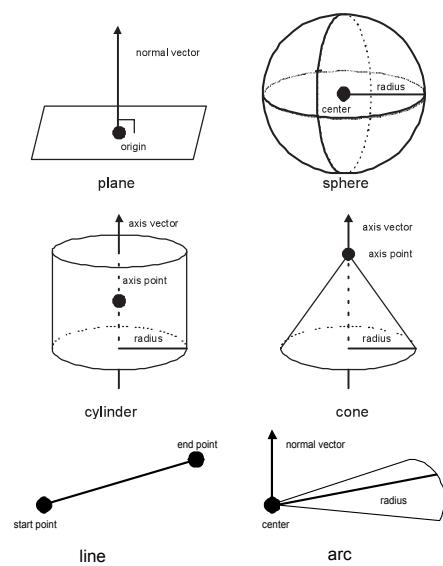


Fig.5 Surfaces and Curves

4. Compiler によるファイル生成

Twister Compiler で行う処理の概要を示す。

- (1) 3D CADシステムのAPI利用により情報取得
- (2) ポリゴンメッシュ簡単化により低LODポリゴンメッシュ生成
- (3) UGT生成
- (4) ファイルにデータを書き出す。

ここでは、Twisterの特徴である(2)と(3)の処理を説明する。

4-1 ポリゴンメッシュ簡単化

ポリゴンメッシュ簡単化の方法は、過去に多くの手法が提案されているが、代表的なものとして、処理速度の速さとポリゴンメッシュの品質を両立したQEM(Quadric Error Metrics)²⁾を利用したものが多い。しかし、QEM利用の簡単化手法のデメリットとして、ポリゴン数などの要素数を指定することは可能であるが、大域的な誤差指定を行うことができない。簡単化前のメッシュとの面数比を一定することで、複数のモデルのLODを一定する考え方もあるが、個々のモデルで構成面数にはばらつきがあるので、必ずしも一定のLODになるとは限らない。

従って、QEM利用の簡単化手法では、複数モデルに対し、一定のLODに揃えるためには、一度にまとめて、ひとつのモデルとして、ポリゴンメッシュ簡単化を行う必要がある。しかし、このような処理は、すべてのモデルを同時にメモリ上にロードして一気にメッシュ簡単化を行うことになるので、非常に多くのメモリが必要となり、仮想メモリを用いることになるため、処理に時間がかかる。また、Twister の運用上、アセンブリの中でも、設計変更や新規に追加されたモデルだけが、Compiler により処理されるため、常にアセンブリモデル全体を一度にまとめて簡単化処理できるわけではない。

そこで、アセンブリモデルのポリゴンメッシュ簡単化にQEMを適用するために、Garlandらの手法²⁾に対し、大域的誤差制御ができるように拡張を施した。Garlandらの手法の手順を示す。

- (1) 積線の端点または近傍に存在する頂点ペア v_s, v_e に対し、頂点結合操作(Fig.6)した場合のコストと最適化頂点位置 v をQEMを用いて計算する。

- (2) コストの小さい順に、頂点ペア v_s, v_e に対し頂点結合操作を繰り返し、要素数を減らしていく。頂点結合操作のたびに、変化した頂点ペアのコストは再計算を行い、更新する。

我々の拡張では、大域的な誤差指定を行うため、Garlandらの手法におけるコスト値として、(1)式に示すような最適化頂点と頂点のまわりの面の距離の最大値を採用した。Garlandらの手法ではコストの小さい順に頂点結合操作を行うため、前述の距離の最大値であるコストが予め指定した誤差値を越えないような範囲で頂点結合操作を行うことにより、大域的誤差の制御が可能になる。頂点結合操作で頂点ペア v_s, v_e が、最適化頂点 v に結合されるときのコスト値は、

$$c(v, v_s, v_e) = \max_{f \in \{fa(v_s), fr(v_s), fa(v_e), fr(v_e)\}} (a_f x + b_f y + c_f z + d_f) \quad (1)$$

となる。ここで、 a_f, b_f, c_f, d_f は、三角形 f が簡単化前の初期状態での面をあらわす係数である。 x, y, z は、最適化頂点の座標である。 $f_a(v)$ は頂点 v に隣接する面の集合を、 $f_r(v)$ は頂点 v の頂点結合操作にかかわり除去された面の集合を、それぞれ表す。

頂点結合操作の際には、削除面集合は(2)式に示すように、最適化頂点 v の削除面集合として保持しておく。

$$f_r(v) = f_r(v_s) + f_r(v_e) \quad (2)$$

以上により、QEM利用のポリゴンメッシュ簡単化手法における大域誤差制御が可能になった。

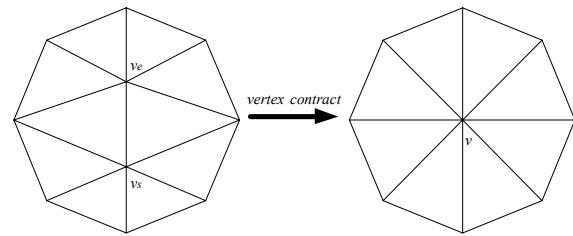


Fig.6 Vertex contract operation

4-2 Unified Geometry Table 生成

Twister では、幾何データとして表示用のポリゴンメッシュで使われる座標、ベクトル以外に、属性要素の座標、ベクトル、スカラも多くの扱うため、それらの圧縮手段も必要である。また、属性要素の幾何データは表示用の幾何データと異なり高い精度が要求されるため、Twister ファイルに占め

る割合は大きくなる。

近年、ポリゴンメッシュやその属性の圧縮の研究^{4,5)}が盛んであるが、多くの場合、形状が複雑で密度の高いポリゴンメッシュに対して効果が高いものである。一方、CAD から生成したポリゴンメッシュの要素数は必要最小限に最適化されており、密度が低い場合が多いため、高い効果は見込めない。また、高圧縮率を得るための複雑なアルゴリズムは、伸長時の処理速度も遅くなる傾向があるので、Viewer のような応用には適していない。

そこで、伸長時の速度面でも有利な圧縮手法である幾何テーブルを用いた幾何要素の共有を次のように拡張し、より圧縮効果を高めて、用いることとする。

- ・統合化共有：表示用の形状データや計測用の属性データなどの用途に関わらず、基本的な幾何要素である座標、ベクトル、スカラにわけて統合して共有する。用途による区別を撤廃することで、より高い圧縮効果を得る。
- ・精度管理：精度に対する制限は存在するため、高精度、低精度の幾何テーブルを座標、ベクトルごとにわけて管理する。但し、低精度の幾何要素と高精度の幾何要素との間で共有化を行い、共有できたものは高精度の幾何要素のほうを残すようとする。このようにすることにより、低精度の幾何テーブルのサイズをより小さくすることができる。

以上を考慮したUGT の構成をTable 1に示す。UGTは、6つのテーブルからなる。幾何要素はもともと精度の高い浮動小数点形式(float)であるが、さらにサイズをコンパクトにするため、上限、下限が明確なモデル境界箱と単位ベクトルについては、量子化形式(quantized)により格納されている。量子化形式とは、上限と下限の間で正規化し、精度に応じて不要桁を削除したものである。

Table 1 Unified Geometry Tables

table	type	content
c_h	quantized	モデル境界箱内部の高精度座標
c_l	quantized	モデル境界箱内部の低精度座標
c_o	float	モデル境界箱外部の座標
u_h	quantized	高精度の単位ベクトル
u_l	quantized	低精度の単位ベクトル
s	float	スカラ

5. Viewer による3Dモデル表示

Twister Viewerは、高速な3D表示により、3Dモデルの形状評価を支援するツールである(Fig.7)。Viewerの特徴は、次の通りである。

- ・視点操作
- ・表示方法
- ・計測
- ・モデル移動
- ・注記
- ・アニメーション

インタラクティブな操作感はViewerツールにおいて必須条件である。そのため、LOD、Culling(裏側を向いた要素や視野に入らない要素を表示しないこと)を利用している。LODは、高・低の2つのレベルを使用し、通常の静止状態では高LOD、ダイナミックな視点移動状態では低LODへと、自動的に切り替えを行うことにより、大規模なモデルにおいてもインタラクティブな操作感を実現している。

視点操作では、目の前にあるモデルを見るような操作感を提供する。そのほか、回転、パン、ズーム、注視点設定、フィット、軸回りの回転、6面図切り替え、視点操作履歴を用意した。

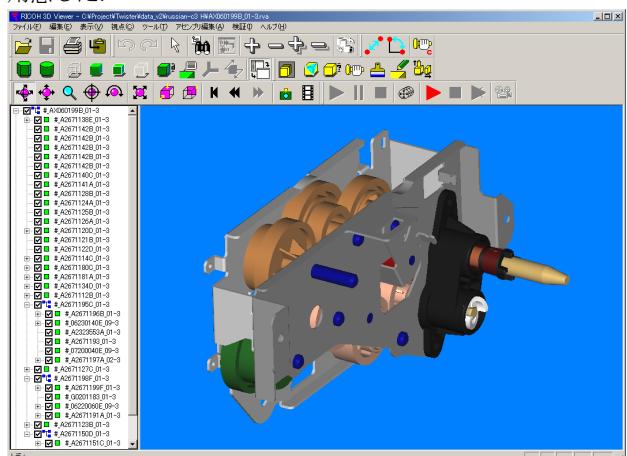


Fig.7 Viewer Overview

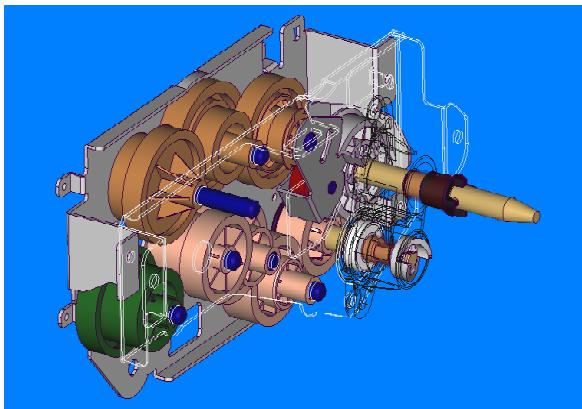


Fig.8 wire-frame and shading

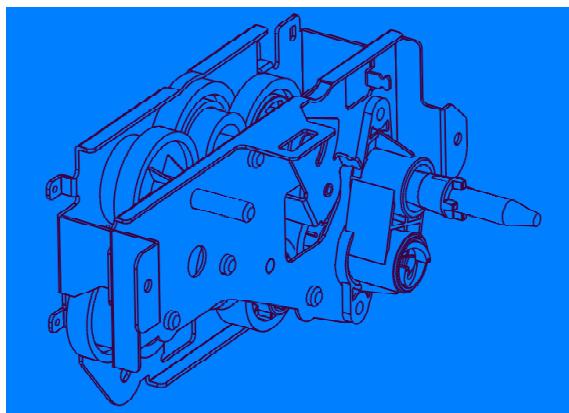


Fig.9 Hidden line elimination

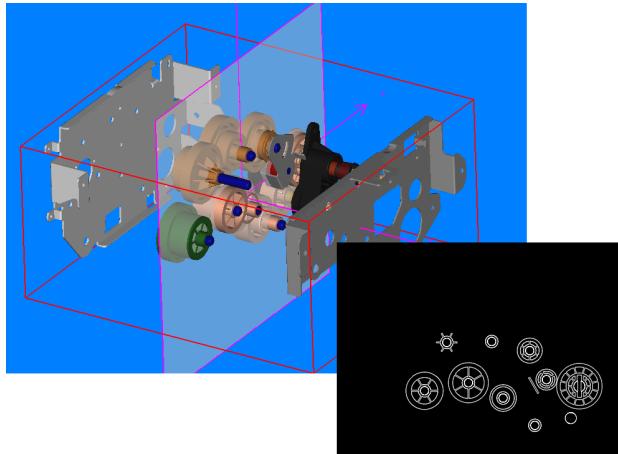


Fig.10 Cross sectional display

表示方法としては、ワイヤーフレーム表示、シェーディング表示、ワイヤー・シェーディング同時(Fig.8)、隠線処理表示(Fig.9)、断面表示(Fig.10)、表示色設定が用意されており、さまざまな表示を行うことにより、徹底的にモデルを検証することが可能である。これらの設定はすべてのモデルの単位

で切り替え可能である。

計測方法としては、モデル計測、点座標、稜線長、2点間距離、3点間角度、線・線間、面・面、点・面間距離、円弧、円柱、円錐、球などを提供している。これらの計測機能を使って、3D CADシステムで行うと同様な計測をTwister上で行うことができる。

モデル移動方法としては、基本動作として平行移動、回転移動がある。また、応用機能として以下が用意されている。

拘束条件付き平行移動 —— 移動方向と範囲を制限、

拘束条件付き回転移動 —— 回転軸と角度範囲を制限、

アセンブリ自動分解 —— CADのアセンブリ配置の情報を使い分解する。

注記機能としては、モデル、面、稜線、頂点などの要素ごとにメモを記入・表示することができる(Fig.11)。注記は、ユーザーごとにファイルに保存することも可能である。

スナップショット機能として、今まで説明した操作のほとんどを表示状態として保存することができる。また、その間を補間してアニメーションとして保存・再生することができる。

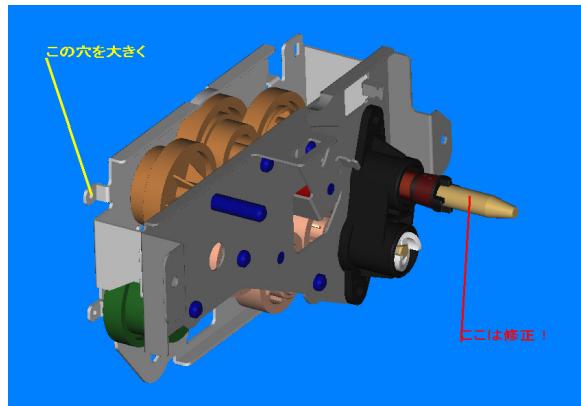


Fig.11 Annotation

6. 結果

ここでは、Twisterによる表示速度やデータ圧縮の結果について説明する。すべてのCADモデルは、Pro/Engineer V20あるいはSolidEdge V8にて設計した。時間測定に用いたPCのスペックは、PentiumIII 1 GHz CPU, 512MB RAM, Ultra ATA 100 HDD, nVidia Geforce2 Ultra(250MHz) VGA である。

6-1 データサイズ

Twister のアセンブリファイルとCADファイルとの比較をTable 2に、Twister のパーツファイルとIGESファイル (CADデータ交換のための標準的なフォーマット規格)との比較をTable 3に示す。Table 3のパーツモデルはすべて、Table 2のcopierモデルの中で使われているモデルの中から代表的なサイズのものを選択したものである。

CADファイルとの比較では、圧縮率 1.32 % から 5.78 % を達成した。IGESファイルとの比較においても、最も小さなものの、圧縮率 1.09 % を達成した。Twister 利用により、データサイズが非常にコンパクトになることがわかる。

Table 2 Assembly model Size

model	# of assembly	# of part	Twister (MB)	CAD (MB)	ratio (%)
copier	321	684	13.82	1046.08	1.32
engine	6	39	4.98	86.13	5.78

Table 3 Part model size

model	Twister (byte)	IGES (byte)	ratio (%)
A	137,687	7,669,378	1.79
B	17,461	1,103,310	1.58
C	3,107	283,884	1.09

6-2 速度

Twister とCADのそれぞれにおいて、ファイルを読み込んでから表示するまでの初期表示時間の比較をTable 4に示す。Table 4の3列目のengine(on network)は、リモートマシンにかけられたengineファイルをネットワーク越しに読み込んで表示するまでにかかった時間である。時間を速度比に換算すると、5.5 倍から 32.6 倍を達成した。Twister 利用により、初期表示時間が大幅に短縮されることがわかる。

Table 4 Assembly model time

model	CAD (sec)	Twister (sec)	ratio
copier	980.0	30.0	32.6
engine	29.5	3.3	8.9
engine (on network)	33.0	5.9	5.5

6-3 表示品質

Twister のアセンブリファイルとCADファイルの表示イ

メージをFig.12に示す。Twisterの高LODのモデルは、CADのモデルと比較して、遜色ない品質を持つことがわかる。Twisterの低LODのモデルは、視点移動時などの指標としては十分な品質を持つことがわかる。また、Fig.13には、LODの面数比指定と誤差指定の違いを示した。誤差指定のほうが全体として均一なLODが実現されているのがわかる。

7. まとめと今後の展開

コンパクトなデータと高速な初期表示速度を実現した3DモデルビューワーTwisterを提案し、実装した。Twister Compilerは、3D CADシステムからデータを取得し、表示品質を保ったまま、IGES比1%程度まで圧縮できるTwisterファイルに変換する。また、TwisterファイルをTwister Viewer で読み込むことにより、CADシステムに比べて最大約 30 倍の初期表示速度を実現した。リコー社内のさまざま部署において、既に設計部署で作成した3Dモデルの有効活用がTwisterを利用して行われている。

今後の展開として、以下を検討している。

- ・干渉計算、衝突判定、検図など検証機能の追加
- ・圧縮率、初期表示速度など基本性能向上
- ・他システムとのリンク

参考文献

- 1) 三浦憲二郎： OpenGL 3Dグラフィックス入門、朝倉書店、pp.18-19, 1995.
- 2) Michael Garland, Paul S. Heckbert: Surface Simplification Using Quadric Error Metrics, SIGGRAPH'97 Conference Proceedings, pp. 209-216, 1997.
- 3) Peter Lindstrom, Greg Turk: Fast and memory efficient polygonal simplification, IEEE Visualization '98 Proceedings, pp.279-286, 1998.
- 4) Gabriel Taubin, Jarek Rossignac: Geometric compression through topological surgery, ACM Transactions on Graphics, 17(2), pp.84-115, 1998.
- 5) Martin Isenburg, Jack Snoeyink: Compressing the Property Mapping of Polygon Meshes, Proceedings of Pacific Graphics 2001, pp. 4-11, 2001.

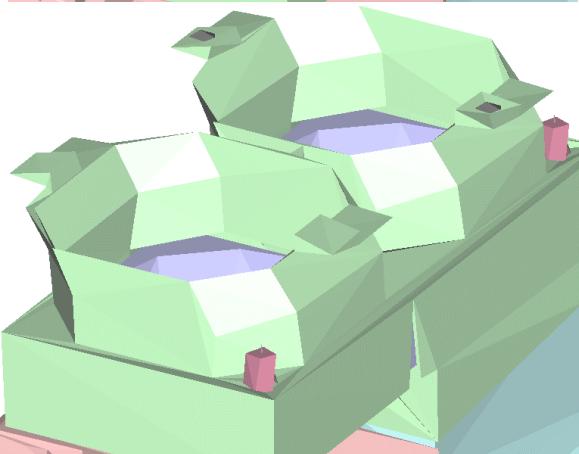
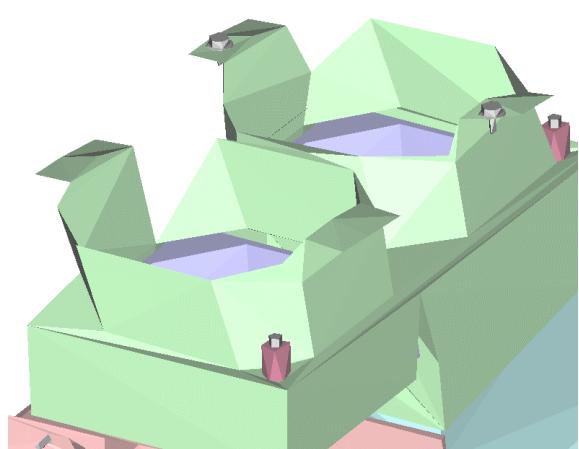
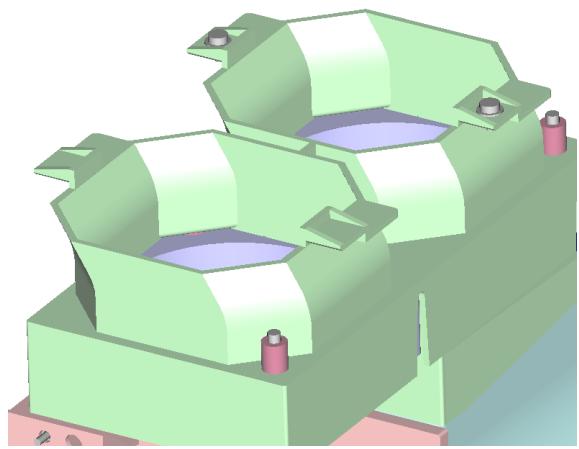
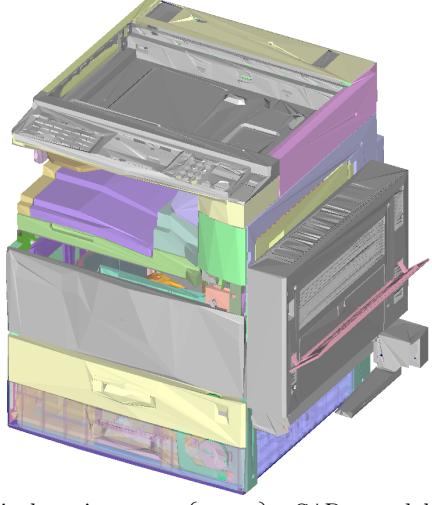
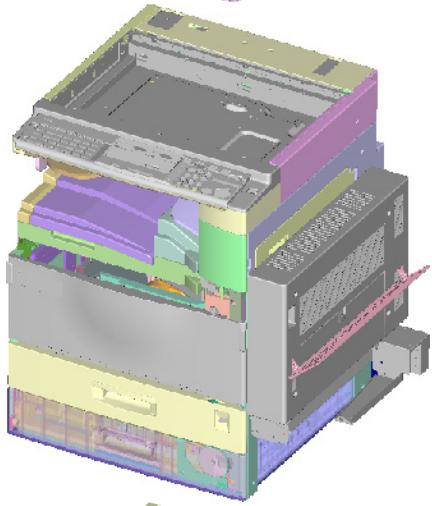
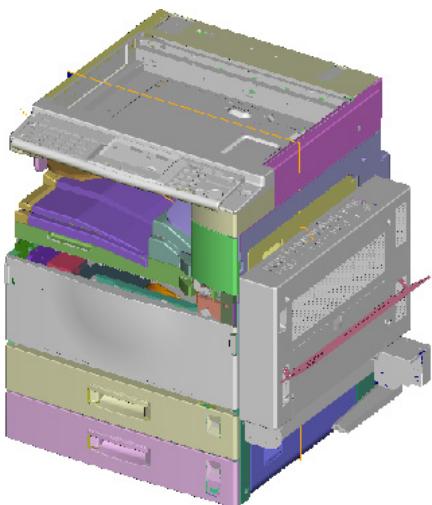


Fig.13 Mesh simplification: (upper) original model(14004 faces), (middle) face ratio=0.1(1438 faces), (lower) maximAum error=3.0(1353 faces).

Fig.12 Display images: (upper) CAD model,(middle) Twister high LOD model, (lower) Twister low LOD model.