

三角形と四角形で構成される3Dモデルの圧縮

遠藤 隆也[†]

築地 立家[‡]

東京電機大学 理工学部

1. はじめに

今日、3DCGは技術革新によってあらゆる製品に用いられ、次第に身近な技術となりつつある。そのため3Dモデルデータにおいては複雑な形状の表現のために容量が増加している。1995年にDeeringが3Dポリゴンモデルの可逆圧縮の研究として最初の論文を発表し、現在までに3Dモデルの可逆圧縮法は幾つも提案されてきた[1][2]。

その中でも位相圧縮という種類に分類される可逆圧縮法は、3Dモデルを構成する頂点や面に着目し、重複を減らす等を行うことで位相情報を圧縮する方法である。しかし、位相圧縮に関するこれまでの研究では、UV情報を含まない、三角形の面で構成された3Dモデルの圧縮を行っており、60%程度の圧縮率が報告されている[3]。

一方、Metasequoiaなど、近年普及しているモデリングツールでは、四角形も使って3Dモデルを表現している。そこで、本研究では、位相圧縮の中で最も簡単な手法である一般化三角形ストリップを拡張し、三角形と四角形が混在する3Dモデルデータの圧縮及び復元を可能にした。UV情報を含む場合、元容量が20Kbyte以上のほとんどのデータにおいて、45~50%程度の圧縮率となった。例えば、3DモデルのQRコード化[4]では、四角形の混在したUV情報付きの3Dモデルを、位相を伴わない文字データとして圧縮しており、本研究で提案する位相圧縮を行えば、圧縮率のさらなる向上が期待できる。

2. 圧縮の提案手法

2.1 三角形と四角形のストリップ

一般化三角形ストリップは同じ頂点を共有する三角形を個別で表記せずにそれらの三角形を三角形ストリップという単位で扱い、頂点の重複を減らすことで容量を削減する方法である。三角形ストリップ内の座標を記述する際は、三角形の連結情報を示す「R」、「O」、「M」の3つのコードを頂点情報とともに表記する。

2.2 面情報圧縮の手順

面情報は、マテリアル別に隣接する面をまとめ、三角形と四角形の面が混在するストリップを構成する。そのストリップを元に、面をノードとして、隣接する面で親と子の関係にした木を形成し、ノードには面を構成する頂点の番号を入れ、枝には親と子で共通する頂点番号、つまり隣接する面で共有している頂点の番号を入れていく。その後、ノードと枝が必要に応じて頂点の順番を入れ替えた後、ノードの頂点番号の中で親か子と繋がっている枝に入っている番号は削除する。頂点を並び替えたとき、逆順にした際にはノードに「R」を付加、ノードの番号削除時に番号が残ったノードには位置に応じて「F」か「L」を付加する。最後に残った頂点番号とノードに付加された「R」「F」「L」順番にストリップを表すコードを面情報として表記する。

2.3 UV情報圧縮の手順

立方体を例として、提案手順を述べる。図5は圧縮前のデータ、図6は圧縮後のデータである。最初に、全ての面のUV情報を頂点ごとにまとめる[図1]。このとき、各UV値に対応する接続頂点を記憶する。次に、頂点が保有するUV値でU値、V値、接続頂点の重複部分をまとめる[図2]。図2において、「a&b&c」は接続頂点の「a&b」と「b&c」をまとめたものである。特に、U値またはV値が一つしかない場合は、値がどの面のものか判別する必要なくなるため、接続頂点の記憶は不要となる。次に、頂点nが持つU値またはV値が頂点0から頂点n-1のどれかと等しい場合、頂点nのU値またはV値を等しい値を持つ頂点番号で表す[図3]。最後に、ファイルに出力する形式に変換する[図4]。例えば、図4の頂点6の「U:v0_7-0-4,4-7 V:v4」の表記について、「U:v0」は、頂点0とU値が等しいため頂点0のU値を用いること、「7-0-4,4-7」は、頂点0のU値で最初の値(0.00000)が保有する接続頂点は7&0と0&4であり、頂点0の次のU値(1.00000)が保有する接続頂点は4&7であること、「V:v4」は頂点4とV値が等しいため頂点4のV値を用いることを表す。

3. 圧縮性能の検証

図7では元のデータ容量に対する各圧縮処理後の容量を比較した。計測したモデルは立方体、球体、トーラスの3種類で全ての面にUV情報を持ち、それぞれ3KByteから始めて5KByteから30KByteまで5KByteごとに計測し、以降は10KByteごとに110KByteまでの15種類、全部で45個のデータを計測、元データの容量別に3種類の平均値をグラフにした。

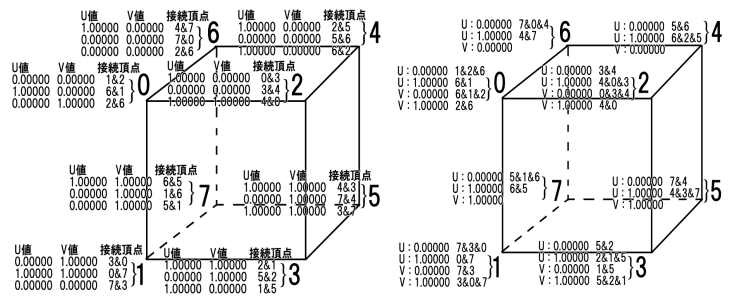


図1: UV情報圧縮1

図2: UV情報圧縮2

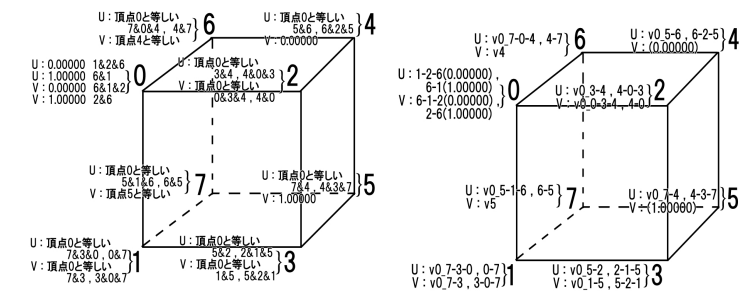


図3: UV情報圧縮3

図4: UV情報圧縮4

```
vertex 8 {
-100.0000 100.0000 100.0000
-100.0000 -100.0000 100.0000
100.0000 100.0000 100.0000
100.0000 -100.0000 100.0000
100.0000 100.0000 -100.0000
100.0000 -100.0000 -100.0000
-100.0000 100.0000 -100.0000
-100.0000 -100.0000 -100.0000
}
face 6 {
4 V(0 2 3 1) M(0) UV(0.00000 0.00000 1.00000 0.00000 1.00000 1.00000 0.00000 0.00000 1.00000 1.00000)
4 V(2 4 5 3) M(0) UV(0.00000 0.00000 1.00000 0.00000 1.00000 1.00000 0.00000 0.00000 1.00000 1.00000)
4 V(4 6 7 5) M(0) UV(0.00000 0.00000 1.00000 0.00000 1.00000 1.00000 0.00000 0.00000 1.00000 1.00000)
4 V(6 0 1 7) M(0) UV(0.00000 0.00000 1.00000 0.00000 1.00000 1.00000 0.00000 0.00000 1.00000 1.00000)
4 V(6 4 2 0) M(0) UV(0.00000 0.00000 1.00000 0.00000 1.00000 1.00000 0.00000 0.00000 1.00000 1.00000)
4 V(1 3 5 7) M(0) UV(0.00000 0.00000 1.00000 0.00000 1.00000 1.00000 0.00000 0.00000 1.00000 1.00000)
}
```

図5: 圧縮前のファイル内容

```
vertex 8 {
-100.00000 100.00000 100.00000 U:-2-6(0.00000) 6:-1(1.00000) V:-6-1-2(0.00000) 2-6(1.00000)
-100.00000 -100.00000 100.00000 U:v0_7-3-0,0-7 V:v0_7-3,3-0-7
100.00000 100.00000 100.00000 U:v0_3-4,4-0-3 V:v0_0-3-4,4-0-1
100.00000 -100.00000 100.00000 U:v0_5-2,2-1-5 V:v0_1-5,5-2-1
100.00000 100.00000 -100.00000 U:v0_7-4,4-3-7 V:(1.00000)
100.00000 -100.00000 -100.00000 U:v0_7-4,4-3-7 V:(1.00000)
-100.00000 100.00000 -100.00000 U:v0_7-0-4,4-7 V:v4
-100.00000 -100.00000 -100.00000 U:v0_5-1-6,6-5 V:v5
}
face 6 {
M(0)
R7.5 3.1 F0 2.3 R5.4 6.7 RF1 0.6 4.2
}
```

図6: 圧縮後のファイル内容

図8では図7のグラフから元容量に対する各圧縮の圧縮率と全体での圧縮率の変化を検証した。また、圧縮はUV情報圧縮後に面情報圧縮の順序で実行した。

元の容量が20KByte以上のほとんどのデータでUV情報圧縮の圧縮率は61%~65%になり、面情報圧縮での圧縮率は73%~77%、全体での圧縮率は45%~50%になった。

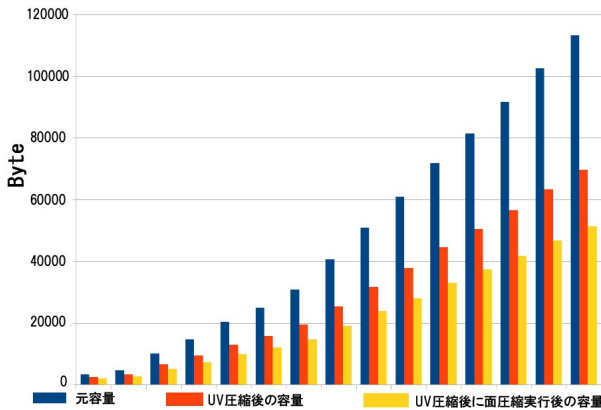


図7: 元データの容量に対する圧縮後の容量比較

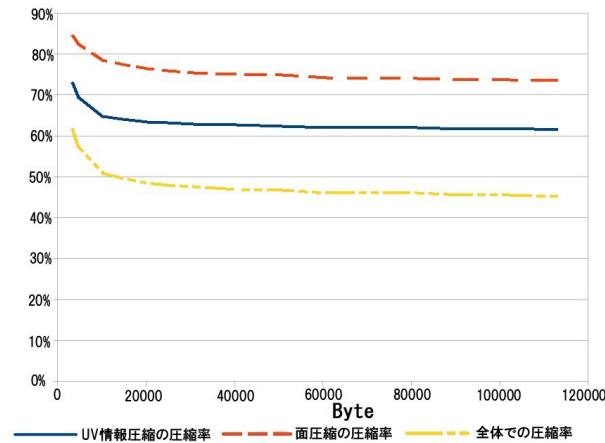


図8: 各圧縮の圧縮率及び全体での圧縮率

図9では圧縮に要した時間を、図10では復元に要した時間を計測した。計測した3Dモデルデータは図7、図8で用いたデータと同じもので各モデルデータで10回計測し、元の容量ごとに平均値を算出した。また、圧縮はUV情報圧縮後に面情報圧縮の順序で実行し、復元は面情報復元後にUV情報復元の順序で実行した。

圧縮時間の合計は元の容量が50KByteでは200ミリ秒程度、100KByteでは450ミリ秒程度を記録した。復元時間の合計は元の容量が50KByteでは120ミリ秒程度、100KByteでは230ミリ秒程度を記録した。

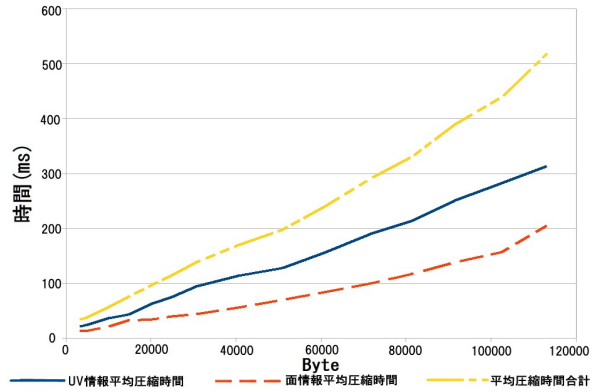


図9: 圧縮時間

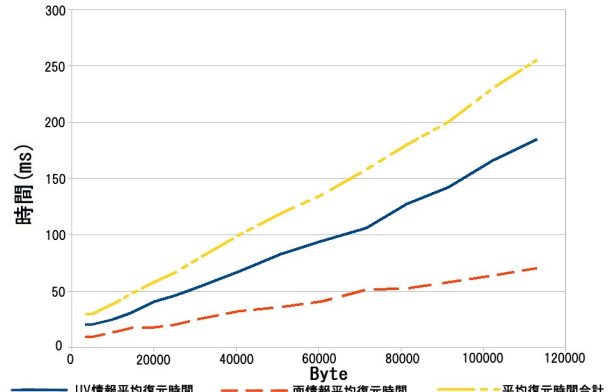


図10: 復元時間

4. 考察・結論

圧縮性能の検証から全ての面でUV情報を持つ3Dモデルデータに対しては半分程度まで圧縮する音が出来た。だが、面情報の圧縮のみでは圧縮率は7割を超えてしまうため改良の余地があると考えられる。また、元の容量が100KByte程度なら圧縮には0.5秒、復元には0.3秒あれば処理を完了できるため、処理に要する時間としては問題無いと考えられる。

また、本研究ではMetasequoiaで生成されたファイルを圧縮したが、本手法による面情報圧縮は面を構成する頂点が判明していれば別のモデリングツールにも適用可能であり、UV情報圧縮はUVマッピングを用いてテクスチャを貼り付けるモデリングツールであれば適用可能であると考えられる。

本手法では位相圧縮の中でも簡単な圧縮法である一般化三角形ストリップを拡張した手法を用いたが、三角形ストリップを用いる圧縮法に本研究で用いた手法を組み合わせることで、圧縮性能の向上とそれらの圧縮法を三角形と四角形が混在する3Dモデルデータに対応させることが可能であると考えられる。

今後の課題としてはこれまでに提案された位相圧縮の手法を本研究で用いた手法を組み合わせるさらに面情報圧縮の圧縮性能を伸ばすこと、UVマッピング以外の手法でテクスチャマッピングを行うものに対する圧縮法の提案が挙げられる。

参考文献

[1]増田 宏: 「ポリゴンモデルの圧縮技術」『精密工学会誌』第69巻第4号 pp.494-498 (2003年4月5日)
 [2]GABRIEL TAUBIN / JAREK ROSSIGNAC: 「Geometric Compression Through Topological Surgery」『ACM Transactions on Graphics』第17巻第2号 pp.84-115 (1998年4月)
 [3]Fruytoso G. M. Silva / Pranjul Yadav: 「Compression and Progressive Visualization of Geometric Models」『WSCG 2008 communication papers』 pp.71-78(2008年2月)
 [4]深津颯騎 / 築地立家: 「3DモデルのQRコード化」『情報処理学会全国大会講演論文集』第76巻第4号 pp.91-92 (2014年3月11日)