

3DモデルのQRコード化

深津颯騎[†] 築地立家[‡]

東京電機大学 理工学部

1.はじめに

本研究では 3D モデルのデータを QR コード化して AR マーカー上に再現することで、スタンドアロンで動く 3D モデル表示システムを開発する。すなわち、WEB カメラで AR マーカーと QR コードを読み取り、AR マーカー上に 3D モデルを立体的に表示するシステムである(図 1)。また、3D モデルの部位毎に QR コードを作成することで容量の多いモデルも対応できるようにした。



図 1 システムイメージ図

2.QR コード化

2.1 3D モデル

3D モデルは Met sequoia の MQO 形式を採用した。Metasequoia はデータがテキスト形式で書かれている為編集する際も容易で、また色・頂点・面が別々に記録されており、この別々に記録されたフォーマットをそれぞれ「部位の頂点」「部位の面」「マテリアル」として QR コード化することで本システムに沿った QR コードにすることができた。

2.2 独自フォーマット化

MQO 形式のデータ容量を減らす為に MQO 形式のデータの中身を「不必要な部分の削除」「精密さの不要」「文字数を減らす。」のように書き換えた。

「不必要な部分の削除」では 3D モデルを表示するのに関係のない空白タブとカメラの視点を削除した。

「精密さの不要」では、今回 3D モデルを表示するにあたって、0.1cm 以下を切り捨てた。なぜならば、WEB カメラ越しに 3D モデルを見るので物体から視点まで数十 cm 以上離れており、そんな状態で 0.1cm 以下を見極めるのは不可能だからだ。

「文字数を減らす」では面のデータの書式を変更した。従来であれば” 頂点数 V(頂点番号 頂点番号 頂点番号) M(マテリアル番号) ” のところを”

頂点数 頂点番号(142 進数) 頂点番号(142 進数) 頂点番号(142 進数) マテリアル番号” とした。こうすることで頂点数が多くなるほど容量が減り、5 文字程度で面情報を記録することができた。

2.3 QR コード化手順

QR コード化は図 2 の手順に沿って行う。コード化部位抽出とは、QR コード化したい部位の” {” から” }” まで取り出すことである。

図 2 の後半で 16 進数を 3 桁 10 進数に変換しているが、これは QR コードが数字のみの方が情報量を多く記録できる為であり、バイナリデータを読み込むと 00~FF までの 16 進数 2 文字であるが、これを 000~255 までの 10 進数 3 文字にしなかった場合、4,296 文字(英数)×3/2=6,444 文字(英数)となり 7,089 文字(数字のみ)より効率が悪くなる。

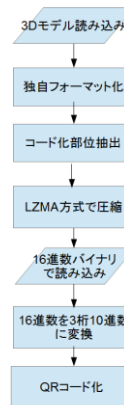


図 2 QR コード化手順

2.4 QR コードでの 3D モデル組み換え

複数に分割した QR コードは組み合わせでひとつの 3D モデルに結合することができる。複数の QR コードから結合する場合 1 個の骨格データと複数の部位(面・頂点・マテリアル)データが必要である。また、骨格データに記載されているオブジェクト名が部位データのオブジェクト名と一緒になければならぬ。裏を返せば同じ名前であれば別形の部位データを使うことができる。例えば、

Development of a stand-alone AR system reading QR codes for 3D imaging

[†] Stauki Fukatsu [‡] Tatuie Tsukiji

Tokyo Denki University, School of Science and Technology

図 3(左)の 3D モデルの中に”耳” というオブジェクトがある場合,同じ”耳” というオブジェクト名を持った別の QR コードを読み込むと図 3(右)のように”耳” の部分だけ形や色を替えることができる。



図 3 耳の部位変更(左 : 変更前 右:変更後)

3.性能検証

図 4では,QR コード一枚に収まる 3D モデルの圧縮前後のデータ容量を比較した.横軸がそれぞれの 3D モデル,縦軸に容量(Byte)のグラフを作成した.折れ線グラフは圧縮率を表している.QR コード一枚に記録できる容量は 7089byte であるが,本システムを使用すれば,テキストチャ無しの場合 37,794byte の 3D モデルデータを記録する事ができた.また,それぞれの圧縮率を比較すると,元々の容量が大きなデータほど圧縮率は高まる傾向が見られるが,大体のデータの圧縮率は 20%前後となった。

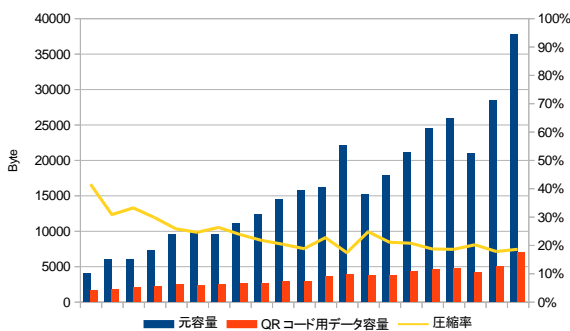


図 4 元データに対する QR コードのデータ量比較

図 5,図 6では復元時間を計測した.ここでの復元時間とは,QR コードをカメラで読み取った直後から,3D モデルを AR マーカー上に表示するまでの時間である. 図 5では QR コードの型番(サイズ)毎に各 500 回復元時間を計測し,グラフにプロットした.その結果,ほとんどのデータが 200 ミリ秒前後に集中しており,非常に高速に復元することができるシステムであると考えられる。

図 6では図 5で計測したデータの度数分布図である.グラフは対数グラフで横軸に中央値,縦軸に出現回数を取った.データは中央値の前後 10 ミリ秒間隔で集計した結果,最頻値は 180~200 ミリ秒

の区間で 3554 回であった.このグラフから圧縮しているにもかかわらず復元に時間が掛からないということが確認できた。

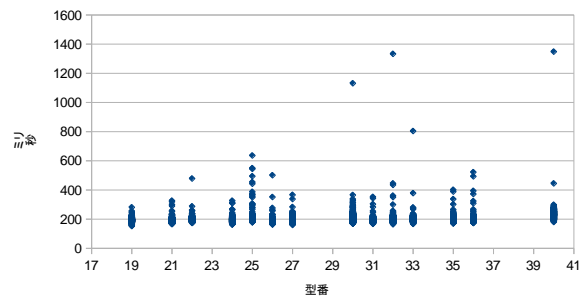


図 5 復元時間(500 回)

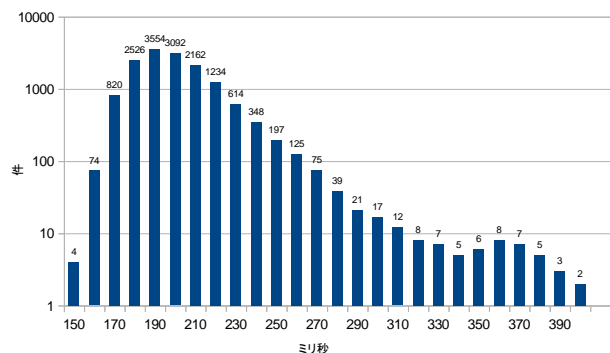


図 6 度数分布図

4.結論・考察

性能検証の結果より,テキストチャ無しの場合に 37KB 程度の 3D モデルデータを一枚の QR コードに記録することができた.容量の大きい 3D モデルも複数枚に分割することができるので,どんなモデルでも柔軟に対応することができると思われる.また,QR コード自体の読み取りも非常に高速で,フォーカスが合ったその瞬間に読み取りが完了する.QR コードから 3D モデルに復元にも時間が掛からなかった.その為読み取り・復元速度面では当初の目標は達成したと考えても良いが,QR コードに収められる容量の増加ではもう少し改良の余地があると考えられる.なぜなら 37KB 程度の 3D モデルではあまり複雑な物は再現できないからだ。

参考文献

[1]橋本直 『3D キャラクターが現実世界に誕生! ARToolKit 拡張現実感プログラミング入門』アスキー・メディアワークス (2008)
 [2] MMD 研究所 「スマートフォン 3G 回線に関する地域別実測比較調査」 <http://mmd.update.ne.jp/news/detail.php?news_id=880> (2014/01/11)