

チェーンコードを用いた3次元2値画像の情報源符号化

西尾孝治[†]手島裕詞[†]金谷孝之[‡]小堀研一[†]大阪工業大学[†]広島国際大学[‡]

1 はじめに

従来、3次元形状モデルの定義には境界表現が用いられてきた。しかし、境界表現は形状操作の際に、形状の構成要素である幾何要素を意識した操作を要求するという問題点があった。これに対し、3次元形状モデルの定義に3次元2値画像を用いて、直感的な操作を提供することが提案されている。このように、今後、3次元形状の定義に空間分割モデルを用いることが予想される。また、遠隔地にいるデザイナーがネットワークを通じて共同で3次元形状のデザインを行うことで、作業の効率化を図ることも考えられている。しかし、3次元2値画像の解像度を高く設定すると、データサイズが大きくなり、ネットワークを通じた転送や、ファイルに保存する際に問題となる。このような問題を解決するために3次元2値画像のコンパクトな保存形式が必要になると考えられる。そこで、本研究では、チェーンコードを拡張することにより3次元2値画像の情報量を削減する手法を提案する。

2 オクツリー

一般に、3次元2値画像はボクセルと呼ばれる単位立方体の集合を用いて、これらに2値情報を保持することで形状を表現する。しかし、ボクセルはモデリング空間の一辺の解像度の3乗に比例した記憶容量を必要とする。このため、一般にはボクセルを圧縮した表現であるオクツリーが広く用いられている。オクツリーは図1(a)に示す形状を、同図(b)に示す木で表現する。なお、簡単のため図は2次元で表している。また、親ノードと子ノードの対応を同図(c)に示す。

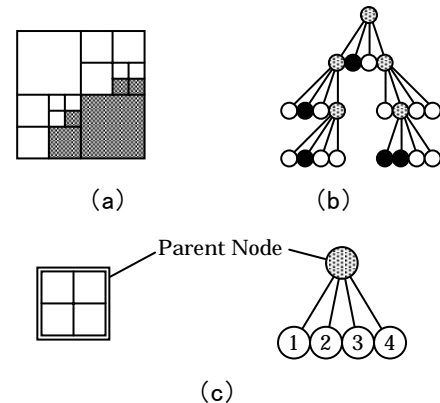


図1 オクツリーによる形状表現

このように、木を用いて形状を表現することで、必要な記憶容量を削減することができる。

3 3次元差分チェーンコード

これまでに、2次元2値画像の符号としてチェーンコードが提案されている。本研究では、チェーンコードを3次元に拡張することにより、3次元2値画像の符号化を行った。2次元の4方向チェーンコードは図2(a)に示すように符号化対象画像の方向を4種類のシンボルを用いて表現する。これを3次元に拡張すると図2(b)に示すように6種類のシンボルを用いることになる。

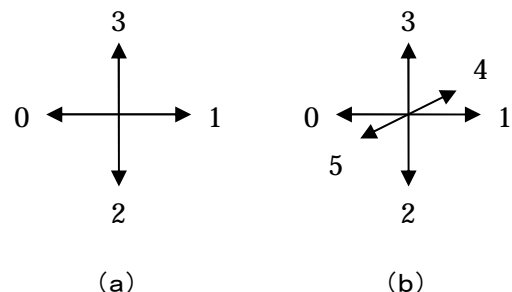


図2 チェーンコードで用いるシンボル

本手法では、符号化後に得られるチェーンコードをハフマン符号などの既存の符号化手法を用いて、さらに圧縮する。このとき圧縮結果のデータサイズを小さく

“Three Dimensional Bi-level Image Encoding Using Chain Code”

Koji NISHIO^{*}, Yuji TESHIMA^{*}, Ken-ichi KOBORI[†]

Takayuki KANAYA[‡]

[†] Osaka Institute of Technology

1-79-1 Kitayama, Hirakata, Osaka, 573-0196, Japan

[‡] Hiroshima International University

555-36 Gakuendai, Kurose, Hiroshima, 724-0695, Japan

するには、圧縮対象であるチェーンコードの情報量を少なくすれば良い。また、情報量を少なくするには、符号列に用いられるシンボルの出現頻度の偏りを大きくすればよい。そこで、図2に示したシンボルをそのまま符号とするのではなく、符号化対象画素の方向を表すベクトルの変化を符号化する差分チェーンコードを用いる。2次元差分チェーンコードの場合、符号化途中のベクトルを基準として図3に示すような4種類のシンボルが用いられる。

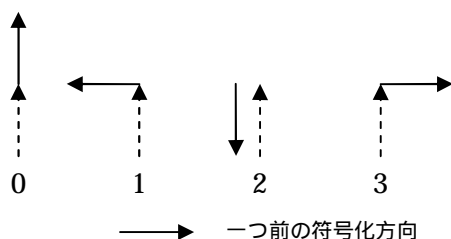


図3 2次元差分チェーンコードで用いるシンボル

ここで、同様に3次元チェーンコードを3次元差分チェーンコードにすると、2つのベクトルの相対的な関係だけでは、復号化後のベクトルが一意に決まらない。そこで、本手法では、Bribiesca のチェーンコード[1]を用いた。

4 情報源符号化

本手法では、3次元差分チェーンコードを用いて3次元2値画像の情報源符号化を行う。符号化は次の手順で行う。

- (1) 4近傍で白画素に隣接する黒画素(形状表面)を探索し、これをシード画素とする。
- (2) シード画素を始点として、XY平面に平行な面上で、2次元チェーンコードを生成すると同様に、シード画素に戻るまで、白画素に隣接する黒画素に対して順次符号化を行う。
- (3) シード画素に対してマンハッタン距離で最も近いシード画素を探索する。このとき、マンハッタン距離が1より大きい場合は、一筆書きで符号化を行えないことになる。そこで、この経路を符号化し、その両端に処理対象画素の移動経路であることを示すシンボルを挿入する。本手法ではこのシンボルを'J'とした。

した。

- (4) シード画素が見つからなくなるまで、(2)～(3)を繰り返す。

以上の結果、形状表面すべての画素の探索が完了する。ただし、文献[1]の符号では、図3のシンボル2に相当するシンボルがないので、新たにシンボルを定義し、'R'で表すことにした。

5 実験

本手法の有効性を検証するために、3次元2値画像の保存形式として一般に用いられるオクトリーを比較対象として情報量の比較をおこなった。実験には解像度が128×128×128の5種類の3次元2値画像を用いた。表1にオクトリーの情報量を、表2にチェーンコードの情報量を示す。

表1 オクトリーの情報量

| 画像 | a | b | c | d | e | |
|----------|--------|--------|---------|--------|---------|--------|
| シンボル | 0 | 17,283 | 18,555 | 37,787 | 11,838 | 33,609 |
| | 1 | 15,800 | 17,202 | 31,815 | 9,219 | 31,142 |
| | 2 | 4,726 | 5,108 | 9,943 | 3,008 | 9,250 |
| 情報量(ビット) | 53,586 | 57,932 | 112,469 | 33,902 | 104,907 | |

表2 チェーンコードの情報量

| 画像 | a | b | c | d | e | |
|----------|--------|--------|--------|--------|--------|--------|
| シンボル | 0 | 6,319 | 13,170 | 29,575 | 8,367 | 19,810 |
| | 1 | 142 | 178 | 196 | 266 | 130 |
| | 2 | 7,106 | 7,327 | 1496 | 3,348 | 7,808 |
| | 3 | 154 | 149 | 149 | 237 | 136 |
| | 4 | 477 | 678 | 883 | 837 | 1,059 |
| | R | 22 | 4 | 22 | 24 | 78 |
| J | 289 | 76 | 319 | 316 | 283 | |
| 情報量(ビット) | 14,509 | 21,582 | 32,640 | 13,395 | 29,304 | |

表1, 2よりオクトリーに比べて本手法では情報量が27～39%に削減されていることがわかる。

6 おわりに

提案手法は3次元2値画像の符号化手法としてオクトリーよりも符号化効率が高いことを示した。今後は、他の符号化手法との比較を行う予定である。

文 献

- [1] E.Bribiesca, "A chain code for representing 3D curves", Pattern Recognition, 33, pp.755-765, 2000.