

## 簡易入力された2次元画像からの3次元立体化手法の提案

鈴木聡 土井章男

岩手県立大学ソフトウェア情報学部

### 1. はじめに

一般に画像による3次元立体化は、複数の異なる角度から撮られた画像が必要である。例えば、両眼立体視では異なる角度から撮影された2枚の画像から共有点を求め、3画測量の原理より立体情報を得る。この場合、共有点を自動的に見つけることは非常に困難である。また、複数のカメラの正確な位置合わせやキャリブレーションも重要になる。そこで、本研究では入力された1枚の画像から立体情報を推定し、3次元オブジェクトの生成を容易に行う手法を提案する。

### 2. 提案手法

入力画像に対してペイントソフト等で陰影付けを行い、その陰影に対応した光源位置を予想して仮想光源を配置する。次に輪郭線で区切られた領域に複数の制御点を配置し、その制御点から曲線を求める。陰影画像の輝度値と曲線からなる面の仮想光源による輝度値が一致するように制御点を移動させる。制御点の移動には遺伝アルゴリズムを適用する。

#### 3次元立体化アルゴリズム

Step1. 輪郭抽出

Step2. 制御点算出

Step3. 遺伝アルゴリズムの適用

Step4. モデリング

Step5. レンダリング

各ステップでの具体的な手続きは次のとおりである。

**Step1. 輪郭抽出:** 入力された画像に対して図1のフィルタを適用し、水平方向(x軸方向)の輪郭線を抽出する。このとき得られた輪郭線データは2値化しておく。

**Step2. 制御点算出:** 画像を水平方向に走査し、始点と終点を見つける。次にこの2点の間に4点の中間点を置き、合計6点をその領域の制御点とする。このとき最初の高さはすべて0とする。図2は、水平方向における輪郭を黒で表した走査線である。領域Aは始点(0,0)、終点(3,0)となり、その中間点は(0.6,0)、(1.2,0)、(1.8,0)、(2.4,0)となる。また、次の領域の始点は前の領域の終点とする。

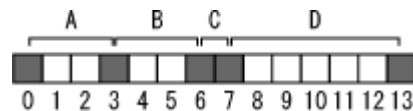


図2 制御点算出例

**Step3. 遺伝アルゴリズムの適用:** 遺伝アルゴリズムの適用の具体的な流れは、以下のとおりである。

a). コード化

求めたい領域の制御点の高さを遺伝子とし、その配列を親配列  $f_1$ ,  $f_2$  とする。

b). 交叉

子配列  $p$  を求める。子配列  $p$  は親配列  $f_1$  と親配列  $f_2$  のどちらかの各状態(ステータス)を持つとする。この時、どちらの親配列が選ばれるかは無作為に選ばれる。

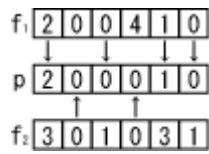


図 3 交叉の例

c). 変異

配列 p の無作為に選ばれたステータス 1 つをランダムに変更する。

d). 評価

すべての領域の配列 f1, f2, p からそれぞれ線形補間によってワイヤーフレームデータを構成する。そのあと各画素に対応するポリゴンの輝度を求め、入力画像と陰影の分布を比較し、誤差の小さい 2 配列を新たな親配列 f1, f2 とする。

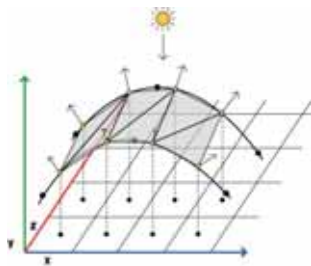


図 4. ワイヤーフレームとポリゴンから対応する画素の

輝度を算出する

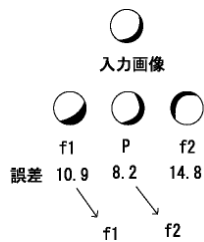


図 5. 評価例

a). から d). の処理を結果が収束するまで繰り返す。

**Step4. モデリング:** 得られた最終的な制御点を元に 3 角形近似し、モデリング行う。

**Step5. レンダリング:** 入力画像をテクスチャとしてテクスチャマッピングを用いたレンダリングを行う。

### 3. 適用例

図 5 に実際の適用例を示す。全体の輪郭は十分イメージに近い形状を推計できたと判断できる。「口」のくぼみも正確に判別できた。ただし、陰影情報の描かれていない「とさか」や「手」の部分は正確な形状を読み取ることができず、やや不自然な形になってしまった。

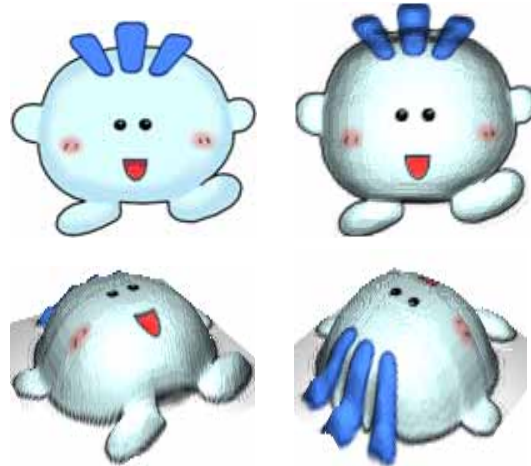


図 5. 実際の適用例

### 4. おわりに

図 5 の画像は単純なため、ほぼ正確な立体オブジェクトを作成できた。しかし、遺伝子アルゴリズムを使用しているため、計算コストが大きい。今回使用した画像は解像度 120 × 100 画素、領域数は 737 個で、処理時間はおよそ 60 秒であった。また本手法では陰影情報をユーザが別途入力しなければならないので、複雑に陰影が分布する画像では正しいモデルが生成できない場合がある。

#### 参考文献

- [1]. 「3次元CGの基礎と応用」千葉則茂・土井章男 株式会社サイエンス社 1997年