

C-04

領域分割に基づく回帰による三次元形状復元

3D-Shape Reconstruction with Regression Approach Based on Segmented Regions

森 義仁[†] 安村 禎明[†] 上原 邦昭[†]
Yoshihito Mori Yoshiaki Yasumura Kuniaki Uehara

あらまし

本稿では、単一の二次元画像から三次元形状を復元する新たな手法を提案する。単一画像から三次元形状を復元するためには、画像情報と形状との間に存在する関係性をモデル化する必要がある。このとき、この関係性は画像の輝度勾配から局所的な形状の変化への写像関数として定義される。既存手法では、物理的な光学特性を考慮した反射モデルを仮定しているが、このモデルは現実的な環境に十分則さないことがある。そこで、我々は回帰を用いて経験的にこの関係性を学習する手法を提案する。はじめに、我々は輝度のパッチと実際の形状から得られる法線ベクトルの関係性を学習し、法線ベクトルの推定器を作成する。次に、この推定器は誤推定を引き起こすことがあるため、これを修正するための新たな推定器を作成する。これは前述の推定器と同様の手続きに則って、輝度のパッチと推定された法線ベクトル、真の法線ベクトル間の関係性を学習することによって得られる。最後に、いくつかの評価実験を通して、本手法の有効性を示した。

1. はじめに

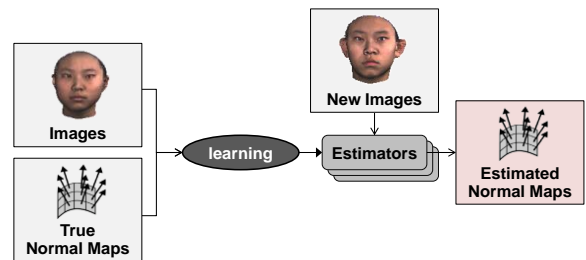
単一画像から三次元形状を復元することは、コンピュータビジョンの分野で最も重要な課題の一つである。この課題は、二次元画像が視点からの距離情報を完全に喪失したデータであるため、実現が困難な課題となっている。

三次元形状復元のための既存手法である Shape-from-shading [1] や、その改良手法 [2, 3]、は光学的な理論に基づいている。これらの手法では、計算の容易さのために理想的な反射モデルや照明条件を仮定している。しかしながら、実画像においてこの制約を満たさないことも多く、復元される三次元形状は精度が低くなることもある。Tal Hassnerら[4]は事例に基づく手法を提案した。これは画像のパッチと対応する形状を保持したデータベースを用いて三次元形状を復元する。形状を復元したい画像を入力すると、データベースから類似した画像のパッチと対応する形状を検索し、これらを統合することによって形状を推定する。しかしながら、この手法は類似したパッチを検索する際に膨大な時間が掛かるという欠点をもつ。

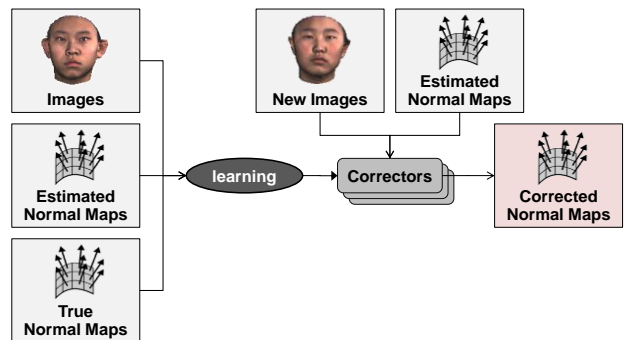
本稿では、光学的な制約を用いず、また高速に三次元形状を復元する新たな手法を提案する。本手法では画像のパッチと形状の法線ベクトルの間に存在する関係性を用いて三次元形状を復元する。我々はこの関係性を、入力画像と

既知の三次元形状から機械学習を用いて獲得し、法線ベクトルの推定器とする。しかしながら、この関係性は画像の被写体とその部位により大きく変化する。この関係性の变化を吸収し、高精度な推定を行うために、我々は画像を類似領域ごとに領域分割することで、複数の推定器からなる一連の推定器セットを作成する。さらに、我々は誤推定された法線ベクトルを修正するための修正器を作成する。これは、推定器を作成する手順と同様に、推定器によって推定された法線ベクトルを学習データに加えたものに対して機械学習を行うことによって得られる。

2. 提案手法の概要



(a) Estimation of normal maps



(b) Correction of erroneous normals

Figure 1: Overview of our approach

本節では我々が提案する三次元形状復元手法の概要を述べる。

Figure 1に示すように、三次元形状の法線マップは単一画像から推定器と修正器を用いて復元される。本手法は大きく二つに分けられる。はじめに、画像のパッチと実際の法線ベクトルからなる学習データを用いて、回帰木によって法線ベクトルの推定器を構成する。推定器は画像パッチとパッチに対応する法線ベクトル間の関係性、すなわち写

像関数を持つ.

この推定器を用いて, 三次元形状の法線ベクトルは推定される (**Figure 1-(a)**). しかしながら, 推定器は推定誤差を生じることがある. このため誤って推定された法線ベクトルを修正するための修正器を構成する. この修正器は推定器と同様の手順で構成されるが, 修正器の構成時と異なる点は, 学習データのレコードに推定器により推定された法線マップのパッチを含めることである (**Figure 1-(b)**).

3. 法線マップの推定手法

次に, 法線マップの推定について述べる.

本節でははじめに法線ベクトルの推定器の構成方法について説明する. 推定器は画像パッチと法線ベクトルの関係性を, これらから作成した訓練データから機械学習を用いて獲得することで構成できる. ここで, 獲得する関係性は, 被写体の部位によって変化するため, 訓練データを部位によって分別して作成する必要がある. 本稿では特に, 顔画像を対象としているため, 顔の構成要素に関して分割を行う. 最後に, 推定器を用いて入力画像から法線マップを推定する方法について述べる.

3.1 推定器の訓練 Training estimators

ここでは推定器の構成方法について述べる. 法線ベクトルは連続値として推定されるため, それぞれの法線ベクトル推定器における画像パッチと法線ベクトルの関係性の獲得に回帰を用いる.

訓練データは画像パッチと法線ベクトルから構成される. 訓練データの一つのレコードは, 説明変数として $k \times k$ ピクセルの窓によって切り出された画像の輝度パッチをもち, 目的変数としてパッチの中心座標に対応する法線ベクトルをもつ. すなわち, データベース中の i 番目の二次元画像と法線マップから, 座標 (x, y) に関して得られるレコード $R_{x,y}^i$ は k^2 次元の輝度ベクトルと一つの法線ベクトルをもっており, Eq.(1)で表される.

$$R_{x,y}^i = [v_0^i(x, y), v_1^i(x, y), \dots, v_{k^2}^i(x, y), \mathbf{M}^i(x, y)]^T \quad (1)$$

ここで, $v_j^i(x, y)$ は座標 (x, y) を中心とする窓内の j 番目の位置における輝度値であり, $\mathbf{M}^i(x, y)$ は法線ベクトルを表している.

この訓練データを用いて画像パッチと法線ベクトル間の関係性を獲得するが, この関係性は被写体の部位によって大きく変化する. 例えば顔画像において, 輪郭付近での関係性は顔内部での関係性と異なる. このため, 顔画像を復元の対象とする場合には, 訓練データの集合を人の顔の構造から得られる三つの領域に分割して訓練を行う. ここで, 我々は訓練データ集合を (1) 輪郭, (2) 肌, そして (3) 顔要素の領域に分割する (**Figure 2**). 領域分割を行うには, はじめに, 入力画像における既知の顔領域から s ピクセルの範囲を輪郭領域として分割する (**Figure 2-(b)**). 次に, 画

像の輝度と色相の情報を用いて Eq.(2) を満たす領域を, 輪郭領域を除いた部分から切り取ることで肌領域を分割する (**Figure 2-(c)**).

$$\text{var}_v(x, y) \times \left(1 - \frac{v_H(x, y) \cdot \bar{v}_H}{|\bar{v}_H|} \right) < \beta \quad (2)$$

ここで, $\text{var}_v(x, y)$ は座標 (x, y) とその近傍から計算される輝度の分散, $v_H(\cdot) = (\cos(2\pi H(\cdot)), \sin(2\pi H(\cdot)))$, \bar{v}_H はすべての座標における $v_H(\cdot)$ の平均値, そして $H(x, y) \in [0, 1.0]$ は座標 (x, y) における色相を表している. 最後に, 残った領域を顔要素の領域として用いる (**Figure 2-(d)**).

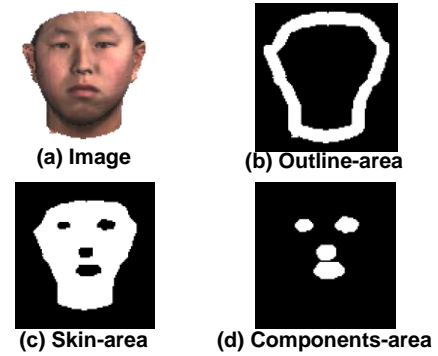


Figure 2: Regions divided with facial components

3.2 法線マップの推定

上述の手順により訓練された推定器を用いて, 入力画像から三次元形状の法線マップを推定する. 法線マップは, 入力画像のそれぞれの座標に関して法線を推定することで得られる. 推定器に対して, 3.1節で述べた訓練データと同様の形式で作成された輝度ベクトルをテストデータとして入力することで法線ベクトルの推定を行う. テストデータは, 訓練の際と同様の三つの領域に応じて推定する.

4. 誤推定された法線ベクトルの修正手法

本章では, 誤推定された法線ベクトルの修正方法について述べる. 法線ベクトル推定器を訓練する際に画像と法線ベクトルの関係性を誤って獲得する場合があります. 前述の推定器は法線ベクトルを誤って推定することがある. このような法線ベクトルを修正するために, 我々は学習によって修正器を構成する.

修正器は推定器を訓練する場合と同様の手法により, 訓練データの形式を拡張することで構成される. 訓練データは推定器を構成する場合と異なり, 属性値ベクトルとして輝度パッチに加えて法線マップのパッチを含める. 真の法線ベクトルはこの場合も同様に, 目的変数として与えられる. また, 推定器を構成する場合と同様に, 訓練データは顔の構成要素に関して分割される.

誤推定された法線ベクトルを修正する場合には, 訓練データと同様の属性値ベクトルを修正器に与え, 目的変数で

ある法線ベクトルを推定することで、より高精度な法線ベクトルを得る。

5. 実験

実際の顔画像データを用いて本手法の評価実験を行った。本章では、はじめに本実験の設定について述べ、次に実験結果について考察する。実験結果から、本手法の復元精度の高さと処理時間の短さ、また、復元された形状の視認性の高さを示す。

5.1 実験設定

本節では実験の設定について述べる。まず、実験に用いたデータと評価指標について述べ、次に復元した法線マップの可視化方法を説明する。最後に本実験を行う際の手順を示す。

評価実験のためのデータには、**Figure 3**に示す北京工科大学によって提供される三次元顔データベース[5]を用いる。このデータベースはテクスチャを含む複数人の顔の形状をもっている。訓練データとテストデータには、**Figure 4**に示す六つの照明条件で作成した顔画像を用いる。このとき、撮影方向は正面で、また、背景は除去されているものとする。三次元形状の法線マップは、法線ベクトルの行列として表される。また、真の法線マップはデータベースの持つ三次元形状から数値微分を用いて作成される。



Figure 3: BJUT-3D Face Database

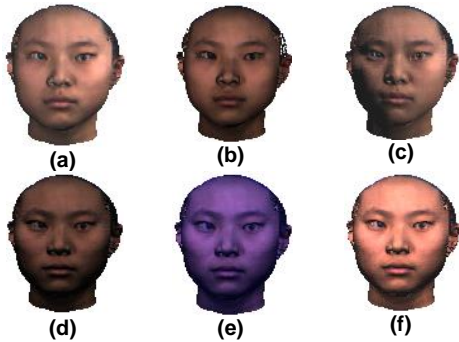


Figure 4: The lighting conditions for experiments

復元される法線マップの評価指標には、復元される法線マップと真の法線マップのコサイン類似度を用いる。この類似度はEq. (3)で与えられる。

$$s(\mathbf{M}^*, \mathbf{M}) = \cos \left(\frac{\sum_{(x,y) \in \{(X,Y)\}} \cos^{-1}(\mathbf{M}^*(x,y) \cdot \mathbf{M}(x,y))}{X \times Y} \right) \quad (3)$$

ここで、 \mathbf{M}^* と \mathbf{M} はそれぞれ復元された法線マップと真の法線マップを表しており、それぞれ Y 行列の行列である。また、 $\mathbf{M}^*(x,y)$ と $\mathbf{M}(x,y)$ はそれぞれ、座標 (x,y) における復元された法線ベクトルと真の法線ベクトルを意味する。

法線マップの可視化の際には、法線ベクトルから計算される勾配を加算することで深さマップに変換を行う。変換は $(x,y) = (\frac{X}{2}, \frac{Y}{2})$ の点を深さ0として始まり、Eq. (4)で与えられる近傍の値を用いた四つの式のうち計算可能なものを平均することで行う。

$$\begin{cases} d_1(x,y) = d(x-1,y) + \delta d_x(x-1,y) \\ d_2(x,y) = d(x,y-1) + \delta d_y(x,y-1) \\ d_3(x,y) = d(x+1,y) - \delta d_x(x,y) \\ d_4(x,y) = d(x,y+1) - \delta d_y(x,y) \end{cases} \quad (4)$$

ここで、 $d(x,y)$ は座標 (x,y) における深さを表し、 $\delta d_x(x,y) = -\frac{n_x(x,y)}{n_z(x,y)}$ 、 $\delta d_y(x,y) = -\frac{n_y(x,y)}{n_z(x,y)}$ である。また、 $n_x(x,y)$ 、 $n_y(x,y)$ 、そして $n_z(x,y)$ はそれぞれ法線ベクトルの x 、 y 、そして z 座標に関する要素である。例えば、既知の二点 $(x-1,y)$ および $(x,y+1)$ があるとすると、座標 (x,y) での深さは次式で与えられる。

$$d(x,y) = \frac{d_1(x,y) + d_4(x,y)}{2}$$

以下に、実験の手順を示す。はじめに、法線マップの推定手順について述べる。まず、**Figure 4-(a)**に示す照明条件で作成された二次元画像と三次元形状を10組用いて訓練データを作成し、推定器を構成する。それぞれの画像から輝度パッチを抽出する際の窓サイズは任意に与えられるが、本評価では 15×15 ピクセルの窓を用いた。このとき、それぞれのレコードは3.1節で述べた顔の構成要素に基づく三つの領域に分類される。また、推定器はWeka [6]に含まれるREPTreeに基づいた回帰木を用いて構成される。次に、訓練に用いなかった440人のデータからすべての照明条件下で作成された二次元画像を用いてテストを行う。

次に誤推定された法線ベクトルの修正手順について述べる。修正器を作成するために、**Figure 4-(a)**に示す照明条件で作成された二次元画像、前述の推定によって得られた法線マップと、真の法線マップを10組用いる。それぞれの画像と推定された法線マップは 15×15 ピクセルの窓によってパッチに分解され、また訓練データのレコードは顔要素を基にした三つの領域に分類される。修正器は修正器と同様に、回帰木として構成される。次に、訓練に用いなかった440人のデータからすべての照明条件下で作成された二次元画像と、そこから推定された法線マップを用いて修正器のテストを行う。

5.2 実験結果

Table 1, 2, および3は三次元形状復元の実験結果を示している。**Table 1**は既存手法と提案手法のそれぞれで復元された法線マップの精度を比較した表である。この実験に用いた二次元画像は**Figure 4-(a)**に示す照明条件で作成された。この結果を見ると、本手法が従来手法と比べより高精度な法線マップを復元できることがわかる。**Table 2**は本手法を様々な照明条件下で復元した場合の精度を示している。前述の通り、訓練に用いた二次元画像は**Figure 4-(a)**に示す照明条件で作成したもののみであるが、様々な照明条件下でも十分高精度な復元を行えていることがわかる。これは、訓練データに輝度のパッチを用いていることから、従来手法と比べ柔軟に画像と法線ベクトルとの対応付けを行えたためだと考えられる。しかしながら、訓練に用いた照明条件とテストに用いた照明条件との間に精度の大きな差があり、この点は改善を要すると考える。また、**Table 3**は三次元形状復元を行うために要する計算時間を従来手法と本手法で比較したものである。本手法は他手法と比べ、推定器および修正器の訓練に多くの時間を要するものの、一度訓練を行うと、三次元形状を短時間で復元できることを示している。

Figure 5は復元された三次元形状を前述の方法で可視化したものである。これを見ると、三次元形状の復元が法線マップの類似度だけでなく、視覚的な面でも高精度に行われていることがわかる。

6. 終わりに

本稿で我々は、機械学習を用いて、制約条件を課すことなく高速で高精度に単一の二次元画像から三次元形状を復元する新たな手法を提案した。三次元形状の法線マップを復元するために、本手法では推定器と修正器を構成する。これらは二次元画像と法線マップの間に存在する関係性を学習により獲得することで構成できる。さらに、我々は被写体の部位によるこの関係性の相違を考慮することでより高精度な復元を行った。また、実験結果から本手法が従来手法と比べより高速で高精度な三次元形状復元を行えることを示した。

今後の課題として、本手法を照明条件に対しより頑健にすることが挙げられる。これはより効率的な学習アルゴリズムを開発することや、より効果的な被写体の構造を採用することによって達成され得ると考えられる。

参考文献

[1] R. Zhang, P.S. Tsai, and J.E. Cryer, M. Shah, “Shape-from-Shading: a survey”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.21, No. 8, pp. 690-706, 1999.

[2] F. Srtori and E.R. Hancock, “Vector transport for shape-from-shading”, *Pattern Recognition*, Vol. 38, No. 8, pp. 1239-1260, 2005.

[3] P.L. Worthington and E.R. Hancock, “New Constraints on

Data-Closeness and Needle Map Consistency for Shape-from-Shading”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 12, pp. 1817-1824, 1999.

[4] T. Hassner and R. Basri, “Example Based 3D Reconstruction from Single 2D Images”, *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pp. 15-15, 2006.

[5] “BJUT-3D Face Database”, The Multimedia and Intelligent Software Technology Beijing Municipal Key Laboratory in Beijing University of Technology, <http://www.bjpu.edu.cn/sci/multimedia/mul-lab/English/>.

[6] I.H. Witten and E. Frank "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

Table 1: The accuracy of the normal maps reconstructed from images in Figure 4-(a)

	[2]	[3]	[4]	Estimation	Correction
s	0.734	0.613	0.949	0.953	0.971

Table 2: Comparison of estimated normal maps among lighting conditions. The upper is estimation result and the lower is correction result.

	4-(a)	(b)	(c)	(d)	(e)	(f)
$s(E)$	0.953	0.901	0.898	0.797	0.884	0.859
$s(C)$	0.971	0.915	0.915	0.817	0.901	0.885

Table 3: Processing time for training learners and reconstructing a shape

	[2]	[3]	[4]	Estimation	Correction
Training	-	-	60	420	840
Testing	5	5	360	4	11

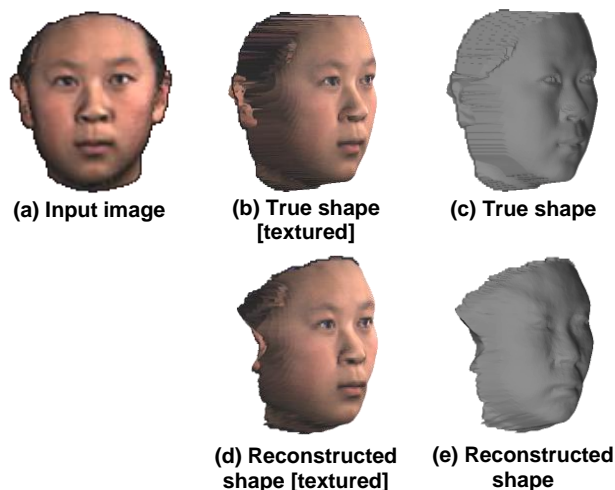


Figure 5: Reconstructed 3D-shapes