画像電子学会
The Institute of Image Electronics Engineers of Japan
画電学会年次大会予稿
Proceedings of the IIEEJ Annual Conference

3 次元ゲームシーンの学習に基づく単一画像の 3 次元化

Single-view Reconstruction by Learning 3D Game Scenes

岡部 誠 [1]　安生 健一 [2]　尾内 理紀夫 [3]

Makoto Okabe[1]　Ken Anjyo[2]　Rikio Onai[3]

1,3 電気通信大学
2 オーエルエムデジタル/科学技術振興機構

1,3 The University of Electro-Communications
2 OLM Digital, Inc./JST CREST

E-mail: 1 m.o@acm.org　2 anjyo@olm.co.jp　3 onai@cs.uec.ac.jp

1. はじめに

映像理解においてシーンの 3 次元形状は大切な手掛かりである．そこでコンピュータビジョン及びコンピュータグラフィクスの分野では映像の 3 次元化が盛んに研究されている．しかし，多くの既存手法はコストが高い．例えば，多視点ステレオは正確な 3 次元形状を測定できるが，複数の異なる視点から撮影した写真を必要とし，撮影機材及びユーザの手間という点でコストが掛かる．高性能距離画像センサは高価で購入が難しい．Microsoft Kinect のような安価な距離画像センサでは測定できる範囲が数メートルに限られる．そこで，画像から 3 次元形状を得るための，もっと安価で使いやすい手法として，単一画像の 3 次元化が盛んに研究されている.解くのが難しい不良設定問題だが，入力として 1 枚の画像しか必要とせず，ユーザにとって使いやすく，また安価である．また，3 次元化した仮想世界でカメラを動かした時の視覚的インパクトも大きくて楽しい[2,3,6,7,8,11,12].本稿では 3 次元ゲームから得た形状データを学習することによる単一画像の 3 次元化手法を提案する．

本手法は次の 3 つの既存手法と関連する．1 つ目は 2 次元線画の 3 次元化手法である．主に建物や部品等の人工物が対象で，その線画で与えるとシステムが自動的に 3 次元形状を復元する．線画はシャープなコーナー及び直線で構成される必要がある [1,9,10,13].2 つ目の関連研究は shape from shading や shape from texture であり，画像と共に物体に当たる照明もしくは物体表面上のテクスチャの情報を与え 3 次元形状を復元する [4]．3 つ目は画像中のシーンが空，建物，地面から成ると仮定して 3 次元化する手法である[6]．しかし，これら 3 つともそれぞれに制限がある．1 つ目の 2 次元線画の場合は入力画像が綺麗で，コーナーをノード，直線をエッジとするようなグラフが完璧に作れる必要があるが，実際の写真等の画像に課すには厳しい条件である．2 つ目の Shape from X では照明やテクスチャ等，シーンの環境に関する情報はいつも簡単に手に入るとは限らない.3 つ目の手法は空，建物，地面から成る画像に適用範囲が限られる．

これら 3 つの制限を緩和しつつ，今まで扱えなかった種類の単一画像を 3 次元化するため，データ駆動型の手法を提案する．画像を入力すると，提案手法はまずシーンの法線情報を推定し，それを積分する事で 3 次元形状（デプス画像）を復元する．

2. システム概要

法線情報の推定はコーナー及び線分を画像特徴量として用いる．あるコーナーや線分が現れた時，その周囲にどういった法線が共起しているか学習する．コーナーや線分がはっきり見えるような綺麗な 3 次元形状を学習データとするため，我々は 3 次元ゲームを用いる．3 次元ゲーム内に出現するシーン形状は複雑で多様性に富むと同時に，ローポリゴンモデルでありコーナーや線分がはっきり見えるので学習に適していると考えた．また，Microsoft DirectX に付属のソフトウェアを用いると，ゲーム内で視点を自由に変えつつシーンの形状データを取得できる．今回は Ubisoft 社の Anno 1404 という都市経営シミュレーションゲーム[14]を遊びつつ，合計 236 のシーンのデプス画像を取得した．このデプス画像を法線画像に変換しデータベースとする(図 1-a)．この法線画像に対しコーナー及び線分抽出器を適用しつつ，そこに現れる法線の状態を学習する(図 1-b と c)．

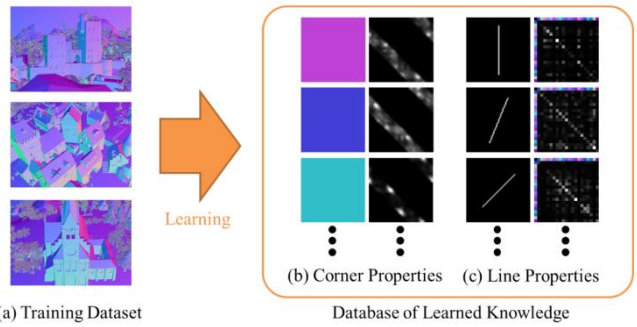


図 1: 学習プロセスの概要．法線画像の色は法線の方向を表す．

3 次元再構築プロセスは単一画像を入力とし(図 2-a)，学習プロセスと同じ手法でコーナー及び線分を抽出する(図 2-b)．抽出したコーナー及び線分が上で学習したような確率分布に従って法線を持つと仮定すれば，入力画像の法線の推定問題は Markov Random Field を用いた法線の割り当て問題として定式化できる (図

2-c)．最後に，推定された法線画像をポアソン方程式を解くことで積分し目的の 3 次元形状を得る(図 2-d)．

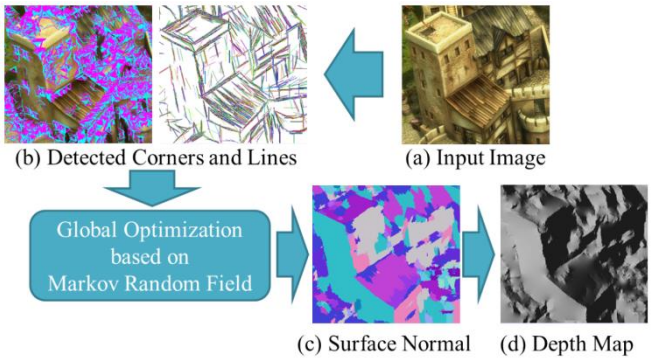


図 2: 3 次元再構築プロセスの概要．

3. コーナー及び線分抽出と法線の関係の学習

本手法で用いるコーナー表現を図 3 に示す．(a)のような画像がある時，(b)のような赤，黄，青の 3 つのコーナーが抽出される．コーナーは 2 つの腕を持ち，それぞれの腕に方向がある．つまり 1 つのコーナーは(c)のように θ，ϕ の 2 つの数値で表現される．

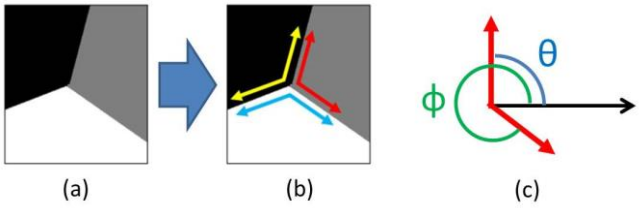


図 3: 本手法のコーナーの表現．

コーナーは入力画像の勾配（エッジ）を解析して抽出する．入力画像をグレースケール変換したものを I とおき，水平方向及び垂直方向の勾配画像をそれぞれ dI/dx，dI/dy とすれば，各ピクセルにおける勾配は方向が $\tan^{-1}\frac{dI/dy}{dI/dx}$ で大きさが $\sqrt{(dI/dx)^2+(dI/dy)^2}$ となる．

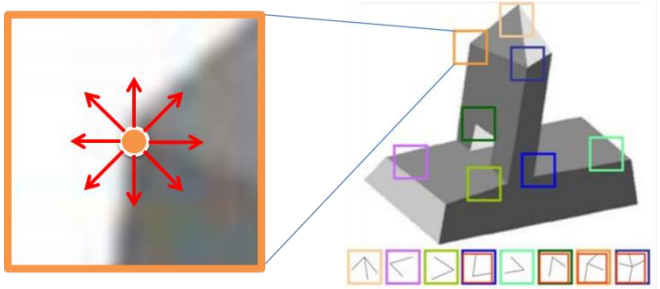


図 4: 移動窓によるコーナー抽出

図 4 のように画像上に正方形の窓(本実験では 64×64)を移動させつつ，窓の中心から赤い矢印の方向にピクセルを観察し，そこでの勾配方向が矢印の方向と一致していて，かつ，勾配の大きさの累積が閾値以上であった場合，そこにコーナーの腕があると判定する．図で赤い矢印は 8 本だが，実験では 32 本の矢印を考えた．

法線画像上でコーナーを抽出しつつ，2 本の腕の間にどのような法線が現れるかを観察する(図 5)．(a)は緑の法線について，データベース中の 236 枚の法線画像に対し，各コーナーの出現頻度を測定した結果である．赤丸で囲んだところが白いのは，緑の法線と共にこの辺りのコーナーが頻出した事を示しているが，これは(b)中の赤いコーナー等に対応する．同様に(c)にはピンクの法線に対して頻出したコーナーの 1 つを黄丸で示したが，これは(b)中の黄色いコーナー等に対応する．

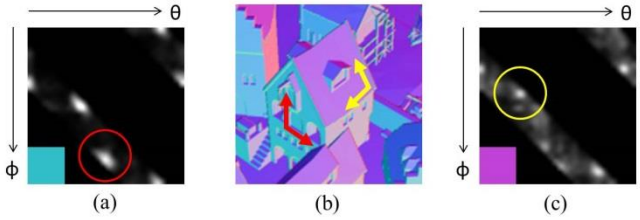


図 5: コーナーと法線が共起する頻度の測定．

線分の抽出はコーナーにおける腕を 1 本抽出する処理と同じである．法線画像上で線分を抽出しつつ，線分の両サイドにどのような法線が何回現れるかも調べる(図 6)．(a)は縦の線分について，各法線の組み合わせの出現頻度を測定した結果である．矢印で示した部分が白いのは，縦の線分の左側に緑の法線がある時，線分の右側はピンクの法線が現れる事が多く，逆に，左がピンクで右が緑の組み合わせも頻出した事を示している(b)．(c)では斜め 45 度の線分に対して頻出した法線の組み合わせの一例を(b)のオレンジの線分と共に示している．

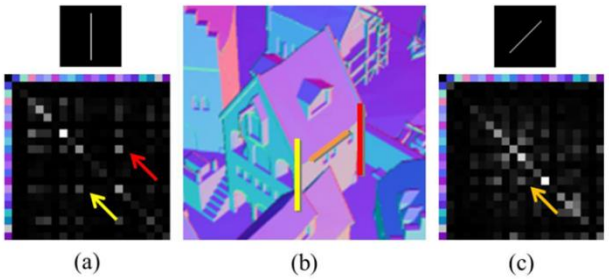


図 6: 線分とその両サイドに出現する法線の組み合わせの測定．

4. 全体最適化による法線推定

3 次元化したい入力画像が与えられたら，その画像内でも上で学習した確率分布に従ってコーナーや線分の周囲に法線が存在しているはずだと仮定することで法線画像の推定を行う．ここではこの問題を，各ピクセルに法線を割り当てる問題として解く．

まず割り当てる法線であるが，データベース中のゲームシーンに出現する全ての法線に対して K-means 法を

$K = 20$として適用し，クラスタ中心となった 20 個の法線を割り当ての候補とする．即ち解くべき問題は，20 個のラベルを各ピクセルに割り当てる問題となる．

次に入力画像をオーバーセグメンテーション[5]する(図 7)．似たような色を持つピクセルが固まり，スーパーピクセルとなる．全ピクセルへの割り当てでなく，スーパーピクセル(図 7-右)への割り当て問題とすることで問題を簡略化し，かつ精度を上げることができる．

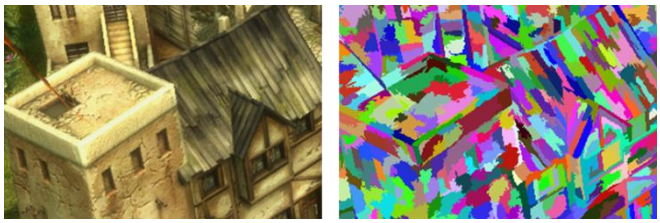


図 7: オーバーセグメンテーション．

各スーパーピクセルをノード，隣接するスーパーピクセル間にエッジがあるようなグラフを考えると，解きたい法線割り当て問題は，以下のようなエネルギー最小化問題を解くことと同じになる．

$$\arg\min_{\vec{n}} E = \sum_{x} V_x(\vec{n_x}) + \lambda \sum_{(x,y)} W_{x,y}(\vec{n_x}, \vec{n_y}),$$

ここで$\vec{n_x}$と$\vec{n_y}$はスーパーピクセルx及びyに割り当てられた法線であり，V_xはxに$\vec{n_x}$が割り当てられることによるコスト，$W_{x,y}$は隣接するスーパーピクセルxとyに$\vec{n_x}$と$\vec{n_y}$が割り当てられたことによるコストを表現する．

ノードのコスト関数V_xは以下のように定義される：

$$V_x(\vec{n_x}) = -\log(\mathrm{P}(\vec{n_x}|c_x)), \quad \mathrm{P}(\vec{n_x}|c_x) \propto \mathrm{P}(c_x|\vec{n_x})\mathrm{P}(\vec{n_x}),$$

ここで$\mathrm{P}(c_x|\vec{n_x})$はスーパーピクセル$x$に法線$\vec{n_x}$が割り当てられたとき，スーパーピクセル$x$でコーナー$c_x$が観測される確率を表しているが，これは図 5 の a と c で表現されるような確率密度関数そのものである．スーパーピクセルxで複数のコーナー$C_x = \{c_{x1}, \ldots, c_{xn}\}$が検出される場合は

$$\mathrm{P}(\vec{n_x}|C_x) \propto \mathrm{P}(C_x|\vec{n_x})\mathrm{P}(\vec{n_x}) = \mathrm{P}(c_{x1}|\vec{n_x})\ldots\mathrm{P}(c_{xn}|\vec{n_x})\mathrm{P}(\vec{n_x}),$$

を考える．エッジのコスト関数$W_{x,y}$を以下のように定義する．

$$W_{x,y}(\vec{n_x}, \vec{n_y}) = -\log(\mathrm{P}(\vec{n_x}, \vec{n_y}|e_{x,y})),$$

ここで$\mathrm{P}(\vec{n_x}, \vec{n_y}|e_{x,y})$は隣接するスーパーピクセル$x$とyの間で線分$e_{x,y}$が観測された時，$x$とyに$\vec{n_x}$と$\vec{n_y}$が割り当てられる確率を表す．これは図 6 の a と c で表現されるような確率密度関数そのものである．

以上のエネルギー最小化問題はグラフカットアルゴリズムの α 拡張法もしくは α-β swap 法を用いて効率よく解ける．ただし，現在の我々のコスト関数はグラフカットで最適解を得るために必要な劣モジュラ性を$W_{x,y}$が満たしていないため，局所解しか得ることができない．そこで，グラフカットの初期値をランダムに変えつつ複数回問題を解き，最小のエネルギーを与えた法線を最終出力としている．

5. 結果と考察

データベースの画像に対してコーナー及び線分の抽出を施した例を図 8 に示す．コーナーについては 2 本の腕を水色，腕の先端を結んだ弦を紫色で表現している．線分はランダムな色で重ねて表示している．図に示すように細かなコーナーや，コーナーの作る角度が極端に大きい、もしくは小さいものは取り逃がしてしまっているものの，代表的なコーナーは概ね正しく抽出できている．

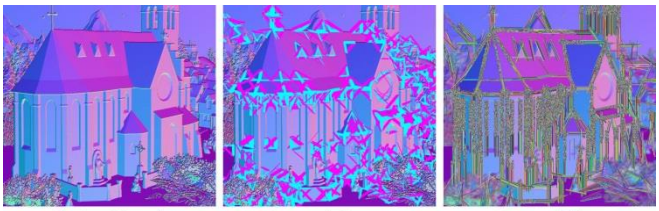


図 8: 3 次元ゲームシーンの法線画像に対しコーナー及び線分の検出を適用した結果．

提案手法で推定した法線画像の例を図 9 に示す．また，法線画像を積分して 3 次元形状を推定した結果をビデオで示す．1 行目の様なシンプルな画像では綺麗な法線画像が得られている．一方，実際の写真では結果の法線に間違い多く見られるが，これはコーナー及び線分抽出の間違いに起因する．コーナー及び線分抽出の精度を高める事が，提案手法を今後改善していくにあたって最も大切なことだと考えている．

コーナー及び線分の抽出に関しては他のアルゴリズムとの比較実験が今後の課題である．例えばハフ変換の利用が考えられ，実際に筆者らも実装して実験を行ってみたものの，Canny エッジ検出器のパラメータやハフ変換自体のパラメータ，更に線分 2 つの距離がどの程度近いときにコーナーとみなすかの閾値など，提案手法と同じかそれ以上のパラメータが存在する上，検出された線分の質を見ても提案手法と比べて顕著な改善が見られなかったので，利用には至っていない．

また提案手法の限界の 1 つとして，コーナーがはっきり見えるような画像が対象であるという点がある．例えば，建物を正面から撮影してしまえば，見えるコーナーは 90 度のものばかりとなり，その場合は提案手法は上手く法線を推定することができない．結果に示すように斜め 45 度から対象を見たような構図が最もバリエーションに富んだコーナーをはっきりと見せてくれるため，今回はこのような画像のみを対象に実験を行っている．今後はより適用範囲を広げるための研究が必要である。

6. 参考文献

[1] Cao, L., Liu, J., and Tang, X. 2005. 3d object reconstruction from a single 2d line drawing without hidden lines. In ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, 272–277.

[2] Criminisi, A., Reid, I., and Zisserman, A. 2000. Single view metrology. Int. J. Comput. Vision 40, 2, 123–148.

[3] Debevec, P. E., Taylor, C. J., and Malik, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. Computer Graphics 30, Annual Conf. Series, 11–20.

[4] Durou, J.-D., Falcone, M., and Sagona, M. 2008. Numerical methods for shape-from-shading: A new survey with benchmarks. Comput. Vis. Image Underst. 109, 1, 22–43.

[5] Felzenszwalb, P. F., and Huttenlocher, D. P. 2004. Efficient Graph-Based Image Segmentation. International Journal of Computer Vision, Volume 59, Number 2.

[6] Hoiem, D., Efros, A. A., and Hebert, M. 2005. Automatic photo pop-up. ACM Trans. Graph. 24, 3, 577–584.

[7] Horry, Y., Anjyo, K., and Arai, K. 1997. Tour into the picture: using a spidery mesh interface to make animation from a single image. In Proc. SIGGRAPH '97, 225–232.

[8] Kang, S. B. 1998. Depth painting for image-based rendering applications. Tech. report, CRL, Compaq Cambridge Research Lab, 1998.

[9] Lipson, H., AND Shpitalni, M. 1996. Optimization-based reconstruction of a 3d object from a single freehand line drawing. Computer-Aided Design 28, 651–663.

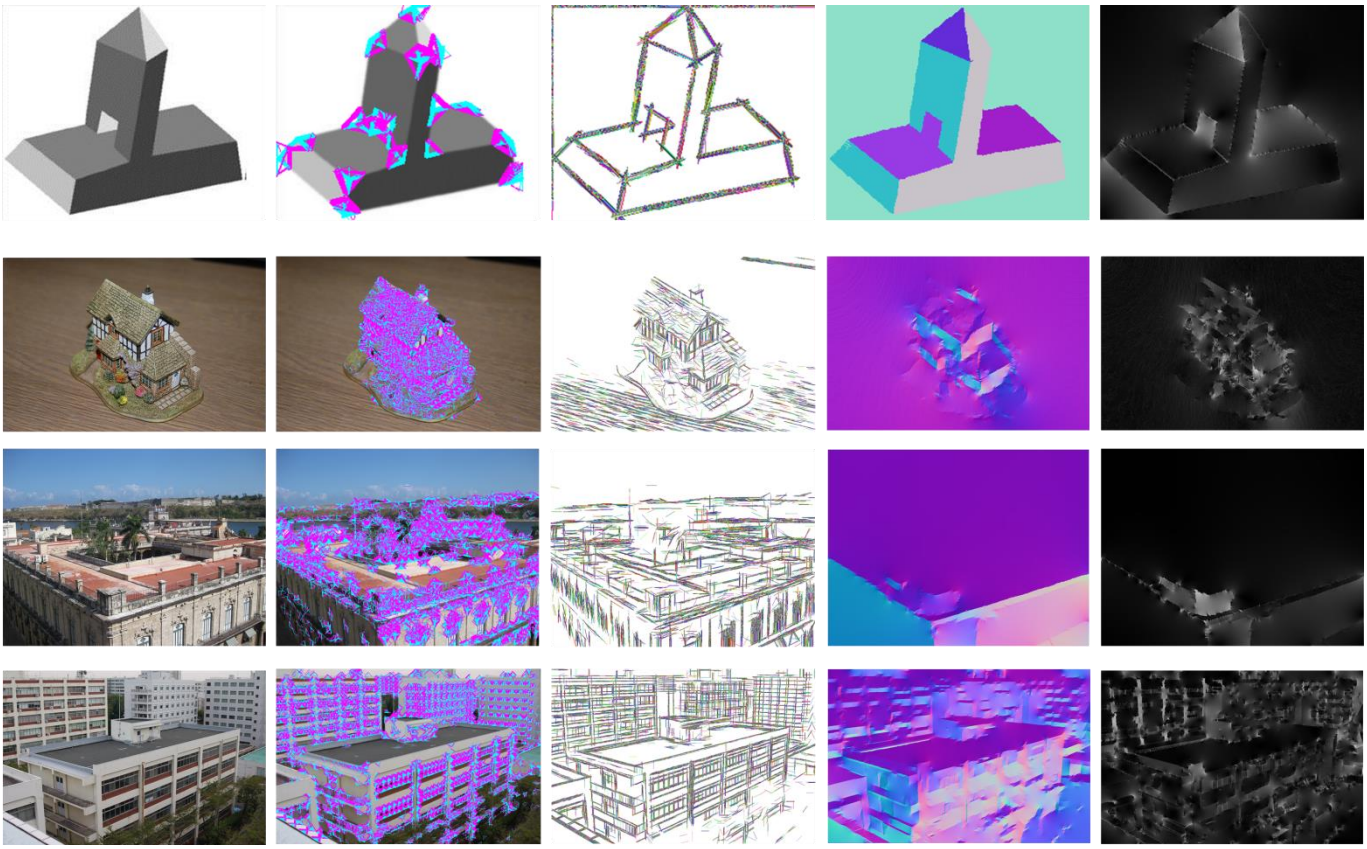
[10] Liu, J., Cao, L., Li, Z., and Tang, X. 2008. Plane-based optimization for 3d object reconstruction from single line drawings. IEEE Trans. Pattern Anal. Mach. Intell. 30, 2, 315–327.

[11] Oh, B. M., Chen, M., Dorsey, J., and Durand, F. 2001. Image-based modeling and photo editing. In Proc. SIGGRAPH 2001, 433–442.

[12] Saxena, A., Chung, S. H., and Ng, A. Y. 2008. 3-d depth reconstruction from a single still image. Int. J. Comput. Vision 76, 1, 53–69.

[13] Sugihara, K. 1986. Machine interpretation of line drawings. MIT Press, Cambridge, MA, USA.

[14] UBISOFT. 2010. Anno 1404.



(a) (b) (c) (d) (e)

図 10: 結果．(a)入力画像，(b)検出されたコーナー，(c)検出された線分，(d)推定された法線，(e)積分により得られた 3 次元形状から測定し直した法線と推定した法線(c)との誤差．