Vol.2020-CVIM-222 No.22
2020/5/15

疎なメッシュモデルの分割と変形によるメッシュ超解像

田村 稜[1]　伊東 聖矢[1]　金子 直史[1]　鷲見 和彦[1]

概要：本研究では，多視点画像から再構成された 3 次元形状の粗いメッシュモデルと画像から推定した法線を入力とし，精細なメッシュモデルを生成する手法を提案する． メッシュモデルは比較的少数のパラメータで形状を表現できる 3 次元データ構造 であり，多視点画像から精細なメッシュモデルを再構成するには Multi-View Stereo (MVS) が適しているが，その計算時間は使用する画像の解像度や枚数が 増えると急激に増大する．本研究では Structure from Motion とドロネー分割で得られる粗いメッシュモデルに対し，画像から推定した法線方向が内部で変化するメッシュを見つけて，それを直接分割することで，MVS より高速に精細なメッシュモデルを生成する．提案手法は単一物体と一般シーンのデータセットを用いて有効性を検証し，平均 60 倍の高速化を実現した．

Super-Resolution of Coarse Mesh Model via Mesh Deformation

1. はじめに

多視点からの 3 次元空間の再構成はコンピュータビジョン分野における困難な課題であり，正確で細部まで表現された 3D モデルの生成はロボティクスや AR など幅広いアプリケーションで使用される．

画像を用いた 3 次元再構成手法に Structure from Motion (SfM) がある．SfM は画像からキーポイントを抽出し，画像間で対応を取ることで 3 次元点を復元する．3D モデルは 3 次元点の集合である点群で表現される．SfM は 3 次元点群だけでなく，カメラの内部パラメータ同時にも推定でき，これらは Multi-View Stereo (MVS) に用いられる．SfM は疎な点群をキーポイントから復元する一方で，MVS は画像間で一致した領域を密な点群として復元する．したがって，MVS は複数視点からの詳細なメッシュモデルの復元に適している．しかしながら，MVS は画像の枚数や解像度が上昇すると計算量も増大するという問題点がある．

この問題を解決するために，MVS の再構成手法について検討する．MVS は画像間のパッチの対応を取るため計算量が多くなる．また，点群は 1 つの平面を大量の 3 次元点で表現するため，詳細なモデルを表現するためには冗長である．そのため，本研究では 3D データ構造として効率的な表現が可能なメッシュモデルを採用する．さらに，画像パッチ間の対応関係を参照するのではなく，疎な点群から構築した粗いメッシュモデルを直接操作し，効率的に詳細なメッシュモデルを再構成する．

本研究では疎なメッシュモデルを変形することでより詳細なメッシュモデルを構築する手法を提案する．まず最初に SfM を用いて疎な点群を構築し，そのモデルに対してドロネー三角分割を行う．ドロネー三角分割によって疎な点群を三角メッシュに変換する．. また，画像から法線を推定し，各メッシュと推定法線を関連付けるために，SfM によって取得されたカメラパラメータを使用してモデルを各カメラに投影する．各三角形メッシュについて，推定法線の変化に基づいてメッシュを変形する．この方法は，メッシュ法線と推定された法線の差が小さくなるまで繰り返される．本研究では基となる 3D メッシュモデルの構造を維持しながら詳細なメッシュモデルを構成することをメッシュ超解像と呼ぶ．

2. 関連研究

2.1 3D モデルの表現方法

典型的な表現の 1 つはボクセルで，これは 3D 空間の規則的なグリッドで区切られ表現される．ボクセルのデータ構造はピクセルの 3D 拡張であるため，画像処理のさまざまなアプローチを 3D データに簡単に拡張できる [1], [2], [3]. ボクセルは解像度とメモリ消費の間のトレードオフのた

[1] 青山学院大学
Aoyama Gakuin Uniersity

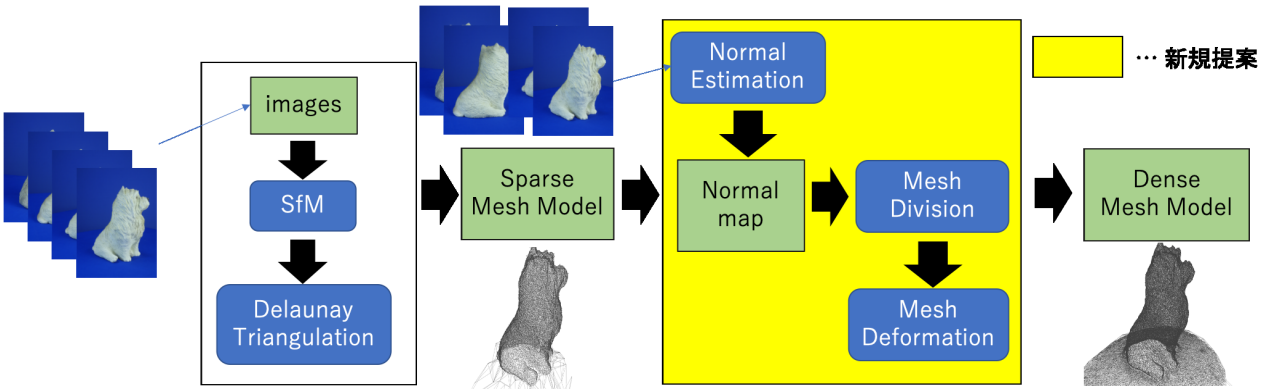


図 1: 提案手法の全体図
Fig. 1 System Overview

め，詳細な 3D 形状を表すのには適していない．

また，3D 空間の点の集合である点群データもある． 点群データを直接取得できるセンサがあるため，分類 [4], [5] に一般的に使用され，点群のメモリ消費はボクセル表現よりも効率的である．しかし，構造化されていないデータのため，点群データの処理が難しい場合がある．

そこで，頂点と面で構成されるポリゴンメッシュが広く使われている [6], [7]．ポリゴンメッシュは点群データと同等なメモリ効率で，頂点と面に関連性がある．ポリゴンメッシュは点群よりも少ないパラメータで平面が表現できるため，詳細な 3 次元形状を表現するのに適している．本研究では，3D モデルをポリゴンメッシュの中でもでもシンプルな三角メッシュで表現する．

2.2 Multi-View Stereo での 3 次元再構成

MVS では，既知のカメラモデルとカメラ位置姿勢を使用して多視点画像から 3D モデルを再構成することを目的としている．また，構造化されていない多視点画像からカメラ位置姿勢を得るために，SfM はよく用いられる．SfM には Incremental な方法 [8] と Global な方法 [9] に分けられる．Incremental SfM では初めに 2 枚の画像から 3D マップを再構成し，その後新しい画像を追加することで 3D マップを更新する．一方 Global SfM では同時にすべての画像からカメラ位置姿勢を推定する．また，Incremental SfM と Global SfM を組み合わせた Hybrid SfM [10] も提案されている．

従来の MVS は隣接する画像を選択し [11] 画像パッチ同士の相関を測る [12] という手法である． しかし，Schönberger はピクセルごとのビュー選択に測光，幾何学的な要素を利用して，深度マップと表面法線を推定する MVS の手法 COLMAP[12], [13] を提案した．COLMAP は様々な MVS ベンチマークにおいて高精度な復元結果を達成したが，3D モデルの再構成には長時間を要する．本研究では COLMAP の Incremental SfM [13] を用いて 3D モデルを再構成し，カメラの内部パラメータ及び外部パラメータを推定する．

2.3 メッシュモデルに対する深層学習

近年，メッシュ表現のための深層学習ベースの手法が提案されている [14], [15], [16]．これらの手法はメッシュモデルをグラフとして扱い，グラフの畳込み，グラフプーリング，グラフアンプーリングを行う．Wang らは単画像から 3D モデルを再構成する Pixel2Mesh[14] を提案した．Pixel2Mesh は初期状態として楕円形のメッシュモデルを使用し，画像の特徴量からメッシュモデルを変形する．Wen らは Pixel2Mesh の拡張で複数視点画像から 3D モデルを再構成する Pixel2Mesh++[15] を提案した．どちらの手法も初期メッシュと同じトポロジのみ再構成可能であるため，様々なオブジェクトが混在する一般的なシーンに適用することは困難である．

3. 提案手法

本研究では，疎なメッシュモデルを変形することで詳細なメッシュモデルを構築する方法を提案する．従来手法である MVS と同等な性能を保ちつつ，計算量を少なくするために表面法線を用いてメッシュモデルを直接変形する．**図 1** に提案手法の概要を示す．入力となる疎なメッシュモデルは，複数視点画像使用して再構成されるとする．本手法では各メッシュの表面法線を画像から推定した表面法線と比較し，それらの違いに基づいてメッシュを変形する．今回は推定法線は既知と仮定する．

まず，複数視点画像から疎なメッシュモデルを再構成する．COLMAP の SfM を用いて疎な点群モデルとカメラパラメータを取得する．得られた疎な点群モデルはドロネー三角分割を行うことでメッシュモデルへと変換される．メッシュモデルと画像から推定された表面法線を関連付けるために，SfM から得られたカメラパラメータを用いてメッシュモデルを各視点から投影される．画像から推定された法線マップは各ピクセルに表面法線を持っている．一方メッシュモデルはメッシュに対して 1 つの表面法線し

か持っていない．これらの法線を比較しメッシュを変形するかを判断する．1 つのメッシュ内の推定された表面法線の変化が閾値以上であるかどうかで変形が必要かどうかを判断する．メッシュ内のすべての表面法線の変化が閾値より小さくなるまでメッシュ変形は繰り返される．以下に提案手法の詳細を示す．本研究では法線の推定結果は既知とし，推定法線マップは多視点で整合性が取れていると仮定する．

3.1 メッシュ分割

メッシュの変形は，メッシュ分割とメッシュ変形の 2 段階の処理で行われる．メッシュ分割では表面法線をもとに 1 つのメッシュを 2 つのメッシュに分割する．この処理はメッシュの表面法線と画像から推定した表面法線との差が小さくなるまで，推定法線マップを順番に使用することで行われる．

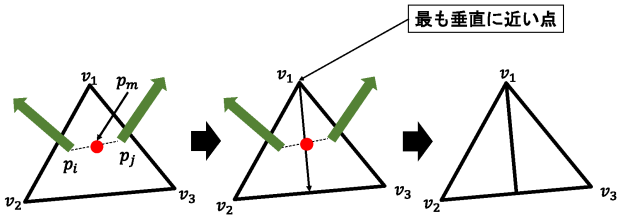


図 2: メッシュ分割の処理フロー
Fig. 2 Mesh Division Overview

図 2 は分割処理の概要である．ここで，メッシュモデルのターゲットメッシュを f，推定法線マップを N，メッシュ f が投影されるピクセルを P とする．メッシュ f の表面法線を n_f とし，ピクセル $i \in P$ の推定表面法線を n_i とする．推定された法線マップ N について，2 つの法線 n_i と n_j の角度 θ_{ij} を計算する．ここで，$i, j \in P, i < j$ である．θ_{ij} の最大角度がしきい値角度 θ_ϵ を超えると，ターゲットメッシュ f が分割される．このとき法線は，最大角度の座標間で線形に変化すると考える．法線の変化はメッシュの中点で分割することで最小化されるため，中点 p_m を計算する．

$$p_m = (p_i + p_j)/2 \tag{1}$$

ここで，p_i および p_j はそれぞれのピクセル i, j の 3D 座標である．また，v_k $(k = 1, 2, 3)$ をメッシュ f の頂点とする．このとき，頂点から中点を通る垂直な直線で分割することが最適だと考えられるため，中点 p_m から p_i，および v_k までの線分の角度を算出する．

$$\cos\phi_{ik} = \frac{(p_i - p_m)\cdot(v_k - p_m)}{|p_i - p_m||v_k - p_m|} \tag{2}$$

そして，中点 p_m と頂点を通る直線でメッシュを分割する．このとき，頂点 $\tilde{v}$ は 2 つの法線の座標を通る線分に最も垂直 に近づく．

$$\tilde{v} = \arg\min_{v_k} |(\cos\phi_{ik})| \tag{3}$$

3.2 メッシュ変形

変形処理では頂点 $\tilde{v}$ と中点 p_m を通る分割線を使用する．$\hat{v}$ は $\tilde{v}$ とは異り，分割線とメッシュのエッジの交点である．**図 3** に示すように，頂点 $\hat{v}$ をメッシュ f に対して垂直に移動させ，2 つのメッシュの法線と 2 つの推定法線の差を最小化する．このとき，目的関数は次のように表される．

$$F = \frac{n_i \cdot (p_i - \hat{v})}{|n_i||p_i - \hat{v}|} + \frac{n_j \cdot (p_j - \hat{v})}{|n_j||p_j - \hat{v}|} \tag{4}$$

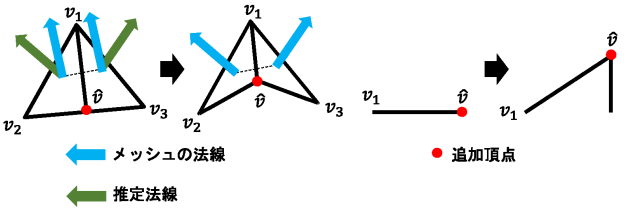


図 3: メッシュ変形の処理フロー
Fig. 3 Mesh Deformation Overview

4. 評価実験

実験では，THU-MVS Dataset [17] を単一オブジェクトのデータとして用い，SfM の評価用データセットである Benchmarking Camera Calibration 2008 [18] を屋外の建造物のデータとして使用した．また，分割と変形にもちいる閾値を 10 度としている．

また，本手法の有用性を示すために以下の 3 つのモデルを用意した．

評価指標として Earth Mover's Distance (EMD) と実行時間を定量評価の指標として用いる．EMD は以下の式 5 で表される．

$$d_{EMD}(X, Y) = \sum^{x \in X} \left[\min_{y \in Y} \{d(x, y)\} \right] \tag{5}$$

点群 X の最短距離の点が点群 Y に含まれる点から算出される．これは点群 X に属するすべての点に対して行われ，この合計を算出する．この合計が点群 X から Y への EMD となる．

本実験では MVS で生成したモデルを Groud Truth として扱い，提案手法で分割したメッシュモデル，および一様分布でランダムに生成した法線をもとに分割したメッシュモデルの 2 つと比較する．

4.1 定量評価

表 1 に定量評価を示す．結果として提案手法がすべてのデータにおいて MVS の生成よりも高速であった．さらに，同じ頂点数である Benchmarking Camera Calibration

2008 データセットの結果ではすべての EMD の値において一様分布のほうが提案手法よりも優れている．頂点数では THU-MVS データセットに比べて Benchmarking Camera Calibration 2008 データセットのほうが増加数が少ないことがわかる．

4.2 定性評価

定性評価では 4 章で示した 3 つのモデルと入力データを示す．図 4(b)，図 5(b)，図 6(b)，図 7(b) の左から，入力モデル，一様分布で分割されたモデル，提案手法で分割されたモデル，MVS で生成されたモデル (GT) である．図 5(b)，図 6(b)，図 7(b) のでは，提案手法で分割されたモデルは一様分布で分割されたモデルよりも平面部分のメッシュ分割が少ない事がわかる．

4.3 考察

定量評価より，Benchmarking Camera Calibration 2008 データセットでは一様分布の EMD の値が提案手法よりも優れており，さらに頂点数も同等であることがわかる．これは含まれるデータが建造物であり，単一オブジェクトに比べて平面が多いことが理由として考えられる．法線の推定結果が一定の精度を保っており，疎なメッシュモデルも平面を正しく表現できているため，分割対象と判断されていないと考えられる．この場合，提案手法では分割されないが，一様分布では分割されるということが起こるため，EMD の値が一様分布よりも劣っていると考えられる．図 5(b)，図 6(b)，図 7(b) の一様分布での分割のように，平面を分割するということは，分割する必要のない部分を分割していることとなる．これは過分割であり，メッシュ表現として冗長な表現となり，またテクスチャ貼り付けなどで複雑な作業となってしまう．この点に着目すると，一様分布を用いるよりも提案手法のように推定法線を用いた分割のほうが効率的であると考えられる．

5. おわりに

本研究では推定法線マップを用いた疎なメッシュモデルの分割と変形によるメッシュモデルの超解像を提案した．実験結果として，表面法線がメッシュモデルの分割と変形に効率的に働くことが示された．今後はバッチ処理や再帰的な処理の実装が考えられる．本研究では推定法線画像は既知としたが，実際の場面では画像からピクセル単位で法線を推定する必要がある．近年，深層学習を用いた単眼画像からの表面法線推定手法 [19] が提案されているため，こういった手法を適用することが考えられる．深層学習で推定された表面法線は多視点での整合性がとれていないため，整合性を考慮しつつ推定法線を利用するアルゴリズムの提案も必要である．

参考文献

[1] D. Maturana and S. Scherer: “Voxnet: A 3d convolutional neural network for real-time object recognition”, 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 922–928 (2015).
[2] C. Ruizhongtai Qi, H. Su, M. NieBner, A. Dai, M. Yan and L. Guibas: “Volumetric and multi-view cnns for object classification on 3d data”, pp. 5648–5656 (2016).
[3] G. Riegler, O. Ulusoy and A. Geiger: “Octnet: Learning deep 3d representations at high resolutions”, Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, Piscataway, NJ, USA, IEEE (2017).
[4] C. R. Qi, H. Su, K. Mo and L. J. Guibas: “Pointnet: Deep learning on point sets for 3d classification and segmentation”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017).
[5] M. Lhuillier and L. Quan: “A quasi-dense approach to surface reconstruction from uncalibrated images”, IEEE Transactions on Pattern Analysis and Machine Intelligence, **27**, 3, pp. 418–433 (2005).
[6] E. Kalogerakis, M. Averkiou, S. Maji and S. Chaudhuri: “3d shape segmentation with projective convolutional networks”, CoRR, **abs/1612.02808**, (2016).
[7] L. Yi, H. Su, X. Guo and L. J. Guibas: “Syncspeccnn: Synchronized spectral CNN for 3d shape segmentation”, CoRR, **abs/1612.00606**, (2016).
[8] C. Wu: “Towards linear-time incremental structure from motion”, 2013 International Conference on 3D Vision - 3DV 2013, pp. 127–134 (2013).
[9] C. Sweeney, T. Sattler, T., M. Turk, M. Pollefeys : “Optimizing the viewing graph for structure-from-motion”, 2015 IEEE International Conference on Computer Vision (ICCV), pp. 801–809 (2015).
[10] H. Cui, X. Gao, S. Shen and Z. Hu: “Hsfm: Hybrid structure-from-motion”, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2393–2402 (2017).
[11] F. Langguth, K. Sunkavalli, S. Hadap and M. Goesele: “Shading-aware multi-view stereo”, Proceedings of the European Conference on Computer Vision (ECCV) (2016).
[12] Johannes, E. Zheng, M. Pollefeys, J.-M. Frahm : “Pixelwise view selection for unstructured multi-view stereo”, 第 9907 巻 (2016).
[13] J. L., J. Frahm : “Structure-from-motion revisited”, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4104–4113 (2016).
[14] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu and Y. Jiang: “Pixel2mesh: Generating 3d mesh models from single RGB images”, CoRR, **abs/1804.01654**, (2018).
[15] C. Wen, Y. Zhang, Z. Li and Y. Fu: “Pixel2mesh++: Multi-view 3d mesh generation via deformation”, ICCV (2019).
[16] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman and D. Cohen-Or: “Meshcnn: A network with an edge”, CoRR, **abs/1809.05910**, (2018).
[17] S. SAKAI, K. ITO, T. AOKI, T. WATANABE and H. UNTEN: “Phase-based window matching with geometric correction for multi-view stereo”, IEICE Transactions on Information and Systems, **98**, 10, pp. 1818–1828 (2015).
[18] C. Strecha, W. von Hansen, L. Van Gool, P. Fua and U. Thoennessen: “On benchmarking camera calibration and multi-view stereo for high resolution imagery”,

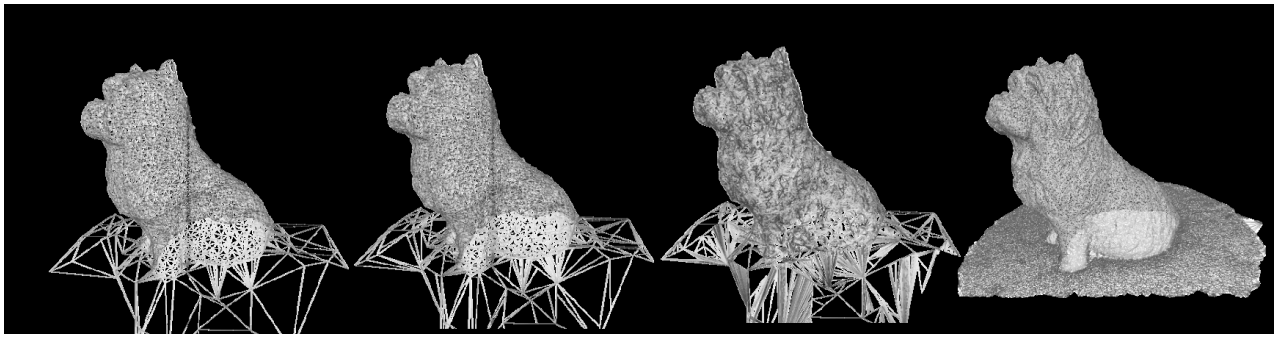
表 1: 各手法の EMD と実行時間
Table 1 EMD and execution time of each method

手法	頂点数	EMD 平均	EMD 分散	実行時間 (s)
Dog RGB				
入力	3.381×10^3	-	-	-
一様分布	1.008×10^4	1.763×10^{-2}	9.063×10^{-3}	-
提案手法	1.220×10^5	1.807×10^{-3}	6.248×10^{-3}	2.384×10^2
MVS	1.870×10^5	-	-	5.490×10^5
Herz-Jesus-P8				
入力	8.072×10^3	-	-	-
一様分布	8.517×10^3	$2.541 \times 10-2$	0.999×10^{-2}	-
提案手法	8.192×10^3	2.605×10^{-2}	1.049×10^{-2}	2.628×10^1
MVS	1.472×10^5	-	-	3.758×10^2
entry-P10				
入力	1.080×10^4	-	-	-
一様分布	1.197×10^4	2.298×10^{-2}	7.467×10^{-3}	-
提案手法	1.143×10^4	2.404×10^{-3}	7.555×10^{-3}	8.937×10^0
MVS	1.305×10^5	-	-	3.959×10^2
fountain-P11				
入力	1.380×10^4	-	-	-
一様分布	2.156×10^4	2.432×10^{-2}	4.275×10^{-2}	-
提案手法	2.105×10^4	2.548×10^{-2}	3.322×10^{-2}	4.256×10^0
MVS	1.837×10^5	-	-	7.999×10^2

2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008).

[19] H. Zhan, C. S. Weerasekera, R. Garg and I. D. Reid: “Self-supervised learning for single view depth and surface normal estimation”, CoRR, **abs/1903.00112**, (2019).

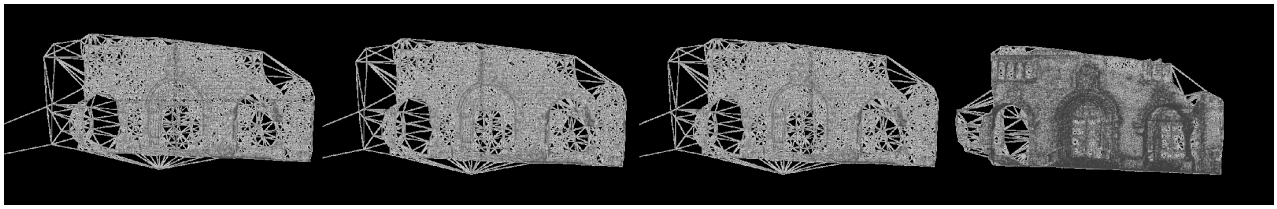
(a) 入力画像



(b) 実行結果

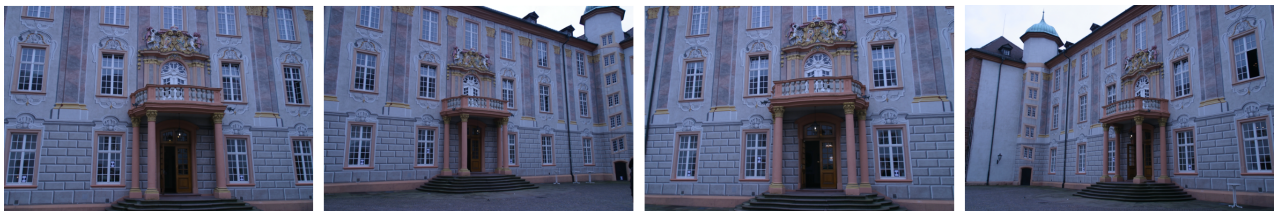
図 4: THU-MVS Dataset
Fig. 4 THU-MVS Dataset

(a) 入力画像

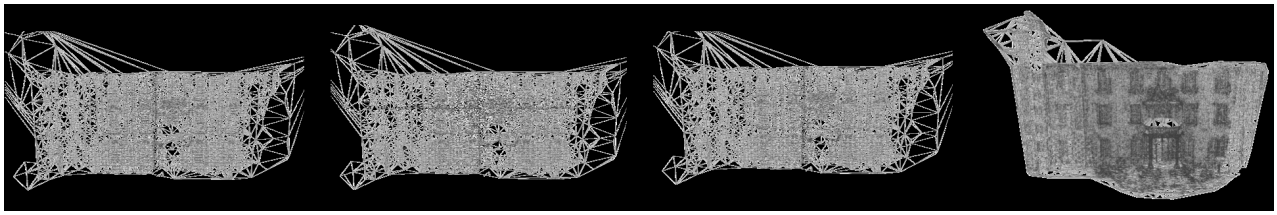


(b) 実行結果

図 5: Herz-Jesus-P8
Fig. 5 Herz-Jesus-P8



(a) 入力画像

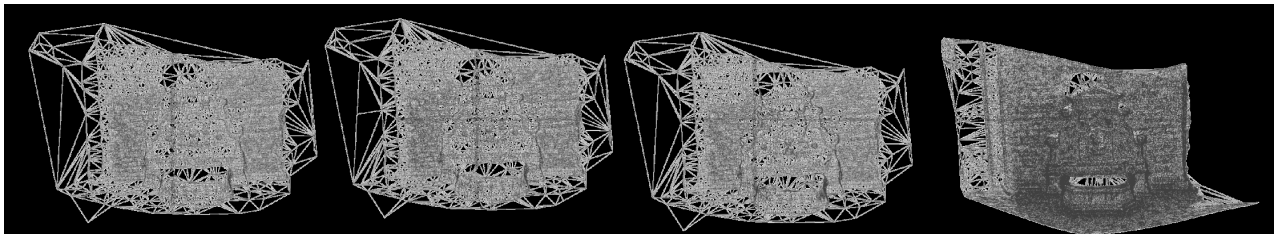


(b) 実行結果

図 6: entry-P10

Fig. 6 entry-P10

(a) 入力画像



(b) 実行結果

図 7: fountain-P11

Fig. 7 fountain-P11