平成20年度卒業論文

三次元スキャニングにおける
曲率極値線を用いた溶接部分解

Sheet Metal Assembly Decomposition along Welded Lines
Using Ridge-Valley Curves in 3D Scan Mesh Processing

指導教官
鈴木宏正教授

東京大学工学部システム創成学科
知能設計コース

工070894
石井遼介

概要

現物形状をスキャンしたデータを用いる現物ベースの CAE シミュレーションでは CAD データをベースとしたシミュレーションに比べ, 予測精度の画期的向上·試作品の完成度向上など多くの利点があるが, 問題点も多い. その問題点の 1 つは, アセンブリ部品を丸ごとスキャンした際,CAE のモデルを作成するためには, それを部品ごとに分解し, かつ部品の接合部分を認識する必要がある点である. 現物ではそれぞれ異なった部品であっても, 溶接などで接触していた場合, スキャニングでは, 1 つのオブジェクトとして認識されてしまう. 現在この分解工程はすべて人の手によって行われており, そこにかかる時間とコストが現物ベースの CAE シミュレーションの大きなボトルネックとなっている. 本研究では, そのような分解問題のうち, 自動車ボディの内板構造のような, 板金部品をスポット溶接によって接合したアセンブリを対象とし, 溶接部の認識と, 溶接部でのポリゴンデータの分離を自動的に行う事を考える.

目次

1 序論：現物ベースの CAE シミュレーション 1–1
1.1 背景：CAE シミュレーションの概要 1–2
1.1.1 本研究のねらい：作業効率の向上と目視確認 1–3
1.2 対象領域 1–4
1.2.1 現物 1–4
1.2.2 スキャニング 1–4
1.2.3 データ加工 1–5
1.3 解決の基本方針 1–5
1.3.1 問題 1. 溶接部の検出・抽出と曲率極値線 [1] 1–6
1.3.2 問題 2. 抽出したものの切り分けとグラフカット [5] 1–6
1.3.3 解決手法の適用 1–6
1.4 目標と評価の枠組み 1–6
1.4.1 効率:時間と手間の評価 1–6
1.4.2 品質:極値線の品質とカットの品質 1–7
1.5 本論文の構成 1–7

2 関連研究 2–1
2.1 曲率極値線計算手法 [1] 2–2
2.1.1 陰関数曲面 2–2
2.1.2 フィッティング関数 [2] 2–3
2.1.3 曲面の特徴線 [3] 2–7
2.1.4 曲率極値線 2–12
2.2 グラフカット 2–14
2.2.1 グラフ理論 [4] 2–14
2.2.2 最大流最小カット定理 2–16
2.2.3 グラフカット [5] 2–21
2.2.4 インタラクティブ・グラフカットの研究 [5] 2–22

3 分割手法 3–1
3.1 分割法の概要 3–2
3.2 (1) 極値線の導出 3–4
3.2.1 トレードオフパラメータ t と極値線の関係 3–4
3.3 (2) コスト設定 3–11

3.3.1 メッシュとグラフ 3–11
3.3.2 (i) ターミナル・ノード間の設定 3–11
3.3.3 (ii) ノード・ノード間の設定 3–12
3.4 (3) グラフカット 3–17
3.4.1 カット目標 3–17
3.4.2 3 領域へのカット 3–18
3.5 なぜ提案手法なのか 3–19

4 実装と結果 **4–1**
4.1 スキャンと現物 4–2
4.2 グラフカット 4–4
4.3 評価と改善 4–5
4.3.1 時間の評価 4–5
4.3.2 品質の評価 4–6
4.3.3 曲率極値線の導出と評価 4–6
4.4 ソースとシンク指定の工夫 4–27
4.4.1 ユーザ指定の最適化 4–27
4.4.2 ファジー領域 4–41

5 結論と展望 **5–1**
5.1 結論 5–2
5.2 課題および展望 5–4
5.2.1 アーク溶接 5–5
5.2.2 複数回の分割 5–5
5.2.3 ペイントから選択形式へ 5–5

謝辞 **A–i**

図目次

1.1 本研究の対象と位置づけ 1–3
1.2 薄板部品を溶接した組み立て構造 1–4
1.3 現物の模式図 1–4
1.4 溶接部品スキャンデータ例と溶接端およびカット目標 1–5
1.5 溶接部分解の模式図 1–5
1.6 評価の枠組み 1–7

2.1 陰関数によって表現された 3 次元空間上の表面 2–2
2.2 関数による点群のフィッティング 2–4
2.3 フィッティング円の大小 2–4
2.4 フィッティングのトレードオフ 2–5
2.5 陰関数による点群の近似 2–6
2.6 曲面：法線ベクトルと接ベクトル 2–7
2.7 曲率 κ の,κ_{max}, κ_{min} 2–9
2.8 接ベクトル t の,t_{max}, t_{min} 2–11
2.9 山谷線 2–12
2.10 山谷線 2–13
2.11 グラフの種類 2–14
2.12 同型グラフ 2–15
2.13 部分グラフの種類 2–15
2.14 有向グラフの概要 2–16
2.15 ネットワーク 2–17
2.16 ネットワークとその値 2–18
2.17 ネットワークのカット 2–19
2.18 グラフの増大道 2–19
2.19 グラフカットの概要 2–21
2.20 ソースとシンク 2–21
2.21 グラフとその隣接行列.N の頂点傍の数値は頂点番号. 弧傍の数値は容量. . . . 2–22
2.22 グラフカットの 2 次元画像への適用 [5] 2–23
2.23 3 × 3 ピクセルとグラフ 2–24
2.24 グラフカット及びその応用研究と他セグメンテーション手法の比較 [7] . . . 2–25

3.1 解決フローチャート 3–3
3.2 サンプルデータ 1 の全体像と拡大図 3–4

3.3 山谷線の精度 (サンプルデータ 1). 3–6
3.4 サンプルデータ 2 の全体像と拡大図 3–7
3.5 山谷線の精度 (サンプルデータ 2) 3–8
3.6 サンプルデータ 3 の全体像と拡大図 3–9
3.7 山谷線の精度 (サンプルデータ 3) 3–10
3.8 3 次元ペイントシステム 3–12
3.9 通常の Edge/山線通過 Edge/谷線通過 Edge 3–12
3.10 コスト設定とパラメータ 3–13
3.11 閾値の設定 3–15
3.12 $\mathcal{A}$ の変化と結果 3–16
3.13 分解目標の詳細な模式図 3–17
3.14 スポット溶接を関数当てはめした時の特徴 3–18
3.15 山線で 1 回, 谷線で 1 回カットし, 狭間領域を捨てる 3–18
3.16 当初想定していたコスト設定 3–19

4.1 非接触光学式 3 次元デジタイザ 4–3
4.2 スキャンのピッチと深度 4–3
4.3 カット結果 (サンプルデータ 1) 4–4
4.4 カット結果 (サンプルデータ 2) 4–4
4.5 カット結果 (サンプルデータ 3) 4–5
4.6 各プロセスと所要時間 4–5
4.7 品質の良い t の値とサンプルデータ 1,2,3 4–7
4.8 サンプルデータ 1,2,3 における t 値と極値線 ($t = 0.001 - 0.005$) 4–9
4.9 サンプルデータ 1,2,3 における t 値と極値線 ($t = 0.006 - 0.01$) 4–10
4.10 サンプルデータ 1 における t 値と時間, 山谷線の本数, エラー値の関係 4–11
4.11 サンプルデータ 2 における t 値と時間, 山谷線の本数, エラー値の関係 4–12
4.12 サンプルデータ 3 における t 値と時間, 山谷線の本数, エラー値の関係 4–13
4.13 理想メッシュ 4–14
4.14 理想メッシュの高さと t 4–15
4.15 理想メッシュの粗密と t 4–16
4.16 Laplacian Flow 法の概要 4–17
4.17 Laplacian Flow によるスムージング結果 (サンプルデータ 1) 4–18
4.18 Laplacian Flow によるスムージング結果 (サンプルデータ 2) 4–19
4.19 Laplacian Flow によるスムージング結果 (サンプルデータ 3) 4–20
4.20 サンプルデータ 1 の $N = 1$ スムージングへの t の変化 4–22
4.21 サンプルデータ 2 の $N = 1$ スムージングへの t の変化 4–23
4.22 サンプルデータ 3 の $N = 1$ スムージングへの t の変化 4–24
4.23 サンプルデータ 2 の特徴線に沿ったカット 4–27
4.24 2 枚のスポット溶接データの構造 4–28
4.25 サンプルデータの極値線とエリア 4–29
4.26 ユーザ指定条件の 3 パターン 4–31
4.27 特徴エリア導入の目標 4–32

4.28 特徴エリアの例 4–33
4.29 エリア分け問題 4–33
4.30 台形領域 4–34
4.31 エリア分け問題 2 4–34
4.32 全ての特徴エリアをシンク/ソース指定 4–36
4.33 溶接端 (B) の特徴線と接する特徴エリアのみを指定 4–37
4.34 ルール (2') を満たすように指定した, 溶接端部特徴線と接する特徴エリア . . 4–37
4.35 ソース指定位置の変更と結果 4–39
4.36 溶接端部における極値線の欠け 4–40
4.37 ファジー領域の概要 4–41
4.38 ファジー領域の適用 4–42

5.1 パラメータの変化と所要時間および品質の関係 5–3
5.2 スポット痕と曲率極値線 5–4
5.3 アーク溶接のカット 5–5
5.4 サンプルデータ 3 の 4 部品への分解 5–6

表目次

1.1 形状データとして現物と CAD データを用いた CAE の一般的な比較 1–2

3.1 サンプルデータ 1 の極値線の計算 3–4
3.2 サンプルデータ 2 の極値線の計算 3–7
3.3 サンプルデータ 3 の極値線の計算 3–9
3.4 2 回カットした際の分類 3–18

4.1 品質の良い t の値と計算時間 4–6
4.2 サンプルデータ 1, $t = 0.001$ から $t = 0.01$ まで 4–8
4.3 サンプルデータ 2, $t = 0.001$ から $t = 0.01$ まで 4–8
4.4 サンプルデータ 3, $t = 0.001$ から $t = 0.01$ まで 4–8
4.5 $t = 0.005$ に於けるサンプルデータ 1,2,3 のスムージング N=1,3,5 表 4–21
4.6 サンプルデータ 1 の $N = 1$ スムージングへの t の変化 4–25
4.7 サンプルデータ 2 の $N = 1$ スムージングへの t の変化 4–25
4.8 サンプルデータ 3 の $N = 1$ スムージングへの t の変化 4–25

第 1 章

序論：現物ベースのCAEシミュレーション

第 1 章

序論：現物ベースのCAEシミュレーション

1.1　背景：CAE シミュレーションの概要

Computer Aided Engineering (CAE) シミュレーション技術では, 形状データを基にし, 有限要素法, 有限差分法, 境界要素法などの手法により, 構造解析, 応力解析, 衝撃解析など, さまざまな解析を行う事ができる.
その中でも, 本研究では現物ベースの CAE シミュレーションを行う. こ れは, 一般的な CAE シミュレーションで用いられる Computer Aided Design(CAD) データを基にした解析ではなく, 実際の物をスキャンした,3 次元スキャンデータを用いた解析手法である.

現物ベースの CAE シミュレーションは, 一般的な CAE シミュレーションに比べ,
(1)CAD データが存在しない場合でもシミュレーションできる.
(2) 予測精度が画期的に向上する. [1]
(3) 試作品の完成度を向上させられる.
といった利点がある. 反面, アセンブリ部品を丸ごとスキャンした場合,CAE のモデルを作成するためには, それを部品ごとに分解し, かつ部品の接合部分を認識する必要がある. 現物ではそれぞれ異なった部品であっても, 溶接などで組み立てられていた場合, スキャニングでは,1 つのオブジェクトとして認識され, そのままでは解析はままならない. 元のパーツへと分解すること, すなわち, **データの加工**が必要である.

表 1.1: 形状データとして現物と CAD データを用いた CAE の一般的な比較

	現物ベース	CAD データ
データ処理	必要	不要
解析精度	現物に近い	データ通り

このような現物ベース CAE シミュレーションのプロセスは,3 段階で成り立っている. すなわち:
(1) 光学式スキャナによるスキャニング
(2) データ加工

[1]CAD データからオブジェクトを作る際に, 生産現場で現物合わせを行っているような場合特に有効である.

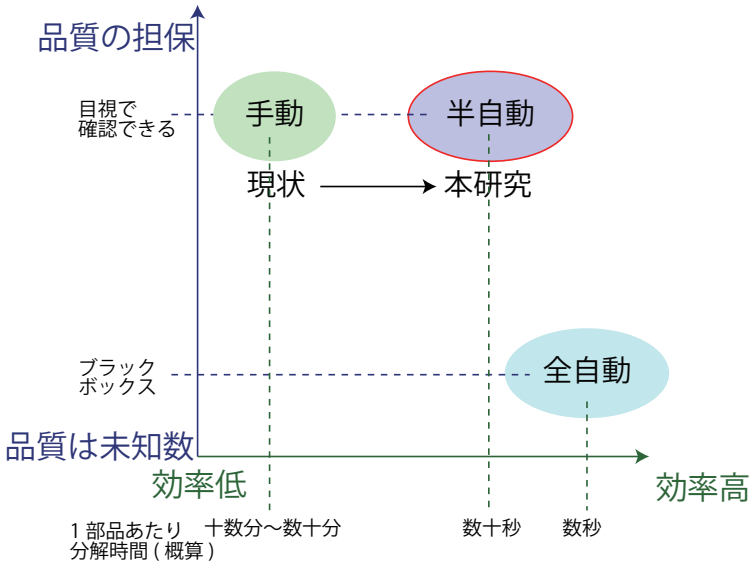


図 1.1: 本研究の対象と位置づけ

(3) 解析 (シミュレーション)
の 3 段階である.

本研究では, この現物ベース CAE シミュレーションのボトルネックである, データ加工のプロセスの改善を目指した. 特に, スポット溶接された金属板のスキャンデータに対し, 金属板を半自動で分離する事に集中した.

1.1.1 本研究のねらい：作業効率の向上と目視確認

CAD 上での溶接部での切断は, 自動車の性能評価などの解析の為, 重要ではあるが人の手で行うには煩雑かつ創造的でない作業である. 本研究によって, そのような労力を削減し, 現物ベースの CAE シミュレーションのボトルネック解消の一助としたい.

本研究のねらいは, 今まで手動で行われていたデータ加工のプロセスを, 半自動化することによって, 現物 CAE シミュレーションの効率化を図る. 同時に, このようなプロセスは, 間違った処理を行っていた場合, 自動車等のシミュレーションによる性能評価に影響を及ぼすため, 目視による確認が不可欠である.

すなわち, 本研究では, 図 1.1 に表現されているように, 品質を維持しながら, スポット溶接部分での部品分解の効率を飛躍的に高める事を目指した.

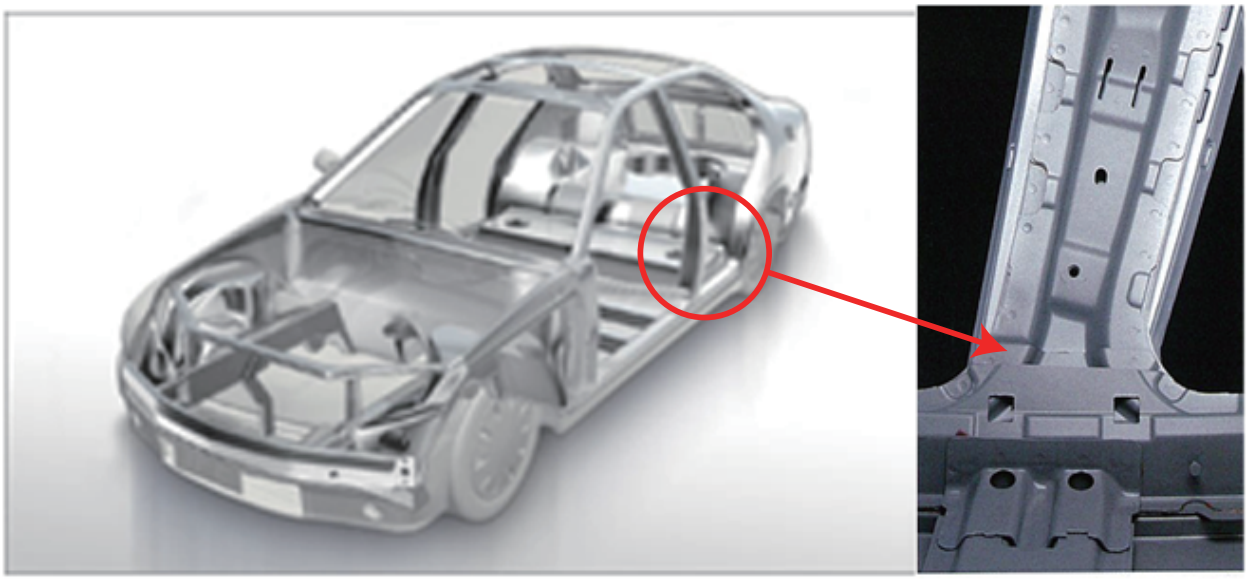


図 1.2: 薄板部品を溶接した組み立て構造

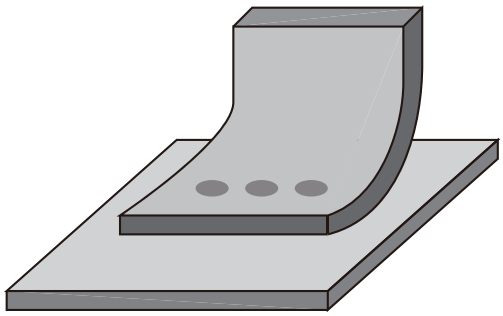


図 1.3: 現物の模式図

1.2　対象領域

1.2.1　現物

本研究で対象とする現物は, 自動車ボディの内板構造のような, 板金部品をスポット溶接した組立構造を想定する.

図 1.2 のようにスポット溶接されたものであり, 模式的には図 1.3 のようなものである.

また, 本研究では, 個々の部品の形状が溶接した部分以外は滑らかで, 曲率が大きな部分は存在しないことを仮定する. 一般的なプレス加工品に対して, この仮定はよく当てはまる.

1.2.2　スキャニング

現物スキャンのスキャン方式には主に,X 線 CT 方式, 接触式, 光学式の 3 つの方式があるが, 本研究では光学式スキャンによるデータを対象とする.

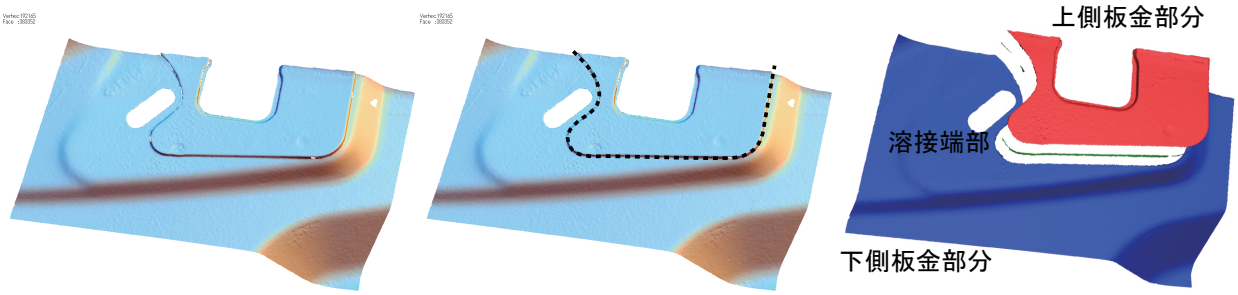


図 1.4: 溶接部品スキャンデータ例と溶接端およびカット目標

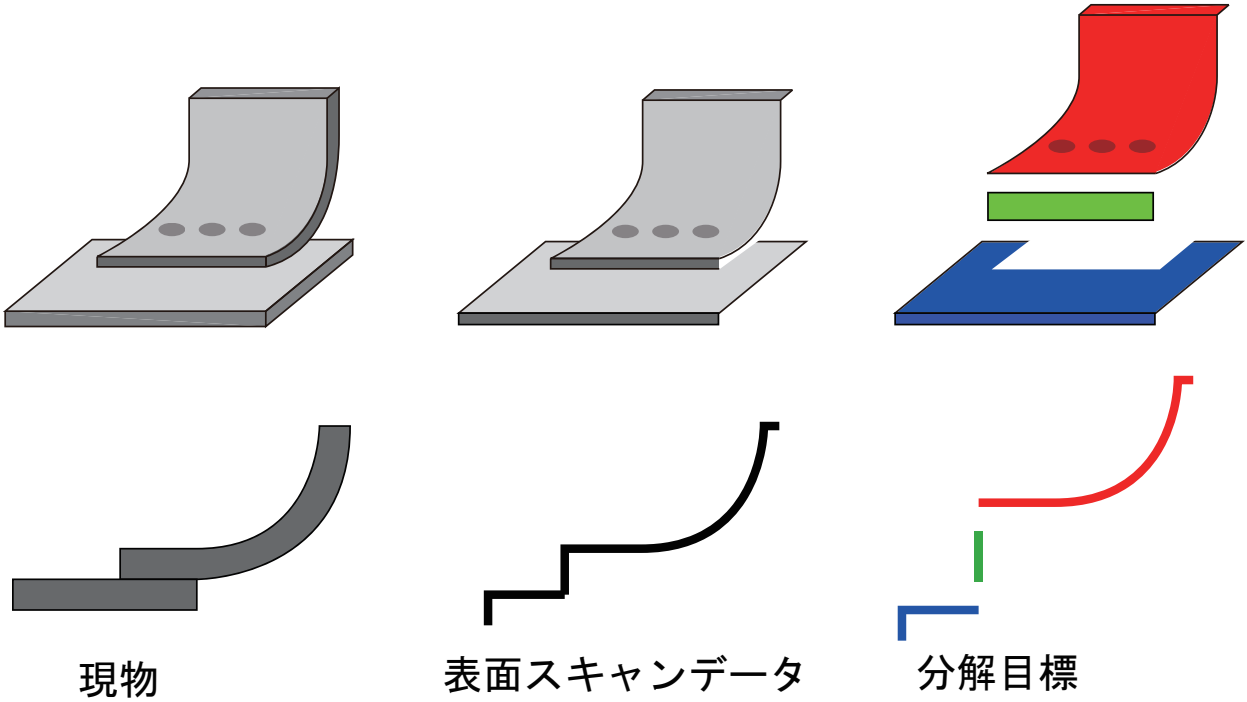


図 1.5: 溶接部分解の模式図

1.2.3　データ加工

こうして光学式 3 次元スキャナにより, 計算機上に取り込んだ 3 次元メッシュデータを, 解析に使えるように加工したい. 図 1.4a のような, スキャンデータを, 図 1.4b の点線でカットし, 図 1.4c で赤で塗り分けられた上側板金部分と青で塗り分けられた下側板金部分, および緑色に塗り分けられた部分へと,3 分割することを目指した. なお, 本論文では, 図 1.4b 点線部のような, 二枚の板金が接合されている端部の事すなわち, 図 1.4c で緑色に塗り分けた部分を, 溶接端と呼ぶ.

また, この分解を模式的に表すと図 1.5 のようになる.

1.3　解決の基本方針

以上のような溶接端による分解を行うため,2 つの問題を解くことで, 解決を試みる. **問題 1. 溶接部の検出/抽出を行うこと. 問題 2. 抽出した溶接部で,3 次元メッシュを 2 つに切り分ける**

事である.

1.3.1 問題 1. 溶接部の検出・抽出と曲率極値線 [1]

板金部品の溶接部では, 板が 2 枚重ねになっており, その内の一極の端部が現れ, その板端で形状が大きく変化する. このような溶接部では, 形状の曲率が大きく変化すると期待される. そのような曲率の変化の検出に適した手法に, 曲率極値線に関する研究 [1] がある.[1] では頂点群に対し陰関数曲面によるフィッティングを行い, その曲面の曲率の極値を求め, その極値を繋いで線状にする, という手法が提案されている. 溶接する時, 特にスポット溶接であれば, 接着部分が非連続的に変化し, 関数をフィットさせた時にその溶接端で極値になりやすい. 従って, 溶接端の検出・抽出に適した手法だと言える.

1.3.2 問題 2. 抽出したものの切り分けとグラフカット [5]

抽出した溶接端を用い,3 次元メッシュを分解するのに適した手法がある．グラフカット [5] である．グラフカットは, グラフ理論における最大流最小カット定理 (Max-flow min-cut theorem) を応用することにより,1 つのグラフを,2 つの部分グラフに分解することができる. スキャンした三角形メッシュは, 頂点を稜線で接続したグラフであり, グラフカットは必ず 2 つの部分グラフに分解できるという点で本研究に適している.

1.3.3 解決手法の適用

溶接部は, 曲率の変化が大きいところとして認識できるので, メッシュ上の曲率極値線を求め, さらに, メッシュの頂点を, グラフの頂点としてグラフカット法によって分割した. この際に, ユーザーが補助的に部品に関する情報を入力することによって, 多数の極値線の中から適切なもので分割できるようにした. 実際のデータ例に適用し, 有効性を確認するとともに, ユーザー指定のガイドラインも明らかにした.

1.4 目標と評価の枠組み

本研究はそもそも, 現物ベースの CAE シミュレーションにおける, 品質を損なわないまま, 効率の改善を目指したものであった. そこで, これまでの結果を, その**品質と効率という 2 軸**で捉え, 評価と改善を行う. 品質というのは,**(i) 極値線の場所は溶接端部に沿って適切に検出できているか (ii) カットはそのような極値線に沿ってきちんと行われているか**を問うものであり, 効率というのは,**(i) 計算機による時間が短縮できるか (ii) 人の手間を省けるか**を考えるものである.

1.4.1 効率:時間と手間の評価

現物ベース CAE による, 溶接端部での分解は, 従来手法も, 本手法も, 人と機械がインタラクションを行うシステムであるため, 個人差があり, 定量的な評価は難しい. だが, 図 1.4 に示したサンプルデータ 1 のようなデータ (約 80[mm]×150[mm]) のものでも, 十数分程度かかり, その時間全てにわたって人の手によるデータ加工が行われる. このような時間と手間を 1 部を機械に担わせ, 時間と労力を減らす事を目指す.

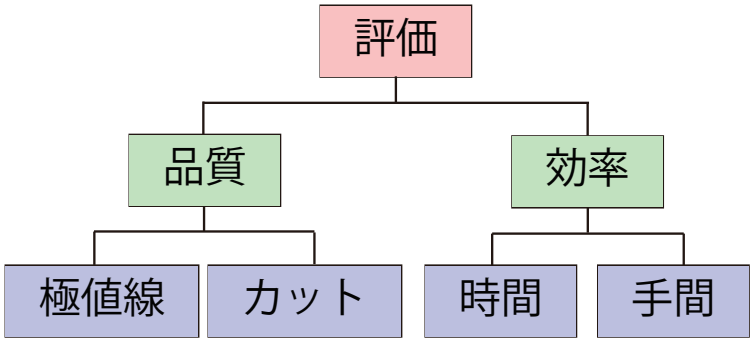


図 1.6: 評価の枠組み

1.4.2　品質:極値線の品質とカットの品質

品質には 2 段階あり, それは既にみたように, 極値線の品質とカットの品質である. また, 極値線の品質がカットの品質を制約する. 極値線がうまく溶接端部を検出しているかという**極値線の品質**と, 極値線がうまく検出できている場合に, その極値線に沿ってカットがうまく出来ているかという**カットの品質**はそれぞれ独立に考える事ができる. 例えば, 極値線の品質が悪くても, その極値線に沿ってカットが成されていれば, カットの品質は良いと言ってよい.

1.5　本論文の構成

以上見てきたように, 本研究では, スポット溶接された現物からスキャンを行った 3 次元メッシュに対し, 自動的に部品分解を行うことを目標とする. その目標を達成する為に, **問題 1. 溶接部の検出/抽出を行うこと. 問題 2. 抽出した溶接部で,3 次元メッシュを 2 つに切り分ける事**という 2 つの問題を設定し, 解決を試みた.

本論文では, その 2 つの問題を解決する礎となる 2 つの手法について, まず第 2 章で関連研究として解説する. 具体的には, 問題 1 に適した手法である, 曲率極値線の計算手法についてみる. そのため, 陰関数曲面, 点群への関数フィッティング, 曲面上の特徴線, について解説し, 溶接部など曲率の大きな部分をうまく検出する曲率極値線という手法について解説する. 同時に問題 2 に適した手法である, グラフカットという手法について解説する. 後に見るが, 曲率極値線は不要線, 及び不連続部分も多く, そのままではうまく溶接端部による分解ができないため, グラフを 2 つの部分グラフへ分解する本手法が適している. 具体的には, 一般的なグラフ理論について解説を行い, 本研究への応用を考えるきっかけとなった,[5] グラフカットによるセグメンテーションについて見る.

次に, 第 3 章では, 第 2 章でみた手法について, それらをどのように用い, 分割を行うのかを見る. そのプロセスは, 曲率極値線を計算し, その精度を決めるパラメータ t についての議論を行い, 品質の良い曲率極値線, すなわち, 溶接部分によくフィットする曲率極値線を導出したのち, その曲率極値線の値とユーザ指定を参考に, グラフ理論における容量を設定し, うまくカットが行われるよう, パラメータの調整を行う.

第 4 章では実装した結果およびその評価と改善を行う. 評価軸は, 効率と品質であったが, 前者については 1 連のカットプロセスの所要時間において, 占める割合の大きな曲率極値線の計算およびユーザの指定における時間の改善を目指し, また, 後者においてはスムージング, ファジー領域といった工夫を行い, その結果の進歩を見る.

第 5 章では, 結論と展望を述べる. 第 4 章までの議論から発見した知見や, 今後の課題, 展望などについて述べる.

第 2 章

関連研究

第 2 章

関連研究

2.1 曲率極値線計算手法 [1]

この手法は，メッシュ頂点において曲率の微分値を算出し，隣接する頂点間において曲率の極値の存在 (微分値が零) を検出する．メッシュの辺上で検出された曲率極値線上の点は，三角形上で線分として結ばれる．そのような曲率極値線は, 溶接端部をよく検出する．

2.1.1 陰関数曲面

3 次元空間 xyz が定義された, 連続な陰関数 $\mathcal{F}(x,y,z)$ が与えられた時,

$$\mathcal{F}(x,y,z) = 0 \tag{2.1}$$

は曲面を表す.

例えば, 関数 $\mathcal{F}(x,y,z) = x^5 + x^4 + y^4 + z^4 + x^2 - x - y - 3 = 0, |_{(x,-2,2),(y,-2,2),(z,-2,2)}$ は,3 次元空間上では, 図 2.1 のようになる.

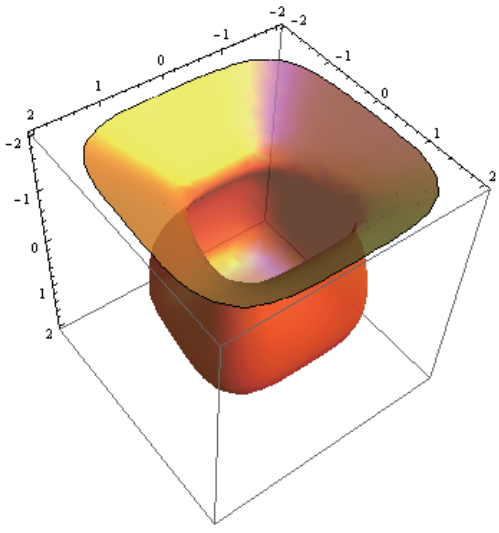


図 2.1: 陰関数によって表現された 3 次元空間上の表面

2.1.2 フィッティング関数 [2]

スキャン点群から, 点群が曲面上にくるような陰関数曲面 $\mathcal{F} = 0$ を作ることができる. その手法の一つに, 3 次元空間上の点群データが与えられた時, その点群を重複を許すいくつかの半径 R_i 内の領域で被覆し, その小領域を最小自乗法を用いて決定する陰関数であてはめる事ができる. i 番目の領域にあてはめられた関数を $f_i(x, y, z) = 0$ と書くと, 当該領域と重複部分を持つ領域でのフィッティング関数と, 滑らかに繋ぐ必要がある. その為, 重み関数 $w_i(x, y, z)$ を用いて, 全点群データをカバーするフィッティング関数 $F(x, y, z)$ として,

$$\mathcal{F}(x, y, z) = \frac{\sum_i w_i(x, y, z) f_i(x, y, z)}{\sum_i w_i(x, y, z)} = 0 \tag{2.2}$$

のように定義できる. また, 重み関数 w_i は, 円の中心で 1, 終端で 0 となるような関数で, 本研究では, 中心からの距離を引数とする 2 次の B スプライン曲線を用いた.
2 次の B スプライン曲線 $B(x)$ は, 式 2.3 のように定義される.

$$B(x) = \begin{cases} -x^2 + 0.75 & if\ 0 \le x \le 0.5 \\ 0.5x^2 - 1.5x + 1.125 & if\ 0.5 \le x \le 1.5 \\ 0 & otherwize \end{cases} \tag{2.3}$$

また, フィッティングの誤差 ϵ は, 元々の頂点 P_j の座標を $P_j = (x_j, y_j, z_j)$ とおくと, $f_i(P_j) = 0$ となるはずであるから, 以下のように定める.

$$\epsilon^2(R_i) = \sum_j (w_i(P_j) f_i(P_j))^2 \tag{2.4}$$

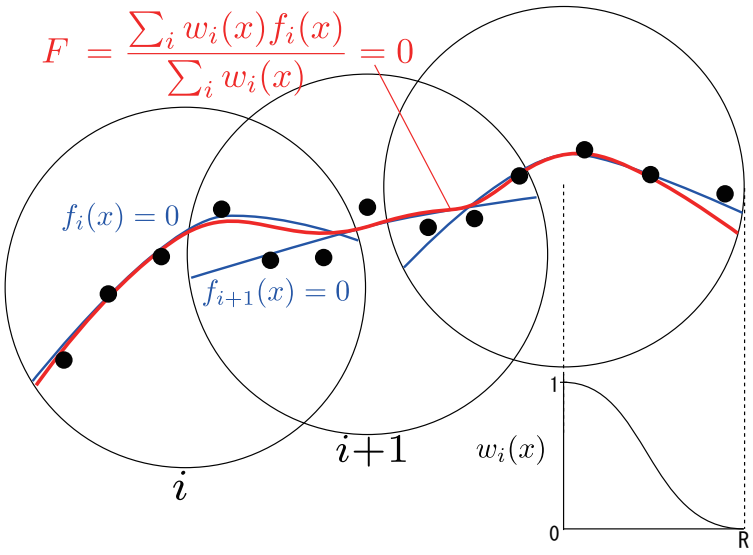


図 2.2: 関数による点群のフィッティング

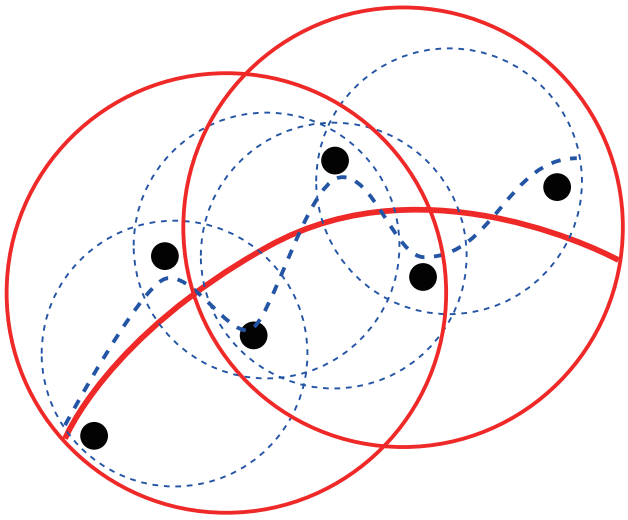


図 2.3: フィッティング円の大小

簡単のため, 以下,2 次元の場合で説明する. 点群データに, 関数 $\mathcal{F}(x)$ をフィットさせる様子を図 2.2 に図示する. 点群に対し, 半径 R_i の領域 i 内に存在するパラメータを陰関数 $f_i(x) = 0$ で, 誤差 ϵ が最小となるようフィッティングし, 領域 i で算出したフィッティング関数と領域 $i + 1$ で算出したフィッティング関数を重み関数 $w_i(x)$ により合成し, 全領域に渡って連続なフィッティング関数を導出する.

誤差と平滑性のトレードオフ

この時, フィッティング円の半径 R_i の取り方によって図 2.3 の赤実線と青点線で示したような, 違いが生まれる.

すなわち, 半径 R_i を図 2.3 の赤円のように大きくすると滑らかに全体をフィッティングさせる事ができるが, 誤差が大きくなる. 青色点線円のように半径 R_i を小さくすると誤差 ϵ は

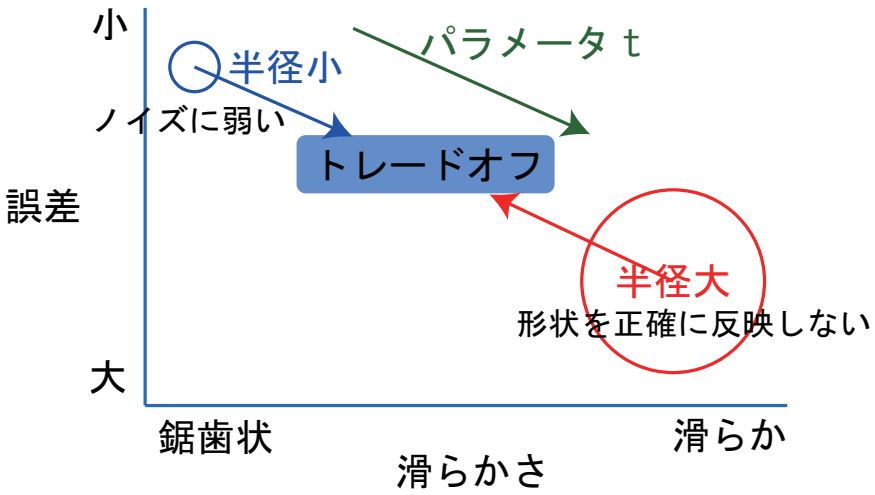


図 2.4: フィッティングのトレードオフ

小さくなるが, 滑らかさが失われる.

ここで, 図 2.4 に示したような, トレードオフを考える必要が生じる. そのためにトレードオフパラメータ t を導入する. ここで, 誤差も滑らかさも, 領域 i を覆う円の半径 R_i によって決まる,R_i の関数だから, 誤差を $\epsilon^2(R_i)$, 滑らかさを $\frac{1}{R_i^2}$ とおくと,

$$E(R_i) = \epsilon^2(R_i) + t \cdot \frac{1}{R_i^2} \rightarrow min \tag{2.5}$$

となるような最適化問題を解き, 点群の近似を行い, 順次 t を変化させていくことでトレードオフの結果を調べる事ができる. 以下, 具体的な t の値と決定方法は 3.2 章にてデータを基に議論する.3 次元に拡張する場合, 円は球と読み替えられる. 図 2.5 に球によるフィッティング及び領域 i 内のフィッティング関数 $f_i(x, y, z) = 0$ を可視化したものを示す. なおフィッティング関数 $f_i(x, y, z) = 0$ は, 可視化のため, 球内の曲面を半分のサイズに縮めて表現をしている.

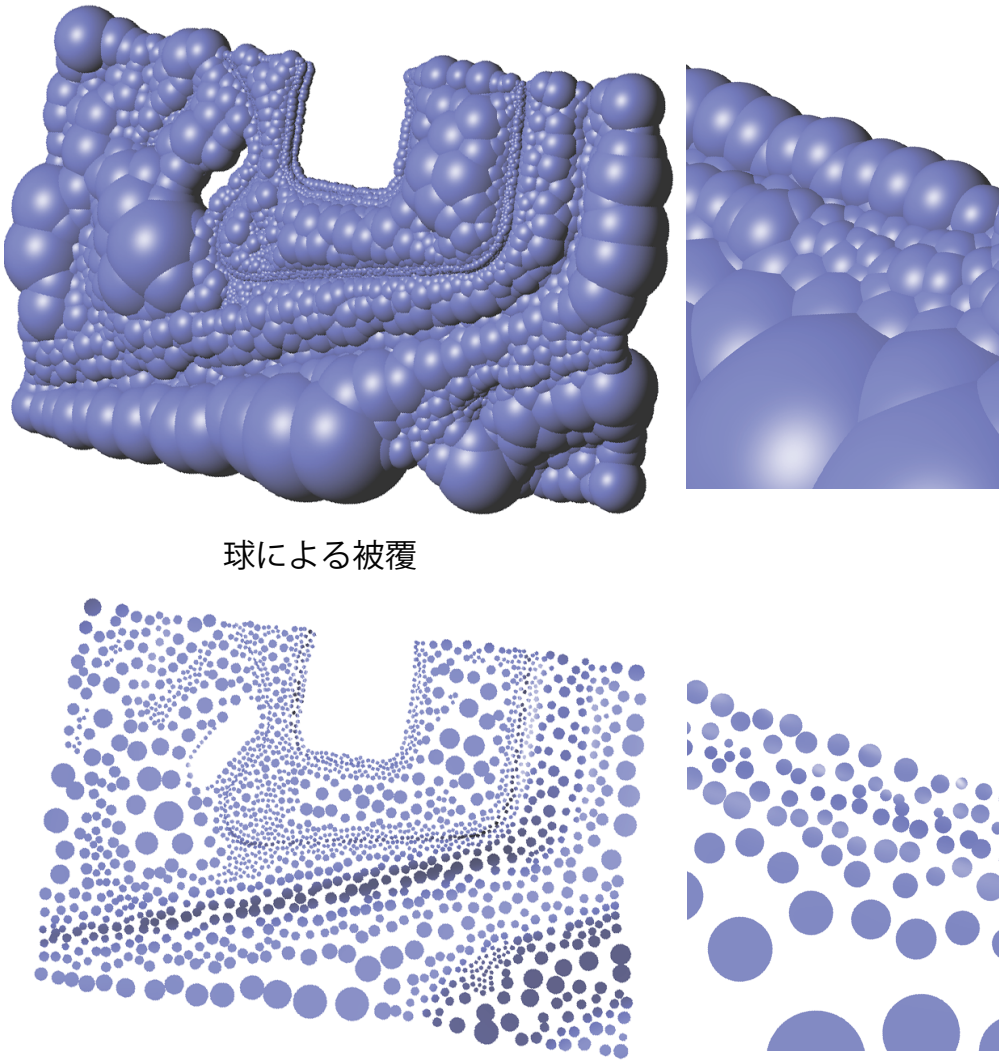


図 2.5: 陰関数による点群の近似

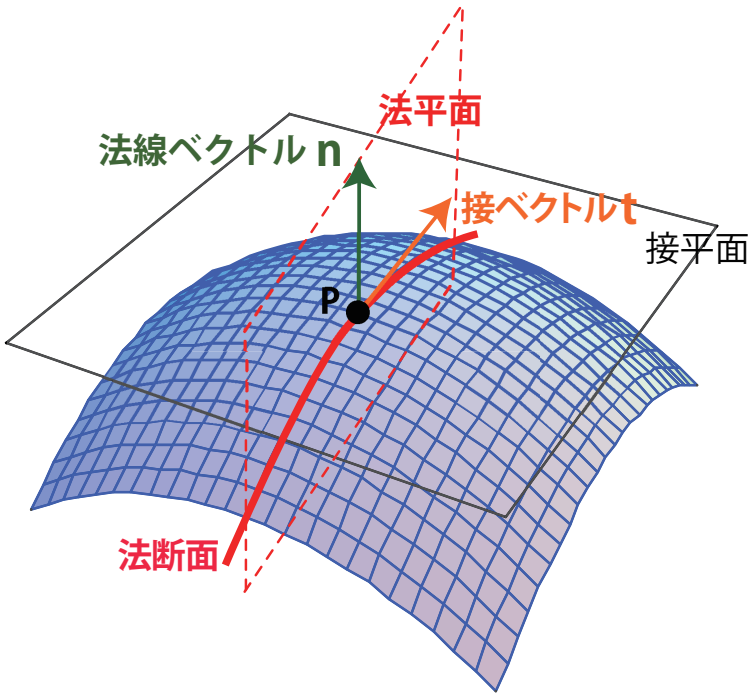


図 2.6: 曲面：法線ベクトルと接ベクトル

2.1.3 曲面の特徴線 [3]

一般に,3 次元空間上の曲面がある時, 曲面上にはその特徴を表す曲線がある. そのような特徴線には曲面の形状把握や曲面の特徴の評価に役立つものがある. 曲面上の特徴線には 3 種類ある. すなわち,**(i) 曲面の幾何形状に固有の曲線**, **(ii) 空間における位置や方向に依存する曲線**, **(iii) 入射光や, 表面反射特性などの環境に依存する曲線**である. ここで, 曲率線とは (i) の分類にあたる. 以下, 曲率極値線について説明するために, はじめに曲面と曲率の定義を見, 曲率線について説明する.

曲面と曲率

曲面は一般的に, $s(u,v) = \{x(u,v), y(u,v), z(u,v)\}, 0 \leq u, v \leq 1$ で表され,u, v に関数関係があると,$s(u,v)$ は曲面上の曲線を表す. すなわち, 曲面上の曲線 $s(u,v)$ は,u と v の関係によって定まる.

Fig.2.6 に示すような曲面 s 上の点 P における単位接線ベクトル $\mathbf{t}$(のひとつ) は,

$$\mathbf{t} = \frac{d\mathbf{s}}{|\mathbf{s}|} = \mathbf{s}_u \left(\frac{du}{ds}\right) + \mathbf{s}_v \left(\frac{dv}{ds}\right) \tag{2.6}$$

ただし,$\mathbf{s}_u = \dfrac{\partial s}{\partial u}, \mathbf{s}_v = \dfrac{\partial s}{\partial v}$ である.

また, 単位法線ベクトル $\mathbf{n}$ は,

$$\mathbf{n} = \frac{\mathbf{s}_u \times \mathbf{s}_v}{|\mathbf{s}_u \times \mathbf{s}_v|} \tag{2.7}$$

$\mathbf{n}, \mathbf{t}$ で張られる面を, 法平面と言う. 法平面と, 曲面の交差線を, 法断面という. 法断面上の点 P における曲率 κ を, 法曲率と呼び, 法曲率は以下のように求まる. まず,$\mathbf{t}$ を法断面の弧長 s で微分すると

$$\frac{d\mathbf{t}}{ds} = \mathbf{s}_u\frac{d^2u}{ds^2} + \mathbf{s}_v\frac{d^2v}{ds^2} + \mathbf{s}_{uu}\left(\frac{du}{ds}\right)^2 + 2\mathbf{s}_{uv}\left(\frac{du}{ds}\right)\left(\frac{dv}{ds}\right) + \mathbf{s}_{vv}\left(\frac{dv}{ds}\right)^2 \tag{2.8}$$

式 2.8 の両辺に $\mathbf{n}$ をかけると, 右辺第 1 項・第 2 項の消去ができる (接平面上にあるため内積 = 0 となる).

いま,$\mathbf{ds}$ の絶対値の 2 乗量を求めると,

$$(ds)^2 = d\mathbf{s}\cdot d\mathbf{s} = \mathbf{s}_u^2(du)^2 + 2\mathbf{s}_u\cdot\mathbf{s}_v dudv + \mathbf{s}_v^2(dv)^2 \tag{2.9}$$

ここで,

$$E = \mathbf{s}_u^2,\ F = \mathbf{s}_u\cdot\mathbf{s}_v,\ G = \mathbf{s}_v^2 \tag{2.10}$$

とおいて (第 1 次規格量), 式 2.9 を書きなおすと,

$$ds^2 = E(du)^2 + 2Fdudv + G(dv)^2 \tag{2.11}$$

また,

$$L = \mathbf{n}\cdot\mathbf{s}_{uu},\ M = \mathbf{n}\cdot\mathbf{s}_{uv},\ N = \mathbf{n}\cdot\mathbf{s}_{vv} \tag{2.12}$$

のように置いて (第 2 次規格量)

$$(\mathbf{n}\cdot\mathbf{n})\kappa = L\left(\frac{du}{ds}\right)^2 + 2M\left(\frac{du}{ds}\right)\left(\frac{dv}{ds}\right) + N\left(\frac{dv}{ds}\right)^2 \tag{2.13}$$

式 2.13 を, 式 2.11 を使って書きなおすと,

$$\kappa = \frac{L(du)^2 + 2Mdudv + N(dv)^2}{E(du)^2 + 2Fdudv + G(dv)^2} \tag{2.14}$$

このとき,E, F, G, L, M, N は, 曲面上の点において定まるため, 点 P を通る法平面の方向には依存しないが,$du : dv$ の比には依存する. 即ち, 法曲率 κ は, 法断面の接線方向に対して異なる値をとる.

主曲率

法平面を, 法線の周りに回転させると, 法曲率は変化する. いま,

$$\gamma = \frac{dv}{du} \tag{2.15}$$

とおいて,κ を γ の関数と考え,$\kappa(\gamma)$ とし, 式 2.14 を書きなおすと,

$$\{L - \kappa(\gamma)\cdot E\} + 2\{M - \kappa(\gamma)\cdot F\} + \{N - \kappa(\gamma)\cdot G\}\gamma^2 = 0 \tag{2.16}$$

$\kappa(\gamma)$ は,π 回転させる事で元に戻る, 連続周期関数だから, どこかで極値をとる.
すなわち,$\dfrac{d\kappa(\gamma)}{d\gamma} = 0$ となるような点が存在する. 式 2.16 をこの極値条件の下で γ に関する微分をすると,

$$(M - \kappa F) + (N - \kappa G)\gamma = 0 \tag{2.17}$$

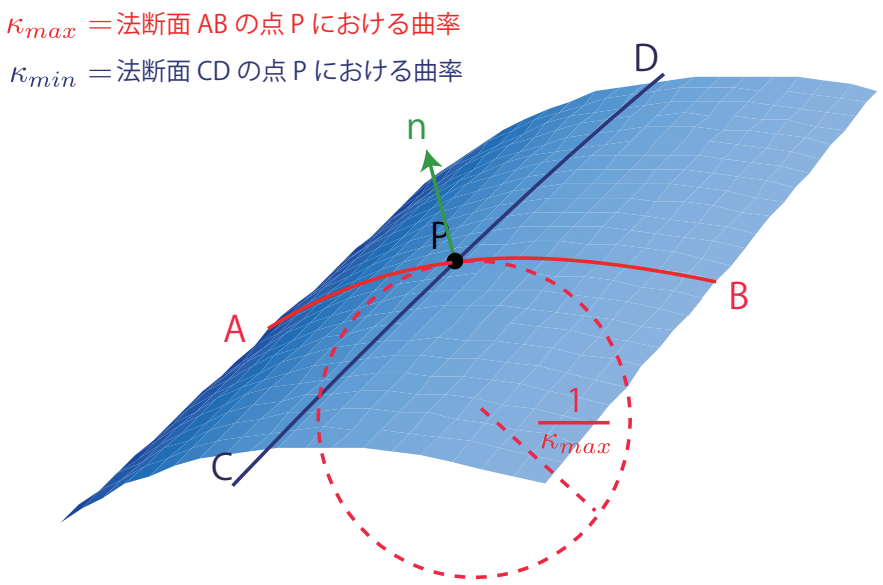


図 2.7: 曲率 κ の,κ_{max}, κ_{min}

を得る. これを, 式 2.16 に戻すと,

$$(L - \kappa E) + (M - \kappa F)\gamma = 0 \tag{2.18}$$

を得る. 式 2.17, 式 2.18 から, 次の関係式を得る.

$$\gamma = \frac{-(M - \kappa F)}{(N - \kappa G)} = \frac{-(L - \kappa E)}{(M - \kappa F)} \tag{2.19}$$

$$\kappa = \frac{(M + \gamma N)}{(F + \gamma G)} = \frac{(L + \gamma M)}{(E + \gamma F)} \tag{2.20}$$

式 2.19 より,κ について,

$$(EG - F^2)\kappa^2 - (EN + LG - 2MF)\kappa + LN - M^2 = 0 \tag{2.21}$$

という式が導かれ, これは κ に関する 2 次式である. **この根を κ_1, κ_2 とし, 主曲率**と呼ぶ. これら κ_1, κ_2 は, 曲面上の点 P に於ける曲率 κ の,κ_{max}, κ_{min} を表す.

なお, 平均曲率 K_m, ガウス曲率 K_g と呼び,

$$K_m = \frac{1}{2}(\kappa_1 + \kappa_2) \tag{2.22}$$

$$K_g = \kappa_1\kappa_2 \tag{2.23}$$

のように書く.

主方向と極値線

式 2.19 から,κ を消去すると,γ の式を得る. すなわち,

$$(MG - NF)\gamma^2 + (GL - NE)\gamma + FL - ME = 0 \tag{2.24}$$

これを, 曲率線の式と呼ぶ.

式 2.21 で見たのと同じく, こちらも 2 次式であるから, その 2 つの根を γ_1, γ_2 のように置けて,

$$\gamma_1 + \gamma_2 = \frac{-(GL - NE)}{(MG - NF)} \tag{2.25}$$

$$\gamma_1\gamma_2 = \frac{(FL - ME)}{(MG - NF)} \tag{2.26}$$

また,κ の時と同じく, どちらか大きい方を γ_{max}, 他方を γ_{min} のように書ける. ここで, 曲面上の点において,γ_1,γ_2 で定まる方向において, 法曲率は極値を取る. 曲面上の曲線の接線ベクトルは,$(S_u du + S_v dv)$ なので,γ_1,γ_2 に対応する 2 つの接線ベクトルの内積は

$$(ds)_1 \cdot (ds)_2 = \{(s_u + s_v\gamma_1) \cdot (s_u + s_v\gamma_2)\}(du)_1(dv)_2 \tag{2.27}$$

この式の $\{(s_u + s_v\gamma_1) \cdot (s_u + s_v\gamma_2)\}$ を式 2.10 の第 1 次規格量と, 式 2.25, 式 2.26 を用いて変換すると,

$$\frac{\{E(MG - NF) - F(GL - NE) + G(FL - ME)\}}{(MG - NF)} = 0 \tag{2.28}$$

ここから, 主曲率の法断面の 2 つの接線方向は, 直交していることがわかる. これらの方向は, **主方向**と呼ばれる. 曲面上の特異点を除き, いかなる点にも主方向が存在する. いま, 曲面上の曲線の接線がそこでの主方向と一致するとき, この曲線は曲率線と呼ばれる.

ここで 2.6 を書き換えると, 接ベクトル $t = \dfrac{\partial \mathbf{s}}{\partial u} \cdot du + \dfrac{\partial \mathbf{s}}{\partial v} \cdot dv$ となるが, この式を γ の定義式 2.15 を用いて $dv = \gamma du$ で書き換えると

$$t = \left(\frac{\partial \mathbf{s}}{\partial u} + \gamma\frac{\partial \mathbf{s}}{\partial v}\right) du \tag{2.29}$$

いま, 式 2.29 に,γ_{max}, γ_{min} を代入すると,2 つの接ベクトル t_{max}, t_{min} が定まる.

これまで, パラメトリック形式で曲面上の特徴線についてみたが, 曲率極値線の研究 [5] では, 陰関数を用いているため, 主曲率および主方向について陰関数形式で書きなおすと

陰関数曲面 $F(x, y, z) = 0$ として, 法線ベクトル n は

$$n = \frac{\nabla F}{|\nabla F|} \tag{2.30}$$

である.

この法線ベクトル $n = (n_x, n_y, n_z)$ の勾配 ∇n は

$$\nabla n = \begin{pmatrix} \frac{\partial n_x}{\partial x} & \frac{\partial n_y}{\partial x} & \frac{\partial n_z}{\partial x} \\ \frac{\partial n_x}{\partial y} & \frac{\partial n_y}{\partial y} & \frac{\partial n_z}{\partial y} \\ \frac{\partial n_x}{\partial z} & \frac{\partial n_y}{\partial z} & \frac{\partial n_z}{\partial z} \end{pmatrix} \tag{2.31}$$

3×3 行列であるから, これを

$$\begin{vmatrix} \frac{\partial n_x}{\partial x} - \lambda & \frac{\partial n_y}{\partial x} & \frac{\partial n_z}{\partial x} \\ \frac{\partial n_x}{\partial y} & \frac{\partial n_y}{\partial y} - \lambda & \frac{\partial n_z}{\partial y} \\ \frac{\partial n_x}{\partial z} & \frac{\partial n_y}{\partial z} & \frac{\partial n_z}{\partial z} - \lambda \end{vmatrix} = 0 \tag{2.32}$$

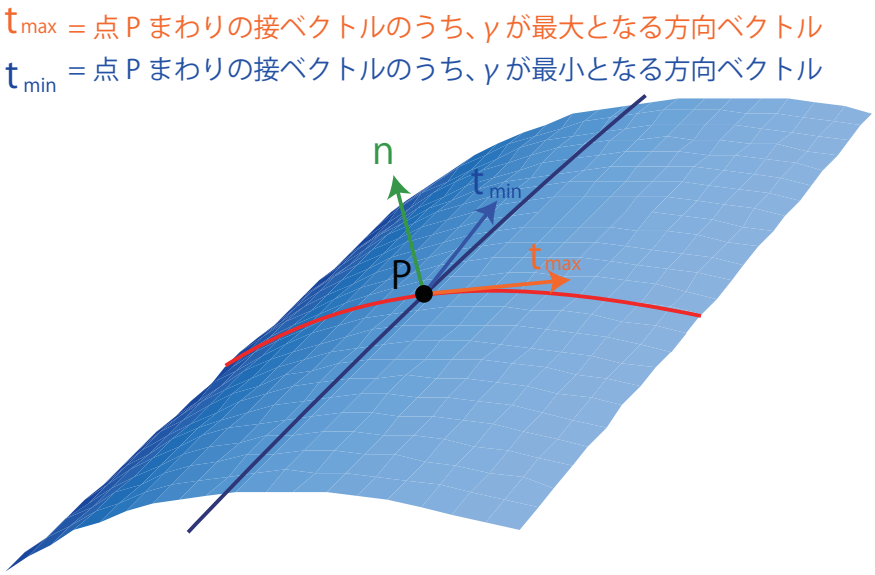


図 2.8: 接ベクトル t の,t_{max}, t_{min}

を解くことによって固有値分解すると,3 つの固有値と, それに対応した 3 つの固有ベクトルが得られる. その固有値のうち,1 つは必ず 0 となり, 対応する固有ベクトルは法線ベクトル n となる. また, 非 0 の 2 つの固有値は, それぞれ主曲率 $\kappa_{max}, \kappa_{min}$ となり, それぞれに対応する固有ベクトルが主方向 t_{max}, t_{min} となる.

また,
$\nabla k \cdot t = \dfrac{\partial \kappa}{\partial t}$ であり, この値が 0 になる点の軌跡が曲率極値線であるが,[5] では,

$$\nabla k \cdot t = \frac{F_{ijl} t^i t^j t^l + 3k F_{ij} t^i n^j}{|\nabla F|} \tag{2.33}$$

を用いて計算を行っている. この記法はアインシュタインの縮約記法 (Einstein summation convention) と呼ばれ, 例えば
$c_i x^i = \sum_{i=1}^{n} c_i x_i$ を表すという.

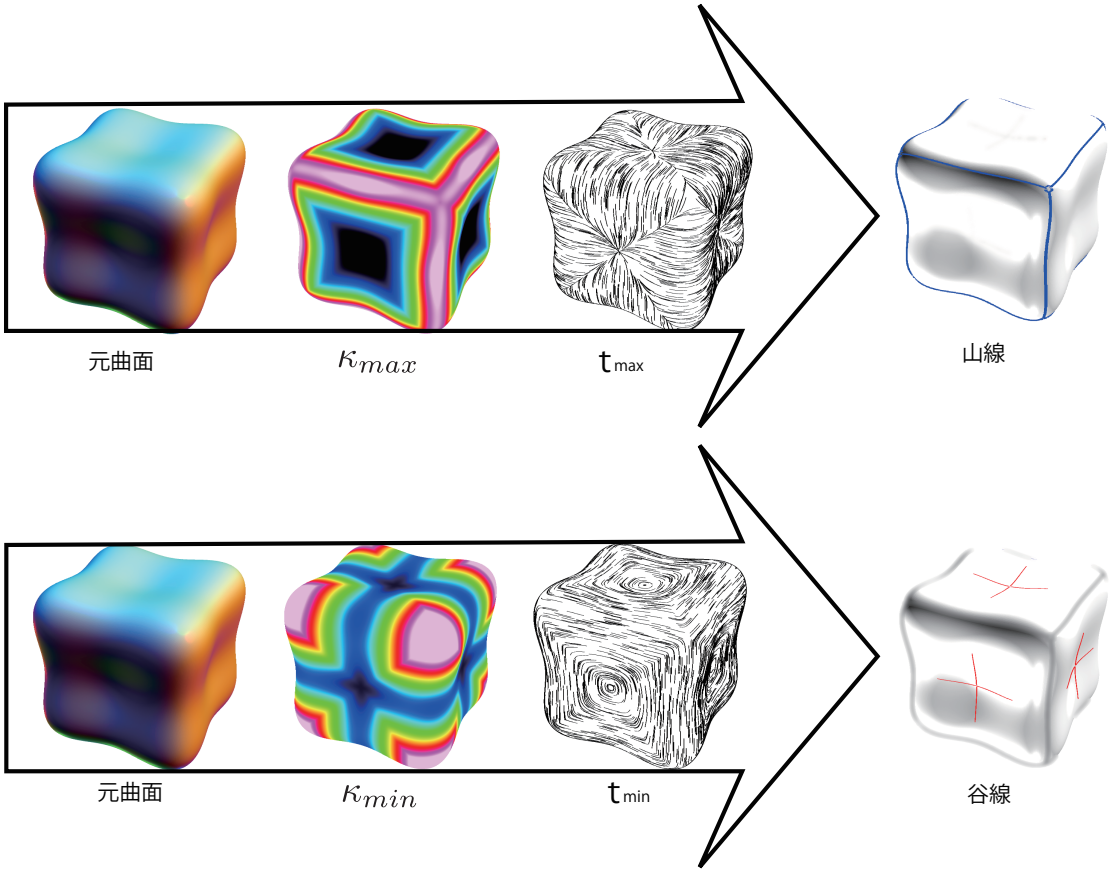


図 2.9: 山谷線

2.1.4 曲率極値線

以上のように定義した, 主曲率 $\kappa_{max}, \kappa_{min}$ および, 接ベクトル t_{max}, t_{min} を用いて, 曲率極値線の山線 · 谷線を, 以下のように定義できる.

山線：k_{max} が,t_{max} に沿って極大になる点の軌跡
谷線：k_{min} が,t_{min} に沿って極小になる点の軌跡

図 2.9 に, その 1 例を示す.

ここまで見た, 点群の陰関数によるフィッティングと, その関数の曲率極値線によって, 問題 1 の溶接部の検出を行う.

この曲率極値線は, 曲率が大きい場所では強く出, 小さい場所では弱く出る. 曲率極値線の強さ k は, 曲率の弧長積分で定義する. すなわち, 極値線の弧長を s, 曲率を κ としたとき, 曲率極値線の強さは式 2.34 のようになる.

$$k = \int_C \kappa ds \tag{2.34}$$

図 2.9 は理想的な 3 次元曲面に極値線を適用した例であったが, 実物スキャンメッシュでは, 問題も多い. 図 2.10 に, スキャンメッシュに対し, 曲率極値線を計算した 1 例を示す. 黄色が山線部, 紫が谷線部分である. 問題とは,**(1) 曲率極値線が出現するのが, 溶接端部分だけではないこと, (2) 溶接端部分の曲率極値線も, 常に連続とは限らないこと**である.

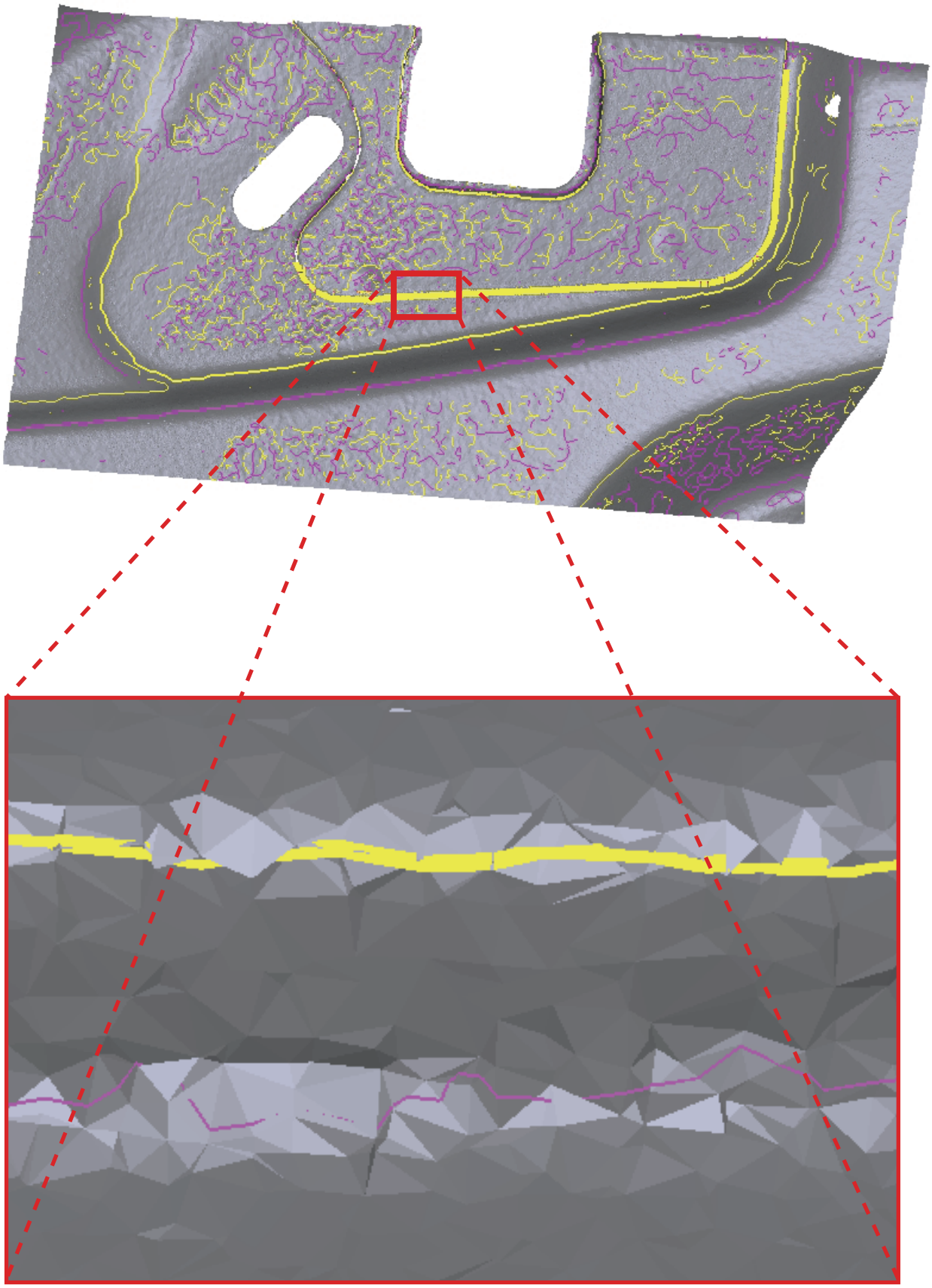
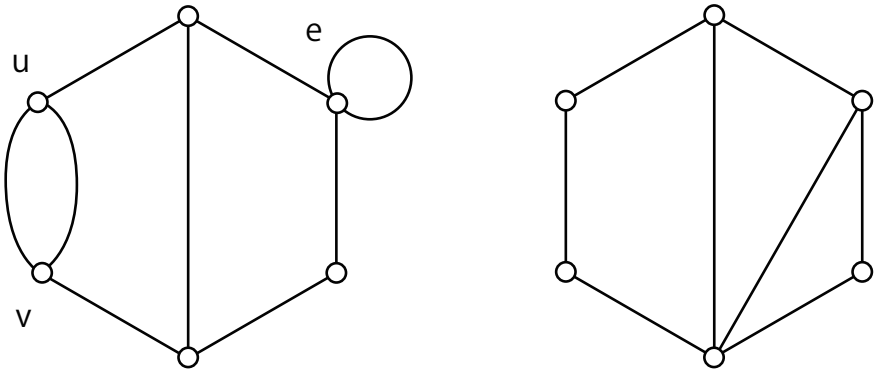


図 2.10: 山谷線



(a) 多重グラフ G　　(b) 単純グラフ H

図 2.11: グラフの種類

2.2　グラフカット

曲率極値線で見た 2 つの問題を, 本手法によって解決したい. グラフカットは, ノード (頂点) とエッジ (辺) を持つグラフを,2 つの部分グラフ：ソース (前景) 側グラフ S と, シンク (背景) 側グラフ T に分ける手法である. その部分グラフへの分割の為に, コストという概念を導入する. コストは, エッジで繋がれた**ノードとノード間**, また, **ソースとノード間**, **シンクとノード間**に設定する. コストは換言すれば, 切るための労力であり, 切れにくさであり, グラフ理論における容量のことである. グラフカットは, ソース側グラフ S と, シンク側グラフ T に分割したとき, このコストの和が最小になるようなカットを行う. 以下, グラフ理論についてまずは見, のちにグラフカットについて説明する.

2.2.1　グラフ理論 [4]

グラフとは何か

グラフ G とは, 要素の有限集合であるノード (頂点)$V(G)$ および,$V(G)$ の相異なる 2 要素の対を要素とする有限集合であるエッジ $E(G)$ から構成されるもののことを言う. グラフ G の点の個数 $|V(G)| = p$ を G の位数, 辺の本数 $|E(G)| = q$ を G のサイズという. また, 辺 e, f が同一の点 v に隣接しているとき,e, f は隣接していると言う. 点 v の隣接点の集合を v の近傍と呼び,$N(v)$, または $N_G(v)$ のように表す.

図 2.11(a) 辺 u-v 間のように,2 点間を結ぶ辺が 1 本以上あるものを, 多重辺, 頂点 e のように, 同一の点を結ぶループを, 多重ループと言う. そのような, 多重辺, 多重ループを持つグラフを, 多重グラフと呼ぶ. 図 2.11(b) のように, 多重辺も多重ループも持たないグラフを, 単純グラフもしくは単にグラフと呼ぶ.

同型グラフ・部分グラフ

2 つのグラフ G, H がある時,
$V(G)$ から $V(H)$ への 1 対 1 対応 f で

$$\{u, v\} \in E(G) \Longleftrightarrow \{f(u), f(v)\} \in E(H) \tag{2.35}$$

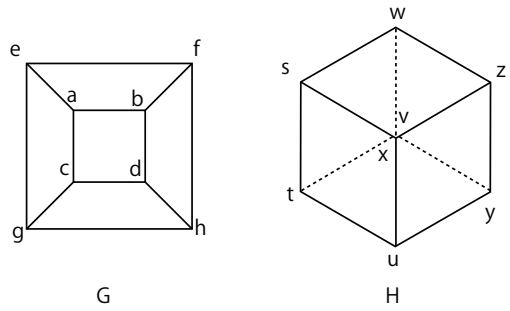


図 2.12: 同型グラフ

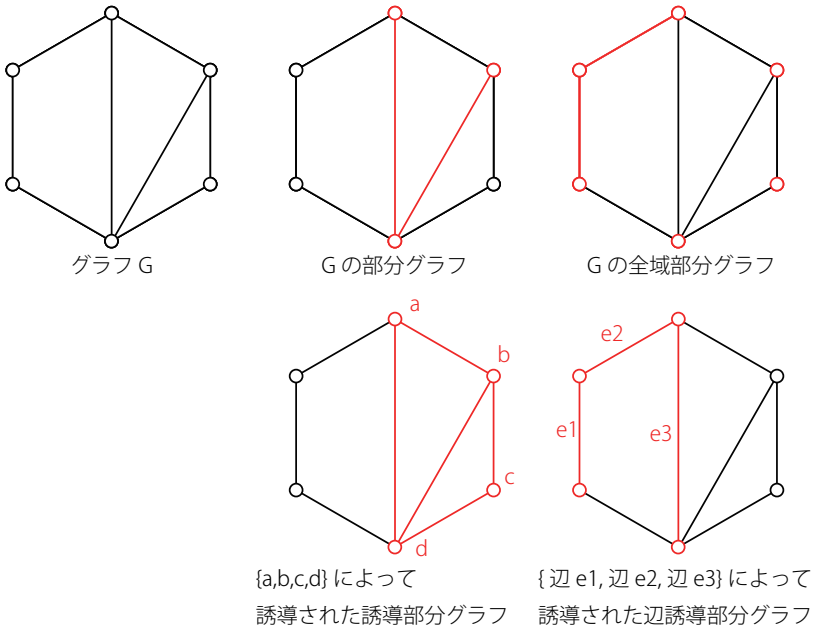


図 2.13: 部分グラフの種類

を満たすものが存在するとき,$G \cong H$ と書き,H における各点の隣接関係が,G における各点の隣接関係と一致する. すなわち,G, H は同型のグラフである.

これに対し,$V(H) \subseteq V(G)$ かつ $E(H) \subseteq E(G)$ を満たすグラフを, グラフ H がグラフ G の部分グラフである, といい, この関係を $H \subseteq G$ と表す. $V(H) = V(G)$ の時,H を G の全域部分グラフという. また,$V' \subseteq V(G)$ に対して,V' を点集合とする G の極大な部分グラフ, すなわち,V' 上の 2 点を結んでいる G の辺が全て含まれているグラフを,V' によって誘導された, 誘導部分グラフといい $< V' >$ で表す.$E' \subseteq E(G)$ に対して,E' に属する辺の端点全体の集合を点集合とし,E' を辺集合とするグラフを E' によって誘導された辺誘導グラフという. それぞれ, 図 2.13 のようなものである.

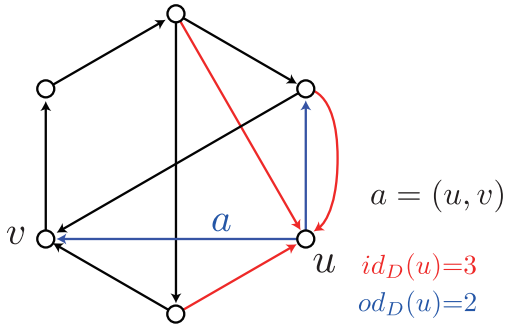


図 2.14: 有向グラフの概要

2.2.2 最大流最小カット定理

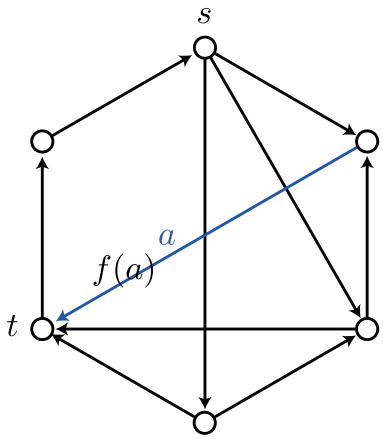
最大流最小カット定理 (Max-flow min-cut theorem) は, グラフ理論における最適化問題の一種である.

有向グラフ

点集合 $V(D)$ と $V(D)$ の要素対からなる有限集合 $A(D)$ で構成される, 有向グラフ D を導入する.$V(D)$ の要素の順序対は弧と呼ばれ, 辺に向きのついたもの (有向) である. また,$A(D)$ は孤集合と呼ばれる. 点 u から点 v に向いている弧 a を,$a = (u, v)$ で表し,u をこの弧の始点,v を終点と呼ぶ. 点 $u \in V(D)$ に対して u に接続し u を始点とする弧の本数を u の出次数といい,$od_D(u)$ で表し, また u に接続し u を終点とする弧の本数を u の入次数といい,$id_D(u)$ で表す. 図 2.14 にこれら概要を示す.

また, この時, 以下の定理が成り立つ.

$$\sum_{u \in V(D)} od_D(u) = \sum_{u \in V(D)} id_D(u) = |A(D)| \tag{2.36}$$



ネットワーク N

図 2.15: ネットワーク

ネットワーク

特定の 2 点 s, t をもつ連結有向グラフ D と,D の各弧 a に付随する数 $c(a)$ の集合が与えられている時, その全体のシステムを**ネットワーク**と呼び,$N = (D, s, t, c)$ で表す.D を N の有向底グラフと呼び, **点 s をソース, 点 t をシンク**と呼び,s, t 以外の点を内点と呼ぶ. また, ソースとシンクを合わせて, ターミナル・ノードと呼ぶ.

弧 a に附随する数 $c(a)$ は, その弧を利用して「輸送」できる物の量の上限に対応しており, 弧 a の容量と呼ぶ. 各弧 a に容量 $c(a)$ が定義されているとは, 有向グラフ D の弧集合 $A(D)$ から数の集合 S への関数 $c : A(D) \to S$ が与えられたと考えられることができ,c を容量関数と呼ぶ.

今, ネットワークを通じた物の輸送を考える. **流れ**と呼ばれる数 $f(a)$ をネットワーク N の各弧 a ごとに定義する. このとき, ネットワーク N の有向グラフ D の弧集合 $A(D)$ から, 数の集合 S への関数 $f : A(D) \to S$ が与えられたと考えることができ,f を流量関数と呼ぶ.

ある物をネットワークを通じて輸送する時, 輸送可能量の上限 c を超えて輸送することは適わず, 各頂点に運び込まれる物の総量と, 運び出される物の総量は, 最終的に等しくならなければならない, という制限から, ネットワークの流れ f を以下を満たすように定義する.

(1) 容量制限:任意の弧 a に対し, $0 \le f(a) \le c(a)$.

(2) 保存条件:任意の内点 v において $\sum_{a \in out(v)} f(a) = \sum_{a \in in(v)} f(a)$

図 2.16 は, ここまで定義してきたネットワークの 1 例である. 弧傍の数値の組は (f, c) を現しており,S, T 以外での内点において, f が保存条件 $\sum_{a \in out(v)} f(a) = \sum_{a \in in(v)} f(a)$ を満たしていることがわかる.

保存条件は, 運んでいる物が途中でなくならないことを保証しているため, ソース s から出たものはすべて最終的にシンク t に届くということになる. 従って, ネットワーク N 全体の流れの総量の変化が s, t に現れる. すなわち,

$$\sum_{a \in out(s)} f(a) - \sum_{a \in in(s)} f(a) = \sum_{a \in out(t)} f(a) - \sum_{a \in in(t)} f(a) \tag{2.37}$$

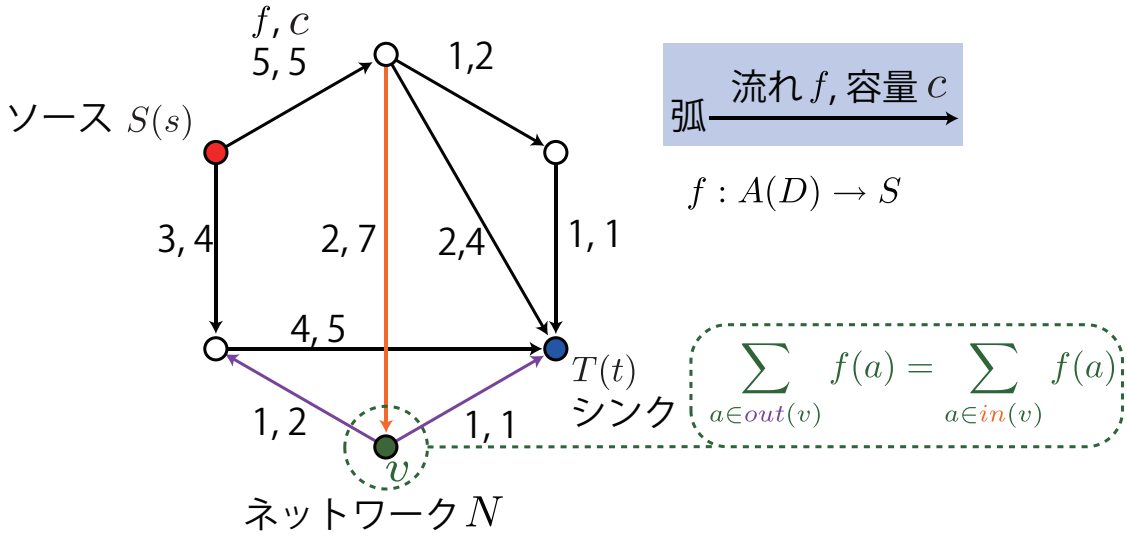


図 2.16: ネットワークとその値

ここで, ネットワークの総流量 *val* を定義する. 2.37 の (左辺)=(右辺) = *val*. すなわち,

$$val(f) = \sum_{a \in out(s)} f(a) - \sum_{a \in in(s)} f(a) = \sum_{a \in out(t)} f(a) - \sum_{a \in in(t)} f(a) \tag{2.38}$$

最大流とグラフのカット

ここで, 図 2.16 のソースから, シンクまで, 最も多く物を運ぶことを考える. それは, このネットワークにおいて, 最大となる $val(f)$, すなわち**最大流**を求めようという試みである. ネットワーク内の, ある弧 a の容量 $c(a)$ を超えて, 物を運ぶことはできないため, 最大流と弧の容量の間には密接な関係があると考えられる.

ネットワーク $N = (D, s, t, c)$ において,$s \in S, t \in \bar{S} (= V(D) - S)$ なる点部分集合 $S \subseteq V(D)$ に対して,S の点を始点とし $\bar{S}$ の点を終点とする弧全体の集合を $(S, \bar{S})$ 逆に, $\bar{S}$ の点を始点とし S の点を終点とする弧全体の集合を $(\bar{S}, S)$ で表す. このとき,$(S, \bar{S})$ を N の**カット** K と呼ぶ. カット K に属する弧の容量の総和を K の容量と呼び,$cap(K)$ で表す. すなわち,

$$cap(K) = \sum_{a \in (S, \bar{S})} c(a) \tag{2.39}$$

このような例を図 2.17 に示す.

ここで, 式 2.38 と保存条件から,
ネットワーク N 内の任意の流れ f と, 任意のカット $K = (S, \bar{S})$ に対して,

$$val(f) = \sum_{a \in (S, \bar{S})} c(a) - \sum_{a \in (\bar{S}, S)} c(a) \tag{2.40}$$

が成り立つ. また, 任意の弧 a に対して, $f(a) \leq c(a)$ であることと,$0 \leq f(a)$ であることと式 2.40 から, 定理

$$val(f) \leq cap(K) \tag{2.41}$$

が成り立つ.

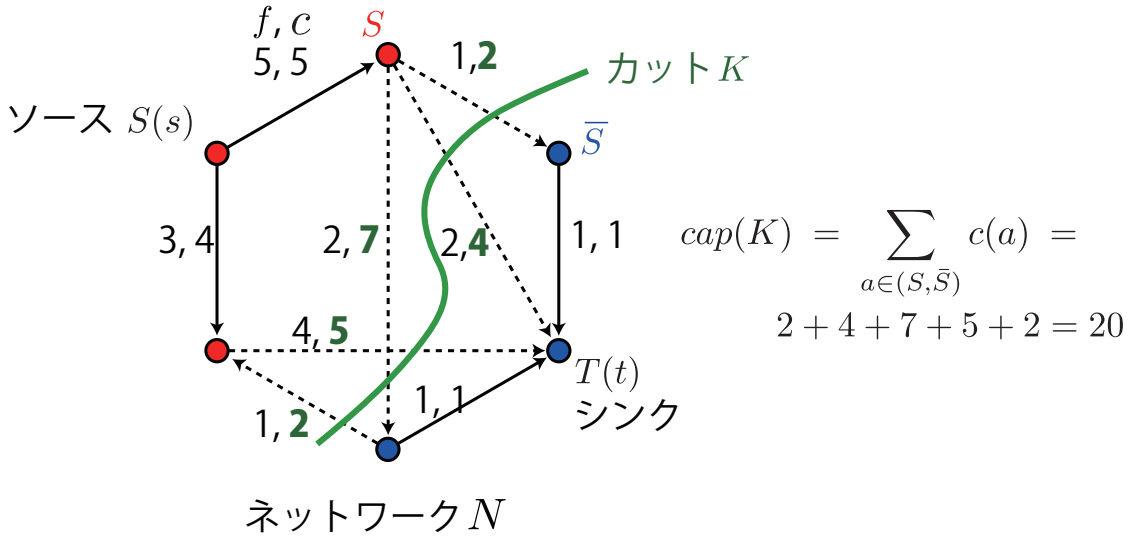


図 2.17: ネットワークのカット

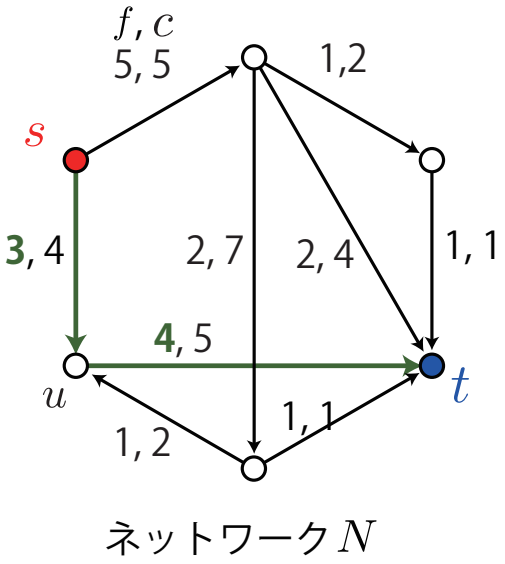


図 2.18: グラフの増大道

ネットワーク内のすべてのカットのうちで, 最小の容量をもつカットを, **最小カット**と呼ぶが, 式 2.41 から,N 内のどのような流れの値も, 任意のカットの容量を越えることができないため, 流れの値は最小カットの容量が上限となる. すなわち,

流れ f とカット K が,$val(f) = cap(K)$ を満たすならば,
f は最大流であり,K は最小カットである.

最大流・最小カットの定理

図 2.18 において,$s \to t$ にモノを運ぶ事を考える. 弧上の数値は流れ f, 容量 c をあらわす. $s \to u, u \to t$ において, 両者とも $f < c$ であるから,s から t の道 s, u, t に沿って弧 $(s, u), (u, t)$ の流れ $f(s, u), f(u, t)$ を 1 づつ増やすことができる. このように, ある道に沿って, 流れ f を増や

すことのできる $s-t$ 道について考える.N 内の各 $s-t$ 道 P に対し, 非負整数 $\tau(P) = \underbrace{min}_{a\in A(P)} \tau(a)$ を定義する. ただし,

$$\tau(a) = \begin{cases} c(a) - f(a) \cdots \text{弧 } a \text{ が } P \text{ と同じ向きのとき} \\ f(a) \cdots \text{弧 } a \text{ が } P \text{ と逆の向きのとき} \end{cases}$$

のようにする. $\tau(P)$ は容量制限を保ちながら P に沿って f を増量できる量の最大値である.$\tau(P) > 0$ であるとき, 道 P は $f-$ **非飽和**であるといい,$\tau = 0$ のとき道 P は $f-$ **飽和**であるといい,$f(a) = 0$ のとき,$f-$ **零**であるという.$f-$ 非飽和な s-t 道を f **増大道**という.f **増大道**が見つかったとき, 次のようにして f より値の大きい流れ $\tilde{f}$ が得られる.

$$\tilde{f} = \begin{cases} f(a) + \tau(P) \cdots a \in A(P) \text{ かつ弧 } a \text{ が } P \text{ と同じ向きのとき} \\ f(a) - \tau(P) \cdots a \in A(P) \text{ かつ弧 } a \text{ が } P \text{ と逆の向きのとき} \\ f(a) \cdots a \notin A(P) \text{ のとき} \end{cases}$$

この $\tilde{f}$ を P に基づく修正流と呼ぶ.

ここから, 次の定理 2.42 が成り立つ.

N の流れ f が最大流であるための必要十分条件は,N が $f-$ 増大道を含まないことである. (2.42)

定理 2.42 は, 以下のように示される.N が $f-$ 増大道を含むとすると, この道に基づく修正流 $\tilde{f}$ は f より大きい値を持つから,f は最大流ではない. 逆に,N が $f-$ 増大道を含まないならば f は最大流であることを示す:

N 内の $f-$ 非飽和道によって,s と結ばれる点の全体を S とする.$s \in S$ であり, また N が $f-$ 増大道を含まないことから,$t \in \bar{S}$ である.

従って, カット $K = (S, \bar{S})$ は,N のカットになる. このとき $(S, \bar{S})$ に含まれる各弧は $f-$ 飽和であり,$(\bar{S}, S)$ に含まれる国こは $f-$ 零であることを示す:弧 $a \in (S, \bar{S})$ の始点を u, 終点を v とする.$u \in S$ だから S の定義より $f-$ 非飽和な $s-u$ 道が存在する. 弧 a が $f-$ 非飽和ならば,P に弧 a を加えて $f-$ 非飽和な $s-v$ 道が得られるが, これは $v \notin S$ に反する. 従って,a は $f-$ 飽和でなければならない. $a \in (\bar{S}, S)$ ならば a は $f-$ 零であることも同様に示せる.

上記の議論と, カットの容量の定義から,

$$\sum_{a\in(S,\bar{S})} f(a) - \sum_{a\in(\bar{S},S)} f(a) = cap(K) \tag{2.43}$$

が成り立つ. ここで, 式に表された定理を用いると

$$val(f) = cap(K) \tag{2.44}$$

を得る. 今,$\tilde{f}$ を N 内の最大流,$\tilde{K}$ を N 内の最小カットとすると, 定理 2.41 より,

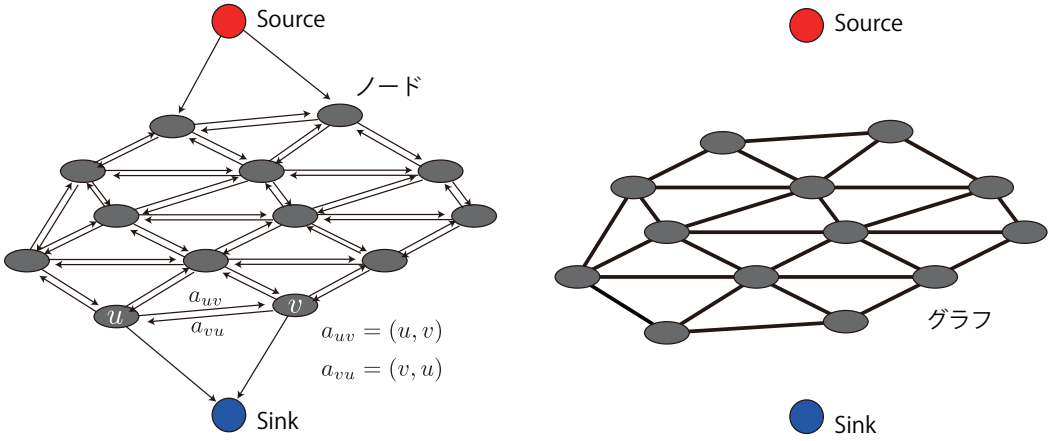
$$val(f) \le val(\tilde{f}) \le cap(\tilde{K}) \le cap(K) \tag{2.45}$$

ところがこれらは, 式 2.44 より等式でなければならないから,

$$val(f) = val(\tilde{f}) \tag{2.46}$$

すなわち,f は N 内の最大流である. この証明の過程で,$val(\tilde{f}) = cap(\tilde{K})$ を満たす最大流 $\tilde{f}$ と最小カット $\tilde{K}$ の存在も同時に示せた. 従って, 以下の**最大流・最小カットの定理**が得られる:

任意のネットワークに対して, 最大流と最小カットの容量は等しい (2.47)



(a) 有向グラフ　　(b) グラフの抽象化

図 2.19: グラフカットの概要

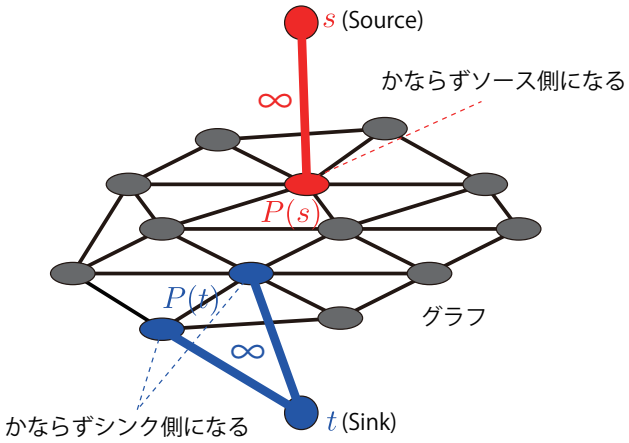


図 2.20: ソースとシンク

2.2.3　グラフカット [5]

以上, グラフ理論における, グラフの定義, 有向グラフ, ネットワーク, グラフのカット, 最大流・最小カットの定理についてみた. これらの知識を元に, 本研究で使う, グラフカット [5] という手法について解説する.

はじめに, 図 2.19(a) のようなグラフを定義する. すなわち, ノード u・ノード v 間にはお互いに弧 $a_{uv} = (u, v)$, 弧 $a_{vu} = (v, u)$ がついており, 容量が等しく $c(a_{uv}) = c(a_{vu})$ であり, また, ソースからは弧が出るのみ, シンクへは弧が入るのみといったグラフである. 同じノード・ノード間の 2 つの弧の容量は等しいので, 簡便の為, 図 2.19(b) のように抽象化する.

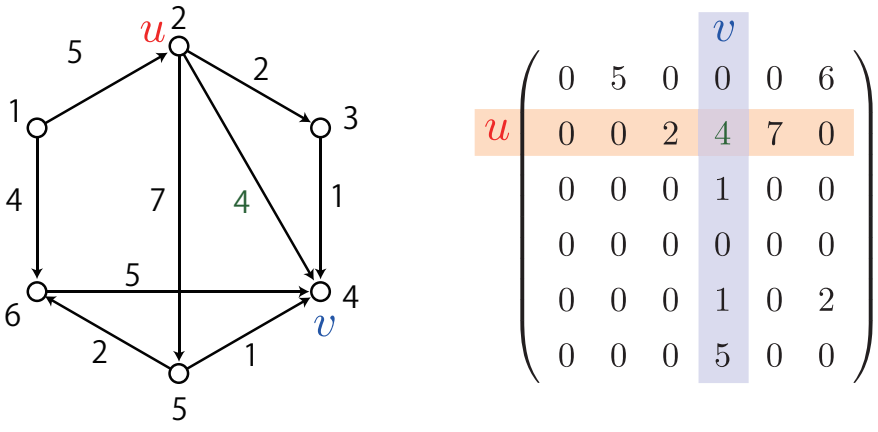
そのような抽象化を行うと, このグラフは,3 次元の頂点群とその点群間に走るエッジ, およびその上下に浮かぶソースとシンクからなる集合のように見ることができる. このグラフを, ソース側, シンク側の 2 つの部分グラフに分解したい.

はじめに, 図 2.20 のように, 分割後にソース側, シンク側に必ず含まれるノード $P(s), P(t)$ をそれぞれ 1 つ以上選択する. そして, ソースと選ばれたノード間, および, シンクと選ばれたノード間に, 大きさ ∞ の**コスト** k を導入する. **コスト**は, ソース・ノード間, シンク・ノード間,

および, ノード・ノード間に導入でき, その切れにくさを表す値で, 容量 $c(a)$ にあたる.

このコスト $k = c(a)$ を, 設定したとき, 定理 2.47 の最大流最小カット定理から, このネットワークを 2 つの部分グラフへとカットする最小のカットは, 最大流を求める事で導出できる.

さらに, グラフはグラフを表現する為に用いる行列:**隣接行列**を用いる事で行列形式に表現する事ができる. 具体的には, ある頂点 u と v の間の枝の本数を行列の (u, v) 成分に割り当てる. 単純グラフであれば, 枝があるとき (u, v) を 1 に, 枝がないとき (u, v) を 0 にする. 有向グラフの場合,u から v に向かう枝があるときのみ (u, v) を 1 に, そうでないとき (u, v) を 0 にする. また, 枝に重みがついているグラフの場合は, (u, v) に容量を代入する. その 1 例を図 2.21 に示す.



ネットワーク N　　隣接行列

図 2.21: グラフとその隣接行列.N の頂点傍の数値は頂点番号. 弧傍の数値は容量.

行列形式に表現されたグラフの最大流・最小カット問題は線形計画問題の特殊形式として定式化する事ができ, 計算機上で容易に導出する事ができる. 今回は,Y. Boykov, V. Kolmogorov[5] によるグラフカットプログラムを用い, 計算した.

2.2.4　インタラクティブ・グラフカットの研究 [5]

また, インタラクティブグラフカットの研究 [5] では, 図 2.22 のように, 二次元画像のセグメンテーションにグラフカットを応用し, 大きな成功を見ている. 本研究ではこの手法を参考に,3 次元メッシュへ拡張した.

以下, グラフカットによるインタラクティブなセグメンテーションについて見る [5], [8]. 本手法の目的は, 画像を前景 (ソース側) と背景 (シンク側) にセグメンテーションする事である. ユーザは, 前景または背景として取得したいものの一部を指定し, インタラクティブ・グラフカットを行う事で, 前景および背景画像を入手する. 具体的には 2.22 左図のような入力画像と, 右図の O 領域を前景と分かっているものの一部,B 領域を背景とわかっているものの一部として指定し, グラフカットを行う事で,2.22 右図のような釣鐘および人物の前景画像を取得している.

グラフカットによるセグメンテーション手法は, 二次元画像のセグメンテーションを, エネルギー最小化問題として捉え, その最小化の実現をグラフの最大流・最小カットアルゴリズムを用いて行う手法である. インタラクティブ・グラフカットでは, 入力画像からグラフを作

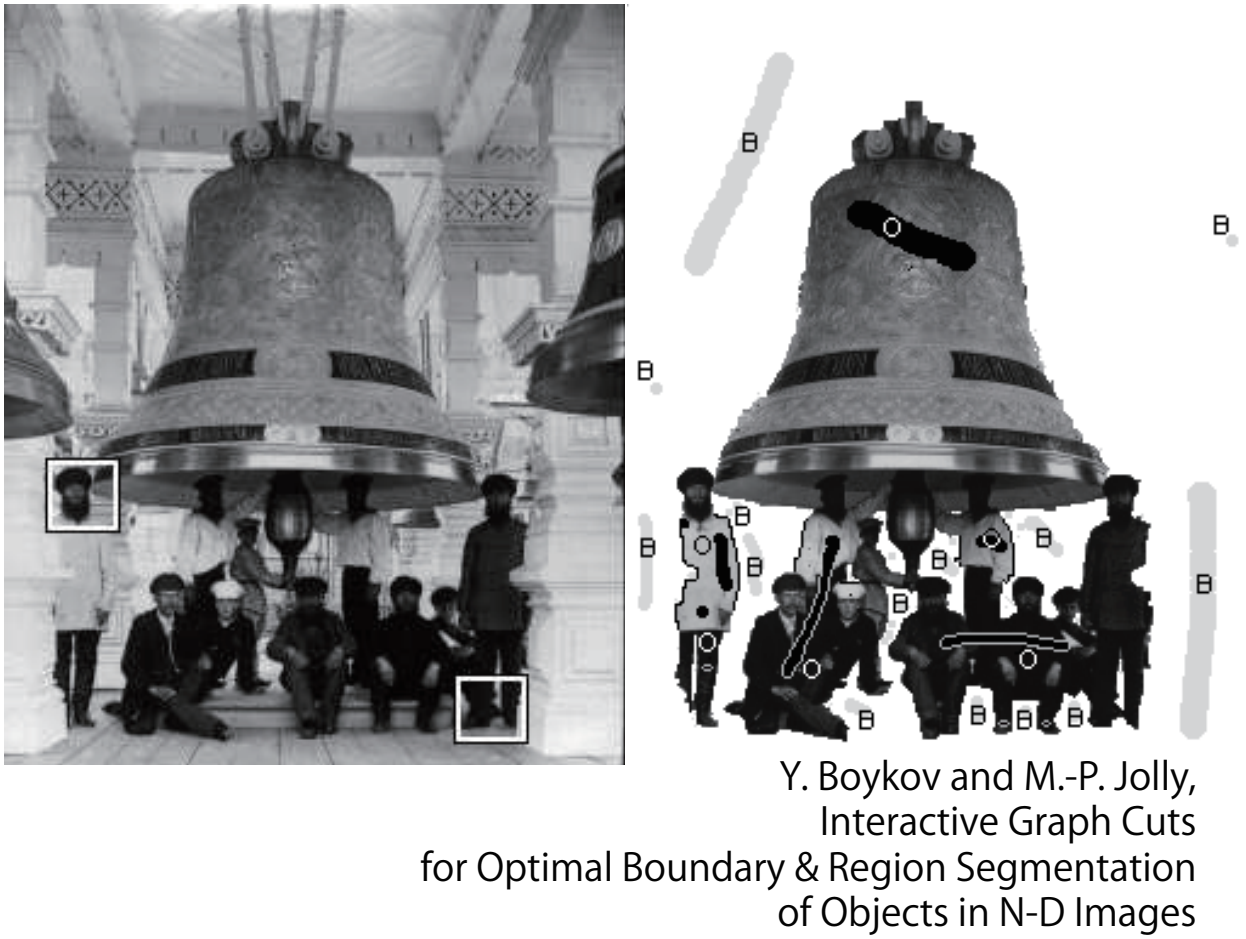


図 2.22: グラフカットの 2 次元画像への適用 [5]

成する. グラフのノードとして各々のピクセルを用いる. また, 全てのノードは, シンク, ソースと繋がるエッジを持ち t-link とよぶ. また, あるピクセルとその近傍ピクセルの関係をグラフのエッジと捉え, こちらを n-link とよぶ.3×3 ピクセル画像の場合の例を図 2.23 に示す. また, ユーザは正解ラベル L_r を指定する. ラベル L とは

$$L = \begin{cases} 0 \\ 1 \end{cases} \tag{2.48}$$

の 2 値を持ち,0 が背景 (シンク側),1 が前景 (ソース側) を示す. すなわち, 正解ラベルを割り当てるとは, 前景あるいは背景に指定したいノードに対し 0(背景),1(前景) の指定をすることである.

このように定めたグラフに対し, 与えたラベル及び, 隣接の関係から, コストを算出し, 最大フロー・最小カット問題へと帰着させる. コストの算出には, 例えば隣接するピクセルの明るさの差を用いたり, 正解ラベルとして指定したものの輝度値と, あるピクセルの輝度値を比べ, 正解ラベルで 1 を与えた (前景側に指定した) ピクセルの明るさの平均値に近ければ, ソースからのコストを大きく設定する.

画像 P に対する各ピクセル $p \in P$ としたとき, ラベルを $L = \{L_1, L_2, \cdots L_{|}P|\}$ とし, 各 L_p に,0/1 のラベルを与える. また,p の近傍ピクセルを $q \in N$ とする.

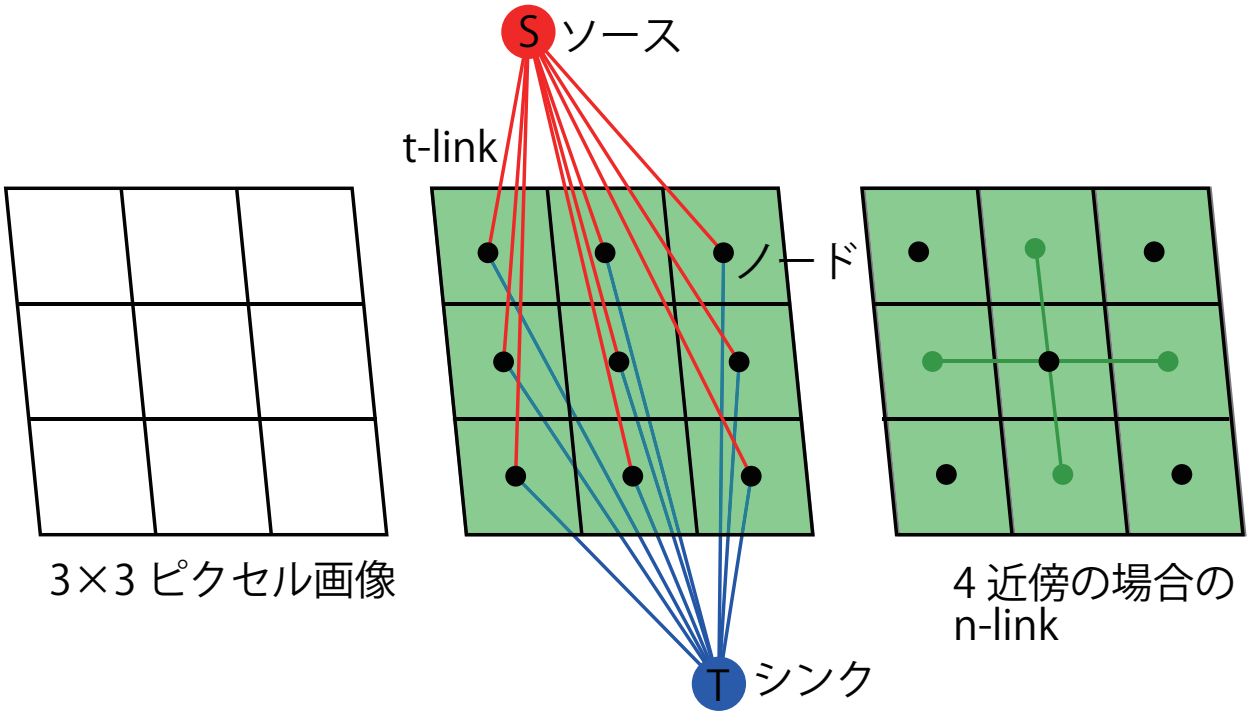


図 2.23: 3 × 3 ピクセルとグラフ

このようなグラフカットは, 次のように定式化することができる. エネルギー関数を $E(L)$ として,

$$E(L) = \lambda \cdot R(L) + B(L) \tag{2.49}$$

λ は,$R(L), B(L)$ の比率を表すパラメータであり,$R(L)$ は領域に対するペナルティー関数,$B(L)$ は物体と背景の境界に対するペナルティ関数で, 以下のように定義する.

$$R(L) = \sum_{p \in q} R_p(L_p) \tag{2.50}$$

$$B(L) = \sum_{(p,q) \in N} B_{(p,q)} \cdot \delta(L_p, L_q) \tag{2.51}$$

$$\delta(L_p, L_q) = \begin{cases} 0 \\ 1 \end{cases} \tag{2.52}$$

ピクセル p がラベル L_p である確率が大きいとき,$R_p(L_p)$ は, 小さくなるような関数として定め,p, q の輝度値が似ているとき,$B_{(p,q)}$ は大きな値になるような関数として定める.

このように定義したエネルギー関数 $E(L)$ が, 最小となるようなラベル L をグラフカットアルゴリズムを用いて解き, グラフを分割する.

その際に用いるグラフのコストとして, 隣接するピクセルの輝度値等を比べ,n-link にコストを与え, ユーザの指定した正解ラベルのピクセルの持つ輝度値等との関係から,t-link にコストを与え, そのようにコスト設定されたグラフを最大流・最小カットアルゴリズムを用いてグラフカットをする.

本手法は従来の二次元画像のセグメンテーション (固有値分解などを用いた手法であった) に比べ, このグラフカットを用いた手法は非常に高速で, 様々な応用研究が成されている. た

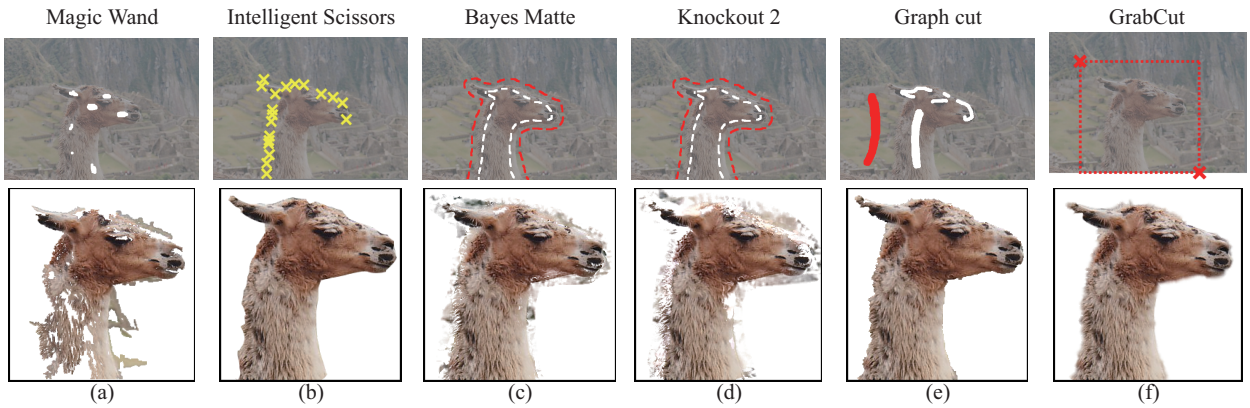


図 2.24: グラフカット及びその応用研究と他セグメンテーション手法の比較 [7]

とえば C.Rother らによる応用 [7] でも, (e) グラフカット, (f) グラブカットの 2 つが特に優れてユーザ指定が少なく, 結果もよい結果が出ている事がわかる.

第 3 章

分割手法

第 3 章

分割手法

3.1　分割法の概要

図 3.1 のフローチャートに示したように, このデータ加工のプロセスを全部で 3 段階に分け, 行った. すなわち,
(1) 極値線の導出
(2) コスト設定
(3) グラフカット
の 3 段階である.

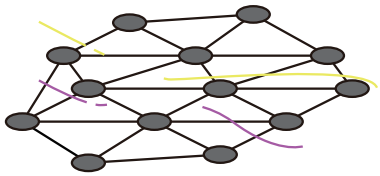
コスト設定では, 曲率極値線を導出し, その際に曲率極値線の品質と計算所要時間を決めるパラメータ, つまり式 2.5 のトレードオフパラメータ t について考える.

コスト設定では, ソースとシンクと繋がるノード, すなわちターミナルノードの指定とその際のルールについて導出し, ノード・ノード間のコスト (容量) と, 曲率極値線の間に定める関係について論じる.

グラフカットの項では, カットの目標を確認し, その目標達成の手段を講じる.

極値線導出

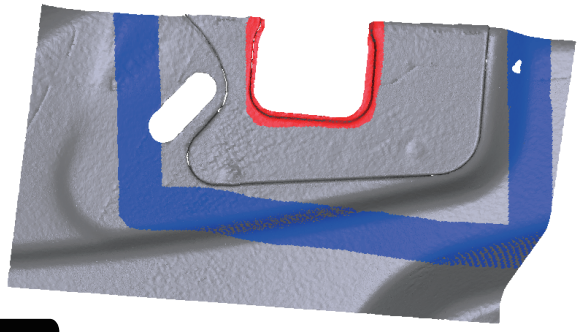
- **曲率極値線の計算**
  - トレードオフパラメータ t



コスト設定

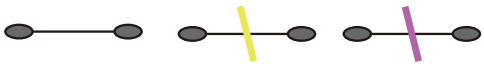
{ ソース | シンク }/ ノードに関わるもの

- **手動によるペイントでソース / シンクの指定 Ps,Pt**



ノード / ノードに関わるもの

- **曲率極値線の強さに基づいた斥力を設定する**
  - コスト倍率 A の設定
  - 閾値 θ の設定



グラフカット

- **山谷線で 2 回カットする**

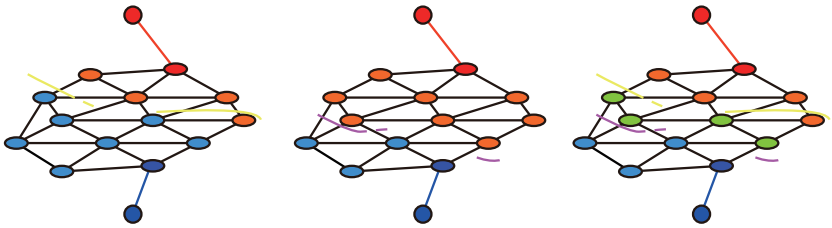


図 3.1: 解決フローチャート

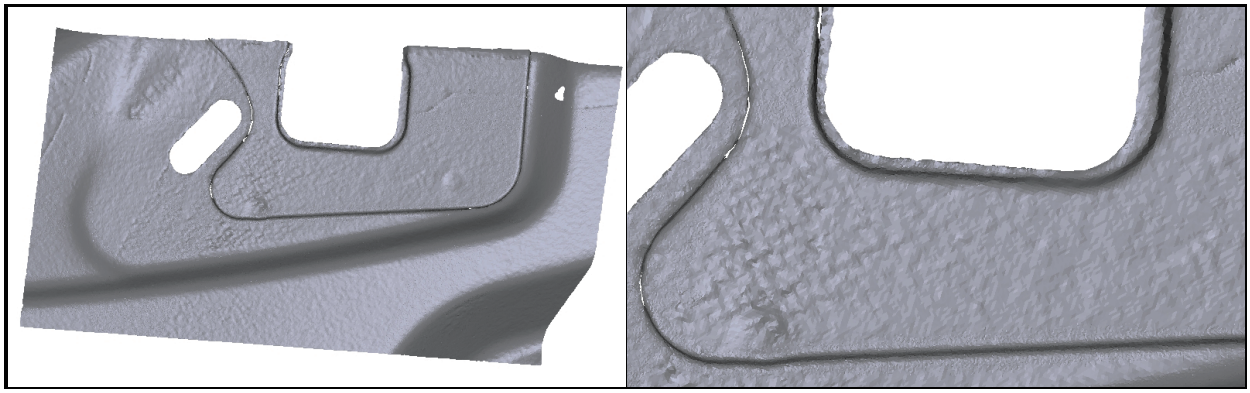
3.2 (1) 極値線の導出

2.1.2 章でみたように, 近似を行い, 曲率極値線を計算した. その際式 2.5 で定義したトレードオフパラメータ t を以下のように変え, 処理に要した時間と結果を比較した. 以降の全ての計算における, 使用マシンの性能及び環境は,
OS:Microsoft Windows XP(32Bit)
CPU:corei7 920, 2.67GHz
RAM 3.00GB
Graphic Board:NVIDIA GeForce 8800GT (512MB RAM)
を用いた. また, 結果を左右しうる特徴線だけを可視化するため,3.3.3 節で後に詳しく議論する閾値 $\theta = 1\%$ 以下の強さの極値線の表示をオフとしている. 表内では処理に要した時間 [sec] を time, 単純平均誤差 [mm] を L1, 自乗平均誤差 [mm] を L2, 誤差の最大値 [mm] を MAX, 山線の本数を Ridge, 谷線の本数を Valley と表記した.

3.2.1 トレードオフパラメータ t と極値線の関係

はじめに, 図 3.2 に示したサンプルデータ 1 を用いた. サンプルデータ 1 の頂点数は 192165, 面数は 383352 である.

この 3 次元データに対し,t を異なる値に設定した際の所要時間及び誤差, 極値線の数を示す.



全体像　　拡大図

図 3.2: サンプルデータ 1 の全体像と拡大図

表 3.1: サンプルデータ 1 の極値線の計算

t	0.0005	0.001	0.005	0.01	0.05	0.1
time	126	93	54	50	48	47
L1	0.002	0.002	0.004	0.006	0.02	0.05
L2	0.004	0.004	0.006	0.008	0.03	0.07
MAX	0.56	0.26	0.59	0.56	0.35	1.24
ridge	14751	8908	1355	447	115	76
valley	14832	9316	1371	504	150	124

図 3.3 には, そうして得られた極値線を示す. 黄色の線が山線, 紫色の線が谷線である.

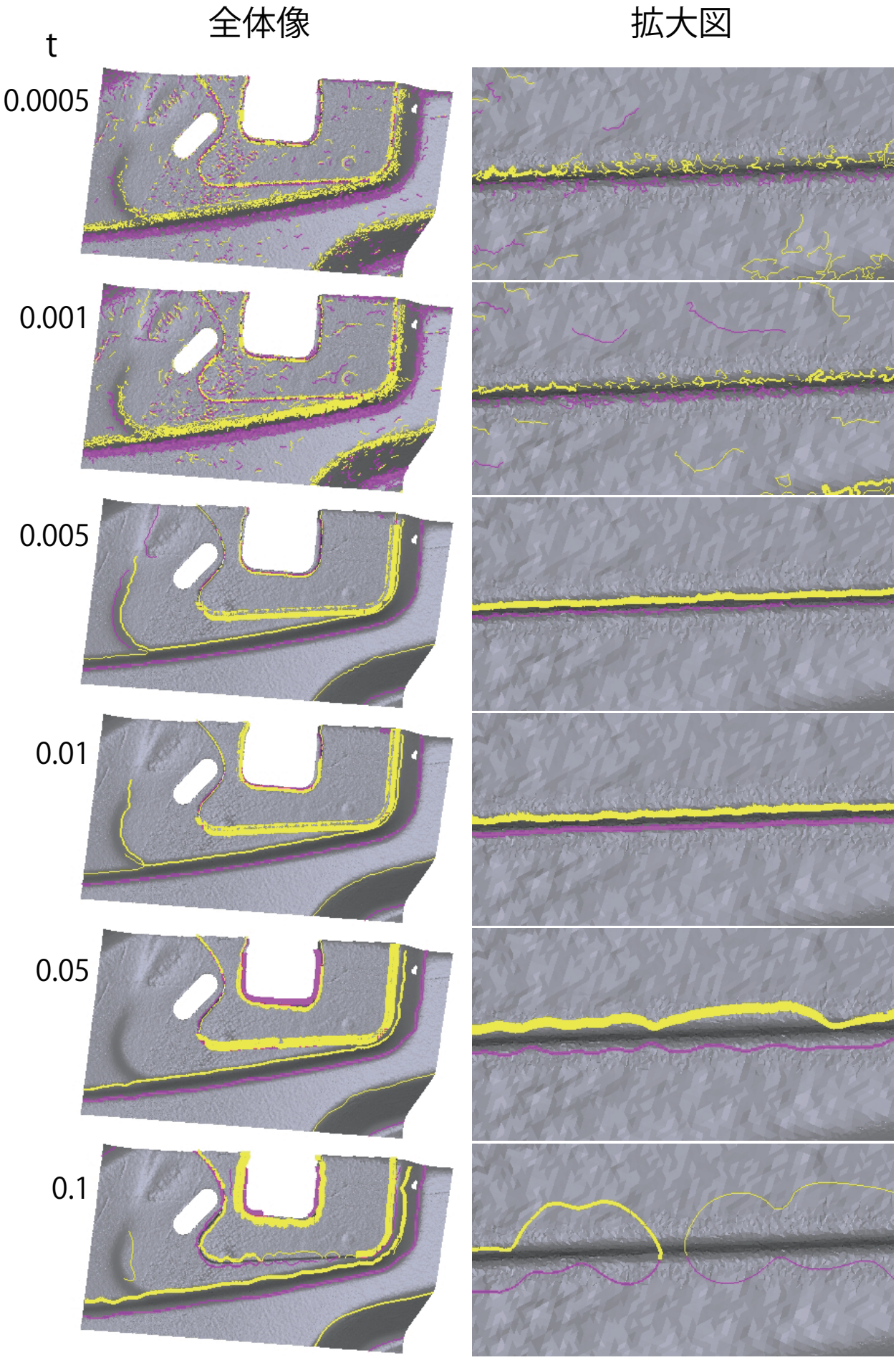
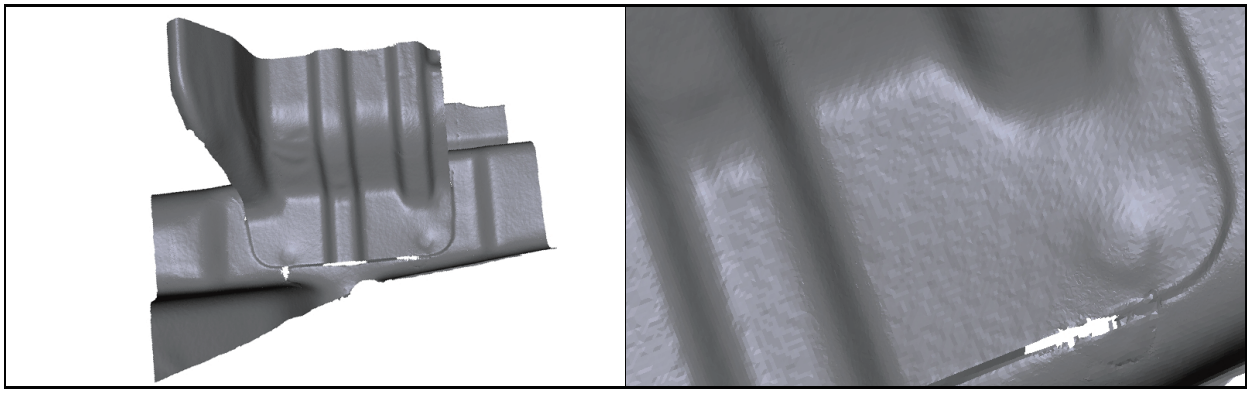


図 3.3: 山谷線の精度 (サンプルデータ 1).



全体像　　拡大図

図 3.4: サンプルデータ 2 の全体像と拡大図

表 3.2: サンプルデータ 2 の極値線の計算

t	0.0005	0.001	0.005	0.01	0.05	0.1
time	105	78	46	44	41	40
L1	0.002	0.002	0.004	0.005	0.02	0.05
L2	0.003	0.003	0.005	0.007	0.03	0.06
MAX	0.2	0.2	0.2	0.18	0.47	0.77
ridge	9981	5133	438	247	129	90
valley	9568	4953	359	238	97	74

サンプルデータ 2 の頂点数は 168494, 面数は 335033 である.

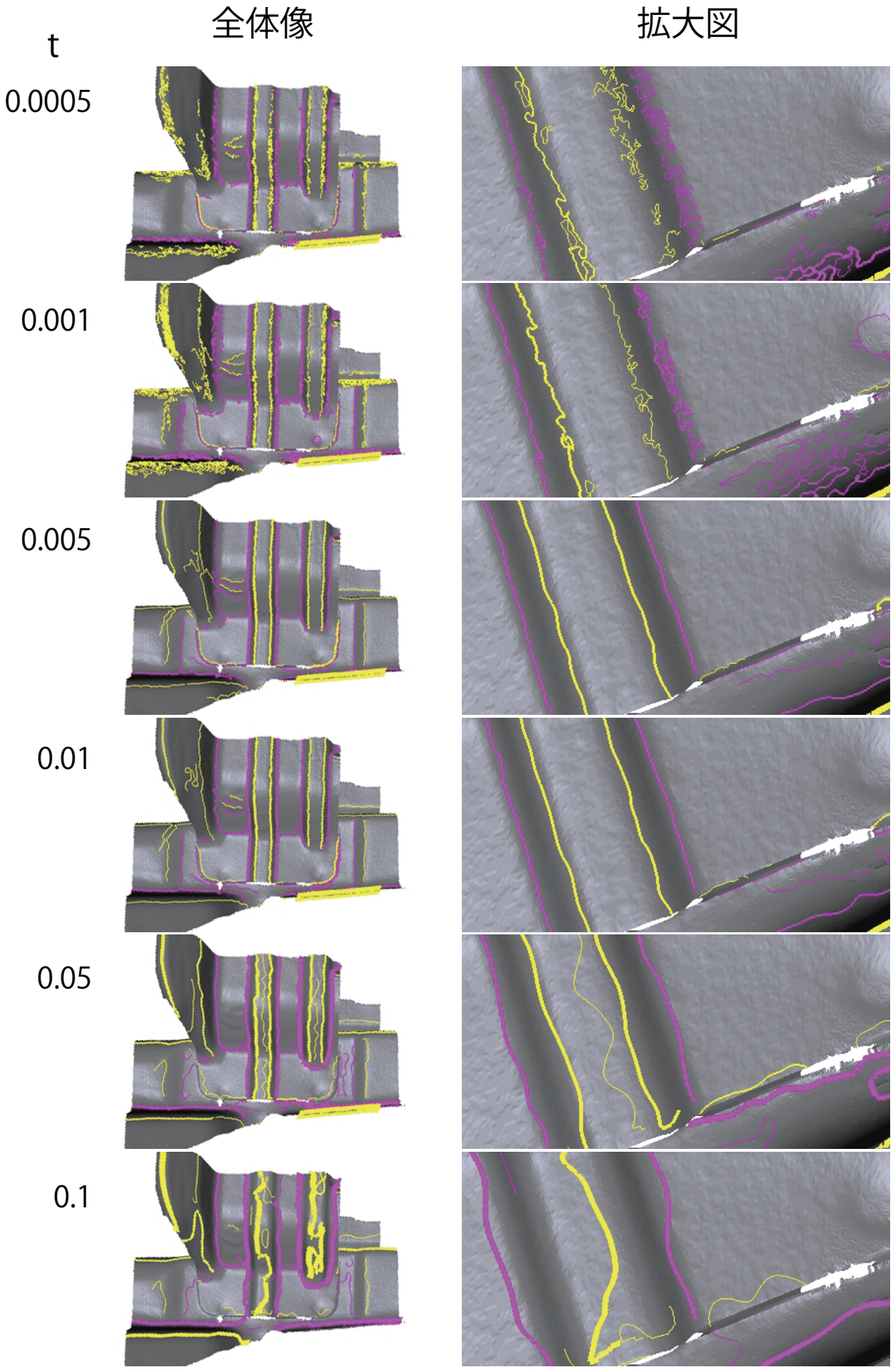
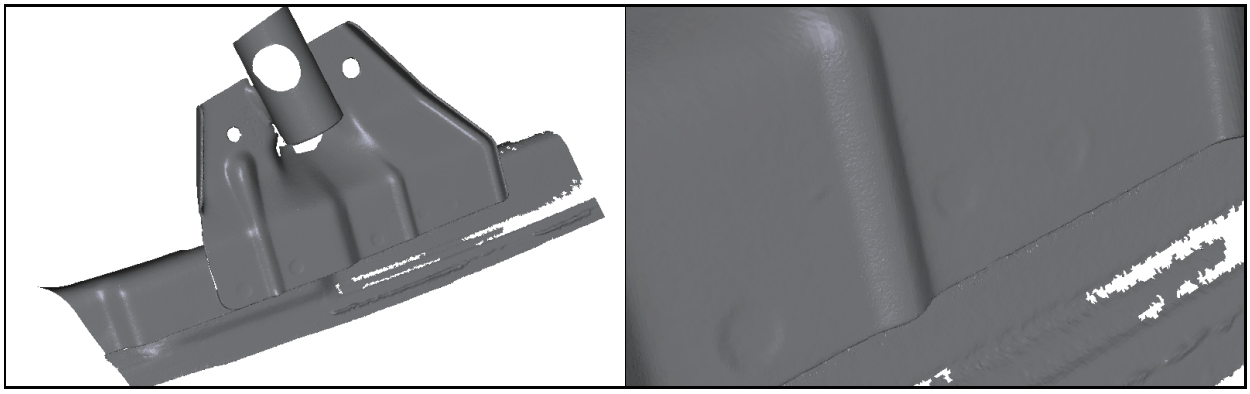


図 3.5: 山谷線の精度 (サンプルデータ 2)



全体像　　拡大図

図 3.6: サンプルデータ 3 の全体像と拡大図

表 3.3: サンプルデータ 3 の極値線の計算

t	0.0005	0.001	0.005	0.01	0.05	0.1
time	248	198	104	98	87	86
L1	0.003	0.003	0.005	0.006	0.02	0.04
L2	0.006	0.007	0.008	0.01	0.03	0.06
MAX	0.39	0.58	0.92	0.42	1.0	2.9
ridge	24480	14699	2552	1518	527	366
valley	18182	11099	1938	1189	528	489

サンプルデータ 3 の頂点数は 314467, 面数は 624093 である.

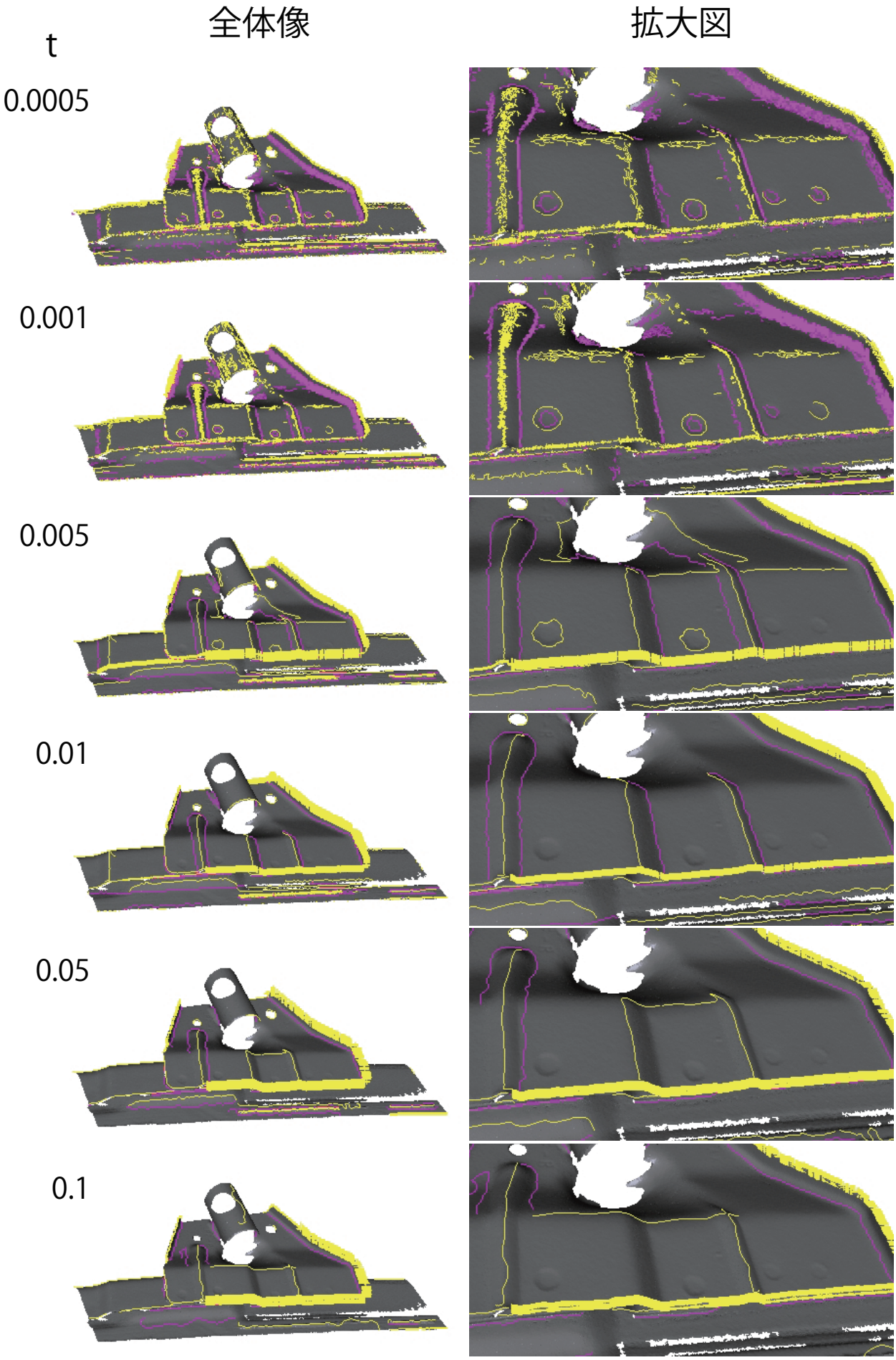


図 3.7: 山谷線の精度 (サンプルデータ 3)

図 3.3 にみられるように, サンプルデータ 1 では,$t = 0.005$ から 0.01 にかけて良い精度の極値線が出ている.$t = 0.0005$,$t = 0.001$ と t の値が小さくなると, ノイズに対して敏感に反応してしまい, うまく溶接端を検出できない. 逆に,$t = 0.05$,$t = 0.1$ と大きくしていくと, もはや溶接端に沿って極値線が検出できているとは言えない.

図 3.5, 図 3.7 も同様で, t が小さくなればなるほど特徴線は短く, 直線的になり, ノイズに対して敏感に反応する傾向にあり,t が大きくなればなるほど特徴線は長く, 滑らかになり, 溶接端部から外れる傾向にある.

サンプルデータ 1〜3 を通して, トレードオフパラメータ $t = 0.005$ がカットに適した結果を出しているため, 今後はこの値を用いて計算を行う (t に関してより詳細な分析は 4.3.3 節で品質と所要時間の評価とともに行う).

3.3 (2) コスト設定

3.3.1 メッシュとグラフ

本研究では, メッシュに対しグラフカットを行う. 本研究で扱う三角形メッシュデータは, 頂点の 3 次元空間上の位置情報,2 頂点を結ぶ辺情報, および 3 辺を結ぶ面情報からなっており, この三角形メッシュをグラフとして捉えると,
メッシュの頂点がグラフのノードに,
メッシュの辺がグラフのエッジ (弧) に
それぞれ対応する.

コスト設定は, さらに 2 つのプロセスから成り立つ.
(*i*) ソース・シンクと繋がるターミナル・ノード間を対話的に指定する
(*ii*) ノード・ノード間に曲率極値線の有無を見つけ, コストを設定する
の 2 つである.

3.3.2 (i) ターミナル・ノード間の設定

ソース及びシンクに接続するノードは,GUI を用いたペイントによって指定する. ユーザ指定の工夫や, ユーザ指定量を変えた際の結果については 4.4 章で詳述する. ソースに繋がれるノードとしてユーザ指定されたノードの集合を P_s, シンクに繋がれるノードの集合としてユーザ指定されたノードを P_t とおく. 図 3.8 のようなインタフェースのシステムを用い,3 次元空間上のメッシュの頂点をペイントする事により, ソースと繋がるノード位置, シンクと繋がるノード位置を対話的に指定できるようにした. 対話的に指定された部分は, グラフカット後確実にソース, シンクのそれぞれに分類されなければならないため, 理想的には無限大のコストを設定する.

また, 指定した部分は, グラフカット後, 確実に指定通りに分類される事を考えれば, 以下の指定ルールを導入する事ができる.

指定ルール (1)**:同じ側に分類したい, グラフのある部分に,ST を同時に指定してはならない.**

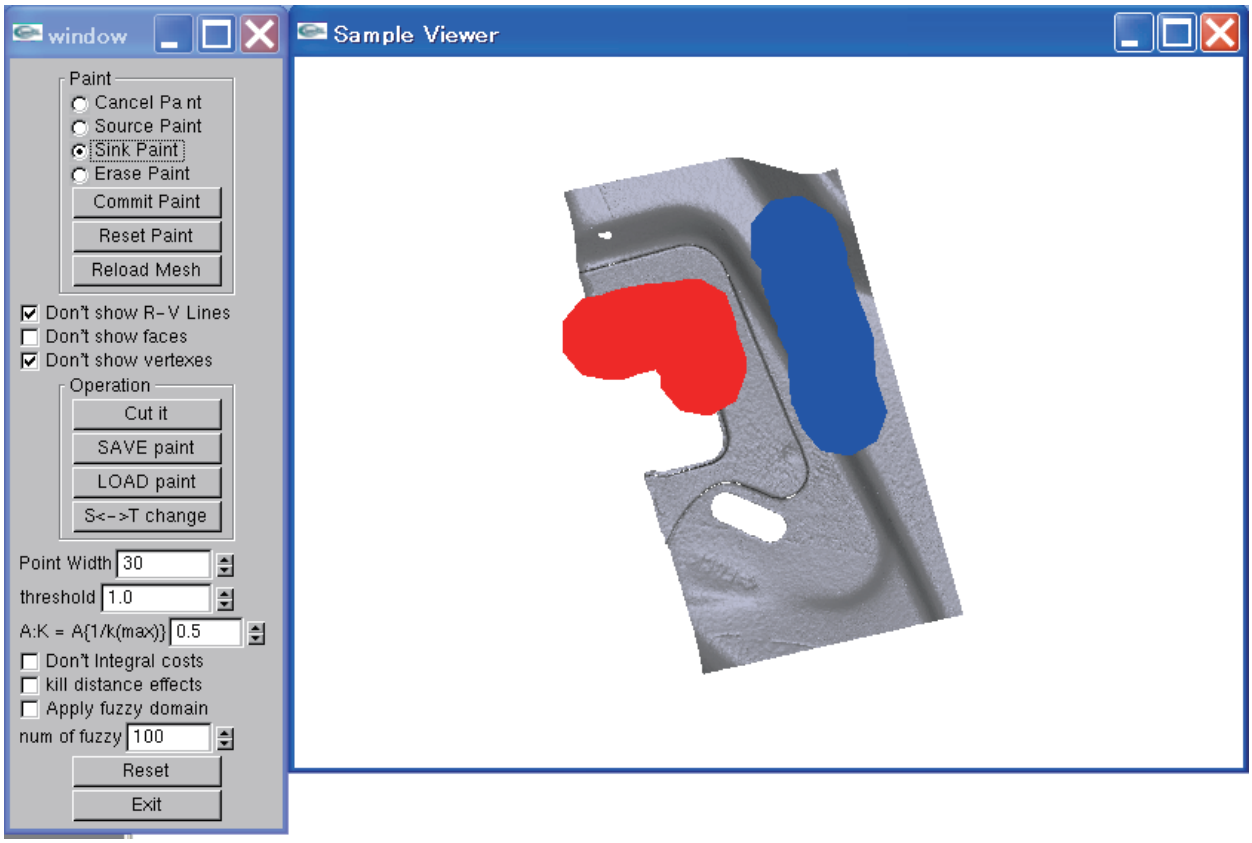


図 3.8: 3 次元ペイントシステム

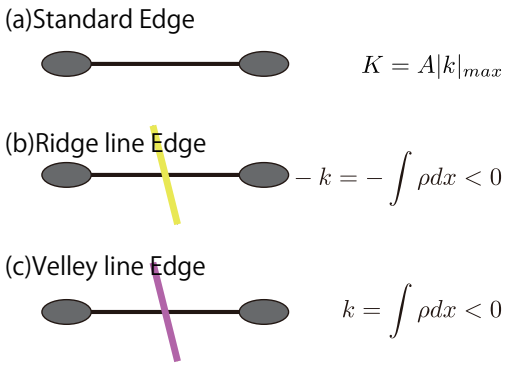


図 3.9: 通常の Edge/山線通過 Edge/谷線通過 Edge

3.3.3 (ii) ノード・ノード間の設定

次に, ノード・ノード間に設定するコスト k について考える. ノードはメッシュの頂点を表すので, ノード・ノード間は, メッシュのエッジに対応する. つまり, ノード・ノード間のコストは, メッシュのエッジに対するコストとなる.

図 3.9 に示すように, エッジには 3 種類の関係しかない. すなわち,

(*a*) 極値線が通らない一般のエッジ
(*b*) 山線の通るエッジ
(*c*) 谷線の通るエッジ

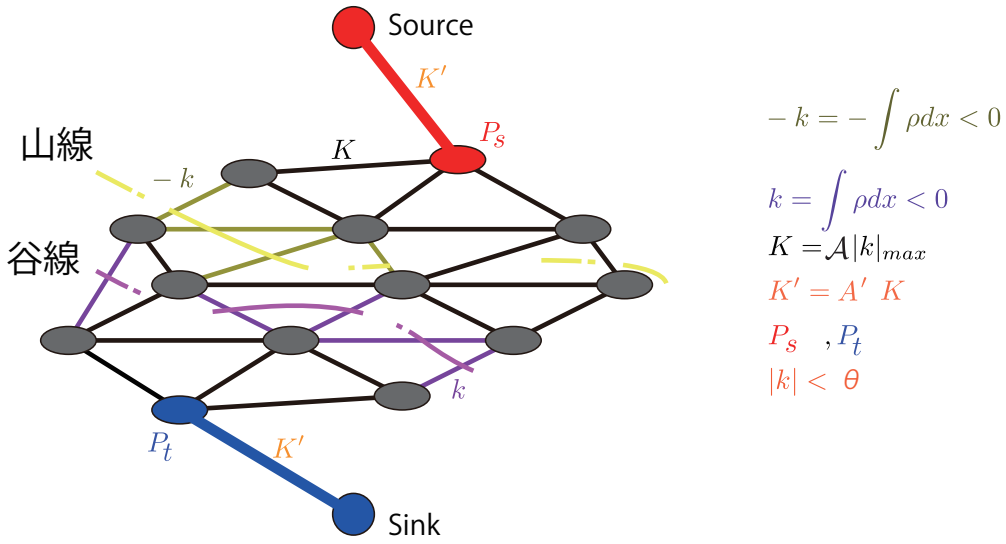


図 3.10: コスト設定とパラメータ

の 3 種類である.

特徴線の曲率を ρ として, 山線 ($k = \int \rho dx > 0$) の通るエッジには,$-k$ のコストを, 谷線 ($k = \int \rho dx < 0$) の通るエッジには k のコストを設定する. すなわち, 極値線の通過するエッジでは, コスト k はどちらも負数をとり, 絶対値が大きくなれば大きくなるほど, そこには強い斥力が働くように設定する. 一般のエッジには, 極値線の強さの最大値を k_{max}, コスト倍率を $\mathcal{A}$ として $K = \mathcal{A}k_{max}$ となるコスト K を設定する. また,$\theta = 5\%$ とし,$|k| < k_{max} \times \theta/100$ となるような極値線を無視した.

また, ソースに分類されるノード, シンクに分類されるノードとして設定したノードと, ソース/シンクそれぞれへのコスト K' は, 理想的には無限大であるが, 計算上 $K' = \mathcal{A}'K$ とし,$\mathcal{A}' = 100$ を用いた. ここで,$k, |k|_{max}$ はメッシュの形状と,3.2 節でみたようにトレードオフパラメータ t のみで決まるから, ユーザ指定位置 P_s, P_t, コスト倍率 $\mathcal{A}$, また, 閾値 θ が, コストの設定という領域で調査すべきパラメータであるといえる.

Y. Boykov, V. Kolmogorov[5] のグラフカットプログラムでは, 負数の入力に対応していないため, 実際には全てのコストが正数となるよう, 全てのコストに k の最小値の絶対値 $|k_{min}|$ を加算し, 負数入力を回避した.[5] によると, そのような処理を行ったときも, 負数での計算結果と変わらない.

閾値の設定　溶接端にそって, 最も大きな特徴線が出るだろうという仮定を置くと, 適切な閾値を設定し, 山谷線の強さの絶対値 $|k|$ が, 閾値以下となるものを無視すると, ノイズが消去でき, 特徴線で綺麗にカットできると推定される. ここで, 極値線の強さやノイズの強さは, 扱うメッシュに依存するはずだから, 閾値 θ を,k_{max} の 100 分率としておく. すなわち,

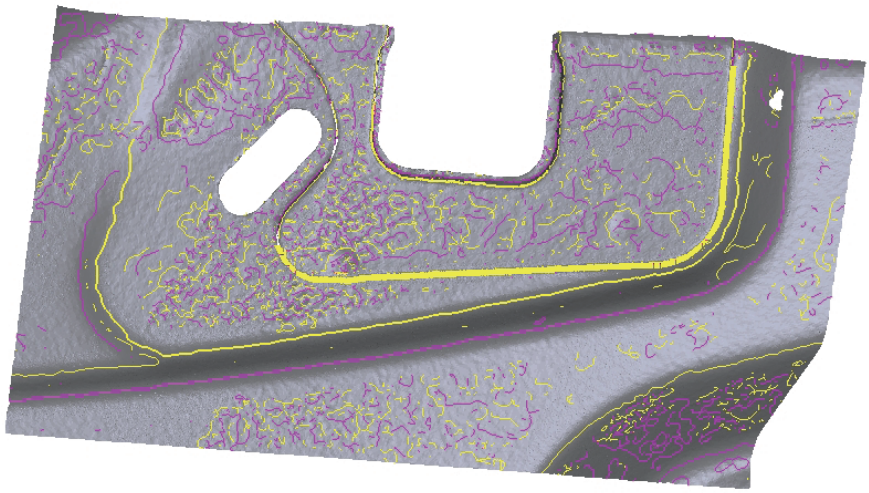
$$|k| < k_{max} \times \frac{\theta}{100} \tag{3.1}$$

となる強さ k の極値線を無視する.

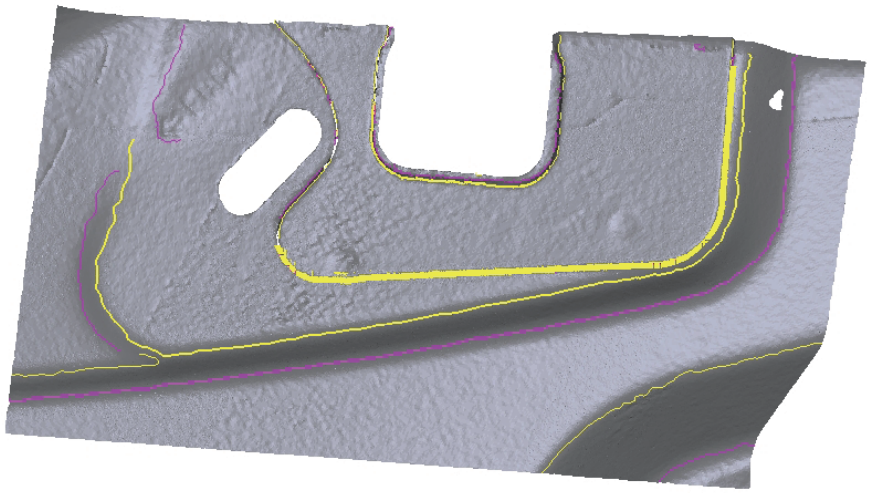
以下, サンプルデータ 1 に対し, メッシュの曲率極値線を $t = 0.001$ で計算し,$\theta = 0\%, 1\%, 5\%$ と変えた結果を示す.

この結果を見る限り,$\theta = 5\%$ のものが溶接端以外の部分での極値線が少ないので, 以下,$\theta = 5\%$ を用いる.

$\theta = 0$



$\theta = 1\%$



$\theta = 5\%$

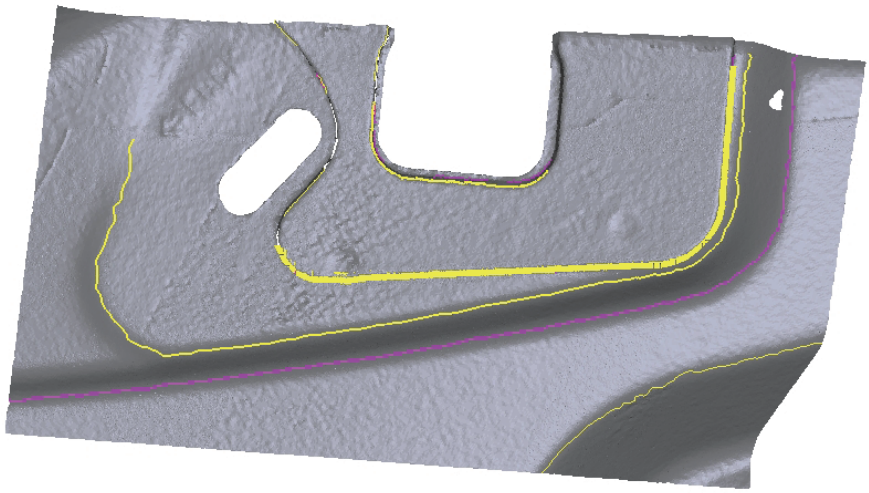


図 3.11: 閾値の設定

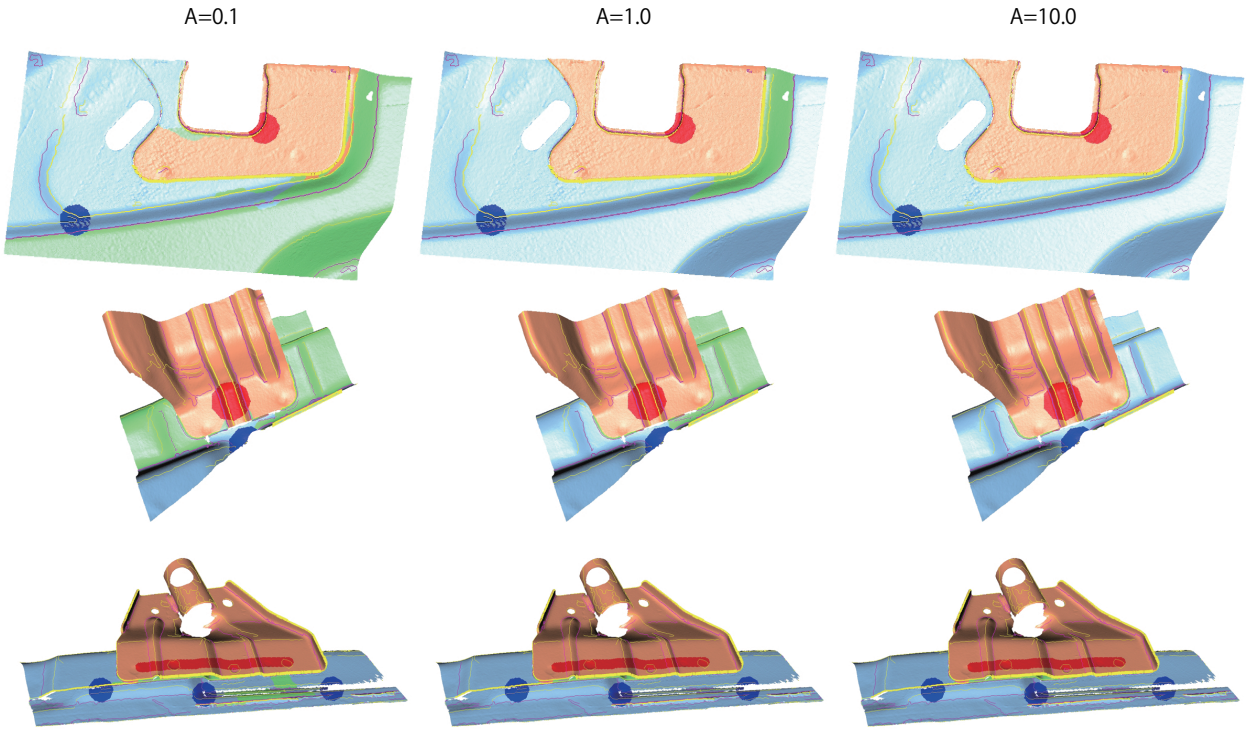


図 3.12: $\mathcal{A}$ の変化と結果

コスト倍率 $\mathcal{A}$ の設定　これまで,3.3.3 節で触れた $K = \mathcal{A} \cdot k_{max}$ の値である. この値を最適化したい.

図 3.12 にサンプルデータ 1,2,3 それぞれに $\mathcal{A} = 0.1, 1.0, 10.0$ で実験した結果を示す.(その他のパラメータはこれまで通りパラメータ $t = 0.005$, 閾値 $\theta = 5\%$ である) 一見して分かるように,$\mathcal{A}$ が大きくなればなるほど, より溶接端に沿った, 分割がなされており, カットの品質が上がっていると言える. また,$\mathcal{A} = 100.0$ でも試したが, 結果は $\mathcal{A} = 10.0$ と変わらなかった. これは, コスト倍率 $\mathcal{A}$ というのは要するに, 極値線が通っていない, 一般のノード・ノード間の結びつきの強さを決定する係数であるから,$\mathcal{A}$ が大きくなればなるほど, 極値線以外の部分では切れにくくなり, より極値線に沿った分割が可能になると言える.

3.4 (3) グラフカット

3.4.1 カット目標

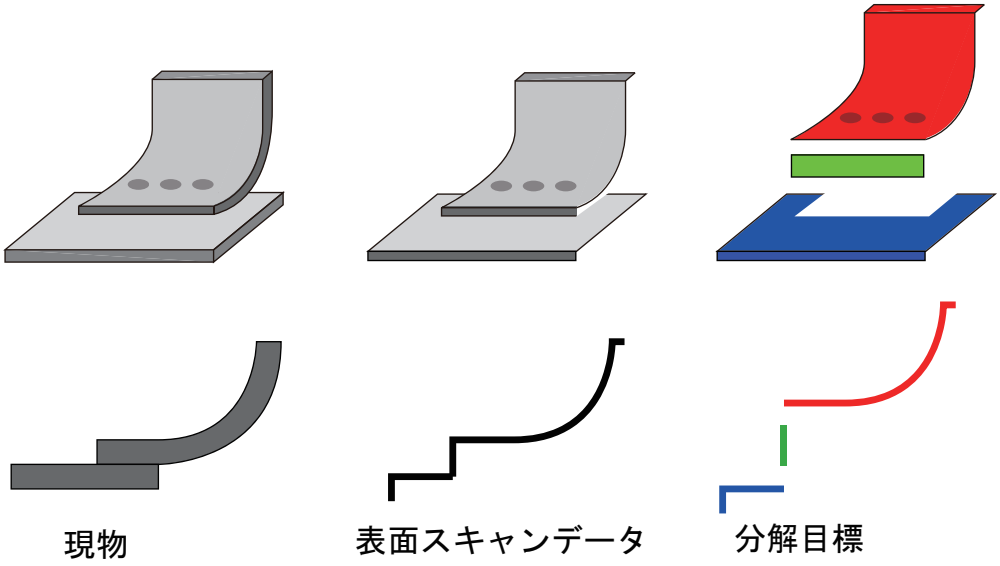
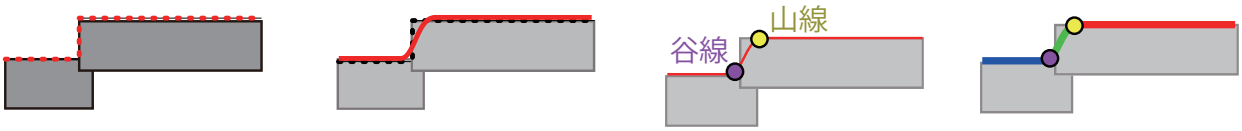


図 3.13: 分解目標の詳細な模式図

カットの目標を今一度確認する. 図 3.13 に示したように, 現物から, その表面スキャンデータを取得し, それを図 3.13 右のような, 上側板金部分である赤色のソース (前景) 部分, 下側板金部分である青色のシンク (背景) 部分, 及び緑色の溶接端部分の 3 つに分ける事を目標とする.



a. 点群をスキャン　b. 関数のフィッティング　c. 山谷ができる　d.3 領域に分けられる

図 3.14: スポット溶接を関数当てはめした時の特徴

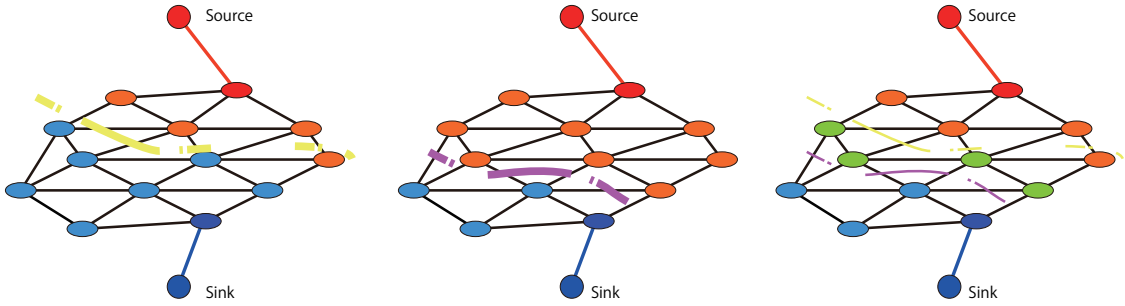


図 3.15: 山線で 1 回, 谷線で 1 回カットし, 狭間領域を捨てる

表 3.4: 2 回カットした際の分類

(1)1 回目	(2)2 回目	最終出力
ソース	ソース	ソース
ソース	ノード	狭間
ノード	ソース	狭間
ノード	ノード	ノード

3.4.2　3 領域へのカット

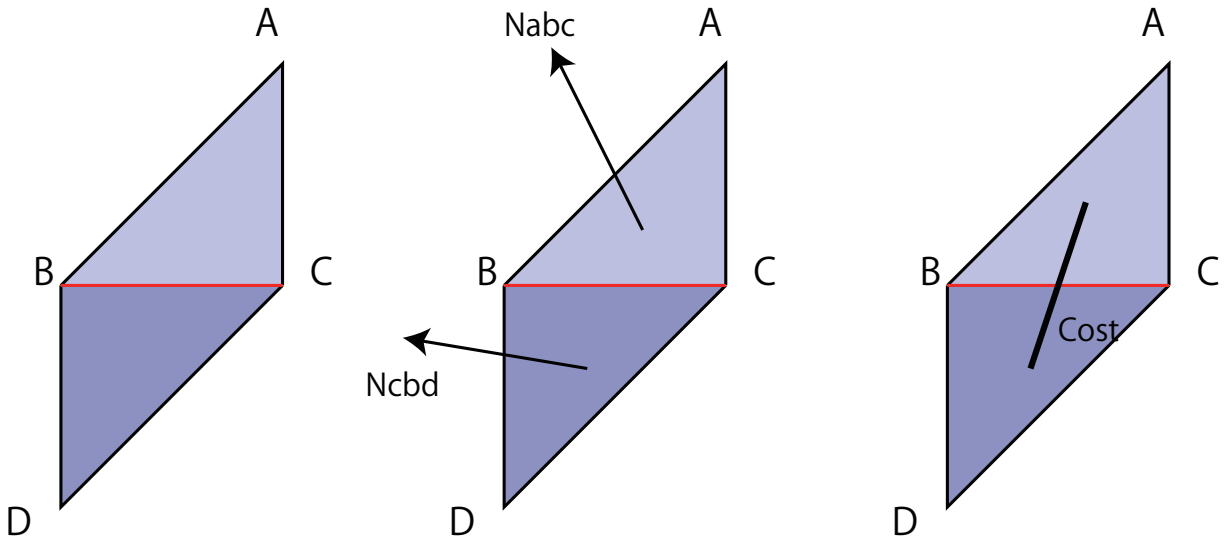
今, 金属板 2 枚をスポット溶接する事を考える.

図 3.14 のように溶接端に沿って, その溶接端を挟み込むように山線と谷線が作らる. 解析には, 板金部品部分の形状データがあればよく, それは図 3.14 の d. に示す赤色部分と青色部分に相当する. 緑色部分に相当する, 溶接端部はむしろ不要である.

そこで,Y. Boykov, V. Kolmogorov[5] のグラフカットプログラムを利用し, 図 3.15 のように, 領域を山線と谷線を利用し 3 つに分ける事を考える. グラフカットは, 一度にグラフを 2 つの部分グラフへと,2 分割する手法であるから, 一度のカットで 3 分割はできない. そこで, 図 3.14d を見ると,

(1)1 回目のカット:山線だけを見てグラフカット
(2)2 回目のカット:谷線だけを見てグラフカット
を行い,

表 3.4 のように, (1) でも (2) でもソースに分類されたものを最終的にソース, (1) でも (2) でもシンクに分類されたものを最終的にシンク, それ以外のものを, 狭間領域として出力するようにすると良さそうである. これによって, 金属板部分 2 つと, 溶接部 1 つ, の 3 領域に分割する事ができるようになる.



a. メッシュの共通パターン　　b. 法線ベクトル　　c. コスト設定

図 3.16: 当初想定していたコスト設定

3.5　なぜ提案手法なのか

はじめ,2.2.4 節, インタラクティブ・グラフカットの研究 [5] でみたように,2 次元画像のセグメンテーションから,3 次元メッシュデータへの応用を着想した.

元々の 2 次元画像へのセグメンテーションにおいて,2 次元ピクセル画像は規則正しく並び, 固定された近傍数を持つなど, そのまま数学的な取扱いが容易であったが,3 次元メッシュデータは, (i) 規則性が無く, (ii) 近傍も多様で, 取扱いが困難であった.

そのため, できるだけ単純化された解法を求め, 探求した. 当初は, どのようなメッシュに置いても成り立つ固有のパターンを探求し, メッシュを観察し 図 3.16a のように, **隣接する 2 つのメッシュは, かならず 1 本のエッジによって分断されている** という事実に着目し, グラフカットを行う事を考えた.

具体的には, 図 3.16b に表した, 三角形 *abc* の法線ベクトル N_{abc}, 三角形 *cbd* の法線ベクトル N_{cbd} を用いて, $|N_{abc} \cdot N_{cbd}|$ を面と面のコスト設定に用い, グラフカットを行う.

以上のような手法を考えたが, **本オブジェクトは平面でも, 計測誤差などで法線方向が一致しない**などの要因により, 上手くいかなかった. その為, 一見, 形状把握に適しているように思われた法線ベクトルの方向の利用ではなく, 少なくとも溶接端部をほぼ通る曲率極値線の研究の応用を考えた.

第 4 章

実装と結果

第 4 章

実装と結果

この章では, 実際にプログラムを作り, カットを行った際の値の設定や, その結果, 及び結果からのフィードバック改善を見る.

4.1　スキャンと現物

本研究で扱った 3 次元データは, 図 4.1 のような光学式スキャナを用い, ピッチ 0.3mm, 深度 0.05mm でスキャンを行ったデータを用いた. ピッチ · 深度とは図 4.2 に示したように, ピッチは現物から取得する点群間の xy 平面内での値, 深度は z 軸方向に取得できる点群の間隔, すなわち z 軸方向の分解能のことである.

図 4.1: 非接触光学式 3 次元デジタイザ

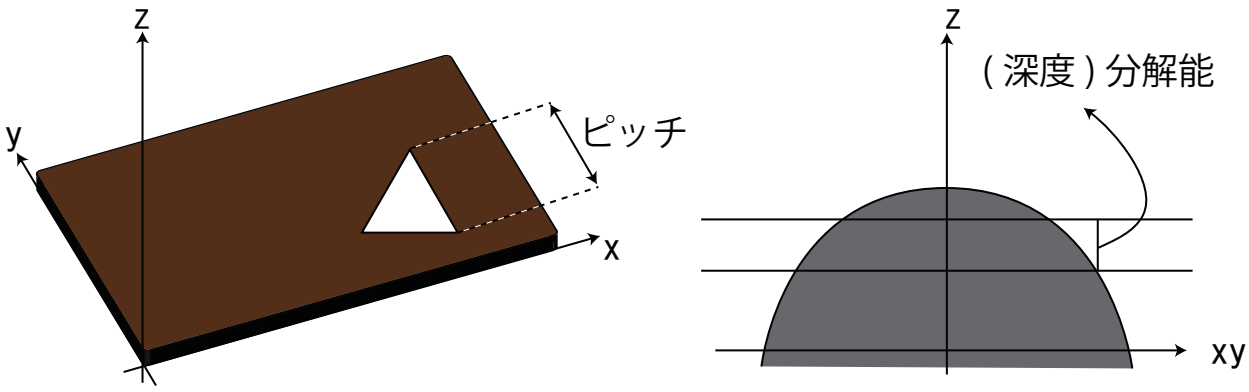


図 4.2: スキャンのピッチと深度

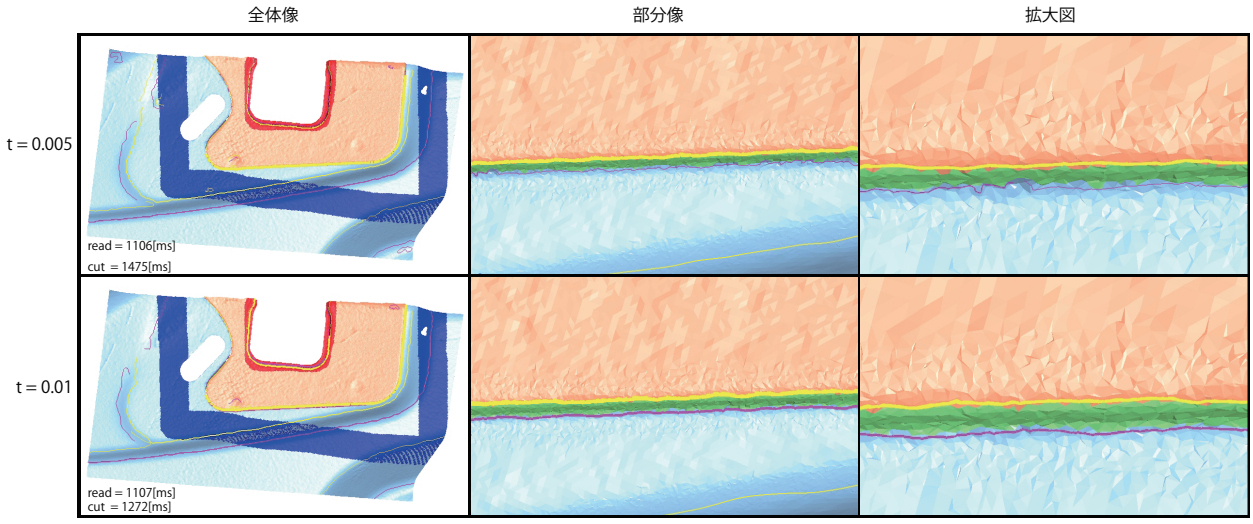


図 4.3: カット結果 (サンプルデータ 1)

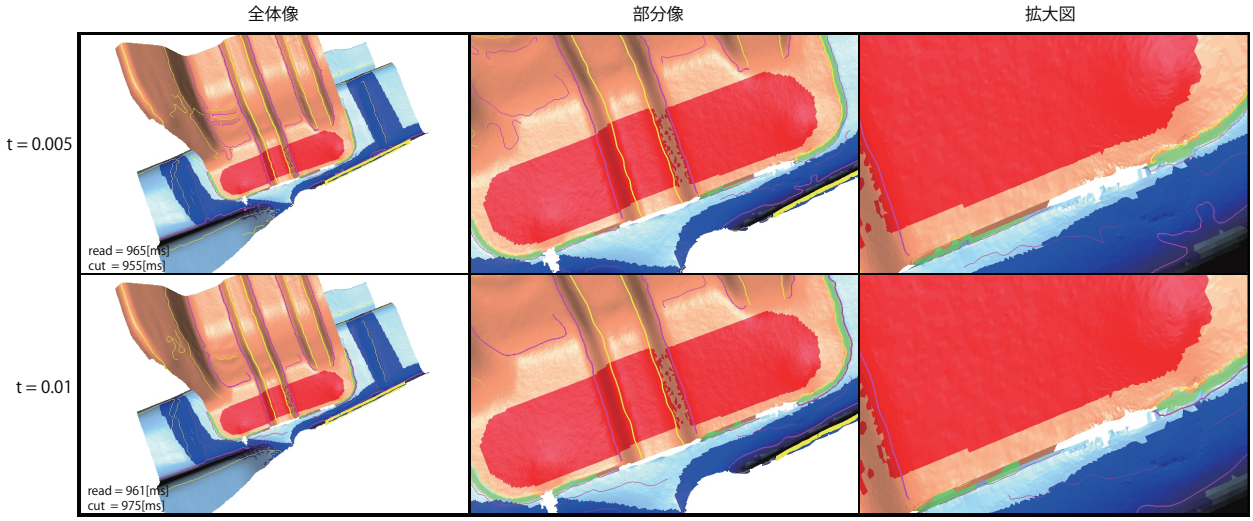


図 4.4: カット結果 (サンプルデータ 2)

4.2 グラフカット

3.2 節でみたように, サンプルデータ 1 に対しては $t = 0.005$ の曲率極値線が, 比較的歪みが少なく, 溶接部分を綺麗にとれている, 良い極値線計算結果であるといえる.

そこで, それらの結果に対して, 実際にグラフカットを適用した結果を以下に示す. 以下, 赤色部分は, 対話的にソースとなるよう指定したノード, 青色部分は対話的にシンクとなるよう指定したノード, また, 橙色部分はカットの結果ソース側に振り分けられたもの, 水色部分はカットの結果シンク側に振り分けられたもの, 緑色部分は, 溶接された部分を示す, 狭間領域である. これまでと同様, 曲率極値線は山線が黄色, 谷線が紫色で示されている.

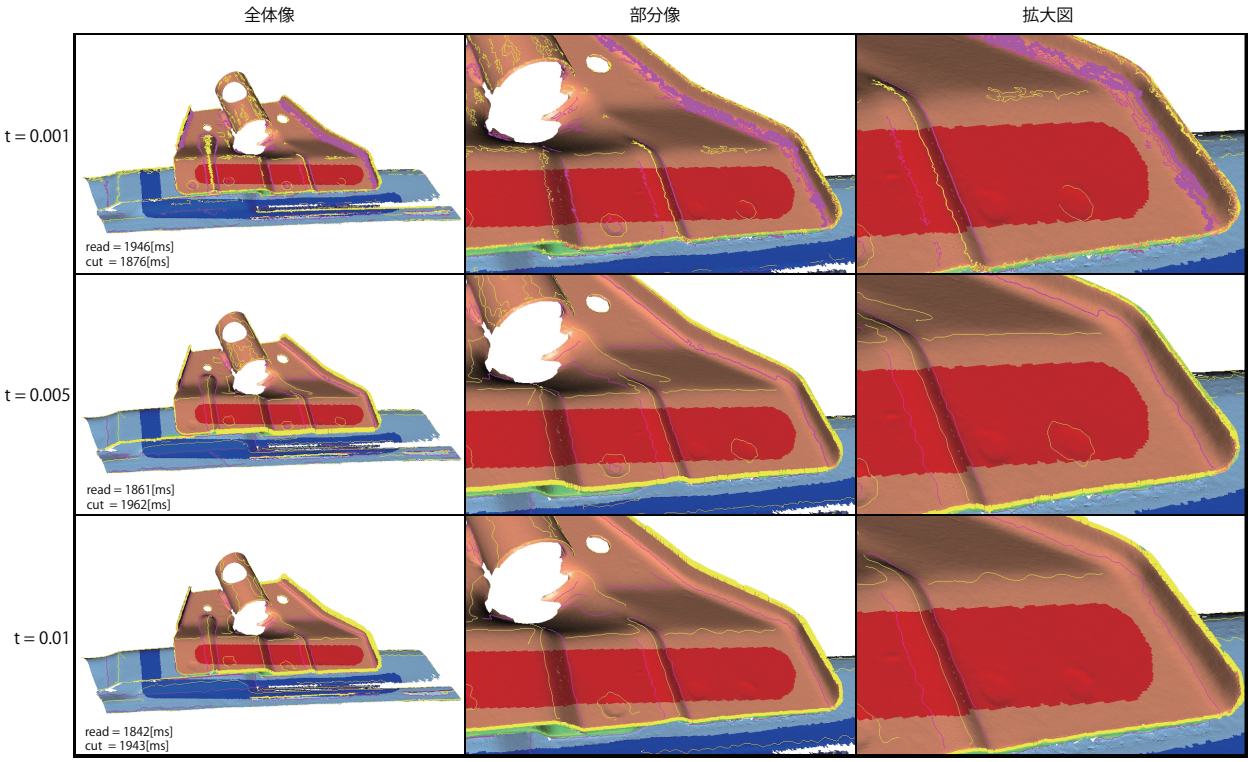


図 4.5: カット結果 (サンプルデータ 3)

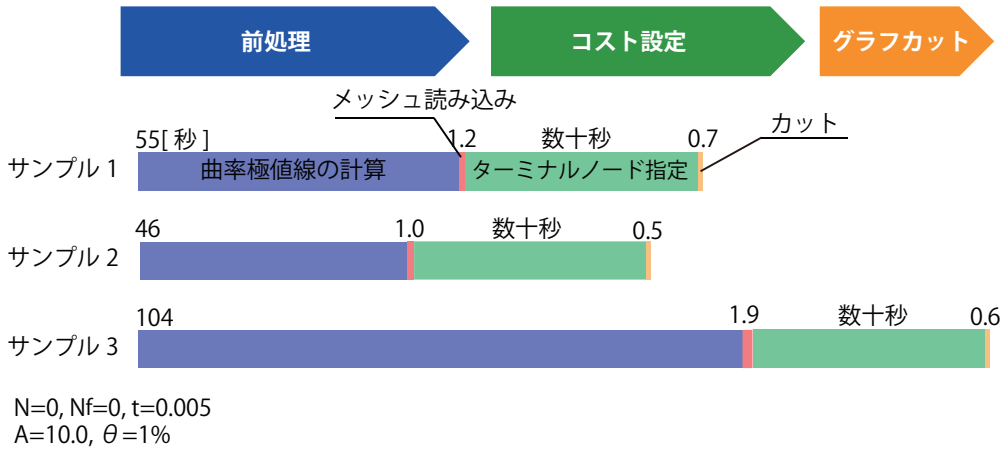


図 4.6: 各プロセスと所要時間

4.3 評価と改善

4.3.1 時間の評価

これまで一部品あたり数分〜十数分かかっていた, 部品分解の時間を, どこまで減らすことができたか見る. 図 4.6 には, 本手法の各プロセスとその所要時間を記した. まず, どのデータをとっても,3 分以下. 特にパーツを読み込んでから分解するまでの時間は数十秒と, 一部品あたりの所要時間はこれまでの手動によるメッシュ加工の時間に比べ, 現時点で既に大幅に減っている事がわかる.

表 4.1: 品質の良い t の値と計算時間

	t	時間 [sec]	t	時間 [sec]
サンプル 1	0.005	55	0.01	52
サンプル 2	0.005	46	0.01	44
サンプル 3	0.001	109	0.005	104

時間に関してのボトルネックがあるとすれば, 図 4.6 の積み上げグラフの割合が示すように, 前処理の曲率極値線の計算と, ユーザのターミナルノード指定の 2 つである. 時間の評価と改善については, この 2 領域に関わる問題に集中して取り組む.

4.3.2 品質の評価

1.4 節でみたように, 品質には,2 段階ある. 特徴線の品質と, カットの品質である.

特徴線の品質は,**1. 溶接端部をきちんと通っているか,2. それは滑らかで連続的に通っているか**の 2 点に尽きる. これは, スムージングなどを行わなければ, 基本的にはトレードオフパラメータ t にのみ依存する.

逆に, カットの品質は

(1). 一部は特徴線を通り, 他は大幅に外れるか

(2). 綺麗に特徴線を通っているか

のどちらかしかなく, 品質の悪い時は大きくカット目標から外れ, 品質の良い時は綺麗に特徴線に沿ってカットができる. つまり, カットの品質は, 一見して溶接端部で切れていれば, 良い品質であると言えるし, そうでなければ悪いといってよい. 換言すれば, 品質は, 閾値 θ, コスト倍率 $\mathcal{A}$, ターミナルノード指定位置 P_s, P_t などのパラメータに敏感に反応するため, 評価は容易である.

4.3.3 曲率極値線の導出と評価

同じスキャナで同じように取得した点群のサンプルデータ 1,2,3 に対するフィッティング関数とそこから導出した曲率極値線について評価する. ここでは, 極値線の結果を左右するのはパラメータ t だけであるから,t について変化させたときの品質と所要時間について考える.

表 4.1 に示したように,$t = 0.005$ が, サンプルデータ 1,2,3 を通して良い結果を出している. 以下,$t = 0.005$ の近傍,$t = 0.001$ から $t = 0.01$ まで,0.001 刻みで変化させ, その結果の差異を探った.

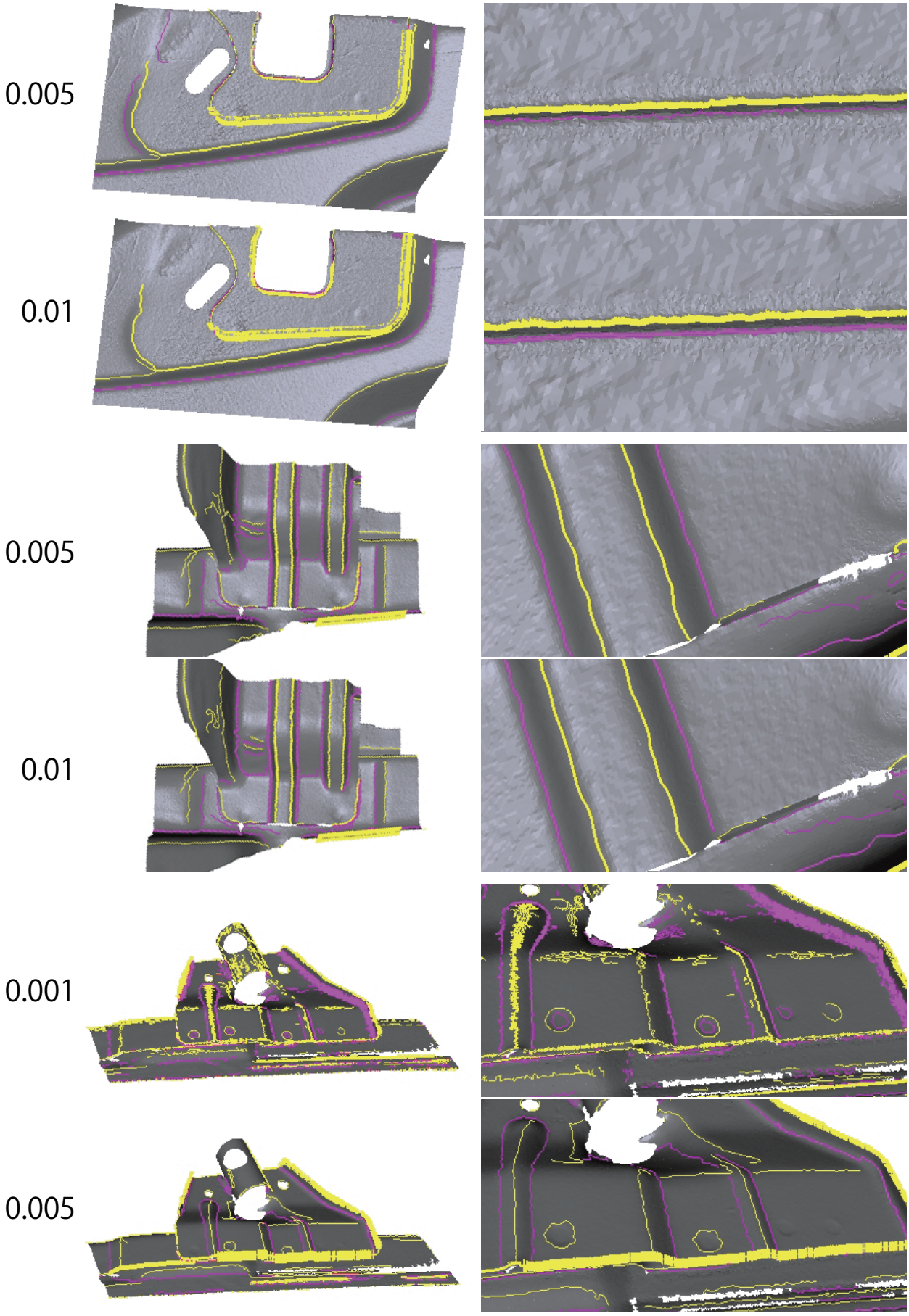


図 4.7: 品質の良い t の値とサンプルデータ 1,2,3

表 4.2: サンプルデータ 1, $t = 0.001$ から $t = 0.01$ まで

t	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009	0.01
time[sec]	93	67	60	55	53	52	52	51	50	50
L1[mm]	0.002	0.003	0.003	0.004	0.004	0.005	0.005	0.005	0.006	0.006
L2[mm]	0.004	0.005	0.005	0.006	0.006	0.007	0.007	0.007	0.008	0.008
MAX[mm]	0.26	0.50	0.42	0.62	0.60	0.50	0.50	0.56	0.57	0.56
ridge	8908	4263	2679	1816	1355	1027	809	611	558	447
valley	9316	4467	2826	1974	1371	1017	788	659	564	504

表 4.3: サンプルデータ 2, $t = 0.001$ から $t = 0.01$ まで

t	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009	0.01
time[sec]	78	55	49	47	45	45	44	44	44	43
L1[mm]	0.002	0.003	0.003	0.004	0.004	0.004	0.005	0.005	0.005	0.005
L2[mm]	0.003	0.004	0.004	0.005	0.005	0.006	0.006	0.006	0.007	0.007
MAX[mm]	0.207	0.227	0.218	0.209	0.205	0.201	0.198	0.210	0.210	0.189
ridge	5133	1818	880	533	438	369	299	265	266	247
valley	4953	1673	787	481	359	325	301	264	246	238

表 4.4: サンプルデータ 3, $t = 0.001$ から $t = 0.01$ まで

t	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009	0.01
time[sec]	134	117	110	105	104	99	98	97	96	96
L1[mm]	0.003	0.004	0.004	0.004	0.005	0.005	0.005	0.006	0.006	0.006
L2[mm]	0.007	0.007	0.007	0.007	0.008	0.008	0.008	0.009	0.009	0.010
MAX[mm]	0.583	0.537	0.510	0.870	0.918	0.762	0.704	0.424	1.171	0.424
ridge	14699	6627	4250	3172	2552	2186	1931	1797	1669	1518
valley	11099	5139	3354	2440	1938	1679	1497	1316	1255	1189

トレードオフパラメータ t の値の変化

これらの結果から, 第 1 に, トレードオフパラメータ t を小さくし, 点群へのより局所的なフィッティングを目指す事で, 指数関数的に計算時間および検出できる山谷線の本数が増大する事がわかった. 誤差は単純平均, 自乗平均ともに t を小さくすることで, 確かに誤差は少なくなっていることがわかった. 誤差の最大値には当然ながらばらつきが出た. 簡単に最適値の t を特定しうる値を探るため, これらの数値の相関や, 山谷線の数について調べたが, 最適値に有意な相関は見つからなかった.

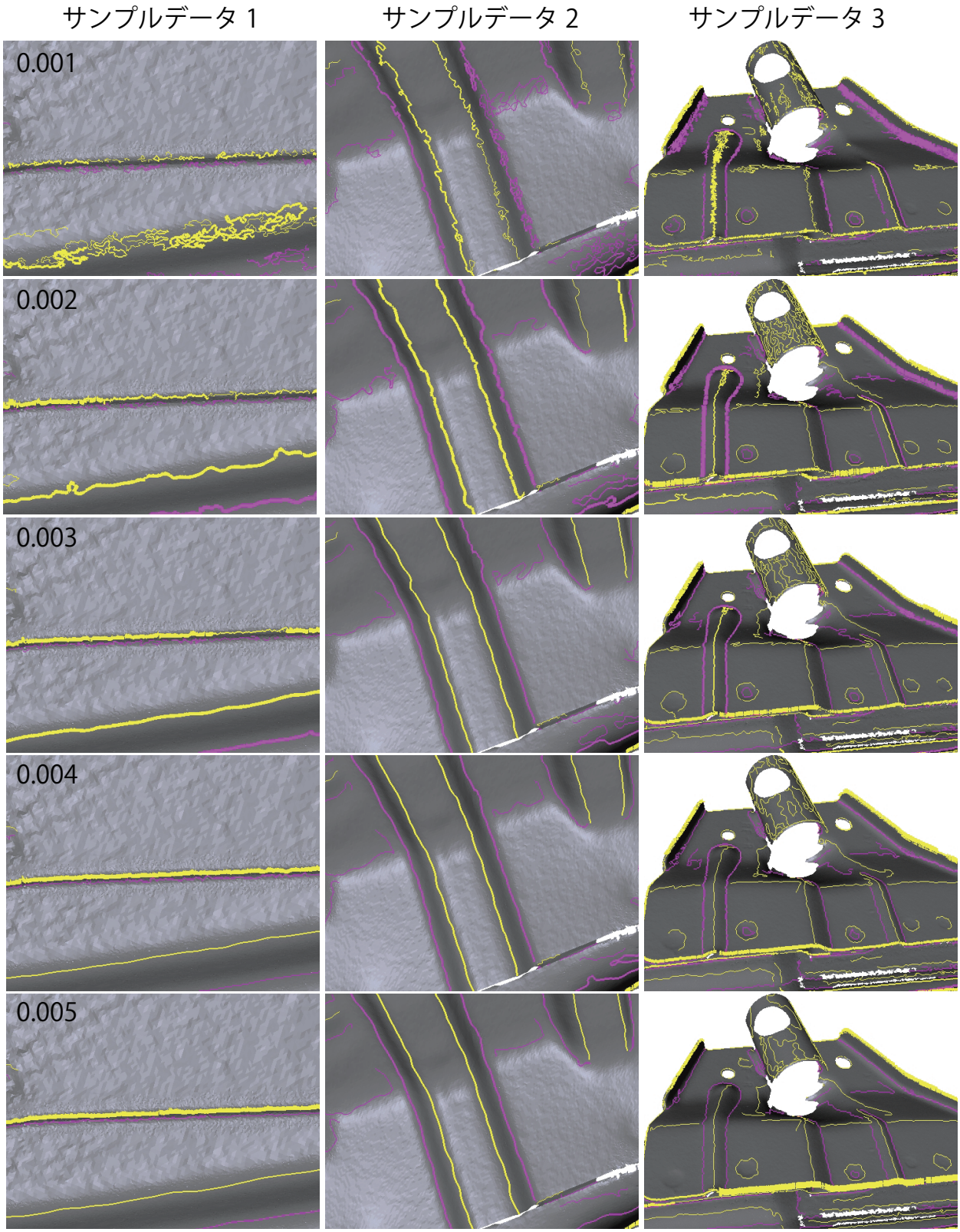


図 4.8: サンプルデータ 1,2,3 における t 値と極値線 ($t = 0.001 - 0.005$)

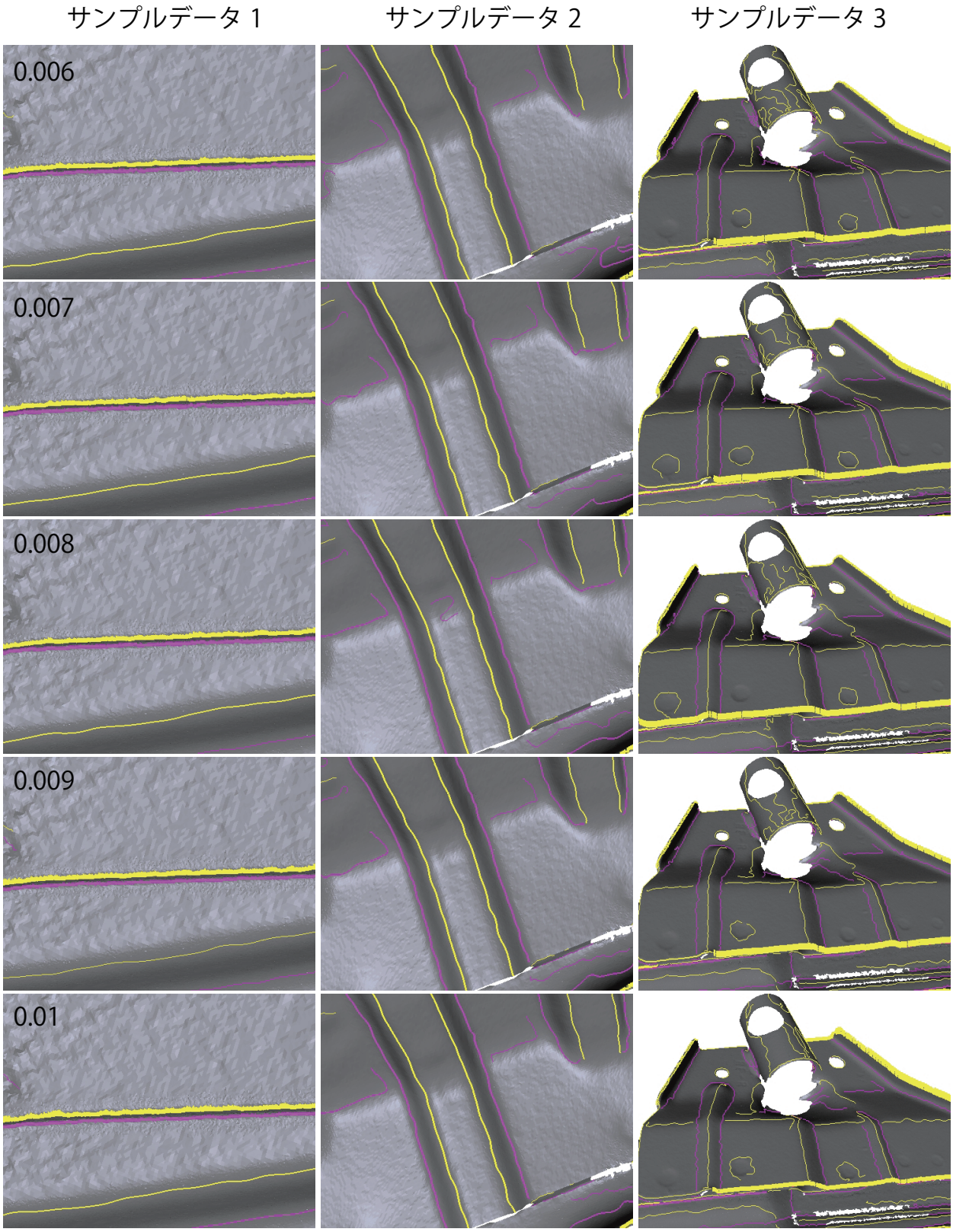
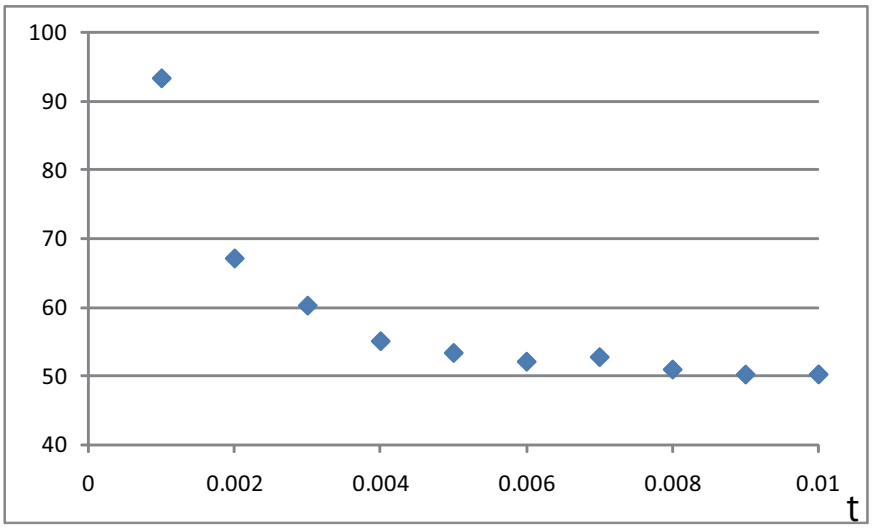
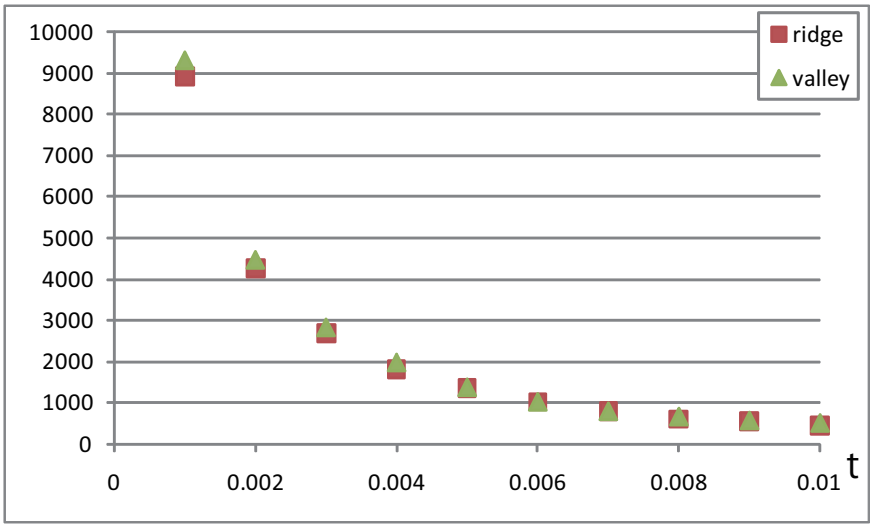


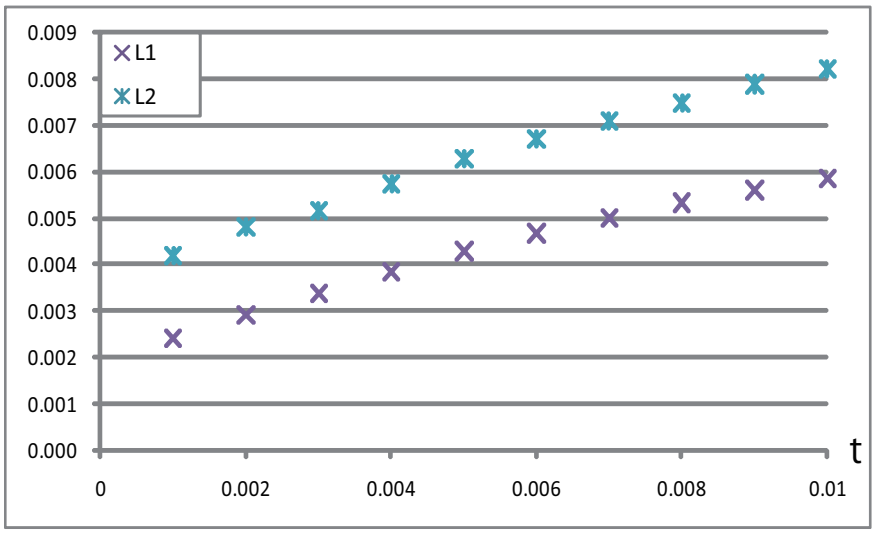
図 4.9: サンプルデータ 1,2,3 における t 値と極値線 (t = 0.006 – 0.01)



t の値と時間 [sec]

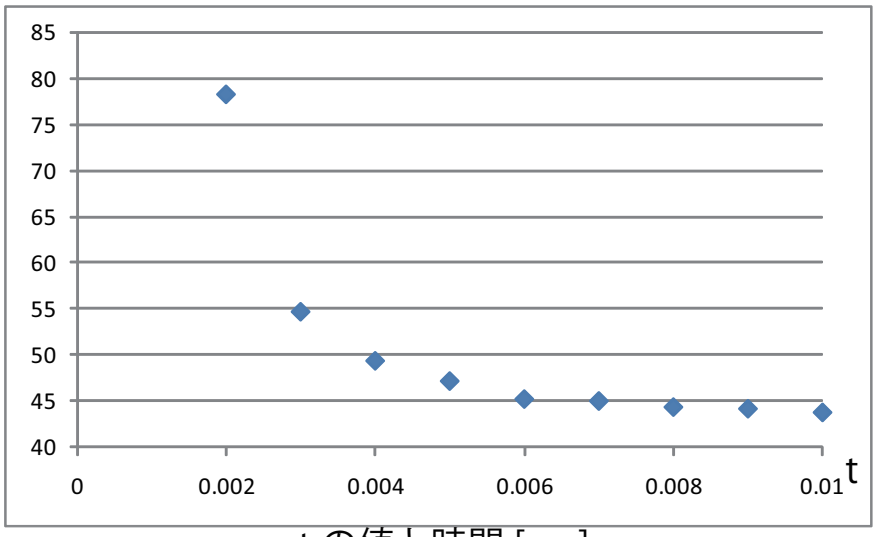


t の値と山谷線の本数

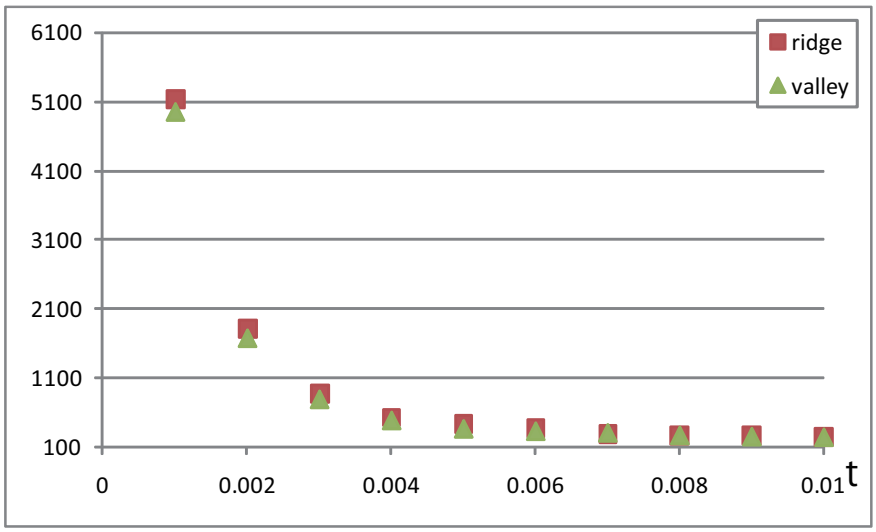


t の値と L1, L2 エラー [mm]

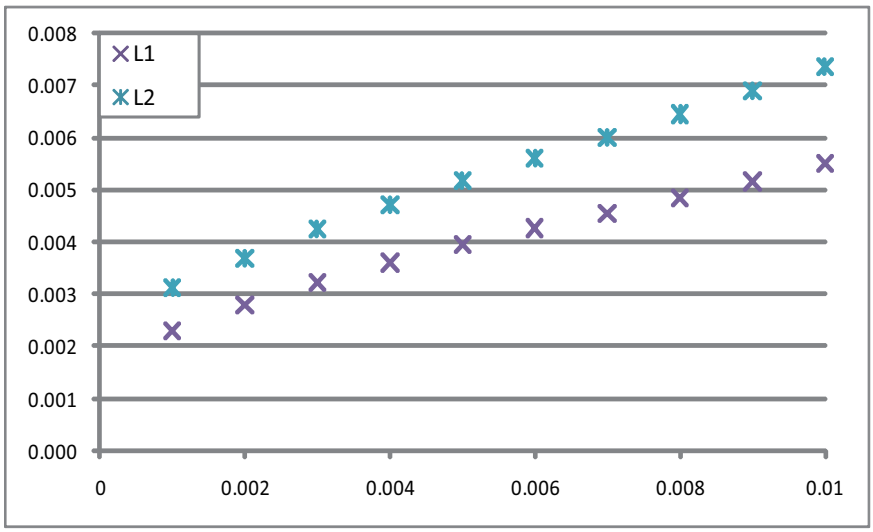
図 4.10: サンプルデータ 1 における t 値と時間, 山谷線の本数, エラー値の関係



t の値と時間 [sec]

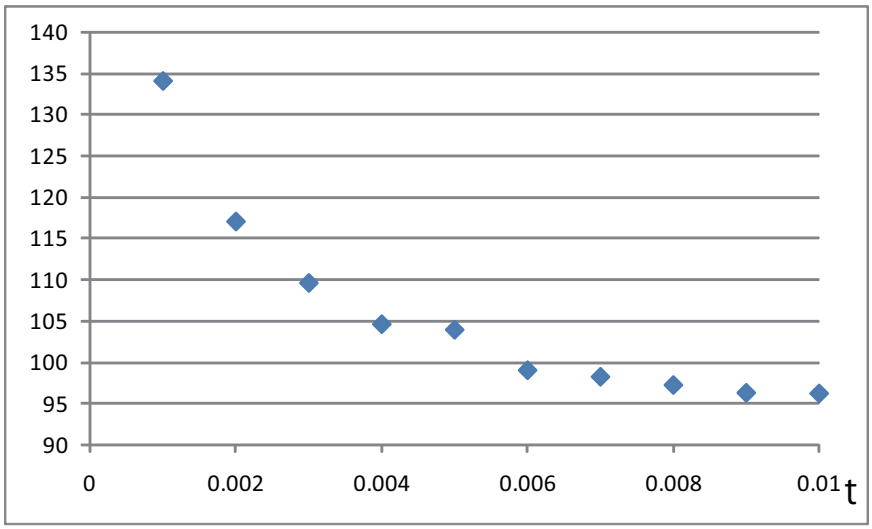


t の値と山谷線の本数

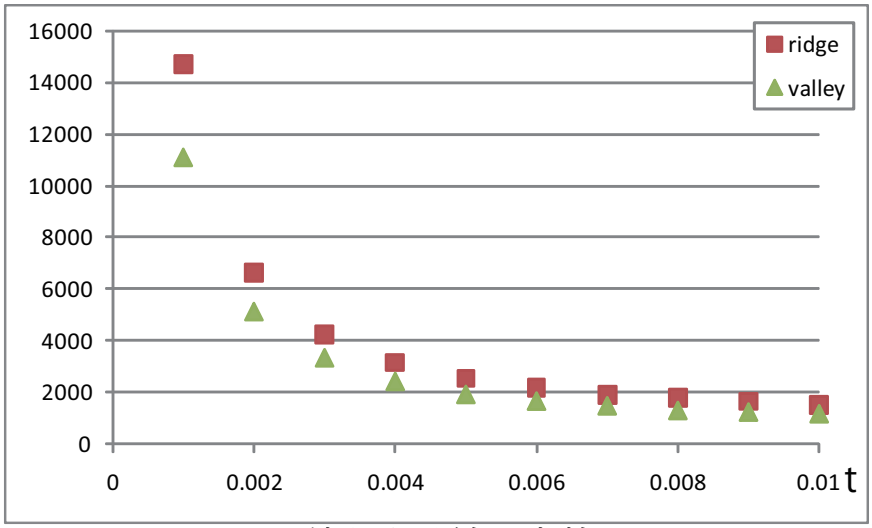


t の値と L1, L2[mm]

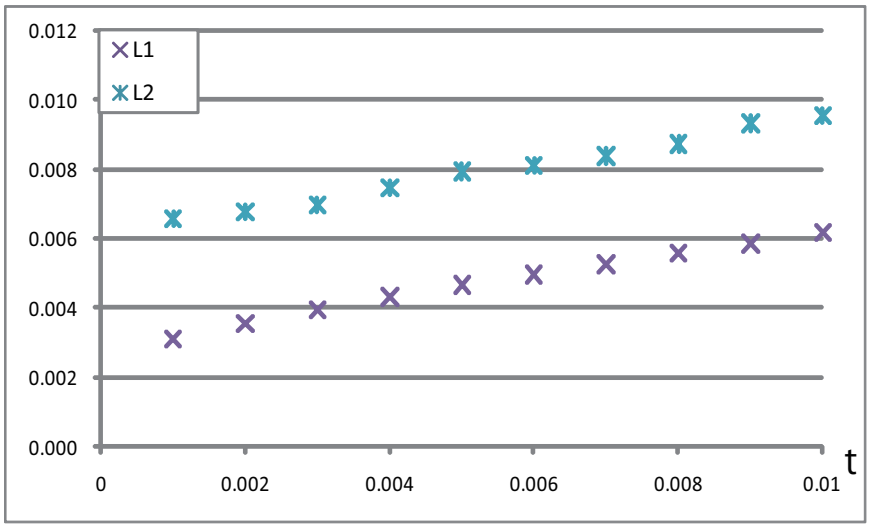
図 4.11: サンプルデータ 2 における t 値と時間, 山谷線の本数, エラー値の関係



t の値と時間 [sec]



t の値と山谷線の本数



t の値と L1, L2[mm]

図 4.12: サンプルデータ 3 における t 値と時間, 山谷線の本数, エラー値の関係

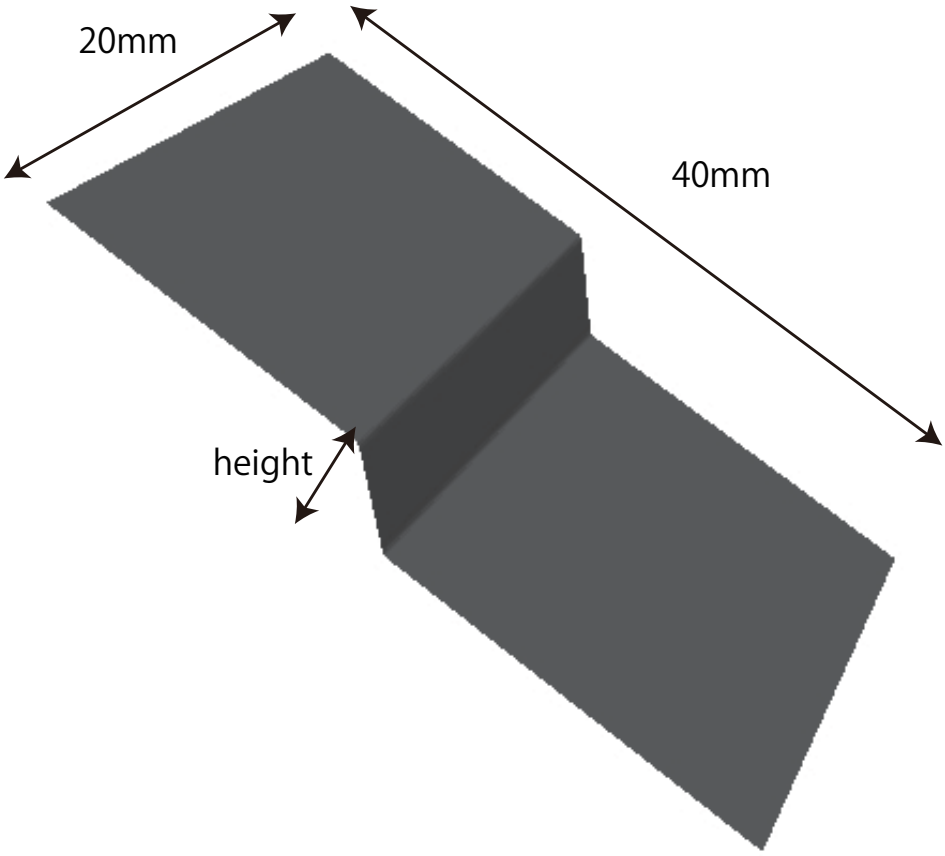


図 4.13: 理想メッシュ

理想メッシュにおけるトレードオフパラメータ t

ここで,t と, メッシュの溶接端部分の高さ *height* の関係について考える. 上側板金の厚さ *height* が, 曲率極値線導出時に相関があるのではないか, と考えたためである.

はじめ図 4.13 のような縦 20mm × 横 40mm の理想的な (ノイズの無い) 平面メッシュに, 高さ *height* をつけて実験を行った. $height = 1mm, 2mm, 3mm, 4mm, 5mm$ における,$t = 0.0001, 0.001, 0.01$ を図 4.14 に示す. 図 4.14 からは, 高さに対する有意な相関は認められない.

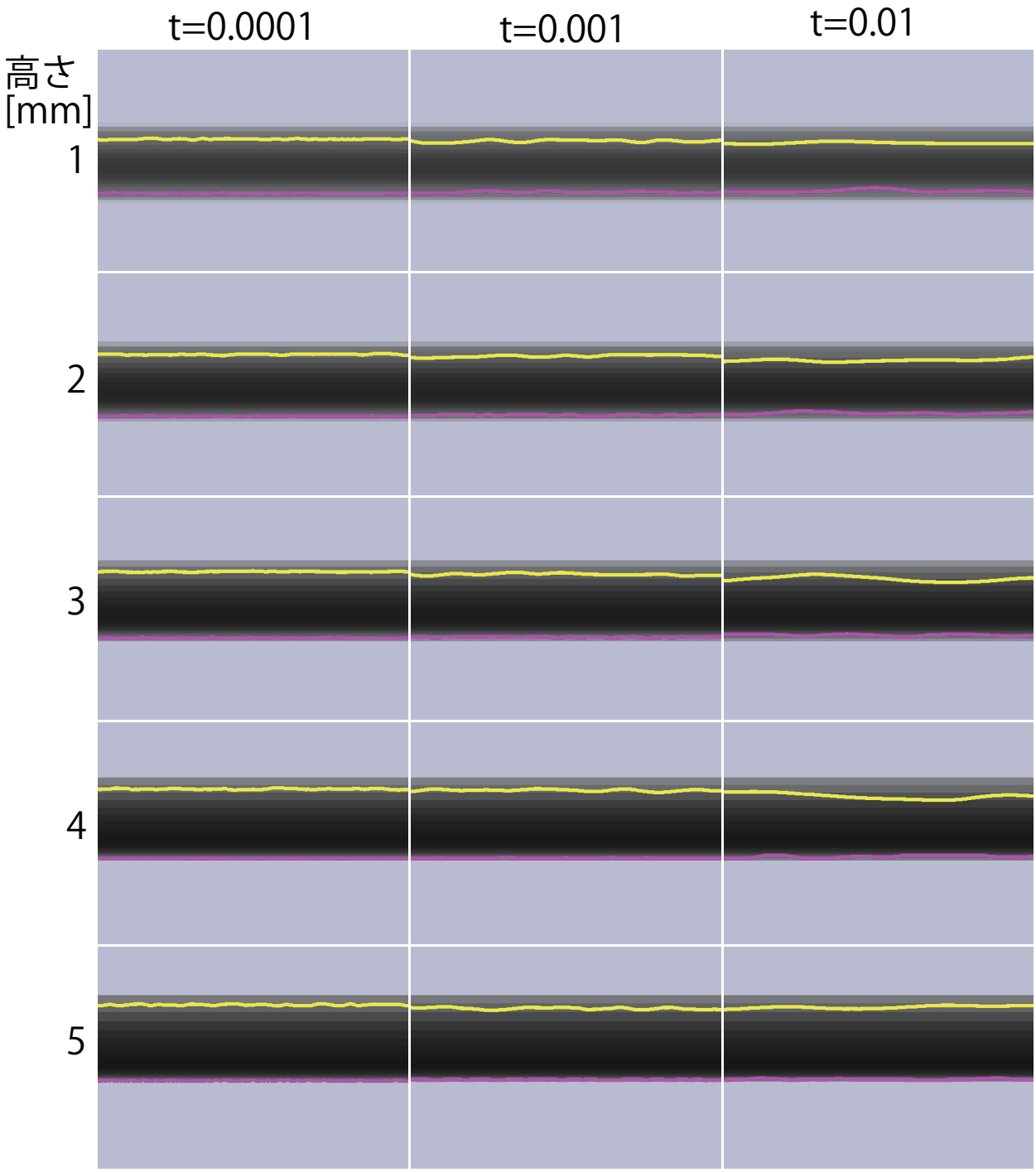


図 4.14: 理想メッシュの高さと t

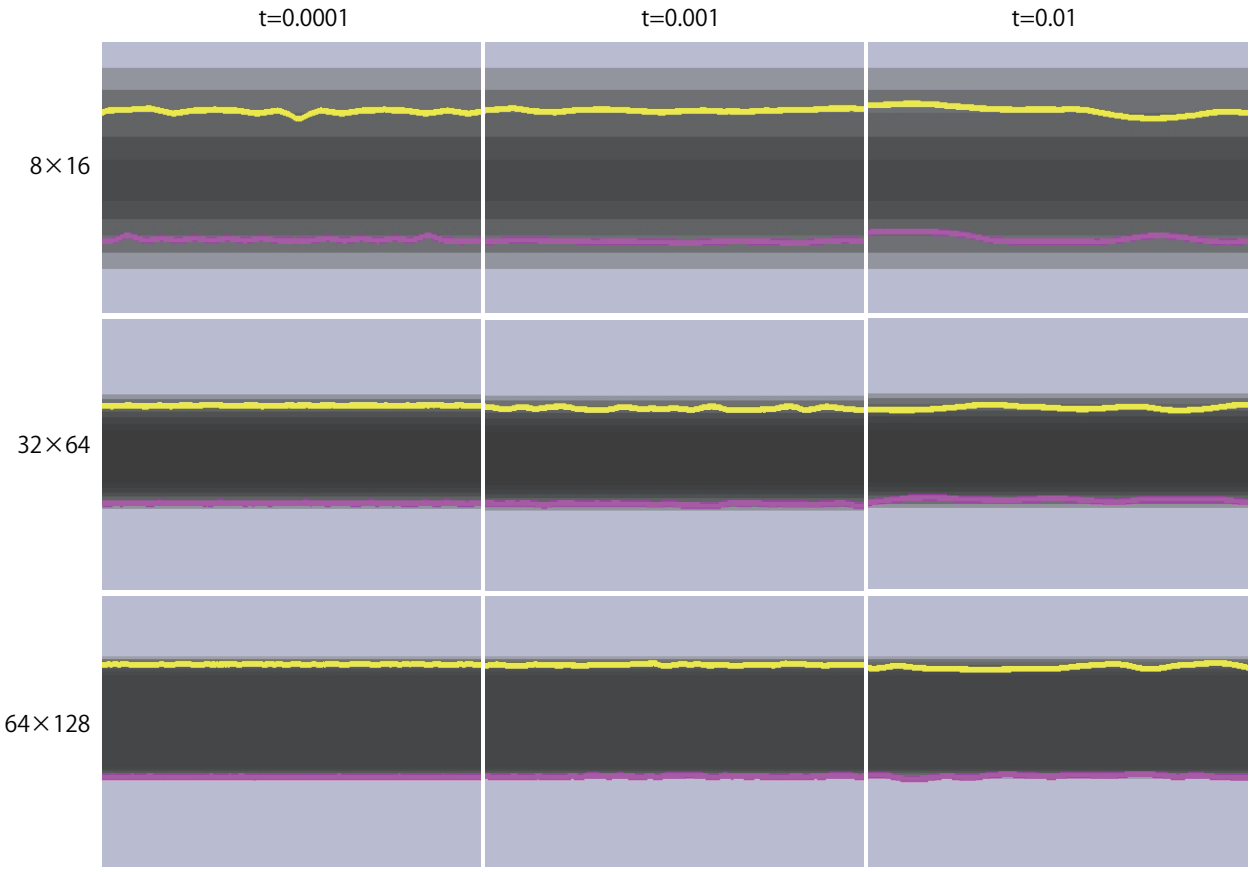


図 4.15: 理想メッシュの粗密と t

そこで, 次に, 理想メッシュの点群の粗密を変え, 実験を行った. 縦 20mm × 横 40mm の平面メッシュに対し, それぞれ, 縦横が 8×16,32×64,64×128 と, 点群の密度を細かくし, 実験を行った.

この場合は, 相関が見られる. すなわち, メッシュの密度が上がれば, より大きな t を用いても, 適した曲率極値線が導出可能である.

これは, 現物スキャンによって取得したメッシュの粗密は 3 次元スキャナの分解能に依存するから, 少なくともノイズが無い理想的な状況下では, あるいは, ノイズが均等に乗る場合, 同じスキャナを用いれば, 同じ t の値を最適値として用いて良いと言える.

実際, サンプルデータ 1〜3 は, 同じスキャナによってスキャンされたデータであるが, そのいずれもがほぼ同じ t の値 ($t = 0.005$ 付近) で良いカット結果を出せる事から,t の最適値の同定は始め一度やればよく, 後は同じスキャナで, 同じ分解能によるスキャンを繰り返し, かつ, ノイズの出方がそれぞれで均一である場合には, 同じものを用いて良いと結論できる.

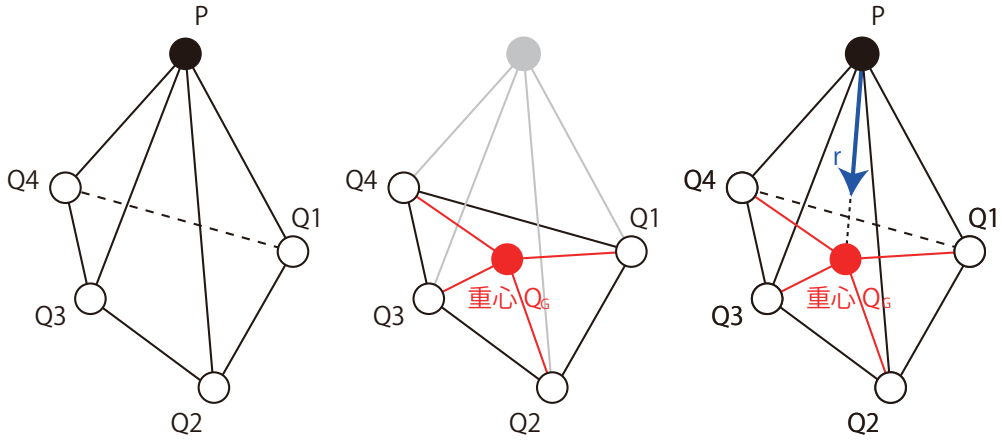


図 4.16: Laplacian Flow 法の概要

フィッティング関数とスムージング

スムージングは, メッシュが滑らかになるように, 頂点をその周囲の頂点位置を参考に移動させる手法である. 自動車などに用いられる溶接部品は, 平な箇所が多いため, 適度にメッシュをスムージングすることで, 計測ノイズや測定誤差に打ち克てる可能性がある. そうして, ノイズの少ない, 溶接端部を中心に極値線を検出できる可能性がある.

そこで,Laplacian Flow 法 [6] によるメッシュスムージングを行う. Laplacian Flow 法とは, 図 4.16 のように, ある頂点 P から繋がる N_q 個の頂点 $Q_i(i = 1, 2, \cdots N_q)$ の重心 $\frac{1}{N_q}\sum_i Q_i$ を Q_G とすると,P から Q_G へ向けて, スムージングファクターを r として,PQ_G を $r : 1 - r$ に内分する点まで動かすという操作を同時に全ての頂点に対して何回か繰り返すという手法である.

すなわち, 三角形メッシュの頂点の一つ P が N_q 個の頂点と隣接するとき, その反復回数 (Iteration) を λ, スムージングファクターを r とすると,

$$P_{\lambda+1} = (1 - r)P_\lambda + r\frac{1}{N_q}\sum_i^{N_q} Q_i \tag{4.1}$$

となるような頂点移動を, 全ての頂点に対し同時に行う手法である.

いま, 反復回数 $Iteration = 10$, スムージングファクター $r = 0.5$ を 1 セットとして,N セット適用し, 比較した. 以下, サンプルデータ 1,2,3 にそれぞれ $N = 0, 1, 5$ セット適用し, 曲率極値線を導出した結果を表 4.5 及び図 4.17 図 4.18 図 4.19 に示す.

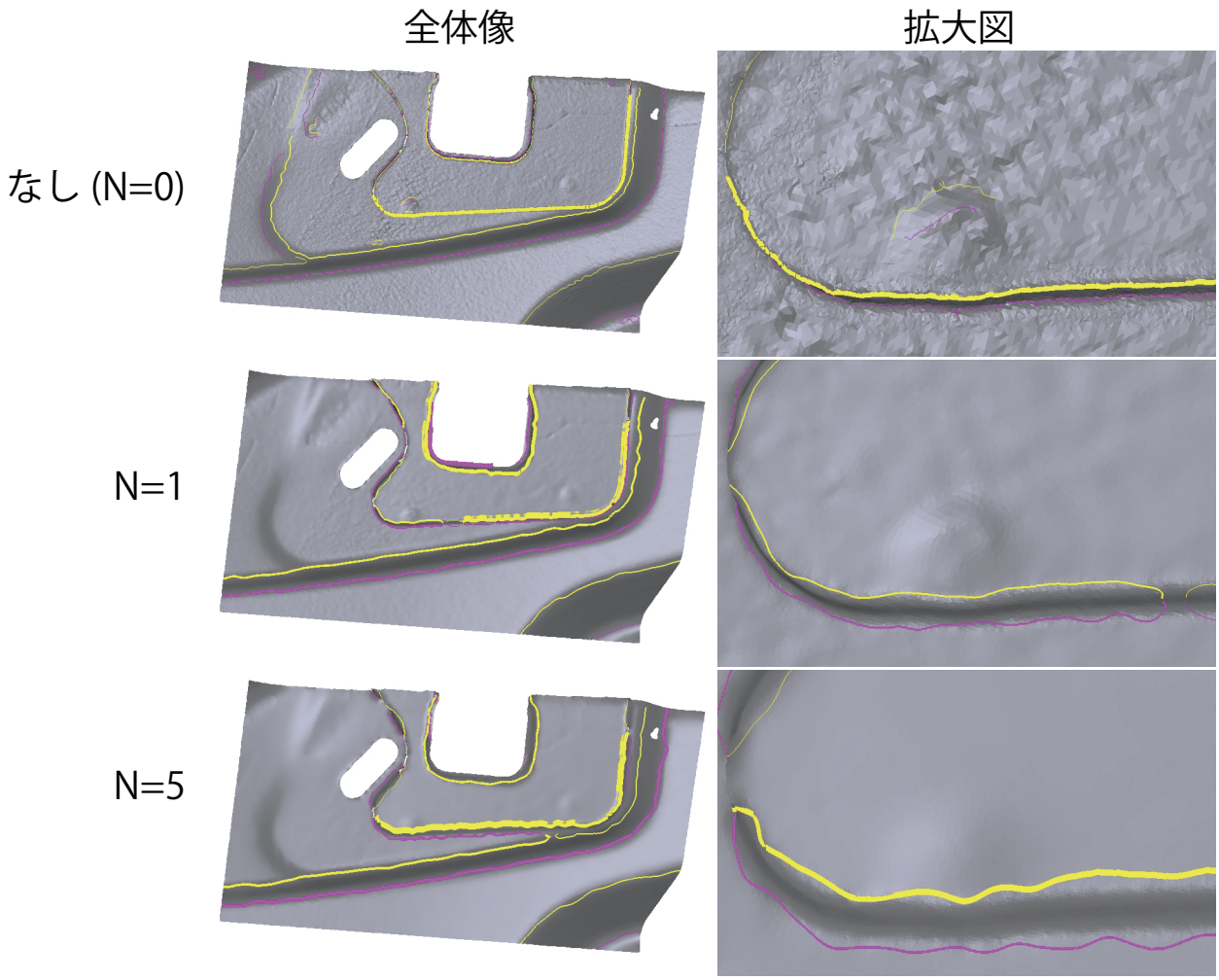


図 4.17: Laplacian Flow によるスムージング結果 (サンプルデータ 1)

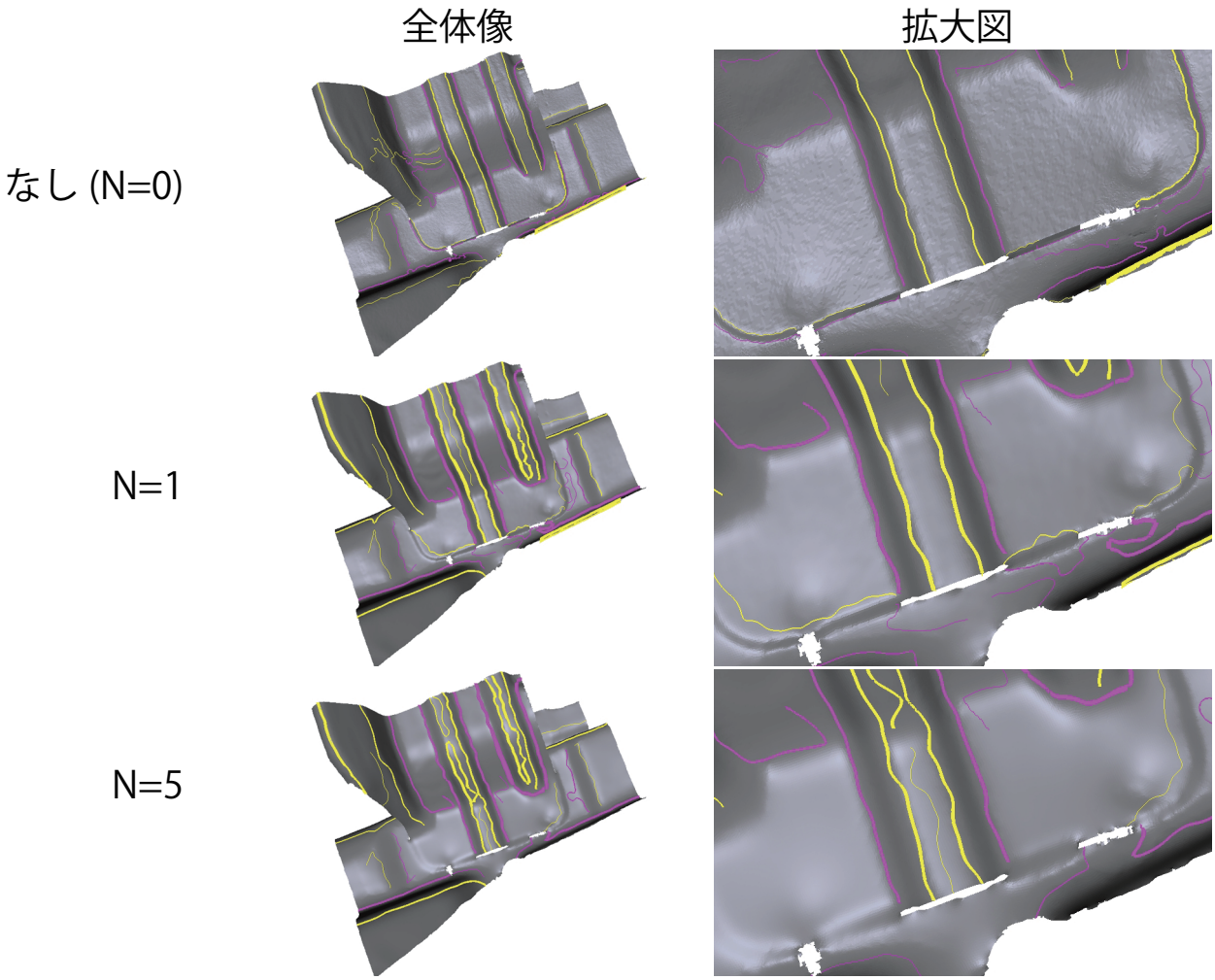


図 4.18: Laplacian Flow によるスムージング結果 (サンプルデータ 2)

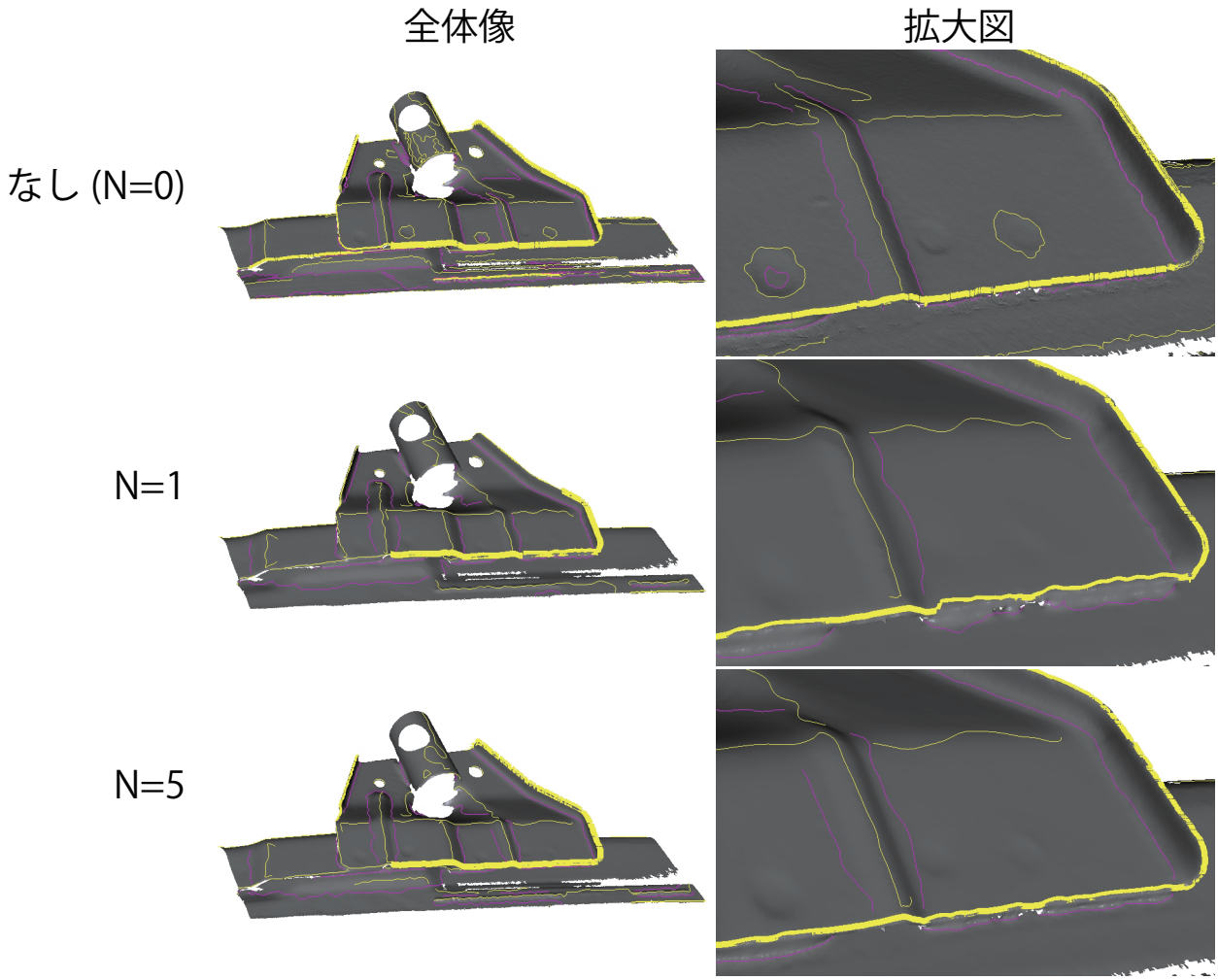


図 4.19: Laplacian Flow によるスムージング結果 (サンプルデータ 3)

表 4.5: $t = 0.005$ に於けるサンプルデータ 1,2,3 のスムージング N=1,3,5 表

サンプル No.	1			2			3		
N	0	1	5	0	1	5	0	1	5
time[sec]	55	45	44	46	38	38	109	78	79
L1[mm]	0.004	0.003	0.002	0.004	0.002	0.002	0.005	0.002	0.002
L2[mm]	0.006	0.004	0.003	0.005	0.003	0.003	0.008	0.003	0.003
Max[mm]	0.56	0.53	0.05	0.21	0.06	0.07	0.37	0.22	0.43
Ridge	1322	355	108	404	127	139	2571	315	340
Valley	1386	416	121	359	90	106	1989	315	353

この結果から, サンプルの 3 次元メッシュ自体は確かに滑らかになっており, 極値線もより滑らかかつ連続的に検出できるが, 極値線がカットに適する品質であるとは言えず, 幾分, カットを行いたいラインから離れてしまっている.

ここで,$N = 1$ 回スムージングを適用したメッシュを観察すると, 確かに $N = 0$ のものに比べて滑らかになっている. ここで,2.1.2 節および 4.3.3 節でみたように,t の値を大きくするのは, ノイズや測定誤差への対策であったから,N を増やす代わりに,t の値を減らす事を考える. 以下,$N = 1$ に固定した上で,t を $t = 0.005, t = 0.001, t = 0.0005, t = 0.0001$ と変化させ計算した様子を示す.

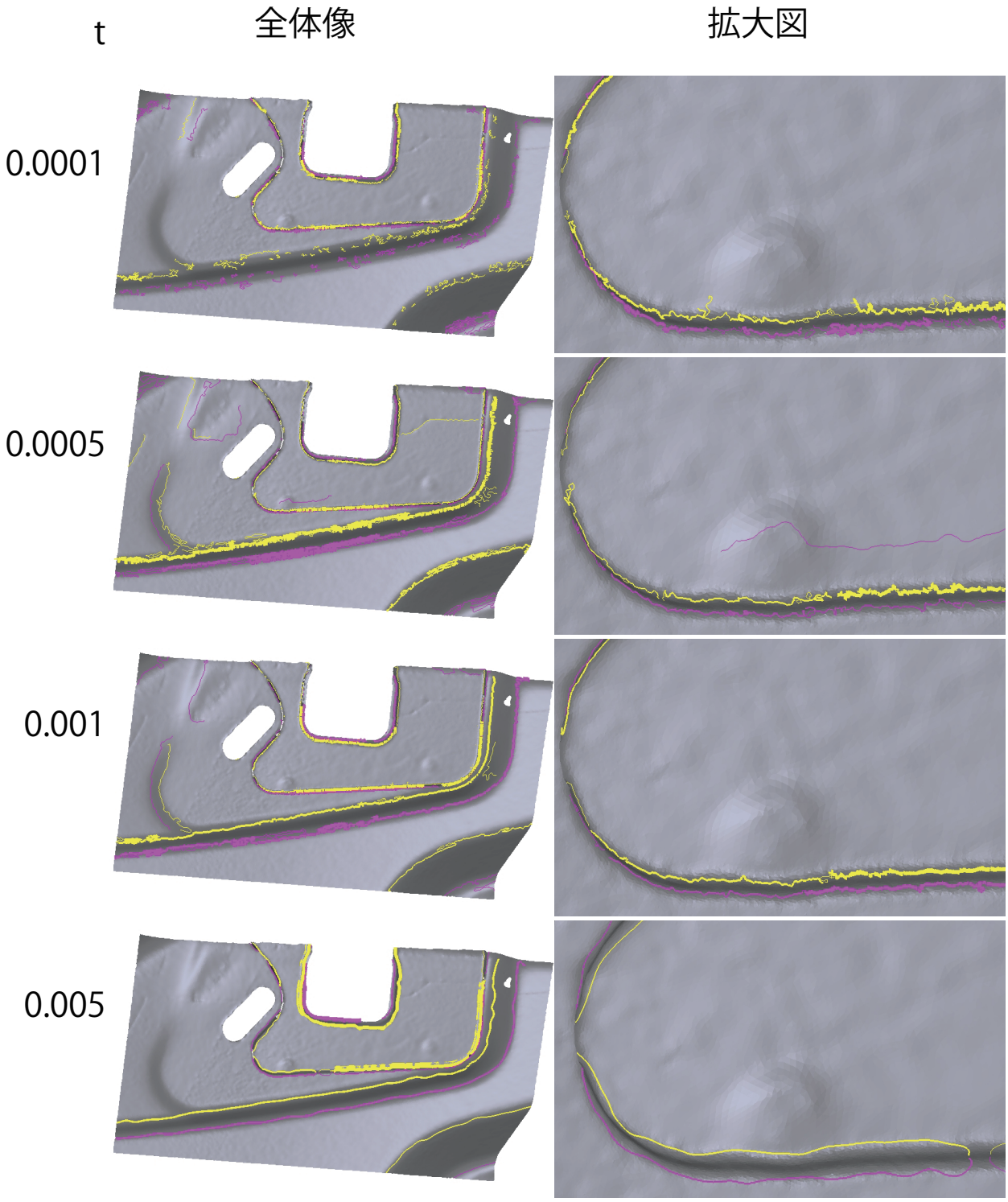


図 4.20: サンプルデータ 1 の $N = 1$ スムージングへの t の変化

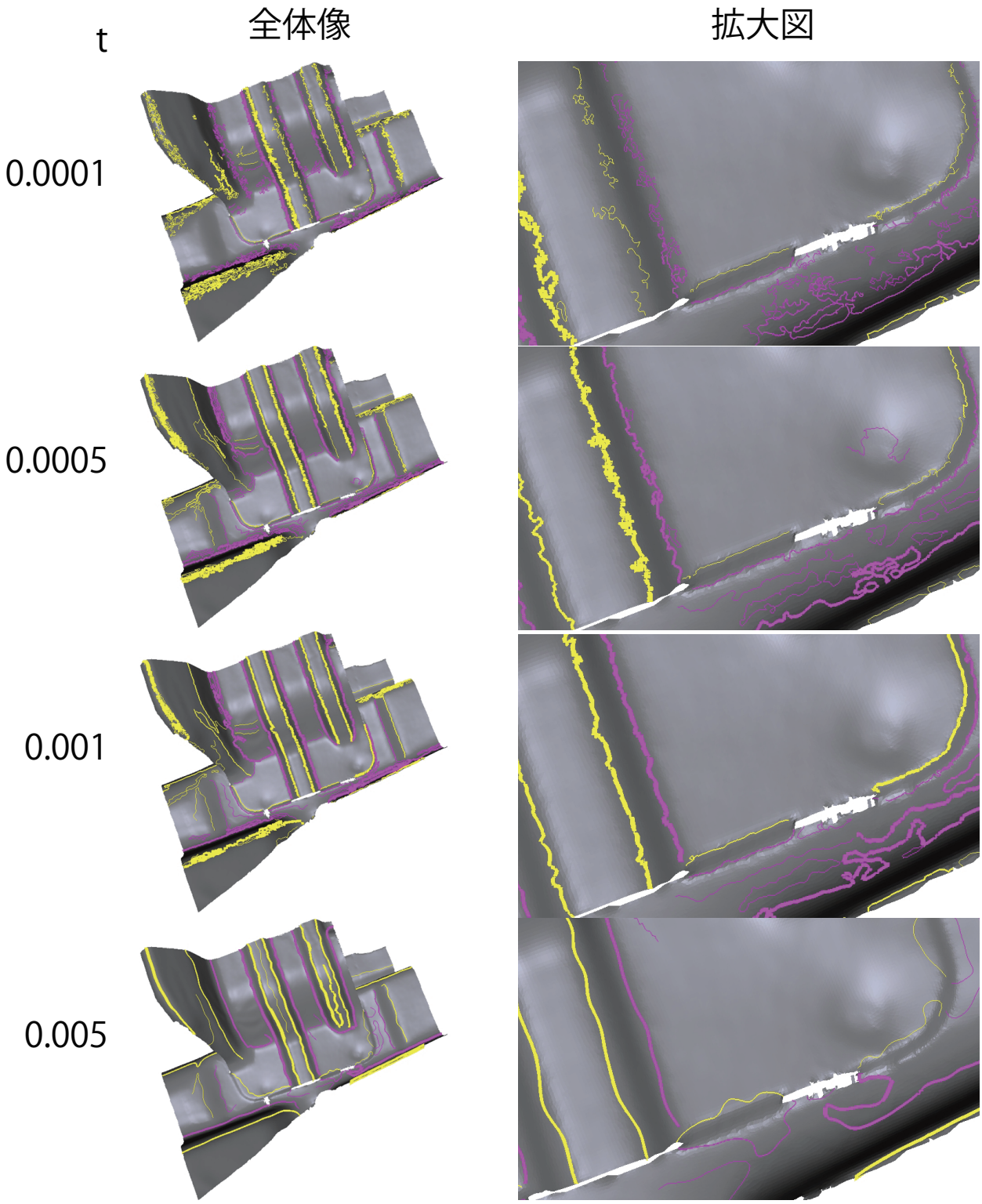


図 4.21: サンプルデータ 2 の $N = 1$ スムージングへの t の変化

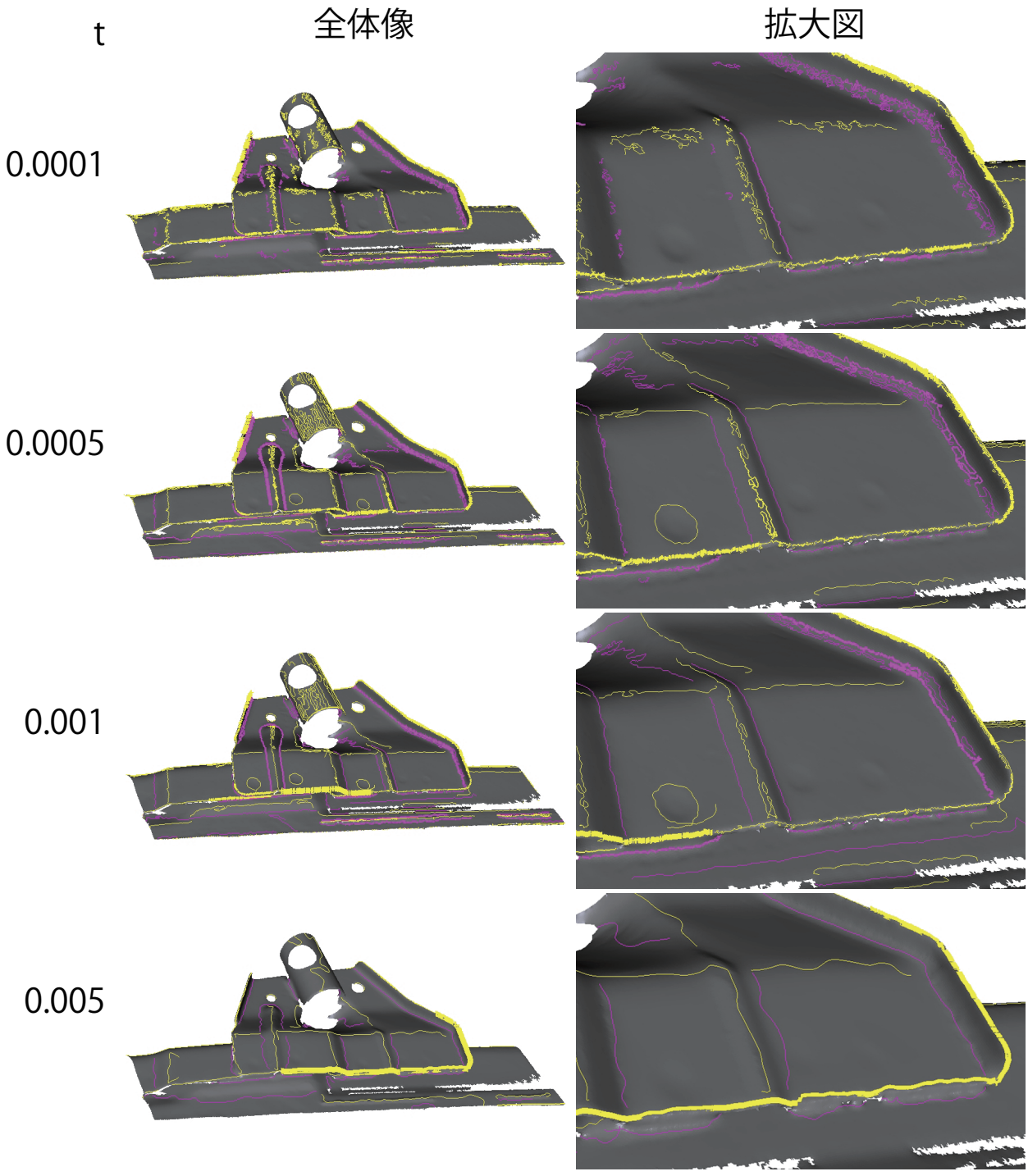


図 4.22: サンプルデータ 3 の $N = 1$ スムージングへの t の変化

表 4.6: サンプルデータ 1 の $N = 1$ スムージングへの t の変化

t	0.0001	0.0005	0.001	0.005
time[sec]	126	67	58	55
L1[mm]	0.001	0.001	0.001	0.004
L2[mm]	0.002	0.002	0.002	0.006
Max[mm]	0.51	0.52	0.49	0.56
Ridge	11766	2264	1404	1322
Valley	12030	2206	1396	1386

表 4.7: サンプルデータ 2 の $N = 1$ スムージングへの t の変化

t	0.0001	0.0005	0.001	0.005
time[sec]	109	60	50	46
L1[mm]	0.001	0.001	0.001	0.004
L2[mm]	0.002	0.002	0.002	0.005
Max[mm]	0.21	0.21	0.21	0.21
Ridge	7798	1323	680	404
Valley	7518	1076	556	359

表 4.8: サンプルデータ 3 の $N = 1$ スムージングへの t の変化

t	0.0001	0.0005	0.001	0.005
time[sec]	252	139	114	109
L1[mm]	0.001	0.001	0.001	0.005
L2[mm]	0.004	0.004	0.004	0.008
Max[mm]	0.4	0.42	0.36	0.37
Ridge	20324	4387	2862	2571
Valley	14249	2772	1703	1989

このように, スムージングをかけると曲率極値線は滑らかな曲線状になるが, 結果として必要な溶接端から外れてしまうことも多い. t の値を N を増やす代わりに小さくしていけば小さくしていくほど曲線状に出現する特徴線が溶接端部から外れる傾向を修正することもできるが, 品質が良いとは決して言えない. また, スムージングを適用してから極値線を計算すると,$N = 1$ 回適用することで, 計算時間がどのデータでも 2 割程度少なくできる事がわかった.$N = 1$ から $N = 5$ に増やしたときの, 計算時間は有意な変化は見られなかった.

ノイズに関しては確かに少なくなる傾向があるが,3.3.3 節で述べた θ の操作ほどの変化・改善は見られなった. 計算時間は 2 割ほど削減できるが, 極値線の品質が犠牲になるため, もしメッシュスムーズを行いたいのであれば, 極値線を導出し, カットが終わってから行った方がよいことがわかった.

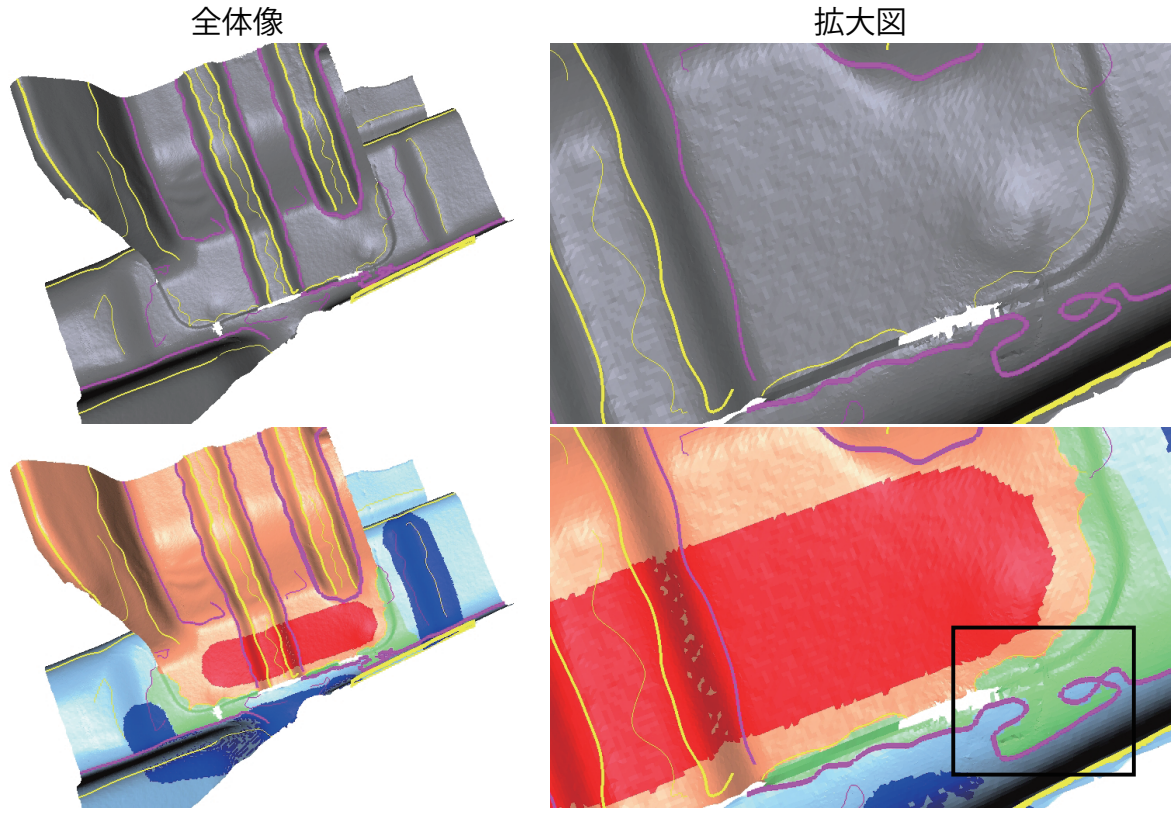


図 4.23: サンプルデータ 2 の特徴線に沿ったカット

4.4　ソースとシンク指定の工夫

ここでは, どのような場所をソースとシンクとして指定すると, できるだけ少ない手間でカットが行えるかを考える. 此処で生じる手間は, **どこを指定するか**と**どのくらい指定するか**すなわち, **場所と面積**である.

まず, そもそも曲率極値線に沿ってカットがきちんと行われているか確認し, その後カットの場所の工夫について考える.

極値線で切れているか　極値線で切れている事を確認する為に, 特徴的な極値線を持つデータを使ってカットを試してみたい. 図 3.5 で用いたサンプルデータ 2, スムーズ無しのうち,$t = 0.05$ のデータでカットを試みる.

図 4.23 の右下に表示したカット後・拡大図の枠内の通り, 確かに主要な特徴線に沿って色分けが成されている. すなわち, **もし十分な品質の特徴線を検出する事ができ, さらに適切な指定を行えば, 十分な品質のカットが行える.**

4.4.1　ユーザ指定の最適化

4.3.1 節で時間の評価について論じた際に, 極値線の計算と, ユーザのターミナル・ノード指定が 2 大ボトルネックであることを見たが, ここではその後者の改善のために, 本節では,3.3.2 節で定めたような「指定ルール」をより意味のある制約条件として定義しなおす事を目指す. そのためにまず, ユーザはどのように, ターミナル・ノード位置を指定すればよいかという問

題を考え, ユーザがメッシュを指定する際の指針を探る.

スポット溶接のパターン

指針を導入するため, 前提となりうる, スポット溶接のパターンを整理する.2 枚の金属をスポット溶接した現物のスキャンデータには通底するパターンがある. 現物は図 4.24a のように, 板 2 枚を重ねたものであり, 従って適切にスキャンが成されれば, 図 4.24b に示したように, メッシュは上側板金 (S) 部分, 下側板金 (T) 部分, 溶接端 (B) 部分の 3 領域から成る. また, ユーザはメッシュを目視することで 3 領域を把握できる.

トレードオフパラメータ t の値を 3.2.1 節で議論したように適切に定めれば, 図 4.24c に示したように, 分割したい上側板金 (S)・下側板金 (T)・溶接端部 (B) はそれぞれ曲率極値線とメッシュの端線 (e) でほぼ囲まれているはずである. なぜならメッシュは溶接端 (B) 部でのみ接合されており, その溶接端部の上端・下端には特徴線が出るからである.

特徴エリア

図 4.25a のような実際のスキャンデータでも, 図 4.25b に示すように, 上側板金 (S) 部分, 下側板金 (T) 部分に分かれているのが見て取れる.

いま, 例えば下側板金 (T) 部分を見たとき, その中にいくつかの特徴線が表れている. このとき, 例えば図 4.25c のようにユーザはその S,T,B 部分が, 特徴線によっていくつかの領域に分断されているように認識する事ができる. そういった認識を持てるのは, その分断された領域の周りを, 曲率極値線や, メッシュの端線がほぼ囲んでいる場合であろう. そこで, そのような**曲率極値線とメッシュの端線 (*e*) でほぼ囲まれた 領域を特徴エリアと呼ぶ**ことにする. 例えば図 4.25c では, 上側板金 (S) 部品が S1,S2 の 2 つ, 下側板金 (T) 部分が T1〜T5 の 5 つの特徴エリアに分割されているようにユーザは**感じる**だろう.

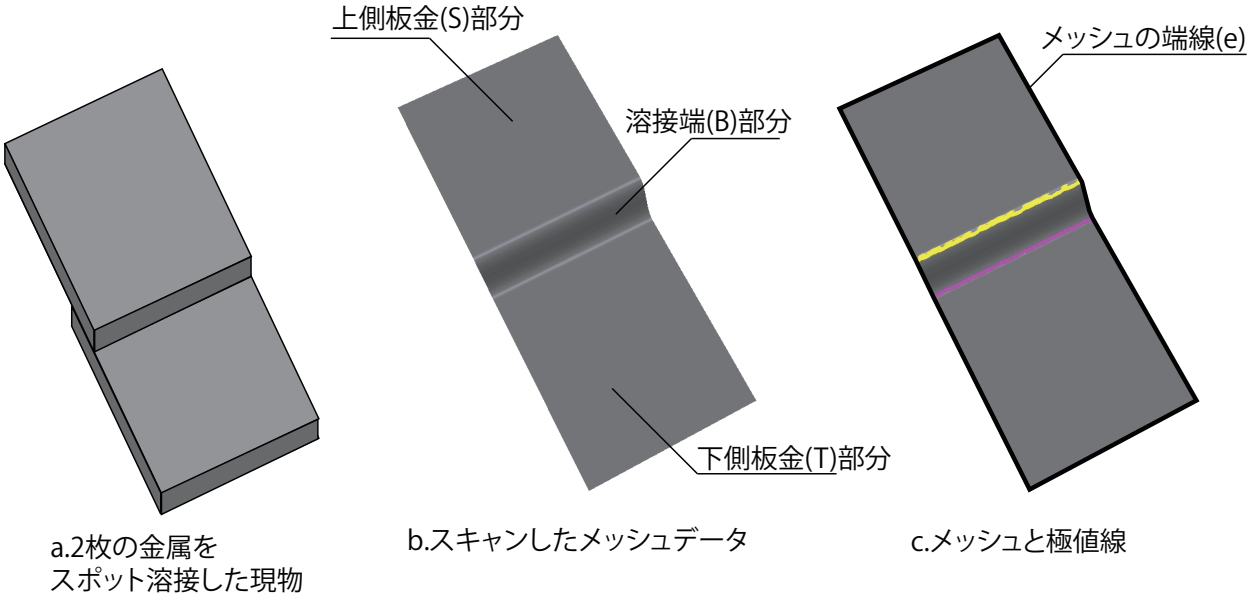
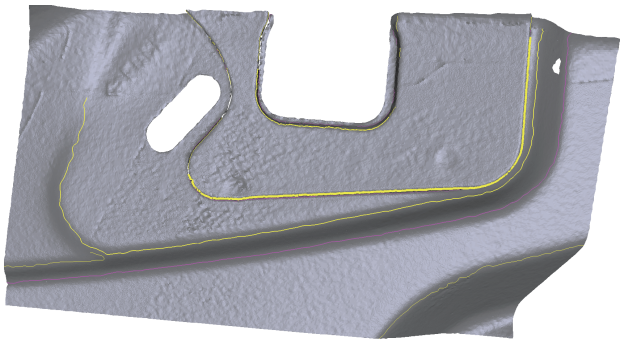
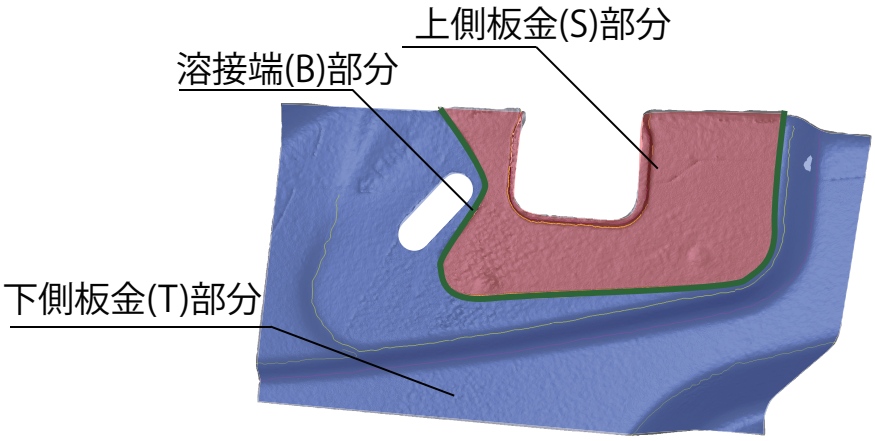


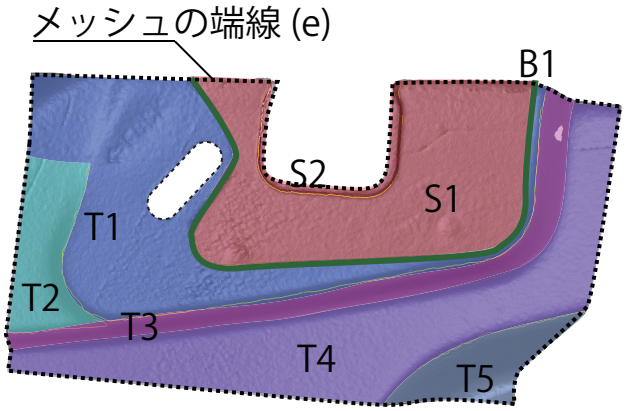
図 4.24: 2 枚のスポット溶接データの構造



a. 特徴線を計算したスキャンデータ



b.3 領域の認識



c. 特徴線による特徴エリア

図 4.25: サンプルデータの極値線とエリア

特徴エリアの性質　このような特徴エリアは, 実際に計算機によってグラフカットを行う前に, 目の前のグラフのどの部分をソースに, どの部分をシンクに指定すれば目指すカットされるかを**予測する**為に導入する. 詳しい定義は 4.4.1 節で行うが, ユーザが簡単に領域を認識でき, かつ, その領域がカットの結果と何らかの関わりを持つような領域として定義する必要が

あるだろう.

特徴線は, その部分でカットされやすいようにコストを設定しており, メッシュの端線 (e) というのはコストが 0 の (あるいは, もともと切れている) 部分だから, そういう特徴線やメッシュの端線で囲まれた領域は, **ちょうどその部分だけを塊として切り出されやすい領域**であるといえる. そもそも分解の目標としていた上側板金部分や, 下側板金部分も, 溶接端部に出る特徴線と, メッシュの端線で囲まれた大きな特徴エリアであり, それがその部分だけで切り出されやすい塊だったからこそ, これまで見てきたように適切にトレードオフパラメータ t や閾値 θ の設定, ターミナルノードの指定をすると, 綺麗なカットを行う事ができたといえる.

また, 特徴エリアはその部分だけで切り出されやすい塊であるが, それにはその特徴エリアを囲む特徴線の強さに応じて, 何段階かのレベル, すなわち, 切り出されやすさに分かれる事が考えられる. 閾値 θ が低ければ, 特徴エリアだらけになってしまい, そもそも特徴エリアを認識する事によってカットの結果を予測する事が不可能になってしまうだろう. そこで, 直観的把握のために, 閾値 θ を調整し, 特徴エリアのレベル差を無視することにする.

さらに, 特徴エリアは, レベル (切り出されやすさ) だけではなく, いくつかの大きさをもつだろう. 図 4.25 の三次元メッシュは, そのメッシュ全体がその大部分が端線で囲まれた特徴エリアであるし, その内部にある上側板金部分, 下側板金部分, および溶接端部分はどれも特徴エリアである (そのように特徴線が出てなければ, 極値線の品質に問題がある). そして, その特徴エリアも, 図 4.25c にみるように, 例えば上側板金部分には S1,S2 の 2 特徴エリアが, 下側板金部分には T1〜T5 の 5 特徴エリアが存在する.

このとき, 実用的には, 特徴エリア成立の条件を定義し, その成立条件を満たす中でのこれ以上特徴エリアに分割できない小領域のみを特徴エリアと呼ぶことにすると便利である.

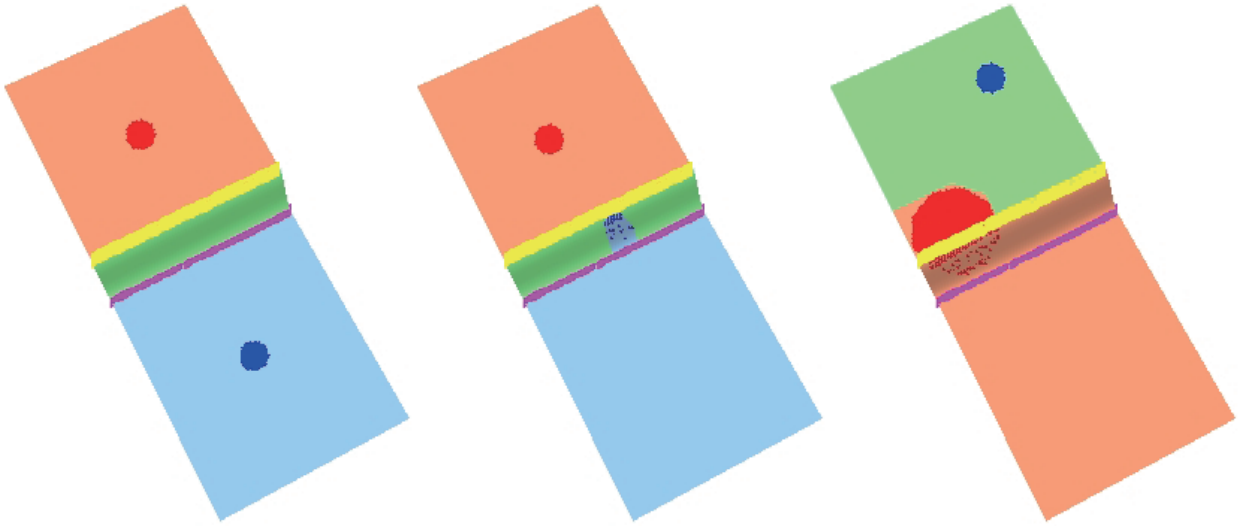
そういった特徴エリアは, ユーザがターミナルノード指定を行う, 指定の一単位となりうる特徴エリアといえるからである.

これらの知見から, 指定ルール (1):同じ側に分類したい, グラフのある部分に,ST を同時に指定してはならない. を修正すると,

指定ルール (1′)**:同一特徴エリア内に, 異なった指定をしてはならない**

というルールを定める事ができる. ある塊として切り出される, これ以上分割できない領域内を,S/T を同時指定すると, 結果が予測不可能になるからである.

図 4.26 に a.(S) 内をソース指定, (T) 内をシンク指定したもの, b.(S) 内をソース指定, (B) 内をシンク指定したもの, c.(S) 内にソースとシンク指定したものを示す. この時, 図 4.26c はルール (1') に違反し, 確かによい結果が出ているとは言えない.



a.(S) 部分と (T) 部分を指定　　b.(S) 部分と (B) 部分を指定　　c.(S) 部分に両方を指定

図 4.26: ユーザ指定条件の 3 パターン

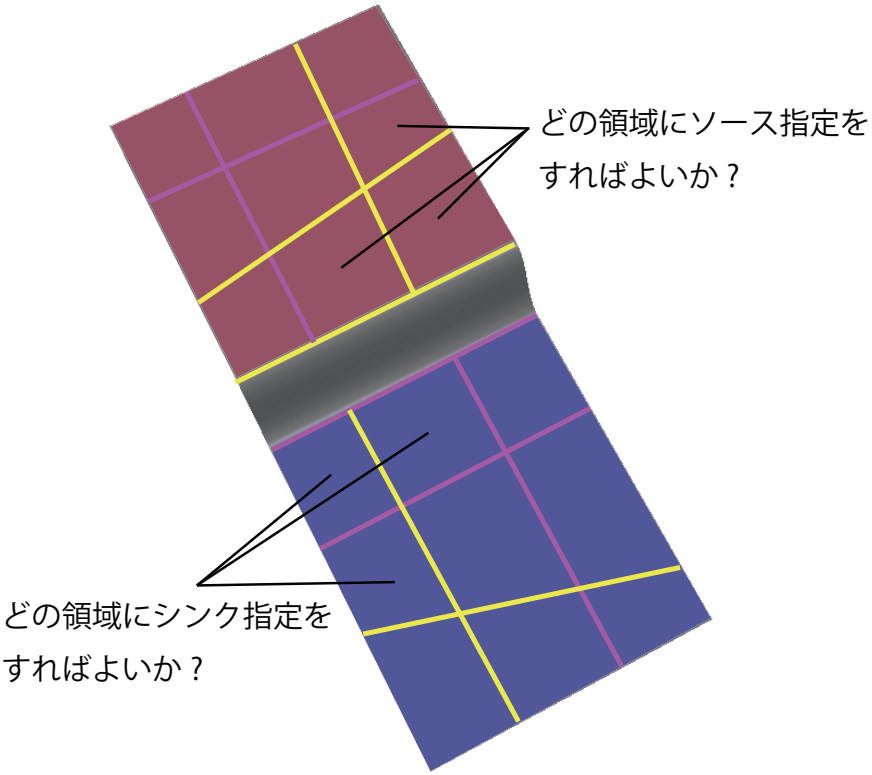


図 4.27: 特徴エリア導入の目標

導入の目標　特徴エリアを導入するのは, ターミナルノード指定問題を, 図 4.27 に指定したように, **上側板金 (S) 部分を構成する, 特徴エリアのうち, どの特徴エリアをソースに指定すべきか. また, 下側板金 (T) 部分を構成する特徴エリアのうち, どの特徴エリアをシンクに指定すべきか**, という領域選択問題へ書き換えるためである.

　このように書き換える事によって, 事実上無限に存在 (ソース側ノード数を N_{VS}, シンク側ノード数を N_{VT} とすると,$(2^{N_{VS}} - 1) \times (2^{N_{VT}} - 1)$ 選択肢存在) したターミナルノードのありうる組み合わせの数が, 有限の組み合わせへと置き換えられる.

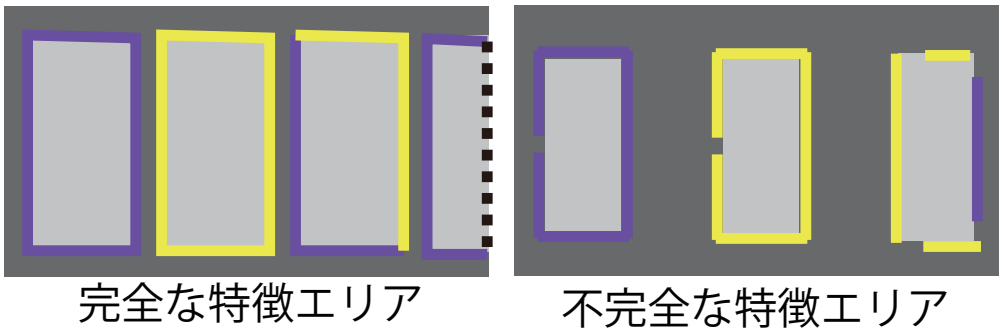


図 4.28: 特徴エリアの例

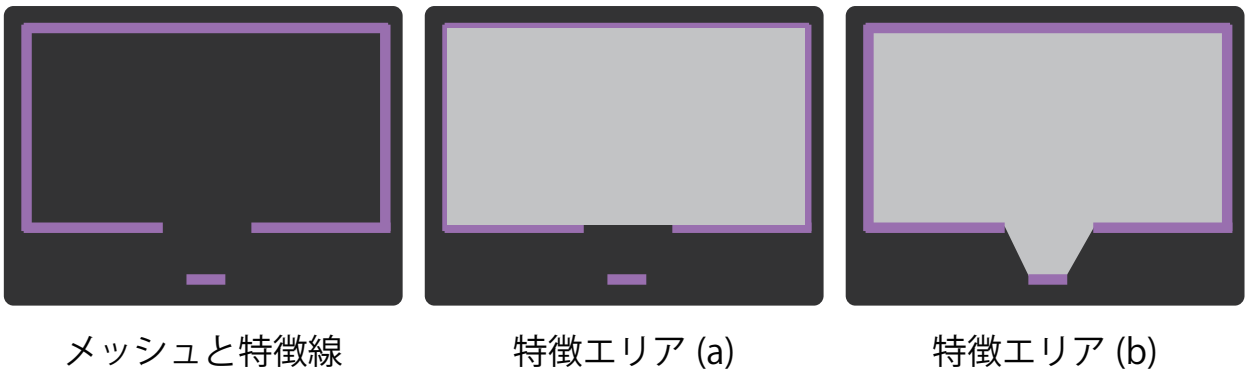


図 4.29: エリア分け問題

特徴エリアの定義とカット

特徴エリアの定義　特徴エリアは, 一言で言うと, 特徴線と, メッシュの端線でほぼ囲まれている領域であった. ここまでは「ほぼ」という言葉を曖昧なまま進めてきており, 何を特徴エリアとし, 何を特徴エリアとしないかはユーザの感覚に依存していたが, 本章でもう少し正確な定義を行う. そのような特徴エリアには, 図 4.28 に示したように, 山谷線およびメッシュの端線で完全に囲まれた**完全な特徴エリア**と, **不完全な特徴エリア**が存在する.

不完全な特徴エリアとは, エリアの周りに山谷線とメッシュの端線以外の部分, すなわち, メッシュ内部を切り取る線を含むようなエリアである.

特徴線の長さ L_l, 端線部の長さ L_e, メッシュ内部を切り取る線 (n と呼ぶ) の長さ L_n のように置く. 不完全な特徴エリアと, 完全な特徴エリアの差は, このメッシュ内部を切り取る線の有無のみだから, 次のような問題を考え,L_l, L_e, L_n の関係を定式化したい.

図 4.29 の左図のようなメッシュ上の特徴線があったとき, ここに生じる特徴エリアは, 特徴エリア (a) のようになるのだろうか, それとも特徴エリア (b) のようになるのだろうか？

ここで,(a), (b) の差異に着目すると,
図 4.30 の台形領域に示したように,
(a):直線 ab
(b):直線 $adcb$
が (a)(b) の差分となる.

このとき, グラフカットの計算時の事を考えると, メッシュ内部を切り取る線 (n) の長さ L_n に, 特徴線部分やメッシュ端部より大きなコスト設定を行っているから,

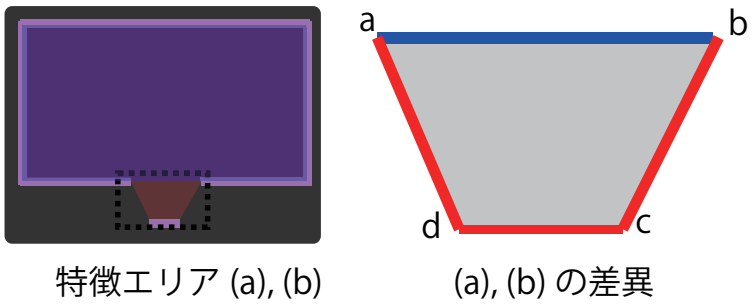


図 4.30: 台形領域

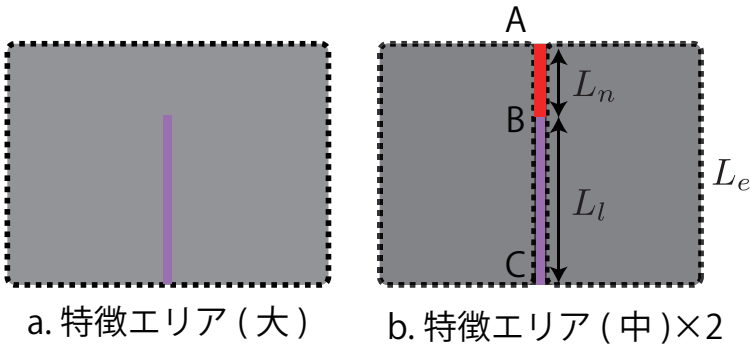


図 4.31: エリア分け問題 2

(n) を含む不完全特徴エリアを作る時,
$L_n - L_l \to min$
となるように, 特徴線およびメッシュ端部 (e) を繋ぐ.
また,L_l が無いとき, 最短距離で繋ぐように, 特徴エリアを定義するとユーザが特徴エリアを識別する助けとなりそうである.

次に, 以下のような問題を考える.

すなわち, 図 4.31a のような, 中に特徴線が走る特徴エリア (大) があったとき, 図 4.31b のように, 特徴エリア (中) のもの 2 つに分断すべきだろうか?という問題である.

$L_n \to 0$ の時は完全特徴エリアとして分断してよいから,$L_n > 0$ の時を考える. 図 4.31 の a と b を較べ, どちらがカットのコストが少ないかを考えれば良い.AB 間長さ L_n の切断に必要なコストを $k(L_n) > 0$ と書き, 極値線が通っている BC 間長さ L_l の切断に必要なコストを $k(L_l) < 0$[1]とすると,

$$k(L_n) + k(L_l) < 0 \tag{4.2}$$

であれば, 図 4.31a のような, 中に特徴線が走る特徴エリア (大) は, 図 4.31b のように, 特徴エリア (中) 二つに分割して良いといえる. このコスト $k(L_n), k(L_l)$ は, 前者は距離に, 後者は極値線の強さに依存する. ここで後者は閾値 θ によって, 弱い極値線は考慮の対象から外れているから, コスト倍率 $\mathcal{A}$ が距離の比率を決める大きな要因となるので, 式 4.2 を距離を用いた表現に書き換えて,

$$(1) : \mathcal{A} \times L_n > L_l \tag{4.3}$$

[1]極値線部分は斥力が働くのでコストは負数. 実装上の負数回避については 3.3.3 節で論じた.

となるような L_l および L_n が特徴エリア内部にあれば, 特徴エリアをその線で分割する. 3.3.3 節で, コスト倍率 $\mathcal{A}$ を大きくすると良い結果が出たのは, この $\mathcal{A}$ が大きくなることで, 極値線の通らないメッシュ間の結びつきが強くなり, **不完全な特徴エリアにおいて, ある特徴線の端点から, 別の特徴線の端点までの距離が, 最短になるようにエリアを**とる傾向が強くなるためであると考えられる.

特徴エリア分割手順　以上の議論から, 特徴エリア分割のための指針を示す.

適切にスポット溶接部品のスキャンを行い, 適切なパラメータで極値線導出を行った, 全ての 3 次元メッシュには,3 つの大きな特徴エリアが存在する. 上側板金 (S) 部分, 下側板金 (T) 部分, 溶接端 (B) 部分の 3 つである. この前提をもとにすると, こういった領域での特徴エリアへの分割は, 以下のような手順で行える.

手順 1. この 3 特徴エリアのうち,1 つを選ぶ.

手順 2. 選んだ特徴エリアの内部に極値線があった場合, その線の長さを L_l とすると, その線上の点から距離 $L_n < L_l/\mathcal{A}$ 以内に (1) メッシュの端部, あるいは (2) 極値線が存在するならば, そのうち L_n が一番距離が短くなるように特徴エリアを分割する.

手順 3. 手順 2 を繰り返し,3 エリア全てについて行う.

このように特徴エリア内をさらに特徴エリアに分割できる時かならず小さな特徴エリアへ分解する. 式 4.3 より,$\mathcal{A}$ を非常に大きくすると, 特徴エリアへの分割数を少なくする事ができる. しかし, トレードオフパラメータ t の指定や, 閾値 θ によっては溶接端 (B) 部の上下に出るはずの極値線がきちんと繋がっていない場合があり, その際に $\mathcal{A}$ を大きな値にしてしまうと, 特徴エリア分割の前提であった上側板金 (S) 部分, 下側板金 (T) 部分, 溶接端 (B) 部分の 3 つが, もはや特徴エリアと言えなくなってしまう可能性があるため, 徒に $\mathcal{A}$ 値を大きくすれば良いわけではない.

特徴エリアを全て選択　特徴エリアは, そこだけでひとかたまりとなって分割されやすいかたまりの事であったから, 以下のようなルールを定める事ができる.

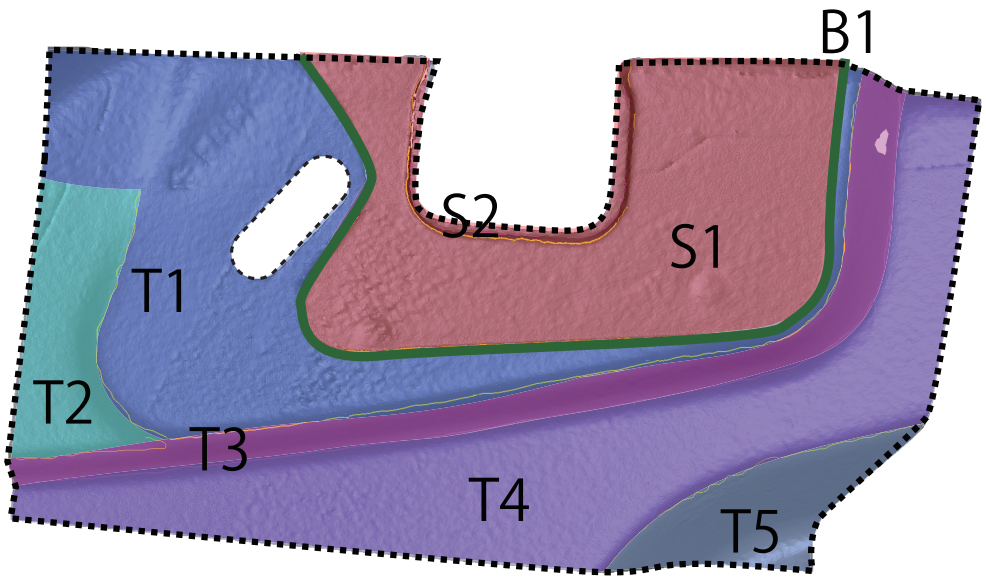
ルール (2)：同じソース (シンク) 側に分類したい特徴エリアを, すべて S(すべて T) となるようように指定する

実際に, このルール (2) を用いて, 図 4.32 のようにターミナルノードの指定と分解を行ったところ, 確かに上手く部品分解ができた.($\theta = 3\%, t = 0.005, \mathcal{A} = 10.0$)

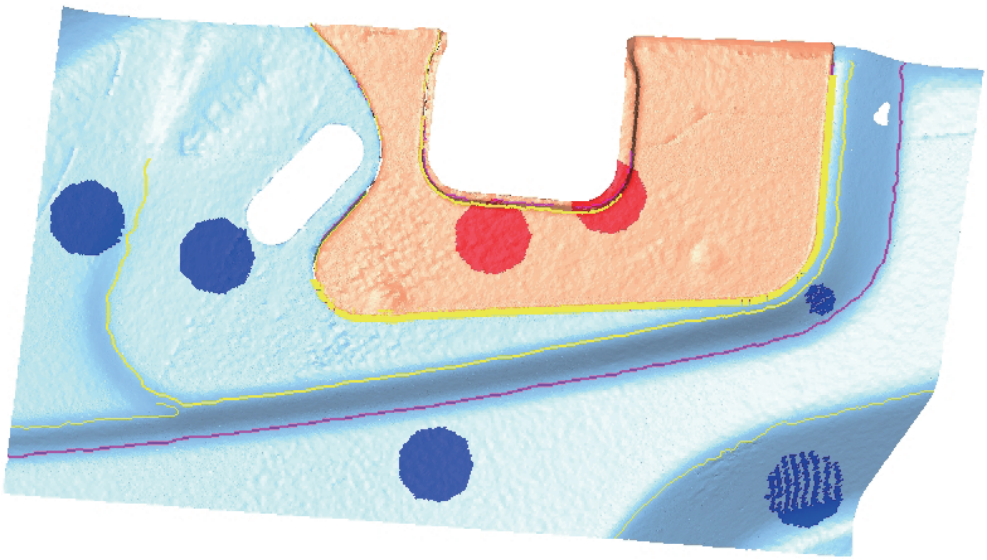
指定の簡略化　溶接端部 (B) を観察すると, **溶接端上下の極値線と接していない部分は, 接している部分が きちんと S/T 側に分割できれば, 同時についてくる**といえる. 従って, 指定ルール (2) は, 次のように書きなおす事ができる.

指定ルール (2′)**:同じソース側 (シンク側) に分類したい特徴エリアのうち, 溶接端 (B) に出た曲率極値線と接するエリアを, すべて S(すべて T) として指定する.**

実際にそのような指定を, サンプルデータ 2 で行った結果を図 4.33 に示す. サンプルデータ 2 のパラメータは $\theta = 1\%, t = 0.005, \mathcal{A} = 10.0$ とした. 図 4.33 のように,2 つの指定ルールを守る事で, 極めて少ない指定でもうまくカットができることがわかった.



特徴エリア分割



すべての特徴エリアを指定

図 4.32: 全ての特徴エリアをシンク/ソース指定

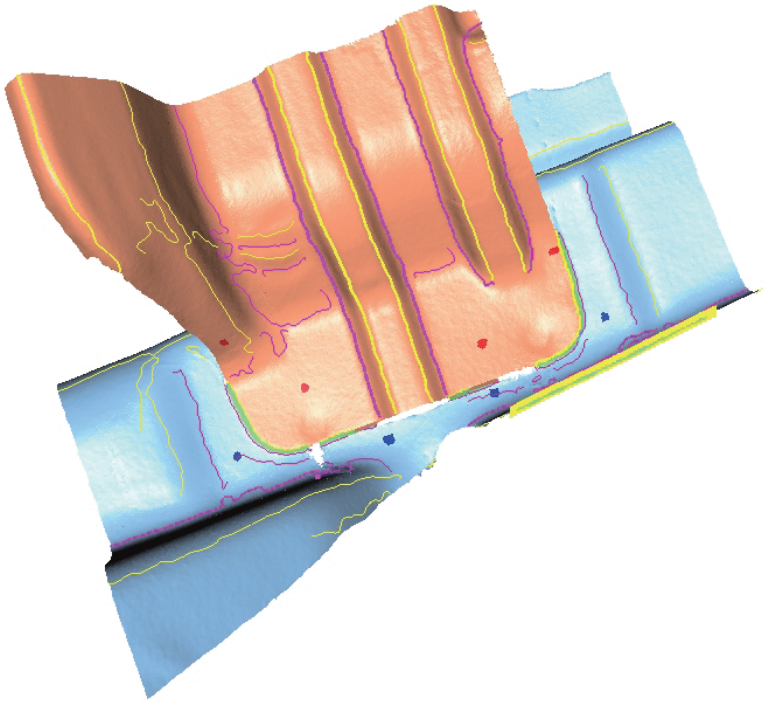


図 4.33: 溶接端 (B) の特徴線と接する特徴エリアのみを指定

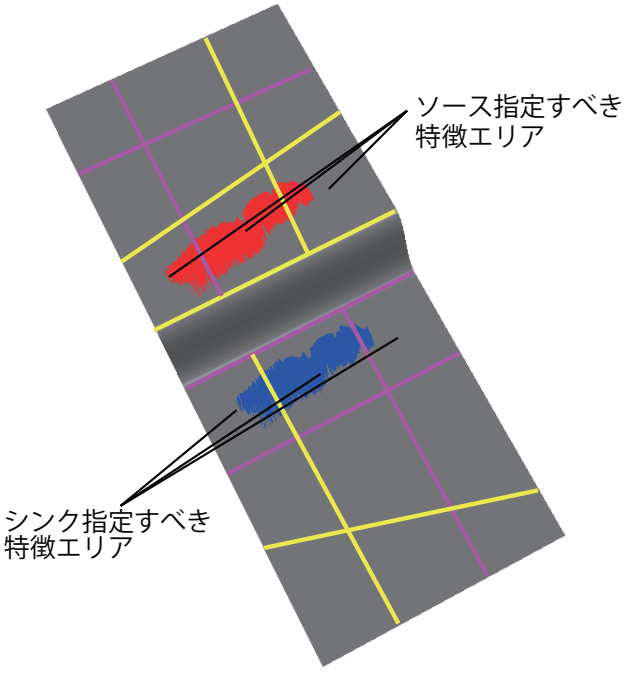


図 4.34: ルール (2') を満たすように指定した, 溶接端部特徴線と接する特徴エリア

以上,2 つの指定ルールを纏めると

指定ルール (1′)**:同一特徴エリア内に, 異なった指定をしてはならない**

指定ルール (2′)**:同じ側に分類したい特徴エリアのうち, 溶接端に出た曲率極値線と接するエリアを, 全て含むように指定する.**

この時,$\mathcal{A}$ を増加させると, メッシュの端線 (e) で切れにくくなるため, 特徴エリア数は減少する傾向にあり, それゆえ一つ一つの特徴エリア面積は増大する傾向にある. また,θ を増やすと考慮対象となる極値線数が減るため, エリア数は減少し, それゆえエリア面積は増大する傾向にあるといえる.

特徴エリア内での変化　特徴エリア内のターミナルノード指定位置を順次変えながら, 特徴エリア内でのふるまいを探った結果を図 4.35 に示す. ユーザ指定面積を極小にして, 変化を探った.$\theta = 1\%, t = 0.005, \mathcal{A} = 10.0$ でサンプルデータ 1 を用いた. サンプルデータ 1 は, ルール (2) を満たす特徴エリアが上側板金 (S), 下側板金 (T) ともに 1 つづつの為, このような試行に適している.

図 4.35: ソース指定位置の変更と結果

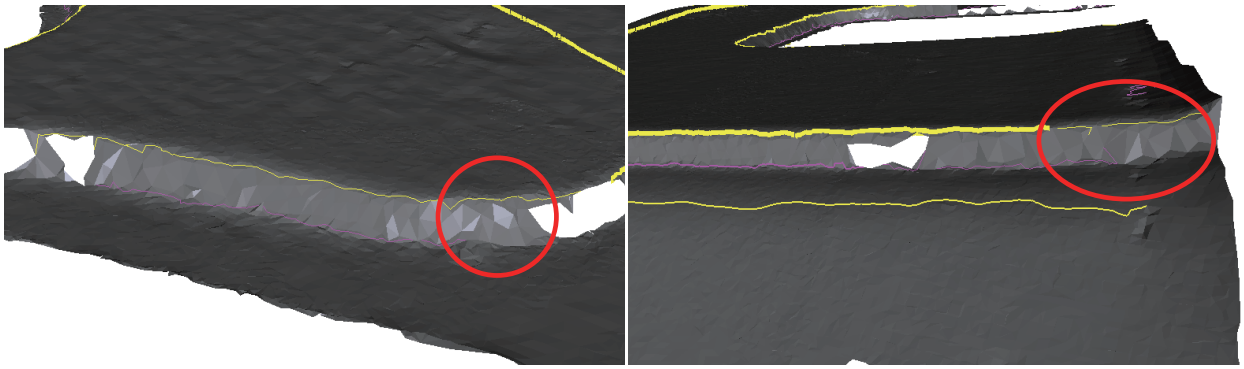


図 4.36: 溶接端部における極値線の欠け

いくつかは特徴エリア内に, それぞれ極少数のターミナルノード指定にも関わらずうまくカットが行えたが, 図 4.36 に拡大図を示したように, 極値線の品質に問題があるため全てのカットで成功とは言えなかった. この部分に関しても, 指定量を増やす事で対応はでき, 事実上は問題ない品質でのカットが行える.

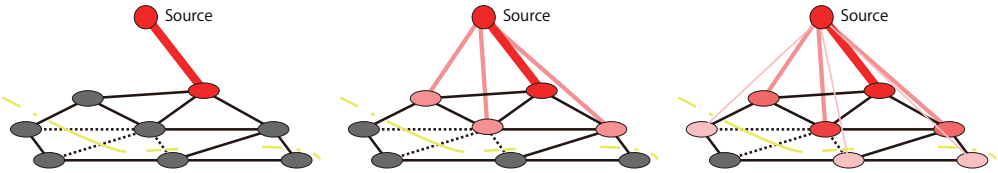


図 4.37: ファジー領域の概要

4.4.2 ファジー領域

この時, できるだけ人の手による介入が少なくなるようにしたい. 具体的には, より少ない指定面積, より少ない指定回数で, 適切なカットが成される事が望ましい. それを実現する為に, ファジー領域の導入を考えた. 図 4.37 にあるように, 対話的にソースへ分類されるノードとして指定されたノードと繋がっていて, かつ, 極値線によって分断されていない近傍ノードに対し, ソースからのコストを設定する.

具体的には

1. ソースとしてユーザが指定したノードには, ソースからのコスト K' を与える.
2. あるノード P と繋がっており, かつ極値線によって分断されていない n_f 個のノード $Q_i(i = 1, 2, \cdots n_f)$ のソースからのコスト $K'_S(Q_i)$ を全て加算する.
3. P, Q_i 全てのノードに, ソースからのコストとしてコストの平均値 $\frac{\sum_i^{n_f} K'_S(Q_i)}{n_f}$ を分配する.
4. 1 に戻る

というプロセスによって, ゆるやかにソースからのコストを持つノードを増やす事を目指している.

これを N_f 回繰り返し, ユーザ指定された部分に近いノードを, ソースに分類されやすくする. ノード・ノード間に曲率極値線が走っていた場合, このような拡張を行わないことによって, 単なる指定面積の拡大とは違った効果が得られる.

実際に, サンプルデータ 1 に対して $\theta = 1\%, t = 0.005, \mathcal{A} = 10.0$, ファジー回数 $N_f = 0, 10, 60$ で試行した結果を 4.38 に示す. カットに要した時間は, $N_f = 0$ で 0.7[sec],$N_f = 10$ で 3.1[sec], $N_f = 30$ で 17.7[sec]$N_f = 60$ で 55.4[sec], $N_f = 100$ で 118.1[sec] だった.$N_f = 60$ 以降, 処理時間は伸びたが, 同じ結果で変わらなかった.

ファジー領域は一見うまくいきそうだったが, 繰り返しの数がある程度以上増えると, 狭間領域に分類されてしまう部分が大半になることがわかった. これは, 繰り返しの回数が増え, それによってソースやシンクから繋がるノードが増えてくると, 極値線の隙間などから, 指定領域が漏れ出し, **指定ルール** (1)**:同じ側に分類したい, グラフのある部分に, 異なった指定をしてはならない.** に抵触してしまうこと, 及び, 図 4.35 で示したように, 少しの漏れや, 少しの位置の差でも, 結果に大きな影響を及ぼすためであろう.

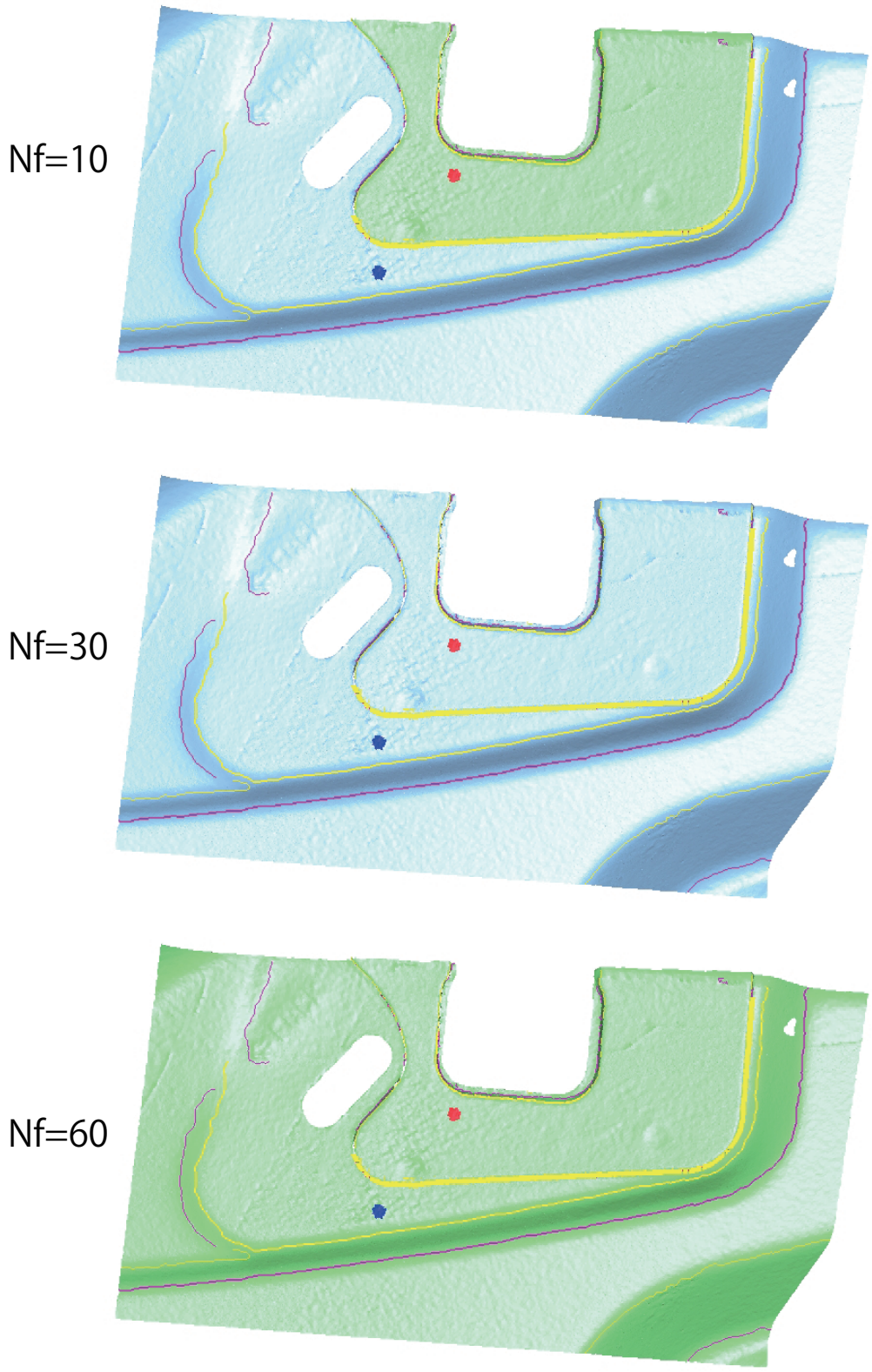


図 4.38: ファジー領域の適用

第 5 章

結論と展望

第 5 章

結論と展望

5.1 結論

本研究では, 現物データからスキャンした 3 次元点群データを陰関数近似し導出した, 曲率極値線を, グラフカットのコストとして適用し, ソースとシンクを対話的に指定する事で, スポット溶接された金属板を, カットの品質を維持しながら, 大幅に時間を短縮できるような手法を提案した.

また, 品質を維持したまま, 効率改善を行う本研究では, 主に 4 つの変数と, ユーザが守るべき 2 つのルールを導入したが, 時間と品質に対して, 図 5.1 のような関係がある事を見出した.

極値線の品質に関してトレードオフパラメータ t, スムージング回数 N, のトレードオフを解消しなくてはならない. また, この t, N には関係があり,N を増やすならば,t を減らさなくては極値線の品質の維持ができない. ノイズのない理想メッシュにおいては適切な t の値は, メッシュの単位面積あたりの点群の粗密に依存する事がわかった.

カットの品質に関して, ターミナルノードの指定ルール (1′), (2′) を導入すると, カットが上手くいく確率が上がり, また, 指定領域も少なくすむ. コスト倍率 $\mathcal{A}$ を大きくすると, 特徴エリアが大きくなる傾向があるため, 指定の手間が省ける傾向にあるが, 特徴線検出が十分でない場合, 品質が悪化する可能性もある. 一般的には, 極値線の品質が良ければ,$\mathcal{A}$ を増加させることは, カットの品質向上に寄与する.

閾値 θ は, 人の利便性および特徴線の品質とのトレードオフを持つ, 特殊な関係である.θ が 0 であれば, 多くのスキャンメッシュにおいて, 人は特徴エリアを見抜くことができず, 塗り分け時の労力が増える. だが, 大きくしすぎると今度はカットに必要な特徴線まで消去してしまい, 特徴線の品質が下がる.

また, 良い品質, すなわち, 溶接端部においてきちんと分解ができていることは, 本研究における必要条件であるから, これら品質のためのトレードオフが, 時間の制約条件となる. 特に計算機側の処理では t の値が全体の時間を大きく左右する事がわかった.

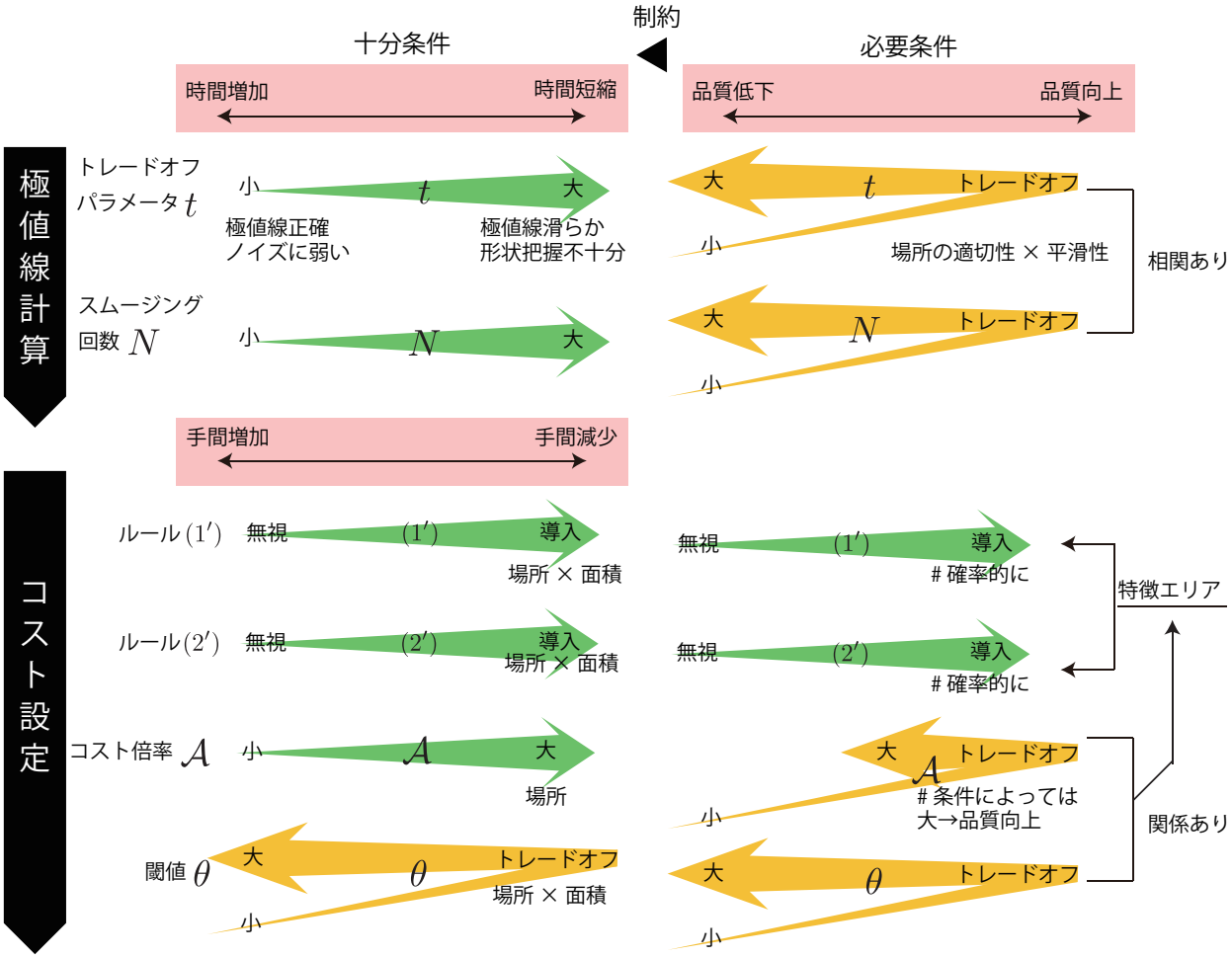


図 5.1: パラメータの変化と所要時間および品質の関係

現場の評価　本手法によりカットを行った結果を解析現場で評価して頂いたところ,CAE 用途として, きちんと分割が成されているとの評価を得た. 効率の改善は十数分から 3 分以下と一般的に改善がみられるといってよいが, 当初の目標通り品質を維持している事が確認された.

5.2 課題および展望

本研究によって, 従来は手動で行われていた 3 次元メッシュのスポット溶接部による分解が, 半自動的に分解できるようになったが, 課題も残る.

トレードオフパラメータ t の自動化 最適な t の値は, 今回大幅にいくつかの値をとって結果を確かめ, 次に適したものの近傍を細かくとって最適なものを選択した. 今回定めた値で他の現物からのスキャンメッシュに適用できるわけではないが, 結論に示したように, メッシュの粗密と適した t の値の組み合わせを順次計算していく事で, スキャナの解像度に応じて適した t の値が決まるような数値表, もしくは近似式を作る事で,t を試行錯誤することなしに, 機械的に決める事ができる可能性がある.

θ の自動決定 θ の値も自動決定できると便利である. 今回は $|k|_{max}$ の 5% 以下はカウントしない, などのように決めたが, 同じ光学式スキャナを用いるという前提を置けば, 平らな金属板をスキャンし, そのメッシュの曲率極値線を計算することで, メッシュが平らな時にも検出されてしまう, スキャナ特有の測定誤差に対して出てくる極値線の強さが求まるのではないだろうか. その値の前後に閾値 (この場合は k_{max} からのパーセンテージではなく, 絶対的な強さとする.) を置く事で, 同一スキャナでスキャンしたデータに対し, 測定ノイズに対してつくられた特徴線の影響を無視できる可能性がある.

スポット痕の検出 また, このような現物ベースの CAE シミュレーションでは, スポット痕に境界条件などを入れ, シミュレーションする事が多いため, スポット痕の検出の要請が強い. 曲率極値線では,t, θ の値によって, スポット痕を囲むように特徴線が出る事があった. スポット痕は丸く圧着し, 凹ませて作られた穴であるから, 図 5.2 のように, 内側に谷線, 外側に山線と, 二重になってスポット痕を囲むような条件がある. そのように囲まれたものを特殊な特徴エリアとして定義できれば, スポット痕の自動検出ができる可能性がある.

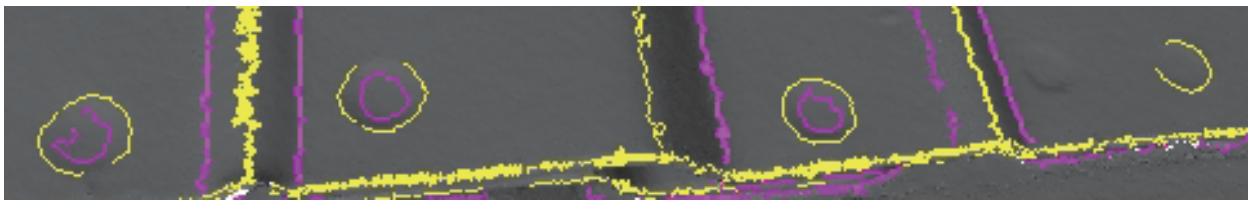


図 5.2: スポット痕と曲率極値線

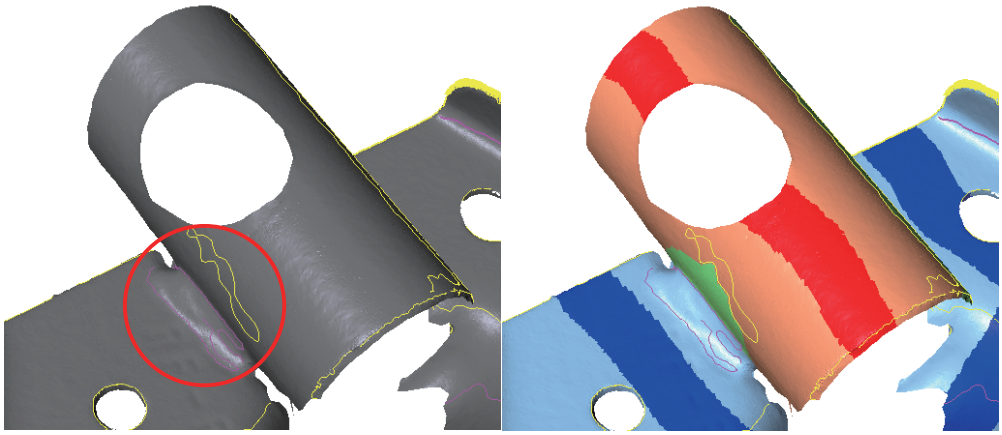


図 5.3: アーク溶接のカット

5.2.1 アーク溶接

本研究の溶接部による部品分解は, スポット溶接を前提としており, アーク溶接には対応していない. だが, 図 5.3 のように, アーク溶接でも, 特徴線の検出はできるため, 本手法の応用により. アーク溶接も上側板金部分, 下側板金部分, アーク溶接部分の 3 部分に分割できるようなカットが行えるのではないか. スポット溶接では, 溶接端は山線と谷線で囲まれていたが, アーク溶接は図 5.3 のように, 谷線と谷線で囲まれる.

5.2.2 複数回の分割

サンプルデータ 3 に関しては, これまでサンプルデータ 3 中央領域のスポット溶接部のカットについて見てきた. サンプルデータ 3 は,1 つのアーク溶接と,2 つのスポット溶接から成っており, 全部で 4 つの部品からなる. 本手法を適用する事により,4 部品を 1 つづつ取得するカットを 4 回行い, 最終的に統合する事で, 図 5.4 に示したように, アーク溶接による溶接端では課題が残るものの, スポット溶接で溶接されていた 2 溶接端ではよい品質でカットができた. 今後, アーク溶接のカットに対する改善が行われれば, アーク溶接・スポット溶接を複数含む部品の分解への応用が期待される.

5.2.3 ペイントから選択形式へ

4.4.1 節では, 特徴エリアの概念を定義し, その手順をみた. この定式化と, ルール (2′) を用いて, グラフをあらかじめ特徴エリアに分割して表示させる事ができれば, ユーザには単に複数の特徴エリアから, 極値線に接する特徴エリアを選択するだけでカットが行える可能性がある.

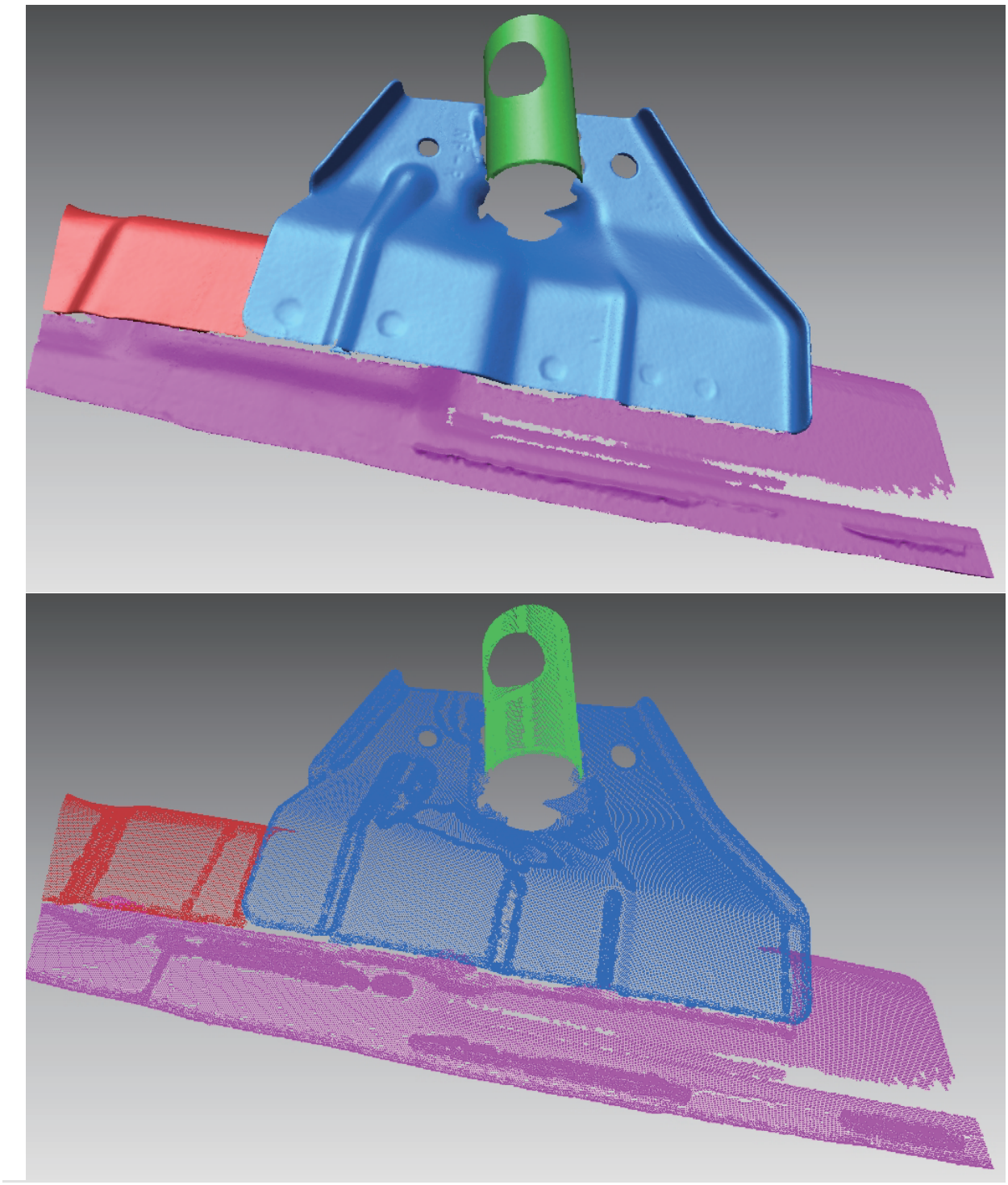


図 5.4: サンプルデータ 3 の 4 部品への分解

参考文献

[1] Y.Ohtake, A.Belyaev, H.-P.Seidel. Ridge-Valley Lines on Meshes via Implicit Surface Fitting. *ACM Trans. Graphics*, pages 609-612,2004

[2] Y.Ohtake, A.Belyaev and M.Alexa, Sparse Low-degree Implicit Surfaces with Applications to High Quality Rendering, Feature Extraction, and Smoothing, *Eurographics Symposium on Geometry Processing*, 2005

[3] M.Hosaka, Modeling of Curves and Surfaces in CAD/CAM (Computer Graphics: Systems and Applications), *Springer-Verlag*, 1992

[4] H.Era, M.Tsuchiya. Graph Theory series-mathematics and information, *Sangyo-Tosho*, 1996.

[5] Y. Boykov and M.P. Jolly, Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images,*In Proc. Eighth Int'l Conf. Computer Vision*, 2001.

[6] Mathieu Desbrun, Mark Meyer, Peter Schröder and Alan H. Barr, Implicit Fairing of Arbitrary Meshes using Diffusion and Curvature Flow *In ACM Siggraph '99 Proceedings*, 1999.

[7] C.Rother, V.Kolmogorov and A.Blake,“ GrabCut ”-Interactive Foreground Extraction using Iterated Graph Cuts *In ACM Transactions on Graphics (TOG)*,pages 309 - 314, 2004.

[8] 石川博, グラフカット, *CVIM 158 (26)* 情報処理学会研究報告,pages 193-204,2007.

謝辞

本研究を行うに当たり, ご指導頂いた東京大学工学部先端科学技術研究センター教授, 鈴木宏正教授に謝意を表します. 鈴木教授には, 研究テーマの設定, 方針や出てきた結果に対する洞察など, いくつもの有益なアドバイスを頂き思考の整理をして頂きました.

大竹豊講師には, 自ら開発した手法である曲率極値線についてご教示頂いたこと,3 次元メッシュから曲率極値線を生成するためのプログラムを頂いたこと, その他多くの示唆を頂いたこと, 感謝しております.

道川隆士助教には,「ペイントによる視点決定手法」のメッシュ上のペイントプログラムを頂いたこと, プログラミングを教えて頂いたこと, 果てなきバグフィクスに根気よくお付き合い頂いたこと, 感謝しております.

鈴木宏正研究室の皆様:特任教員の松崎幸一先生, 協力研究員の森本有紀さん, 秘書の辻口るみさん, 博士課程の Peter Salvi さん, 五十嵐悠紀さん, 長井超慧さん,M. Haitham Shammaa さん, 修士課程の Caiyun Yang さん, 上端純平さん, 中崎瞬さん, 佐藤桂太郎さん, 永井伸直さん, そして同期の大橋憲二, 塚本圭, 金箱春樹の 3 人に, 研究会や輪講のさなか, また研究室でのディスカッションにて多くの有益なアドバイスや洞察, 示唆を頂いたこと, アプリケーションの使い方や文献をご紹介頂いたことなど, 私にとってはじめての研究生活 1 年間を通し, 大きく助けられました. ありがとうございました.

また, お忙しい中, 査読をお引き受け頂いた, 東京大学工学部精密工学科, 太田順准教授から, 論文の構成, 言及範囲など, 論文の改善への貴重なご意見を頂きました. ありがとうございます.

最後になりましたが, 教養学部 2 年生でゼミナールを持って下さり, その後公私に渡り深く教え導いて下さいました, 工学教育推進機構教育プロジェクト室長, 吉田眞教授と,InterStates Consulting Group 代表取締役 永綱浩二先生に感謝致します. 研究室配属の前に研究の一端に触れさせて頂き, 私のものの見方, 考え方に大きな影響を与えて頂きました. 鈴木宏正教授のもとでの成長の礎となる, 思考力を鍛えて下さいました. ありがとうございました.