

測定点群を基にした採寸ソフトの開発 Development of Measurement Software based on Scanning Points

平岡忠志*
Tadashi Hiraoka

抄 録

人体下半身の3次元測定点群から、採寸情報（周囲長，丈）と体積を計算する採寸ソフトを開発した．本ソフトでは，まず，一般的な下半身形状の三角形メッシュを測定点群にフィッティングすることで，測定点群を近似した三角形メッシュを構成する．この手法では，点群の欠落部を自動的に穴埋めするという利点はあるが，つま先付近などの形状変化が大きい部分は，うまく近似されないという問題があることがわかった．次に，利用者が恥骨などの特徴点をマウスやキー入力で指示することで，弾性ストッキングを製造するための採寸情報とリンパ浮腫患者の治療経過を評価するための体積を算出する．三角形メッシュの位相構造を利用することで，これらの処理を正確に計算することができた．

1 はじめに

現在，リンパ浮腫患者用のオーダーメイド弾性ストッキングは，大部分を輸入品に頼っているため，高価で入手するまでに時間がかかる．また，採寸にメジャーを使うので精度が低く，着圧も経験と勘に頼っている．そこで，リンパ浮腫患者に安価で性能の良いオーダーメイド弾性ストッキングを素早く提供するための製造支援システムを開発する．これは，光切断法により測定点群を取得し，採寸情報（周囲長，丈）を計算し，その結果と多層弾性分布計測装置により測定された皮膚の硬さからストッキングの網目数を計算するものである．また，医師が治療経過を評価するために，浮腫部分の体積を計算する必要がある．本報では，これらのうち，測定点群から採寸情報と体積を計算する方法について検討する．

2 方法

測定点群間には表面形状の情報がないため，測定点群から採寸情報と体積を直接計算することは難しい．そこで，まず測定点群を近似する三角形メッシュを構成し，次に採寸情報と体積を計算する（図1）．

2・1 三角形メッシュの構築方法

測定点群から三角形メッシュを構築する方法は，ポロノイ図を利用する補間による方法¹⁾，ポアソン方程式などを利用する近似による方法²⁾，一般モデ

ルを測定点群にフィッティングする方法^{3,4)}などがある．補間による方法では，三角形メッシュを構築した後に平滑化処理が必要になる．また，補間や近似による方法では点群の欠落部をうまく補間することが難しい．フィッティングの方法では，欠落部の無い一般モデルを用意しておけば，点群の欠落部をうまく補間してくれる．そこで，本報では，フィッティングの方法を実装したHBM（Homologous Body Modeling）ソフトを利用する．

HBMソフトはワシントン大学や東京大学で研究された内容を基に開発された（有）デジタルヒューマンテクノロジーの製品である．この技術の原理は，一般モデル M （Generic Model，三角形メッシュ）とその特徴点群 L_m を測定点群 D とその特徴点群 L_d にフィッティングしていくことで近似曲面 $\hat{M}$ を

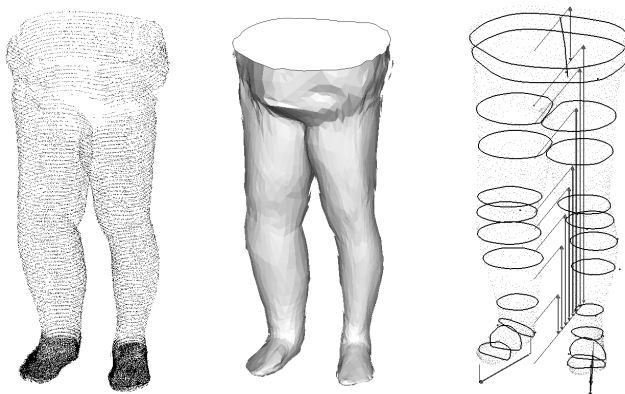


図1 測定点群，三角形メッシュ，採寸結果



図2 一般モデル M ，測定点群 D ，近似曲面 $\hat{M}$

構築するものである（図2，3）．具体的には，式(1)の誤差関数 E の最小値を算出すれば，近似曲面 $\hat{M}$ を求めることができる．

$$E = \alpha E_f + \beta E_s + \gamma E_l \quad (1)$$

$$E_f = \sum_{i=1}^n \|(v_i + \Delta v_i) - d_i\|^2 \quad (2)$$

$$E_s = \sum_{\{i,j\}|\{v_i, v_j\} \in \text{edge}\}} \|\Delta v_i - \Delta v_j\|^2 \quad (3)$$

$$E_l = \sum_{i=1}^m \|(v_i + \Delta v_i) - l_i\|^2 \quad (4)$$

E_f はフィッティングエラー関数と呼ばれ，頂点 $\hat{v}_i \in \hat{M}$ と点 $d_i \in D$ の差の2乗和で定義される． d_i は頂点 $v_i \in M$ と測定点 $d \in D$ との最近点が選ばれる． n は M の頂点数である． E_s は連続性エラー関数と呼ばれ， M の稜線 (edge) の隣接頂点 v_i, v_j の移動ベクトル $\Delta v_i, \Delta v_j$ の差の2乗和で定義される． E_s は v_i, v_j が離れたり，集中することを防ぐ． E_l はランドマークエラー関数と呼ばれ， $\hat{M}$ の特徴点 $\hat{v}_i \in L_m$ と D の特徴点 $l_i \in L_d$ の差の2乗和で定義される． L_m と L_d の数は m でなければならない． E_l は，解が関数の極小値に陥ることを防ぐ． α, β, γ は各関数に掛ける重みである．

しかし，本報で扱う測定点群 D には特徴点群 L_d が無いため，式(4)は使用しない． E_l の重み γ は常に0とする．そのため，入力データは一般モデル M ，測定点群 D ，重み α, β となり，出力データは近似曲面 $\hat{M}$ となる．この近似曲面 $\hat{M}$ が求める三角形メッシュに相当する．

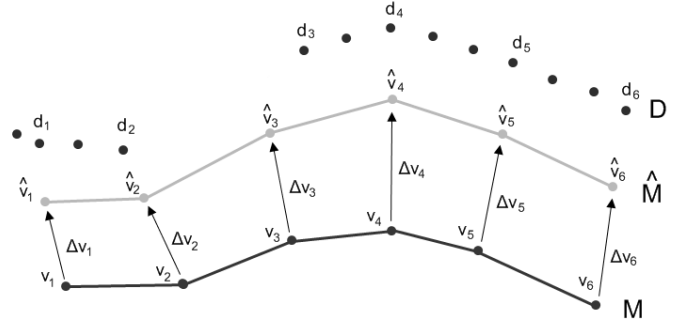


図3 フィッティングの概念図

2・2 採寸情報と体積の計算方法

採寸者が指示した恥骨などの特徴点（17点～24点）と三角形メッシュを基に採寸情報を計算する．また，利用者が指示した上下限值と三角形メッシュから，患者の部分的な体積を計算する．

まず，採寸情報を計算する．採寸は丈と周囲長の2つに分類される．丈は容易に計算できるが，周囲長の計算には工夫が必要である．本報では，周囲長の計算に三角形メッシュの位相構造を利用する．位相構造とは，三角形メッシュの頂点，稜線，三角形の繋がり方に関する構造である．ここで用いた位相構造はハーフエッジ(HE)である（図4）．HE構造は，頂点，稜線，面，HE からなり，以下のデータ構造を持つ．

- ・頂点 {点, HE}
- ・稜線 {HE}
- ・面 {法線ベクトル, HE}
- ・HE {頂点, 稜線, 面, 隣の HE, 次の HE, 前の HE}

頂点は点と HE を持つ．点は XYZ 座標値を持つ．稜線は HE を持つ．面は法線ベクトルと HE を持つ．HE は頂点・稜線・面・隣の HE・次の HE・前の HE を持つ．HE の隣の HE とは，HE から稜線を隔てて存在する HE のことである．HE の次の HE とは，同じ三角形平面の対象 HE の矢印の先側の HE のことである．HE の前の HE とは，同じ三角形平面の対象 HE の矢印の根元側の HE のことである．HE を辿ることにより，三角形メッシュの全ての稜線を調べる必要がないため，任意 Z 値の XY 平面と三角形メッシュの稜線との交点計算に無駄がなくなる．つまり，周囲長算出に利用する断面線を高速に構築すること

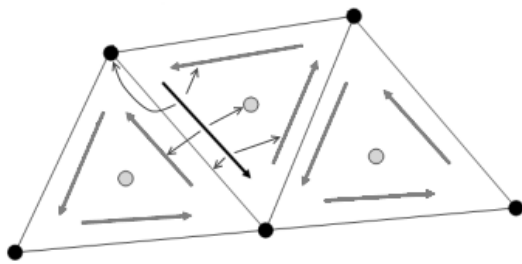


図4 ハーフエッジ構造

ができる。

三角形メッシュに対して、マウスによるピックや高さ指定により周囲長を計算し、画面に断面線を表示する。マウスによるピックの場合、入力が三角形平面とその面上の点となる。高さ指定の場合、入力が高さとして右足か左足のどちらを指定したかという情報になる。まず、問題を統一するため、入力を稜線 BStart とその稜線上の点 Pstart に変換する。マウスによるピックの場合、点の高さ (Z 値) から得られる XY 平面と三角形の稜線との交点が存在する稜線を選びだせばよい。稜線が 2 本あればどちらを選択してもよい。高さ指定の場合、Y 軸の+側に右足があり、Y 軸の-側に左足があるとすれば、右足の周囲長を計算したい場合は、Y 軸の+側の稜線から順に、任意 Z 値の XY 平面との交点を調べていけばよい。出力は、断面線を表示する点リスト PL である。断面線を算出するアルゴリズム (FUNC1) は以下の通りである。これは頂点を通る断面線にも対応する。

FUNC1：断面線を算出するアルゴリズム

入力：稜線 BStart と BStart 上の点 Pstart

出力：断面線を表示する点リスト PL

1. HEnow ← BStart.HE
2. FUNC2(入力 HEnow, Pstart.z,
出力 bE, bEV, Pnow)
if bE=偽 HEnow = HEnow.mate
3. PL.Add(Pstart)
4. FUNC2(入力 HEnow.next, Pstart.z,
出力 bE, bEV, Pnow)
if bE=真 HEnow ← HEnow.next goto 6.
5. FUNC2(入力 HEnow.prev, Pstart.z,
出力 bE, bEV, Pnow)
if bE=真 HEnow ← HEnow.prev goto 6.
else HEnow ← HEnow.mate.next goto 4.

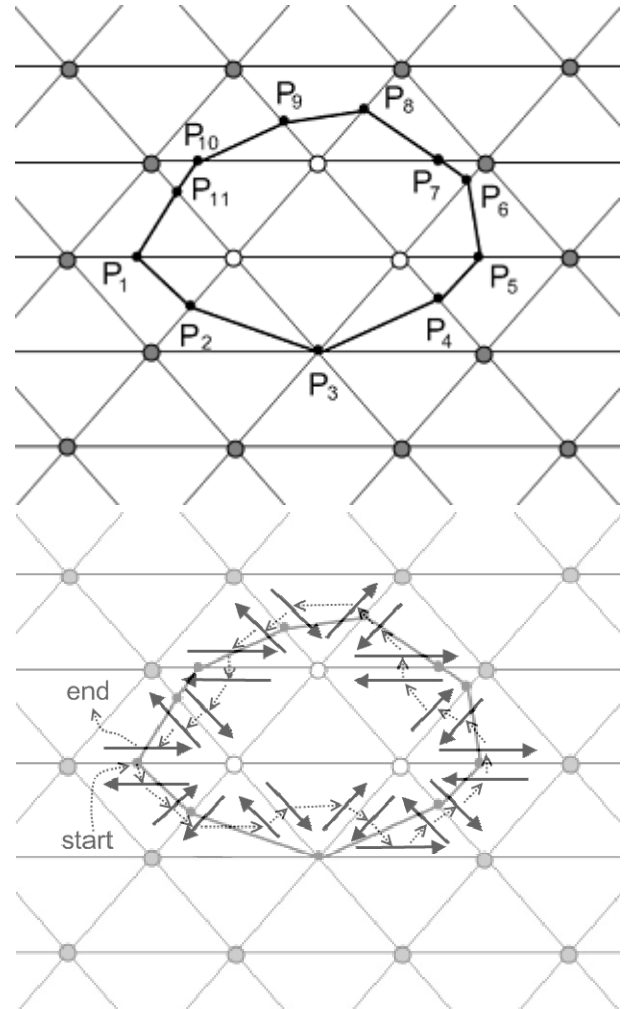


図5 断面線計算のために辿った HE の軌跡

6. if Pstart=Pnow 結果を出力し、処理を終了
7. PL.Add(Pnow)
8. if bEV = 真 HEnow ← HEnow.mate.next
else HEnow ← HEnow.mate
9. goto 4.

FUNC2：HE と任意 Z 値の XY 平面との交点を算出するアルゴリズム

入力：対象 HE, XY 平面の Z 値 Zarb

出力：bE (交点がある→真), bEV (頂点上に交点がある→真), 交点 Pnow

1. bE ← 偽, bEV ← 偽, 点 a ← HE.vertex.point,
点 b ← HE.next.vertex.point, ベクトル ab ← b - a
2. if a.z < b.z goto 3. else if a.z > b.z goto 4.
else if a.z=b.z goto 5.
3. if a.z < Zarb < b.z
bE ← 真, t ← (Zarb - a.z) / (b.z - a.z),
Pnow ← a + t * ab, goto 6.

```

else if a.z = Zarb
    bE ← 真, bEV ← 真, Pnow ← a, goto 6.
else goto 6.
4. if a.z > Zarb > b.z
    bE ← 真, t ← (Zarb - a.z) / (b.z - a.z),
    Pnow ← a + t * ab, goto 6.
else if a.z = Zarb
    bE ← 真, bEV ← 真, Pnow ← a, goto 6.
else goto 6.
5. if Zarb = a.z
    bE ← 真, bEV ← 真, Pnow ← a, goto 6.
else goto 6.
6. 結果を出力し, 処理を終了

```

断面より上の頂点を白色で, 下の頂点を灰色で示すと, 断面線は図 5 上のようになり, 辿った HE の経路は図 5 下のようになる.

メジャーで採寸すると, 断面線が凹凸形状でも, メジャーの形は凸形状となる. そこで断面線より凸包を作成し, これを周囲形状とする. 2 次元凸包の構成問題は計算幾何学で扱われている. 本報では, 杉原の方法⁵⁾を参考にして 2 次元凸包構成プログラムを作成する.

次に部分的な体積を計算する. 発散定理より, 閉じた三角形メッシュの体積 V は, 点 a_i , 点 b_i , 点 c_i の外積と内積の和として, 次の式で表現できる.

$$V = \frac{1}{6} \sum_{i=1}^n \langle a_i, (b_i - a_i) \times (c_i - a_i) \rangle \quad (5)$$

n は三角形の数である. 点 a_i , 点 b_i , 点 c_i と順と順に見たとき, 反時計回りならその三角形平面は表となり, 時計回りなら裏となるように, あらかじめ三角形メッシュを構築しておく.

部分的な体積とは, 下限値 (Z 値) の XY 平面と上限値 (Z 値) の XY 平面との間に囲まれる三角形メッシュの体積のこととする. この上下の XY 平面で三角形メッシュを切断し, その三角形メッシュに蓋をすれば, 発散定理により部分的な体積が計算できる.

まず, 三角形メッシュを任意 Z 値の XY 平面で切断する方法を考える. 断面線を算出するときとほぼ同様のアルゴリズムで 1 つの三角形メッシュを 2 つの三角形メッシュに分割できる. まず, 三角形リス



図 6 一般モデル, 患者 A, 患者 B, 患者 C

トを上下 2 つ (FLt, FLb) 用意する. 次に, HE を伝って断面線を構築するときと同様に, HE を伝って切断面付近の 1 個の三角形を 2, 3 個の三角形に分割し, FLt と FLb に登録していく. 切断面付近の三角形の登録が全て終了した後, それ以外の未訪問の三角形を FLt と FLb のどちらに属するかを判定し, 次々と登録する. 未訪問の三角形がなくなれば, 切断された三角形リスト FLt と FLb が得られる. 後は, 三角形リストに位相を構築し, 三角形メッシュとしてやればよい.

次に, 三角形メッシュの穴埋めをする. これには計算幾何学の Polygon triangulation というポリゴンを三角形群に分解する技術を用いる. 本報では OpenGL に用意されている gluNewTess 関数などを使用する.

3 実験結果

人体下半身の測定点群を三角形メッシュ化し, 採寸情報と体積を計算する採寸ソフトを開発した. 開発には Microsoft Visual Studio 2005 (C#と C++), Xeon 3.2GHz の PC を利用した. 実験にも同様の PC を利用した.

3・1 HBM による三角形メッシュの構築結果

HBM への入力データは, 一般モデル, 測定点群, 重みである. 一般モデルは, 当センターで開発した非接触人体形状測定装置から得られた 1 人の測定点群から, INUS Technology 社の形状処理ソフト Rapid Form を利用して作成した. 穴埋めは SensAble Technologies 社のボクセル処理ソフト Freeform で行った. 作成した一般形状の頂点数は 1017, 稜線数は 3015, 面数は 1999 である.

測定点群はリンパ浮腫患者 3 人 (A, B, C) から得られたデータを用いた. 測定点数はそれぞれ 33653,



図7 フィッティングの様子. 左上から, 処理前の一般形状, 手順 1, 2, ..., 10 を処理後の再構築モデル. 累積計算時間は, 手順 1 (2.6s), 手順 2 (3.5s), 手順 3 (4.2s), 手順 4 (5.0s) 手順 5 (5.7s), 手順 6 (12s), 手順 7 (35s), 手順 8 (43s), 手順 9 (52s), 手順 10 (65s).

35225, 29634 である. 図 6 からわかるように, 測定点群は, 足首から先の密度が高く, 足首から上の密度が低い. 患者 B に顕著に見られるが, 全ての測定点群は, 股部分が欠落している. これは, 非接触人体形状測定装置のカメラ 6 台が全て下向きのため, 股部分が死角になるためである.

フィッティングには式(1)のほかに, 三角形メッシュの細分化も行う. これは, まずは粗いモデルを大まかにフィッティングし, 次に細分化したのちにフィッティングすることで, より細かい形状にフィッティングできるようにするためである. ただし, メモリや計算時間の制約からそれほど頂点数を多くすることができない. HBM には細分化手法として, 補間 (Modified Butterfly) と近似 (Loop) を用意している. 位相的には, 1 つの面を 4 つに分割する. 言い換えると各稜線に頂点を必ず 1 つ生成する. そのため, 細分化を 1 回行うと, 元の頂点数 V_a , 元の稜線数 E_a , 元の面数 F_a と, 後の頂点数 V_b , 後の稜線数 E_b , 後の面数 F_b の関係は, $V_b = V_a + E_a$, $E_b = 2E_a + 3F_a$, $F_b = 4F_a$ となる. 今回作成した一般モデルを用いた場合, 細分化を 1 回適用した三角形メッシュの頂点数は 4032, 稜線数は 12027, 面数は 7996 となる. 補間手法で細分化すると面が荒れ

るため, 近似手法でなるべく滑らかな再構築モデルを生成する. 一度面が荒れてしまうと HBM のフィッティング機能で修正することは困難である.

式(1)と細分化によるフィッティングの手順は以下のとおりである. 最初は連続性を重視し, 徐々にフィッティング精度を高めながら一度細分化し, さらに詳細にフィッティングした. この手順は実験的に決定した.

1. 式(1) $\alpha=1.0, \beta=8.0, \gamma=0$
2. 式(1) $\alpha=1.0, \beta=4.0, \gamma=0$
3. 式(1) $\alpha=1.0, \beta=2.0, \gamma=0$
4. 式(1) $\alpha=1.0, \beta=1.5, \gamma=0$
5. 式(1) $\alpha=1.0, \beta=1.2, \gamma=0$
6. 細分化 近似 (Loop) 1 回
7. 式(1) $\alpha=1.0, \beta=1.0, \gamma=0$
8. 式(1) $\alpha=1.0, \beta=1.0, \gamma=0$
9. 式(1) $\alpha=1.0, \beta=1.0, \gamma=0$
10. 式(1) $\alpha=1.0, \beta=1.0, \gamma=0$

以上の手順で患者 B に一般モデルをフィッティングする様子を図 7 に示す. 手順 1~10 までの計算時間の合計は 65 秒である. 次に, 患者 A,B,C のフィ

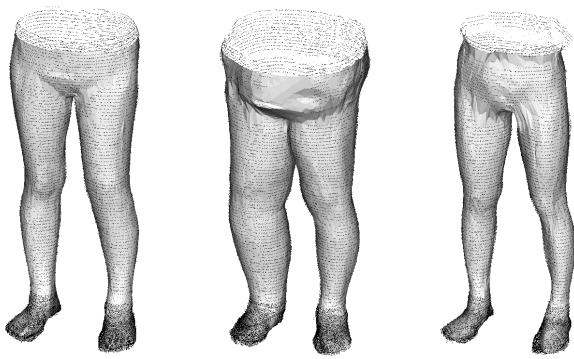


図 8 患者 A, B, C の近似曲面



図 9 足首辺りの近似曲面（患者 A, B, C）



図 10 フィッティング精度の改良（患者 A, B, C）

フィッティング結果を図 8, 図 9 に示す. 図 8 を見ると, 一般モデルが測定点群におおよそフィッティングしていることがわかる. しかし, 図 9 を見ると, 足首から下のフィッティングが悪いことがわかる.

足首から下のフィッティングが悪い原因は, 特徴点を用いなかったため, 解が誤差関数の極小値に落ちこんでしまったことにある. しかし, 非接触人体形状測定装置では特徴点を取得できない. また, 測定点群から特徴点を算出することも困難である. そこで, 一般モデルを変形してから同様のフィッティング手順を踏むことにした. 変形処理は, 足首から下の部位を足首を軸として回転し, 踵からつま先を拡大縮小するというものである. 結果を図 10 に示す. フィッティングは改善されているが, まだ不完全であることがわかる. 例えば, 患者 A の右足の親指はかなりずれている. また, 左足の中指の先もずれが大きい. 患者 C の右足の中指の先や小指の先, 左足の中指の先もずれが大きい. このずれは, これ以上フィッティングを繰り返してもほとんど改善されなかった.

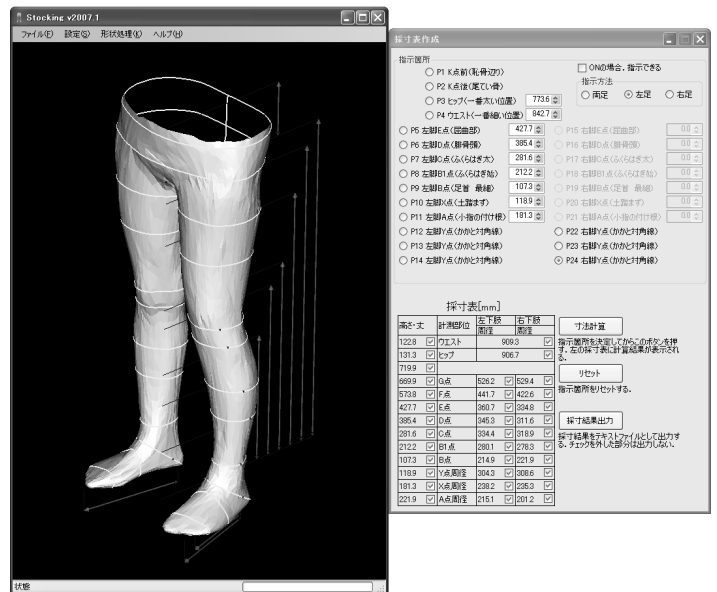


図 11 採寸表作成画面

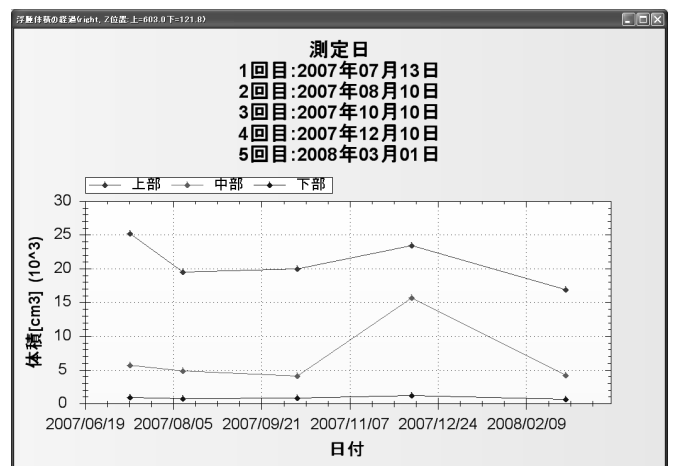


図 12 浮腫体積の経過

3・2 採寸情報と体積の計算結果

周囲長と体積が既知のモデルを利用して, 周囲長と体積が正しく計算できることを確認した.

図 11 に三角形メッシュから採寸表を作成する操作画面を示す. 左側の画面に三角形メッシュが表示され, 右側の画面に採寸表を作成するためのダイアログが表示される. ダイアログからピックする特徴点を選択し, 三角形メッシュをピックしたり, 高さ情報を与えることで周囲形状が計算され, 丈と周囲長が計算される.

体積計算に関しても同様に, 三角形メッシュをピックしたり, 高さ情報を与えることで体積が計算される. 浮腫体積の経過がどのようになっているか一目でわかるように体積計算結果をグラフに表示する機能も加えた (図 12).

4 まとめ

人体下半身の測定点群を三角形メッシュ化し、採寸情報と体積を計算する採寸ソフトを開発した。測定点群を三角形メッシュ化する処理は、指以外はうまくメッシュ化できた。股付近の点群欠落部もうまく補間できた。指のフィッティング精度が悪い原因は、フィッティングエラー関数の点 $d_i \in D$ を頂点 $v_i \in M$ と測定点 $d \in D$ との最近点としているため、他の指や指元などの望まない点を選択されたからだと考えられる。点 d_i の選択方法を改良すればフィッティングが良くなる可能性がある。

採寸情報と体積は正確に計算することができた。しかし、採寸情報の計算には、採寸者がマウスを利用して特徴点を 17 点～24 点選択する必要があるため、作業に時間がかかる。また、人手が介入するため、採寸者により採寸情報が変化する。この作業を自動化するための、特徴点計算方法が必要である。

謝辞

本研究は、平成 18～19 年度地域新生コンソーシアム研究開発事業「リンパ浮腫患者用弾性ストッキング製造システムの開発」において実施しました。四国経済産業局、(財)とくしま産業振興機構を始めとする支援機関の皆様に感謝します。また、HBM ソフトについての助言を頂いた(独)産業技術総合研

究所の小川洋司氏、持丸正明博士に感謝します。また、採寸ソフトの実用試験をして頂いた東光(株)の皆様に感謝します。

参考文献

- 1) Tamal K. Dey, Joachim Giesen and James Hudson: “Delaunay Based Shape Reconstruction from Large Data”, IEEE Symposium in Parallel and Large Data Visualization and Graphics(PVG), pp19-27(2001)
- 2) Michael Kazhdan, Matthew Bolitho and Hugues Hoppe, “Poisson Surface Reconstruction”, Eurographics Symposium on Geometry Processing, pp61-70(2006)
- 3) 稲垣 知大, 浅田 友紀, 倉賀野 穰, 鈴木 宏正, 持丸 正明, 河内 まき子:「解剖学的特徴点による点群からの足形状の再構成と人体寸法の抽出」, 2004 年度精密工学会春季大会学術講演会講演論文集, pp271-272(2004)
- 4) Brett Allen, Brian Curless and Zoran Popović, “The space of human body shapes: reconstruction and parameterization from range scans”, ACM SIGGRAPH, pp587-594(2003)
- 5) 杉原厚吉:「FORTRAN 計算幾何プログラミング」, (株) 岩波書店(1998)