

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7641048号
(P7641048)

(45)発行日 令和7年3月6日(2025.3.6)

(24)登録日 令和7年2月26日(2025.2.26)

(51)国際特許分類 F I
G 0 6 T 7/13 (2017.01) G 0 6 T 7/13
G 0 6 T 7/00 (2017.01) G 0 6 T 7/00 6 6 0 Z

請求項の数 15 (全31頁)

(21)出願番号	特願2023-552543(P2023-552543)	(73)特許権者	522106031 上海慧姿化粧品有限公司 SHANGHAI HUIZI COSMETICS CO., LTD. 中華人民共和国 201401 上海市奉賢区奉金路166号-1第2幢2層A区 Area A, 2nd Floor, Building 2 No. 166-1 Fengjin Road, Fengxi District Shanghai 201401 (CN)
(86)(22)出願日	令和3年4月28日(2021.4.28)	(74)代理人	110003801 KEY弁理士法人
(65)公表番号	特表2024-511563(P2024-511563 A)	(72)発明者	デン ジンピャオ 中華人民共和国 200233 徐匯区宜 最終頁に続く
(43)公表日	令和6年3月14日(2024.3.14)		
(86)国際出願番号	PCT/CN2021/090435		
(87)国際公開番号	WO2022/198745		
(87)国際公開日	令和4年9月29日(2022.9.29)		
審査請求日	令和5年9月25日(2023.9.25)		
(31)優先権主張番号	202110312686.8		
(32)優先日	令和3年3月24日(2021.3.24)		
(33)優先権主張国・地域又は機関	中国(CN)		

(54)【発明の名称】 爪輪郭の抽出方法及びそのシステム

(57)【特許請求の範囲】

【請求項1】

爪輪郭の抽出方法であって、以下の工程：

対象指に対して三次元走査を行い、対象指の指三次元データDataFingerを取得し、前記指三次元データDataFingerは少なくとも、すべての爪の爪三次元データDataを含んでいる、データ収集工程と、

前記指三次元データDataFingerを三次元点群データに変換して、指点群CloudFingerを取得し、前記指点群CloudFingerは少なくとも、すべての爪の爪曲面点群CloudNailsを含んでいる、データ点群処理工程と、

前記指点群CloudFingerに対して分割を行い、前記指点群CloudFingerから前記爪曲面点群CloudNailsを取得する、爪点群取得工程と、

前記爪曲面点群CloudNailsを表面積を変えずに平坦化して、平面状態における爪平面点群CloudNails-pを取得する、爪点群平坦化工程と、

前記爪平面点群CloudNails-pの輪郭に基づき輪郭点群CloudContourの形状をレンダリングし、爪の爪甲型を取得する、爪輪郭レンダリング工程と、を含むことを特徴とする爪輪郭の抽出方法。

【請求項2】

前記データ点群処理工程は、

前記指点群CloudFingerを、姿勢及び位置をプリセットの姿勢及び位置上にある出力指点群CloudFinger-oに変換する、点群方向矯正サブ工程をさらに含み、

前記プリセットの姿勢及び位置とは、前記出力指点群 $CloudFinger-o$ の質量中心が PCL 点群ライブラリ中の座標系の原点と重複し、指の先端が座標系中の x 軸の正の向きを指し、指の背面が座標系中の z 軸の負の向きを指すことを意味し、

前記爪点群取得工程における前記指点群 $CloudFinger$ として前記出力指点群 $CloudFinger-o$ を用いる、ことを特徴とする、請求項 1 に記載の爪輪郭の抽出方法。

【請求項 3】

前記点群方向矯正サブ工程後において、

前記データ点群処理工程は、

変換後の指点群中の指の背面が座標系中の z 軸の負の向きを指しているか否かをチェックし、

指の背面が座標系中の z 軸の正の向きを指している場合、前記変換後の指点群を x 軸まわりに 180 度回転させた上で前記出力指点群 $CloudFinger-o$ とする、又は、

指の背面が座標系中の z 軸の負の向きを指している場合、前記変換後の指点群を前記出力指点群 $CloudFinger-o$ とする、チェックサブ工程をさらに含むことを特徴とする、請求項 2 に記載の爪輪郭の抽出方法。

【請求項 4】

前記爪点群取得工程は、

領域拡張分割法を用いて前記指点群 $CloudFinger$ を分割し、前記指点群 $CloudFinger$ から前記爪曲面点群 $CloudNail-s$ を取得する、点群分割サブ工程を含むことを特徴とする、請求項 1 から 3 のいずれか 1 項に記載の爪輪郭の抽出方法。

【請求項 5】

前記点群分割サブ工程前において、前記爪点群取得工程は、

先に爪の爪甲面エリア $Nrange(x, y)$ を判定して取得した上で、前記爪甲面エリア $Nrange(x, y)$ 中で 1 点をシード点 S_0 として選択し、前記シード点 S_0 によって前記領域拡張分割法における拡張プロセスを開始する、シード点 S_0 選択サブ工程をさらに含むことを特徴とする、請求項 4 に記載の爪輪郭の抽出方法。

【請求項 6】

前記爪点群平坦化工程において、前記爪曲面点群 $CloudNail-s$ を x 軸に沿って横向きに分割して n 層の点群 L_i とし、次に各層の前記点群 L_i を全て同一層上に展開して平坦化し、平面状態における前記爪平面点群 $CloudNail-p$ を取得することを特徴とする、請求項 1 に記載の爪輪郭の抽出方法。

【請求項 7】

前記爪点群平坦化工程は、具体的には、

各層の前記点群 L_i 中における各点 p_j の x 軸座標値を統一し、且つその y 軸及び z 軸座標値は変えずに保持し、1 つの略円弧形の点群集合 $stand_L_i$ を取得することと、

各層の前記点群集合 $stand_L_i$ 中におけるすべての点 p を爪の中心線の y 座標 $Center_y$ に従って 2 つの部分に分け、 L_part1_i 、 L_part2_i とすることと、

各部分の円弧形点群 L_part1_i 、 L_part2_i を爪の中心線に近いものから順番に 1 つの直線上に配列し、 $L_ordered_part_i$ を得ることと、を含み、

前記 $L_ordered_part_i$ を用いて前記爪平面点群 $CloudNail-p$ を取得する、ことを特徴とする、請求項 6 に記載の爪輪郭の抽出方法。

【請求項 8】

前記爪点群平坦化工程後において、

取得した前記爪平面点群 $CloudNail-p$ をエッジ処理して、エッジを平滑化する、エッジ処理工程をさらに含むことを特徴とする、請求項 1 に記載の爪輪郭の抽出方法。

【請求項 9】

爪輪郭の抽出システムであって、請求項 1 に記載の爪輪郭の抽出方法を実現するのに用いられ、

10

20

30

40

50

対象指に対して三次元走査を行い、対象指の指三次元データ `DataFinger` を取得し、前記指三次元データ `DataFinger` は少なくとも、すべての爪の爪三次元データ `DataNail` を含んでいる、データ収集装置と、

データ変換装置であって、前記データ収集装置と通信接続され、前記データ変換装置は前記指三次元データ `DataFinger` を受信し且つ三次元点群データに変換することにより指点群 `CloudFinger` を取得し、前記指点群 `CloudFinger` は少なくとも、すべての爪の爪曲面点群 `CloudNail-s` を含んでいる、データ変換装置と、

前記データ変換装置と通信接続され、PCL点群ライブラリを有するPCL装置と、

分割モジュールであって、前記PCL装置内にあり、前記分割モジュールは、前記指点群 `CloudFinger` に対して分割を行い、前記指点群 `CloudFinger` から、前記爪の前記爪曲面点群 `CloudNail-s` を抽出できるように構成される、分割モジュールと、

平坦化モジュールであって、前記PCL装置と通信接続され、前記平坦化モジュールは、前記爪曲面点群 `CloudNail-s` を表面積を変えずに展開して平坦化し、平面状態における前記爪平面点群 `CloudNail-p` を取得できるように構成される、平坦化モジュールと、

レンダリングモジュールであって、前記平坦化モジュールと通信接続され、前記レンダリングモジュールは、前記爪平面点群 `CloudNail-p` を受信し、且つ前記爪平面点群 `CloudNail-p` の輪郭点群 `CloudContour` の形状に基づきレンダリングを行い、前記爪の爪甲型を取得する、レンダリングモジュールと、を含むことを特徴とする爪輪郭の抽出システム。

【請求項10】

前記データ変換装置は矯正モジュールを有し、前記矯正モジュールは、前記指点群 `CloudFinger` を、姿勢及び位置をプリセットの姿勢及び位置上にある出力指点群 `CloudFinger-o` に変換できるように構成され、

前記分割モジュールは、前記指点群 `CloudFinger` として前記出力指点群 `CloudFinger-o` を用いる、ことを特徴とする請求項9に記載の爪輪郭の抽出システム。

【請求項11】

前記爪輪郭の抽出システムは、

チェックモジュールであって、前記矯正モジュールと通信接続され、前記チェックモジュールは、変換後の指点群中における指の背面が指す方向を受信し且つチェックするように構成され、

前記変換後の指点群中の指の背面が座標系中のz軸の正の向きを指している場合、前記矯正モジュールを用いてその指点群をx軸まわりに180度回転させた上で前記出力指点群 `CloudFinger-o` とし、

前記変換後の指点群中の指の背面が座標系中のz軸の負の向きを指している場合、前記変換後の指点群をそのまま前記出力指点群 `CloudFinger-o` とする、チェックモジュールをさらに含むことを特徴とする請求項10に記載の爪輪郭の抽出システム。

【請求項12】

前記爪点群取得工程は、

領域拡張分割法を用いて前記指点群 `CloudFinger` を分割し、前記指点群 `CloudFinger` から前記爪曲面点群 `CloudNail-s` を取得する、点群分割サブ工程を含み、

前記爪輪郭の抽出システムは、

前記分割モジュールと通信接続されるシード点 S_0 の選択モジュールをさらに含み、前記選択モジュールは、前記爪曲面点群 `CloudNail-s` 上の爪甲面エリア $Nrange(x, y)$ を取得し、且つ前記爪甲面エリア $Nrange(x, y)$ 中で1点をシード点 S_0 として選択し、前記分割モジュールが前記シード点 S_0 によって前記領域拡張分割法における拡張プロセスを開始できるように構成されることを特徴とする請求項9に記載の爪輪郭の抽出システム。

10

20

30

40

50

【請求項 13】

前記平坦化モジュールは、前記爪曲面点群 $Cloud Nail - s$ を x 軸に沿って横向きに分割して n 層の点群 L_i とし、次に各層の前記点群 L_i を全て同一層上に展開して平坦化し、平面状態における前記爪平面点群 $Cloud Nail - p$ を取得できるように構成されることを特徴とする請求項 9 に記載の爪輪郭の抽出システム。

【請求項 14】

前記平坦化モジュールは、各層の前記点群 L_i 中における各点 p_j の x 軸座標値を統一し、且つその y 軸及び z 軸座標値は変えずに保持し、1つの略円弧形の点群集合 $stand_L_i$ を取得し、次に各層の前記点群集合 $stand_L_i$ 中におけるすべての点 p を爪の中心線の y 座標 $Center_y$ に従って2つの部分に分け、 L_part1_i 、 L_part2_i とし、最後に各部分の円弧形点群 L_part1_i 、 L_part2_i を爪の中心線に近いものから順番に1つの直線上に配列し、 $L_ordered_part_i$ を得て、 $L_ordered_part_i$ を用いて前記爪平面点群 $Cloud Nail - p$ を取得することができるように構成されることを特徴とする請求項 13 に記載の爪輪郭の抽出システム。

10

【請求項 15】

前記 PCL 装置は、

取得した前記爪平面点群 $Cloud Nail - p$ をエッジ処理して、エッジを平滑化するのに用いる、エッジ処理モジュールをさらに含むことを特徴とする請求項 9 に記載の爪輪郭の抽出システム。

20

【発明の詳細な説明】**【技術分野】****【0001】**

本願は画像処理分野に関し、より具体的には、三次元点群の結果に基づき爪輪郭を抽出する方法及びそのシステムに関する。

【背景技術】**【0002】**

近年、人々が爪の美しさを追及するようになるにつれて、ネイルチップと呼ばれる手軽なネイルアート製品が市場に登場するようになった。ネイルチップは、成形されたチップであり、接着層を利用して爪に直接貼ることができる。従来のマニキュアによるネイルアートと比べて、ネイルチップは取り扱いが簡単で、価格が安く、いつでもどこでも使用できるなどの利点があり、ファッションやおしゃれを好む人々の間でますます人気が高まっている。

30

【0003】

一般的に、ネイルチップは形状が固定されたものが出荷される。また、製品の美観性のために、その外周輪郭は平面状態で完璧なエッジ半径を有するのが一般的である。しかし、固定された完璧なエッジ半径は使用者の実際の爪輪郭とフィットせず、使用者の様々な爪形状に完全に適応することができないため、使用者の爪に貼るネイルチップには、貼付けが不完全になったり、はみ出したりするという問題が存在していた。

【0004】

40

現在、この問題の主な解決方法としては、画像認識を利用して使用者の爪輪郭を抽出するものがある。例えば、特許文献 1 はネイルアート装置のプリント方法を開示しており、そのネイルアート装置のプリント方法では、撮影装置が使用者の指を撮影し、制御モジュールが撮影装置の撮影した使用者の指の動画又は写真に基づき爪の位置、形状及びサイズを判断し、爪輪郭を認識する。

【0005】

しかし、このネイルアート装置のプリント方法は、抽出するデータが二次元画像の二次元データであり、やはり弧度を有する三次元の爪形状と完全にはフィットせず、抽出された爪輪郭が真の爪輪郭とフィットしていなかった。

【先行技術文献】

50

【特許文献】

【0006】

【文献】中国特許出願公開第109770520号明細書

【発明の概要】

【発明が解決しようとする課題】

【0007】

上述の技術的課題を解決するために、新しい爪輪郭抽出方法により、使用者の爪の周縁によりフィットし、弧度を有する三次元の爪輪郭を抽出する差し迫った必要性があった。

【課題を解決するための手段】

【0008】

第1の態様として、本願が提供する爪輪郭の抽出方法は、以下の工程：

対象指に対して三次元走査を行い、対象指の指三次元データDataFingerを取得し、指三次元データDataFingerは少なくとも、すべての爪の爪三次元データDataNailを含んでいる、データ収集工程と、

指三次元データDataFingerを三次元点群データに変換して、指点群CloudFingerを取得し、指点群CloudFingerは少なくとも、すべての爪の爪曲面点群CloudNailsを含んでいる、データ点群処理工程と、

指点群CloudFingerに対して分割を行い、前記指点群CloudFingerから前記爪曲面点群CloudNailsを取得する、爪点群取得工程と、

爪曲面点群CloudNailsを表面積を変えずに平坦化して、平面状態における爪平面点群CloudNail-pを取得する、爪点群平坦化工程と、

爪平面点群CloudNail-pの輪郭に基づき輪郭点群CloudContourの形状をレンダリングし、爪の爪甲型を取得する、爪輪郭レンダリング工程と、を含む。

【0009】

さらに、好適には、データ点群処理工程は、

指点群CloudFingerを、姿勢及び位置をプリセットの姿勢及び位置上にある出力指点群CloudFinger-oに変換する、点群方向矯正サブ工程をさらに含み、

プリセットの姿勢及び位置とは、出力指点群CloudFinger-oの質量中心がPCL点群ライブラリ中の座標系の原点と重複し、指の先端が座標系中のx軸の正の向きを指し、指の背面が座標系中のz軸の負の向きを意味することをいう。

【0010】

さらに、好適には、点群方向矯正サブ工程後において、前記データ点群処理工程は、変換後の指点群中の指の背面が座標系中のz軸の負の向きを指しているか否かをチェックし、

指の背面が座標系中のz軸の正の向きを指している場合、変換後の指点群をx軸まわりに180度回転させた上で出力指点群CloudFinger-oとし、

指の背面が座標系中のz軸の負の向きを指している場合、変換後の指点群をそのまま出力指点群CloudFinger-oとする、チェックサブ工程をさらに含む。

【0011】

さらに、好適には、爪点群取得工程は、

領域拡張分割法を用いて指点群CloudFingerを分割し、前記指点群CloudFingerから前記爪曲面点群CloudNailsを取得する、点群分割サブ工程を含む。

【0012】

さらに、好適には、点群分割サブ工程後において、前記爪点群取得工程は、

検索半径 R_{rs} をベースとして検索半径 R_{rs} よりも小さい検索半径 R_{rs}' を入力し、検索半径 R_{rs}' にある点と近傍点との法線ベクトルの夾角 R_{rs} を取得し、且つ角度閾値 $thres_2$ を判定条件として追加し、

検索半径 R_{rs} 及び検索半径 R_{rs}' にある点と近傍点との法線ベクトルの夾角の差が角度閾値 $thres_2$ よりも大きい場合、拡張を停止する、クラスター拘束サブ工程をさ

10

20

30

40

50

らに含む。

【0013】

さらに、好適には、点群分割サブ工程前において、前記爪点群取得工程は、先に爪の爪甲面エリア $N_{range}(x, y)$ を判定して取得した上で、爪甲面エリア $N_{range}(x, y)$ 中で1点をシード点 S_0 として選択し、シード点 S_0 によって拡張プロセスを開始する、シード点 S_0 選択サブ工程をさらに含む。

【0014】

さらに、好適には、爪点群平坦化工程において、爪曲面点群 $Cloud_{Nail-s}$ を x 軸に沿って横向きに分割して n 層の点群 L_i とし、次に各層の点群 L_i を全体として等しい長さに展開して平坦化し、平面状態における爪平面点群 $Cloud_{Nail-p}$ を取得する。

10

【0015】

さらに、好適には、爪点群平坦化工程において、具体的には、各層の点群 L_i 中における各点 p_j の x 軸座標値を統一し、且つその y 軸及び z 軸座標値は変えずに保持し、1つの略円弧形の点群集合 $stand_L_i$ を取得することと、各層の点群集合 $stand_L_i$ 中におけるすべての点 p を爪の中心線の y 座標 $Center_y$ に従って2つの部分に分け、 L_part1_i 、 L_part2_i とすることと、各部分の円弧形点群 L_part1_i 、 L_part2_i を爪の中心線に従って近くから遠くまで順序通り配列し、 $L_ordered_part_i$ を得ることと、を含む。

【0016】

さらに、好適には、爪点群平坦化工程後において、取得した爪平面点群 $Cloud_{Nail-p}$ をエッジ処理して、エッジを平滑化する、エッジ処理工程をさらに含む。

20

【0017】

第2の態様として、本願は爪輪郭の抽出システムをさらに提供する。この抽出システムは、上述の任意の技術案中の爪輪郭抽出方法を実現するのに用いられる。具体的には、爪輪郭の抽出システムは、

対象指に対して三次元走査を行い、対象指の指三次元データ $Data_{Finger}$ を取得し、指三次元データ $Data_{Finger}$ は少なくとも、すべての爪の爪三次元データ $Data_{Nail}$ を含んでいる、データ収集装置と、

30

データ変換装置であって、データ収集装置と通信接続され、データ変換装置は指三次元データ $Data_{Finger}$ を受信し且つ三次元点群データに変換することにより指点群 $Cloud_{Finger}$ を取得し、指点群 $Cloud_{Finger}$ は少なくとも、すべての爪の爪曲面点群 $Cloud_{Nail-s}$ を含んでいる、データ変換装置と、

データ変換装置と通信接続される、PCL点群ライブラリと、

分割モジュールであって、PCL点群ライブラリ内にあり、分割モジュールは、指点群 $Cloud_{Finger}$ に対して分割を行い、前記指点群 $Cloud_{Finger}$ から、前記爪の爪曲面点群 $Cloud_{Nail-s}$ を抽出できるように構成される、分割モジュールと、

平坦化モジュールであって、PCL点群ライブラリと通信接続され、平坦化モジュールは、爪曲面点群 $Cloud_{Nail-s}$ を表面積を変えずに展開して平坦化し、平面状態における爪平面点群 $Cloud_{Nail-p}$ を取得できるように構成される、平坦化モジュールと、

40

レンダリングモジュールであって、平坦化モジュールと通信接続され、レンダリングモジュールは、爪平面点群 $Cloud_{Nail-p}$ を受信し、且つ爪平面点群 $Cloud_{Nail-p}$ の輪郭点群 $Cloud_{Contour}$ の形状に基づきレンダリングを行い、爪の爪甲型を取得する、レンダリングモジュールと、を含む。

【0018】

さらに、好適には、データ変換装置中は矯正モジュールを有し、矯正モジュールは、指点群 $Cloud_{Finger}$ を、姿勢及び位置をプリセットの姿勢及び位置上にある出力指

50

点群 $CloudFinger-o$ に変換できるように構成される。

【0019】

さらに、好適には、爪輪郭の抽出システムは、

チェックモジュールであって、矯正モジュールと通信接続され、チェックモジュールは、変換後の指点群中における指の背面が指す方向を受信し且つチェックするように構成され、

変換後の指点群中の指の背面が座標系中の z 軸の正の向きを指している場合、矯正モジュールを用いてその指点群を x 軸まわりに 180 度回転させた上で出力指点群 $CloudFinger-o$ とし、

変換後の指点群中の指の背面が座標系中の z 軸の負の向きを指している場合、変換後の指点群をそのまま出力指点群 $CloudFinger-o$ とする、チェックモジュールをさらに含む。

10

【0020】

さらに、好適には、爪輪郭の抽出システムは、シード点 S_0 の選択モジュールをさらに含み、選択モジュールは、爪曲面点群 $CloudNail-s$ 上の爪甲面エリア $Nrange(x, y)$ を取得し、且つ爪甲面エリア $Nrange(x, y)$ 中で1点をシード点 S_0 として選択し、分割モジュールがシード点 S_0 によって拡張プロセスを開始する。

【0021】

さらに、好適には、分割モジュールは、爪曲面点群 $CloudNail-s$ を x 軸に沿って横向きに分割して n 層の点群 L_i とし、次に各層の点群 L_i を全体として等しい長さに展開して平坦化し、平面状態における爪平面点群 $CloudNail-p$ を取得できるように構成される。

20

【0022】

さらに、好適には、平坦化モジュールは、各層の点群 L_i 中における各点 p_j の x 軸座標値を統一し、且つその y 軸及び z 軸座標値は変えずに保持し、1つの略円弧形の点群集合 $stand_L_i$ を取得し、次に各層の点群集合 $stand_L_i$ 中におけるすべての点 p を爪の中心線の y 座標 $Center_y$ に従って2つの部分に分け、 L_part1_i 、 L_part2_i とし、最後に各部分の円弧点群 L_part1_i 、 L_part2_i を爪の中心線に従って近くから遠くまで順序通り配列し、 $L_ordered_part_i$ を得ることができるように構成される。

30

【0023】

さらに、好適には、PCL点群ライブラリは、

取得した爪平面点群 $CloudNail-p$ をエッジ処理して、エッジを平滑化するのに用いる、エッジ処理モジュールをさらに含む。

【発明の効果】

【0024】

従来技術と比べて、本考案は次の有益な効果を有する。

【0025】

1つ目に、本発明は、三次元点群によって弧度を有する爪点群を取得し、さらにそれを平坦化する。平坦化後、爪点群の外周輪郭点群形状は、弧度を有する爪の外周輪郭とより一層フィットする。

40

【0026】

2つ目に、本発明は、変換後の指点群を所望の姿勢及び位置上に転換することにより、指点群の計算と処理が格段に便利になる。

【0027】

3つ目に、本発明は、クラスター拘束条件を追加することにより、指点群中の点が爪点群に属するか否かを精密に計算し、これにより一層効果的に爪曲面点群を指点群から分割する。

【0028】

4つ目に、本発明は、最初に爪甲エリアを判定し、次に爪甲エリア中から1つの点をシ

50

ード点として選んで拡張プロセスを開始することにより、拡張課程で取得するものが爪曲面点粉であることを保証することができ、拡張プロセスが1回で済むため、拡張アルゴリズムにかかる時間が減り、アルゴリズムの効率が向上する。

【0029】

5つ目に、本発明は、爪の成長方向に沿って層分割を行ったうえで、各層に対して平坦化を行うことにより、取得した爪平面点群が幅方向上で平坦化される。この爪平面点群の輪郭点群に基づきレンダリングして得る形状輪郭は、爪の側面寸法においてより一層爪にフィットすることができる。

【0030】

6つ目に、本発明は、取得した爪平面点群をエッジ処理して、エッジを平滑化することにより、得られる爪甲型の美観性が向上する。

10

【図面の簡単な説明】

【0031】

【図1】第1実施形態中の爪輪郭抽出方法の工程概念図である。

【図2】第1実施形態中の爪輪郭抽出システムのブロック図である。

【図3】第1実施形態中の爪輪郭抽出方法の工程概念図である（点群分割サブ工程）。

【図4】第2実施形態中の爪輪郭抽出方法の工程概念図である（点群方向矯正サブ工程）。

【図5】第2実施形態中の爪輪郭抽出システムのブロック図である（矯正モジュール）。

【図6】第2実施形態中の爪輪郭抽出方法の工程概念図である（チェックサブ工程）。

【図7】第2実施形態中の爪輪郭抽出システムのブロック図である（チェックモジュール

20

）。

【図8】検索半径 R_{rs} にある近傍点群の法線概念図である。

【図9a】検索半径 R_{rs} に基づく指点群の分割結果である。

【図9b】縮小した検索半径 R_{rs} に基づく指点群の分割結果である。

【図10】第3実施形態中の爪輪郭抽出方法の工程概念図である。

【図11】第3実施形態の検索半径 R_{rs}' にある近傍点群の法線概念図である。

【図12】第3実施形態の検索半径 R_{rs}' にある指点群の分割結果である。

【図13】第3実施形態中の爪輪郭抽出方法の工程概念図である（シード点 S_0 選択サブ工程）。

【図14】第3実施形態中の爪輪郭抽出システムのブロック図である（選択サブモジュール）。

30

【図15】第3実施形態中の爪甲エリア下の構造概念図である。

【図16】第4実施形態中の爪曲面点群を爪の成長方向に沿って層分割した構造概念図である。

【図17】第4実施形態中の各層における点群上の点を同一平面下に統一した構造概念図である。

【図18】第4実施形態中の同一平面下に統一した点群を平坦化した構造概念図である。

【図19】第5実施形態中の爪輪郭抽出方法の工程概念図である。

【図20】第5実施形態中の爪輪郭抽出システムのブロック図である。

【発明を実施するための形態】

40

【0032】

以下、図に基づき本願の実施形態中の技術案について説明を行う。

【0033】

本明細書中の具体的な例は、当業者が本願の実施形態をより良く理解できるようにするためのものに過ぎず、本願の実施形態の範囲を限定するものではないことを理解されたい。

【0034】

また、本願の各種実施形態中、各工程の番号の大きさは実行順序の前後を意味してはならず、各工程の実行順序はその機能及び内部論理によって決定すべきであり、本願の実施形態の実施する工程に対して如何なる限定も構成しないことを理解されたい。

【0035】

50

また、本明細書が説明する各種の実施形態は、単独で実施することも、組み合わせて実施することもでき、本願の実施形態はこれについて限定しない。

【0036】

別段の説明がない限り、本願の実施形態が用いるすべての技術及び科学用語は、本願の当業者が一般的に理解するのと同じ意味を有する。本願において使用する用語は、具体的な実施形態を説明する目的で使用したに過ぎず、本願の範囲の限定を意図していない。本願が使用する「及び/又は」という用語は、関連して列挙される1つ以上の項目の任意及びすべての組み合わせを含む。

【0037】

第1実施形態

本願の第1実施形態は爪輪郭の抽出方法を提供するが、それは特に三次元点群の結果に基づいて爪輪郭を抽出する方法に関する。

【0038】

具体的には、図1を参照して、第1実施形態中の爪輪郭の抽出方法は、以下の工程：

対象指に対して三次元走査を行い、対象指の指三次元データDataFingerを取得し、指三次元データDataFingerは少なくとも、すべての爪の爪三次元データDataNailを含んでいる、データ収集工程と、

指三次元データDataFingerを三次元点群データに変換して、指点群CloudFingerを取得し、指点群CloudFingerは少なくとも、すべての爪の爪曲面点群CloudNailsを含んでいる、データ点群処理工程と、

指点群CloudFingerに対して分割を行い、指点群CloudFingerから爪曲面点群CloudNailsを取得する、爪点群取得工程と、

爪曲面点群CloudNailsを表面積を変えずに平坦化して、平面状態における爪平面点群CloudNail-pを取得する、爪点群平坦化工程と、

爪平面点群CloudNail-pの輪郭に基づき輪郭点群CloudContourの形状をレンダリングし、爪の爪甲型を取得する、爪輪郭レンダリング工程と、を含む。

【0039】

従来技術と比べて、第1実施形態が提供する爪輪郭の抽出方法が取得するのは、表面が非平坦で弧度を有する爪の爪曲面点群CloudNailsを平坦化してから形成される爪平面点群CloudNail-pであり、平坦化後の爪平面点群CloudNail-pの外周輪郭点群CloudContourの形状は、弧度を有する爪の外周輪郭とより一層フィットする。

【0040】

このように、平坦化した爪平面点群CloudNail-pの外周輪郭点群CloudContourの形状に基づきレンダリングを行い、平面における爪甲型を取得する。当該爪甲型を成形型として製造したネイルチップによれば、製造されたネイルチップは爪の表面弧度に適合することができ、それを使用者の爪表面に貼ると、ネイルチップの外周輪郭も爪の外周輪郭と一致し、貼付けが不完全になったり、はみ出したりすることがない。

【0041】

言い換えると、第1実施形態が提供する爪輪郭の抽出方法は、爪表面の三次元画像の取得後にそれを点群処理した上で、点群化後の爪曲面点群CloudNailsを分割し且つ平坦化して爪平面点群CloudNail-pとし、爪平面点群CloudNail-pの外周輪郭に基づき輪郭点群CloudContourの形状をレンダリングして、平面の爪甲型を取得する。従来技術中の二次元画像を用いて爪甲型を取得するものと比べて、第1実施形態が抽出する爪輪郭の外周輪郭形状は、使用者の爪の外周輪郭と完全に一致しており、これに基づき爪甲型をレンダリングして製造されるネイルチップは、使用者の爪上に完璧に貼付けることができ、貼付けが不完全になったり、はみ出したりすることがない。

【0042】

図2を参照されたい。第1実施形態は爪輪郭の抽出システムをさらに提供する。この爪輪郭の抽出システムは、上述の爪輪郭の抽出方法を利用して爪の輪郭を取得することがで

10

20

30

40

50

きる。具体的には、爪輪郭の抽出システムは、順に通信接続される、データ収集装置1、データ変換装置2、PCL3、平坦化モジュール5及びレンダリングモジュール6を含み、PCL3中には分割モジュール4が含まれる。

【0043】

具体的には、図1と図2に示す通り、データ収集工程中、データ収集装置1を用いて対象指に対して三次元走査を行い、対象指の指三次元データDataFingerを取得する。対象指の爪輪郭を抽出するために、本データ収集装置1はデータ収集工程において、対象指の三次元走査範囲に対象指の爪全体及び爪周囲の皮膚を少なくとも含む。これにより、指三次元データDataFingerには少なくとも爪全体の爪三次元データDataNailが含まれ、通常はさらに爪周囲の皮膚の皮膚三次元データDataSkinを少なくとも含む。

10

【0044】

データ収集装置1は、レーザーレーダー、深度カメラを利用するか、又はレーザーレーダー、深度カメラを主に利用して作られた三次元スキャナ、構造化照明カメラ、二眼カメラなどでよい。これにより、データ収集装置1はデータ収集工程において、三角測距原理、TOF測距原理などを利用してレーダーが対象指に対して三次元走査を行うことができ、また、構造化照明測距原理、双眼視覚測距原理及びTOF測距原理などを利用して深度カメラが対象指に対して三次元走査を行うこともでき、迅速且つ正確に指三次元データDataFingerを取得することができる。

【0045】

上に挙げた三角測距原理、TOF測距原理、構造化照明測距原理、双眼視覚測距原理などの三次元走査測距原理はいずれも従来技術であり、日常において広く使用されているため、ここではその原理について詳述しない。

20

【0046】

なお、三角測距原理、TOF測距原理、構造化照明測距原理、双眼視覚測距原理などは実際の応用において各々にメリット・デメリットがあり、測定環境、光ファイバー、距離などに対する要求事項や、実際の測定精度、製品コストなどもそれぞれ異なるため、本実施形態の実際の応用においては、データ収集装置1の実際の種類に基づいて選択する必要がある。

【0047】

また具体的には、図1と図2に示す通り、データ点群処理工程中、データ変換装置2は、データ収集装置1が収集して送信する指三次元データDataFingerを受信した後、指三次元データDataFingerをPCD形式の三次元点群データに変換し、これにより指点群CloudFingerを取得する。具体的には、データ変換装置2は、指三次元データDataFingerの形式に対して変換を行い、PCL点群ライブラリ3がサポートしているデータ形式に変換する。それは例えばPCD、PLYなどの形式であり、実際の形式の選択は、データ収集装置1が収集するデータ形式や、分割モジュール4が処理をサポートするデータ形式などに基づき決定する。指三次元データDataFingerには爪の爪三次元データDataNail及び爪周囲の皮膚の皮膚三次元データDataSkinが少なくとも含まれるため、指三次元データDataFingerの形式を例えばPCD形式などのPCL点群ライブラリ3がサポートするデータ形式に変換した後の指点群CloudFingerには、表面が非平坦で湾曲弧度を有する爪の爪曲面点群CloudNails及びその爪周囲の皮膚の皮膚点群CloudSkinが少なくとも含まれる。

30

40

【0048】

もちろん、第1実施形態の他の装置の例において、データ収集装置1にデータ変換機能が搭載されている場合には、データ点群処理工程においてデータ収集装置1にデータ変換装置2を兼任させ、収集した指三次元データDataFingerのデータ形式を直接PCD形式に変換することによって指点群CloudFingerを取得してもよい。

【0049】

PCL(Point Cloud Library)3(点群ライブラリ3)は、前人の

50

点群に関する研究を吸収しそれをベースに構築されたクロスプラットフォームであり、C++言語によるオープンソースの大規模ライブラリである。指三次元データDataFingerの形式をPCD形式に変換する場合を例にすると、PCD形式のデータはPCL点群ライブラリ3がサポートする形式であり、指三次元データDataFingerをPCD形式に変換した後の指点群CloudFingerをPCL点群ライブラリ3中に入力すると、PCL点群ライブラリ3中の関連アルゴリズムを用いて指点群CloudFingerに対して後に続く処理を実行することができる。

【0050】

最初に取得する指三次元データDataFingerには爪全体の爪三次元データDataNail及び爪周囲の皮膚の皮膚三次元データDataSkinが含まれるため、指三次元データDataFingerを点群処理し、さらに変換して取得する指点群CloudFingerには、爪の爪曲面点群CloudNails及び爪周囲の皮膚の皮膚点群CloudSkinが含まれる。爪輪郭の抽出においては、爪曲面点群CloudNailsと皮膚点群CloudSkinを分離することによって必要とする爪曲面点群CloudNailsを抽出する必要がある。

10

【0051】

図3に示すように、これに基づき、爪点群取得工程は、具体的には点群分割サブ工程を含む。点群分割サブ工程において、PCL点群ライブラリ3中の分割モジュール4を用いて指点群CloudFingerに対して分割を行い、指点群CloudFinger中の爪曲面点群CloudNailsと皮膚点群CloudSkinを分割して、弧度を有する爪の爪曲面点群CloudNailsを抽出する。

20

【0052】

より詳細には、図2と図3とに示す通り、第1実施形態中の点群分割サブ工程では、領域拡張分割法(region growing-segmentation)を用いて指点群CloudFingerの分割と校正を行う。具体的な分割方式は以下の通りである。

入力：

点群{P}を入力する

検索半径 R_{rs} を入力する

曲率閾値 cur_{thres} 、角度閾値 $thres$ を入力する

出力：

点群分割後のエリアクラスターリストRを出力する

30

初期化：

エリアクラスターリストRを初期化する

点リスト{A}を初期化し{A} = {1, ..., P_size}、点群のすべての点をリスト{A}にプッシュする

入力点群の各点の法線ベクトル{N}を計算する

入力点群の各点の曲率{c}を計算する

アルゴリズムループ：

リスト{A}が空でない限り：

現在エリアリスト{R_c}と現在シードリスト{S_c}をクリアする

40

リスト{A}中で現在の曲率最小点 P_{min} を見つけ、且つその曲率最小点 P_{min} をリスト{A}から除去する{A} = {A} \setminus P_{min}

同時に、リスト{A}から除去した前述の曲率最小点 P_{min} を現在エリアリスト{R_c}と現在シードリスト{S_c}にプッシュする

{S_c} = {S_c} ∪ P_{min}、{R_c} = {R_c} ∪ P_{min}

現在シードリスト{S_c}中の各点{S_c(i)}に対し、以下をループ：

k-d treeにより点{S_c(i)}のすべての近傍点{N_{bc}}を迅速に探索する

{N_{bc}}中のすべての近傍点の各点{N_{bc}(j)}に対し、以下をループ：

リスト{N_{bc}}に条件を満たす点{N_{bc}(j)}があるかどうかを一つ一つチェック

50

くし、点 $\{S_c(i)\}$ と現在近傍点 $\{Nb_c(j)\}$ の法線ベクトルの夾角が角度閾値 $thres$ よりも小さい場合、即ち $\cos^{-1}(S_c(i), Nb_c(j)) < thres$ である場合には：

条件を満たす点 $\{Nb_c(j)\}$ を現在エリアリスト $\{R_c\}$ に加え、

$\{R_c\} = \{R_c\} \cup Nb_c(j)$

条件を満たす点 $\{Nb_c(j)\}$ をリスト $\{A\}$ から除去する

$\{A\} = \{A\} - Nb_c(j)$

条件を満たす点 $\{Nb_c(j)\}$ の曲率が曲率閾値 cur_thres よりも小さい場合には、その条件を満たす点 $\{Nb_c(j)\}$ を現在シードリスト $\{S_c\}$ にプッシュする

$\{S_c\} = \{S_c\} \cup Nb_c(j)$

分割して出力したエリアクラスターリスト R に現在エリアリスト $\{R_c\}$ を加える

$\{R\} = \{R\} \cup R_c$

【0053】

このように、指点群 $CloudFinger$ を分割し、皮膚点群 $CloudSkin$ を除去し、爪の爪曲面点群 $CloudNail-s$ を拡張し且つ抽出することができる。

【0054】

具体的には、図1と図2に示す通り、爪点群平坦化工程中、分割モジュール4と通信接続された平坦化モジュール5を用いて、爪点群取得工程においてデータ変換装置2を用いて取得した爪曲面点群 $CloudNail-s$ を表面積を変えずに展開して平坦化することにより、平面状態における爪平面点群 $CloudNail-p$ を取得する。

【0055】

最後に、爪輪郭レンダリング工程中、平坦化モジュール5と通信接続されたレンダリングモジュール6を用いて爪平面点群 $CloudNail-p$ を受信した後、爪平面点群 $CloudNail-p$ の輪郭点群 $CloudContour$ の形状をレンダリングして、爪の爪甲型を取得する。

【0056】

後に続く製造において、前述の方法及びシステム中のエッジの輪郭点群 $CloudContour$ の形状に基づきレンダリングを行って取得した爪のエッジ輪郭は、後にそのエッジ輪郭の形状に基づきネイルチップの型を作製し、ネイルチップの製造や、爪形状の研究などの基盤とすることができる。

【0057】

なお、本願が述べる爪は、手の爪でもよいし、足の爪でもよい。

【0058】

第2実施形態

本願の第2実施形態は、爪輪郭の抽出方法及びそのシステムを提供する。当該抽出方法及びそのシステムは、いずれも第1実施形態中の爪輪郭の抽出方法及びそのシステムの設計及び実施に基づくものであり、実際の工程又は構造は似ているものの、異なる点も幾らかあり、以下では異なる点について重点的に説明を行う。異なる点以外の詳細な説明については、いずれも第1実施形態中と同じであると見なすことができ、詳述しない。

【0059】

PCL点群ライブラリ3は、可視化ウィンドウを利用して座標系 (x, y, z) をプロットすることができ、座標系は三次元の世界における点群の位置を特定する。図4と図5を参照して、第2実施形態では、後に続く指点群 $CloudFinger$ の処理を容易にして正確且つ高効率の処理結果を取得するために、データ点群処理工程中に点群方向矯正サブ工程を含み、爪輪郭の抽出システムは矯正モジュール7をさらに含む。点群方向矯正サブ工程において、矯正モジュール7を用いて、データ点群処理工程中のデータ変換装置2が変換して取得する指点群 $CloudFinger$ の姿勢と位置の転換を行い、出力指点群 $CloudFinger-o$ を取得して、出力指点群 $CloudFinger-o$ の姿勢及び位置を所望のプリセットの姿勢及び位置上に転換する。

【0060】

10

20

30

40

50

矯正モジュール7は、独立して設置してデータ変換装置2と通信接続してもよいし、データ変換装置2内に統合するか、又はデータ変換装置2の一部であってもよい。

【0061】

具体的には、所望するプリセットの姿勢及び位置は、指点群 $CloudFinger$ を変えることにより出力指点群 $CloudFinger-o$ の質量中心を PCL 点群ライブラリ3中の座標系の原点と重複させ、指の先端が座標系中の x 軸の正の向きを指し、指の背面が座標系中の z 軸の負の向きを指すようにしたものである。

【0062】

より具体的には、指の先端とは、爪の生え際により近い一端をいい、指の先端が指す方向は爪の成長方向であり、指の背面とは、指の爪が生えている側のことをいい、指の背面が指す方向は指の腹から爪に向かう方向である。

10

【0063】

この要件に基づき、まずデータ点群処理工程において、データ点群処理工程中で得た指三次元データ $DataFinger$ を変換することにより指点群 $CloudFinger$ (以下、入力指点群とも呼ぶ) を取得してから、点群方向矯正サブ工程に進み、矯正モジュール7を用いて入力指点群 $CloudFinger$ の姿勢と位置の転換を行い、これにより所望する姿勢と位置上の指点群 $CloudFinger-o$ (以下は統一して出力指点群と呼ぶ) を新たに得る。

【0064】

本実施形態中、点群方向矯正サブ工程中の矯正モジュール7は、 tf の回転行列によって入力指点群 $CloudFinger$ の姿勢と位置を転換するものであり、入力指点群 $CloudFinger$ を出力指点群 $CloudFinger-o$ に転換する具体的な公式の手順は以下の通りである。

20

【0065】

【数1】

$$CloudFinger-o = CloudFinger * tf$$

$$tf = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$$

$$R = [\vec{v}_1 \quad \vec{v}_2 \quad \vec{v}_3]^{-1}$$

$$T = \begin{bmatrix} - & x_c \\ - & y_c \\ - & z_c \end{bmatrix}$$

30

ここで、

tf は回転行列であり、

$CloudFinger$ はデータ点群処理工程中で得た指三次元データ $DataFinger$ を変換して取得した入力指点群であり、

$CloudFinger-o$ は入力指点群 $CloudFinger$ を tf で回転させた後の出力指点群であり、

R は回転行列の姿勢回転量であり、

T は回転行列の平行移動量であり、

40

【0066】

【数2】

$$\vec{v}_1, \vec{v}_2, \vec{v}_3$$

は入力指点群 $CloudFinger$ に対して PCA (Principal Components Analysis、主成分分析) を行い、共分散行列を求めて得た x、y 及び z 軸方向の特徴ベクトルであり、

C は入力指点群 $CloudFinger$ の質量中心であり、

50

x_c 、 y_c 、 z_c のそれぞれは入力指点群CloudFingerの質量中心座標である。

【0067】

これにより、 t_f で回転変換させた後に取得する出力指点群CloudFinger-oは、質量中心が座標の原点と重複し、指の先端がx軸の正の向きを指し、指の背面がz軸の負の向きを指すという要件を満たすものとなる。プリセットの姿勢と位置に一致する出力指点群CloudFinger-oは、後に続く出力指点群CloudFinger-oに対する処理に対して良好な前提条件を提供することができる。

【0068】

上述の t_f で回転変換させた後に取得する出力指点群CloudFinger-oは、指の背面が座標系中のz軸の正の向きを指すという状況が存在する可能性もある。これに基づき、図6と図7を参照すると、点群方向矯正サブ工程後において、データ点群処理工程はチェックサブ工程をさらに含み、爪輪郭の抽出システムはチェックモジュール8をさらに含む。チェックサブ工程では、チェックモジュール8を用いて変換後の指点群中の指の背面が指す方向に対してチェックを実行し、それが座標系中のz軸の負の向きを指しているか否かをチェックする。

【0069】

チェックモジュール8は矯正モジュール7と通信接続され、チェックモジュール8は、独立して設置してデータ変換装置2と通信接続してもよいし、データ変換装置2内に統合するか、又はデータ変換装置2の一部であってもよい。

【0070】

もしもチェックモジュール8が変換後の指点群中の指の背面が座標系中のz軸の正の向きを指していることを確認した場合、矯正モジュール7を用いて変換後の指点群をx軸まわりに180度回転させた上で出力指点群CloudFinger-oとすることにより、後続の出力に提供する。

【0071】

もしもチェックモジュール8が変換後の指点群中の指の背面が座標系中のz軸の負の向きを指していることを確認した場合、変換後の指点群をそのまま出力指点群CloudFinger-oとすることにより、後続の出力に提供する。

【0072】

このようにして、出力指点群CloudFinger-o中の指の背面が座標系中のz軸の負の向きを指すことを保証できる。

【0073】

バウンディングボックスは最小境界矩形とも呼ばれる、離散点集合の最適な包囲空間を求めるアルゴリズムである。基本的な考えとしては、複雑な幾何学的対象を、体積がやや大きく特性がシンプルな幾何学体(バウンディングボックスと呼ぶ)で近似的に代替するというものである。PCL点群ライブラリ3は、点群のバウンディングボックスを取得し、且つバウンディングボックスを用いてデータを計算し処理する。

【0074】

具体的には、上述のチェックサブ工程の具体的なチェック工程は以下の通りである。

【0075】

t_f で回転変換させた後に取得する指点群CloudFinger中、z座標の平均値

z

と、バウンディングボックスの中央値 z_m 計算し、判定を行う。

【0076】

【数3】

10

20

30

40

50

$$\bar{z} = \frac{\sum_n z_1 + z_2 + \dots + z_n}{n}$$

$$z_m = \frac{z_{max} + z_{min}}{n}$$

もしも、

【 0 0 7 7 】

【 数 4 】

$$\bar{z} - z_m < 0$$

10

である場合は、 t f で回転変換させた後に取得する指点群を x 軸まわりに 1 8 0 度回転させ、

もしも、

【 0 0 7 8 】

【 数 5 】

$$\bar{z} - z_m \geq 0$$

である場合は、 t f で回転変換させた後に取得する指点群を変えずに保持する。

ここで、

\bar{z}

20

は指点群の z 軸座標の平均値であり、

z_m は点群のバウンディングボックスの z 軸における中央値を指し、

z_{max} は指点群の座標の z 軸における最大値であり、

z_{min} は指点群の座標の z 軸における最小値である。

【 0 0 7 9 】

より具体的には、もしも、

【 0 0 8 0 】

【 数 6 】

$$\bar{z} - z_m < 0$$

30

である場合は、 t f で回転変換させた後に取得する指点群の指の背面が座標系中の z 軸の正の向きを指していることを表す。そのため、 t f で回転変換させた後に取得する指点群を x 軸まわりに 1 8 0 度回転させてから、 1 8 0 度回転させた後の指点群を出力指点群 CloudFinger-o とする。

【 0 0 8 1 】

もしも、

【 0 0 8 2 】

【 数 7 】

$$\bar{z} - z_m \geq 0$$

40

である場合は、 t f で回転変換させた後に取得する指点群 CloudFinger の指の背面が座標系中の z 軸の負の向きを指していることを表す。そのため、 t f で回転変換させた後に取得する指点群を変えずに保持し、そのまま出力指点群 CloudFinger-o とする。

【 0 0 8 3 】

第 3 実施形態

本願の第 3 実施形態は、爪輪郭の抽出方法及びそのシステムを提供する。当該抽出方法及びそのシステムは、いずれも第 1 又は第 2 実施形態中の爪輪郭の抽出方法及びそのシス

50

テムの設計及び実施に基づくものであり、実際の工程又は構造は似ているものの、異なる点も幾らかあり、以下では異なる点について重点的に説明を行う。異なる点以外の詳細な説明については、いずれも第1及び第2実施形態中と同じであると見なすことができ、詳述しない。

【0084】

図8に示す通り、爪曲面点群 $Cloud_{Nail-s}$ と皮膚点群 $Cloud_{Skin}$ を第1実施形態中の点群分割サブ工程及び分割モジュール4によって分離するとき、上述のクラスター拘束条件下において、爪曲面点群 $Cloud_{Nail-s}$ の分割・拡張は、隣接する2つの点における法線ベクトルの夾角 θ 及び点の曲率閾値 cur_{thres} のみに依存するため、生じる誤差がやや大きくなる。

10

【0085】

特に、指の爪溝がはっきりしない、又は指点群 $Cloud_{Finger}$ の点群モデルのノイズが顕著である場合には、指点群 $Cloud_{Finger}$ 中の点群の起伏がそれほど大きくなる。即ち、爪曲面点群 $Cloud_{Nail-s}$ と皮膚点群 $Cloud_{Skin}$ との境目エリアの起伏がやや小さいとき、皮膚点群 $Cloud_{Skin}$ 中の近傍点群の法線にもやや大きな誤差が生じ、これによって皮膚点群 $Cloud_{Skin}$ 中の点が誤って現在シードリスト $\{S_c\}$ 中にプッシュされてしまい、且つ最終的には分割後に出力されるエリアクラスターリスト R に追加されてしまう。つまりは、本来なら皮膚点群 $Cloud_{Skin}$ 中に属する点が誤って爪曲面点群 $Cloud_{Nail-s}$ 中に追加されてしまう。

【0086】

20

図9aに示す通り、具体的には、検索半径 R_{rs} において、皮膚点群 $Cloud_{Skin}$ 中の隣接する点と近傍点の法線ベクトルの夾角 θ_1 と θ_2 が同時に角度閾値 θ_{thres} より小さいことを満たすことにより、前述の以下条件が満たされる。

条件を満たす点 $\{Nb_c(j)\}$ の曲率が曲率閾値 cur_{thres} よりも小さい場合には、その条件を満たす点 $\{Nb_c(j)\}$ を現在シードリスト $\{S_c\}$ にプッシュし、
 $\{S_c\} \leftarrow \{S_c\} \cup Nb_c(j)$

これにより以下の動作が実行される。

分割して出力したエリアクラスターリスト R に現在エリアリスト $\{R_c\}$ を加え、
 $\{R\} \leftarrow \{R\} \cup \{R_c\}$ とする。

【0087】

30

このようにして、皮膚点群 $Cloud_{Skin}$ 中の点群がエリアクラスターリスト R に線り入れられ、PCL点群ライブラリ3中の爪曲面点群 $Cloud_{Nail-s}$ のエリアが起伏部の外側、つまり皮膚部分まで拡張され、これにより図9a中の状況が出現し、最終的に爪の分割に失敗してしまう。

【0088】

こうした状況について、本実施形態の幾つかの実施形態において検索半径 R_{rs} を縮小する方法を採用しており、これによって精密化された計算を行い、より正確な n 点の法線ベクトルを計算して取得することができる。

【0089】

但し、図9bに示す通り、検索半径 R_{rs} の縮小は n 点の法線ベクトルを正確に計算する助けにはなるものの、指点群 $Cloud_{Finger}$ モデルのノイズが縮小される R_{rs} に対して増大してしまい、相対的に増大したノイズの影響によってPCL点群ライブラリ3中のエリア拡張時のアルゴリズムの誤差が大きくなり、爪溝部分が不適切に拡張されて、結果の精度に影響が出てしまう。

40

【0090】

これに基づき、図10と図11を参照すると、第3実施形態ではこれを改善しており、爪点群取得工程の点群分割サブ工程後において、クラスター拘束サブ工程をさらに設けており、クラスター拘束サブ工程は、分割モジュール4にクラスター拘束条件を追加することによって計算結果の精度を向上させるものである。

【0091】

50

クラスター拘束サブ工程は、以下の工程を含む。

【0092】

検索半径 R_{rs} をベースとして検索半径 R_{rs}' を入力し、検索半径 R_{rs}' は検索半径 R_{rs} よりも小さく、検索半径 R_{rs}' にある点と近傍点との法線ベクトルの夾角 R_{rs} を取得し、且つ角度閾値 $thres_2$ を判定条件として追加して判定を行い、

もしも現在点が $R_{rs} - R_{rs}' > thres_2$ である場合は、拡張を停止し、

もしも現在点が $R_{rs} - R_{rs}' \leq thres_2$ である場合は、その現在点を現在エリアリスト $\{R_c\}$ にまとめる。

ここで、

R_{rs} は検索半径 R_{rs} にある点と近傍点との法線ベクトルの夾角であり、

R_{rs}' は検索半径 R_{rs}' にある点と近傍点との法線ベクトルの夾角である。

10

【0093】

図11に示すように、2つの点 n_1 、 n_2 を例にすると、点 n_1 は $\alpha_1 = \alpha_1' - \alpha_1$ であり、 $\alpha_1 > thres_2$ であり、 n_1 は現在エリアリスト $\{R_c\}$ にまとめられず、点 n_2 は $\alpha_2 = \alpha_2' - \alpha_2$ であり、 $\alpha_2 \leq thres_2$ であり、 n_2 は現在エリアリスト $\{R_c\}$ にまとめられる。

ここで、

α_1 は検索半径 R_{rs} にある点 n_1 と近傍点との法線ベクトルの夾角であり、

α_1' は検索半径 R_{rs}' にある点 n_1 と近傍点との法線ベクトルの夾角であり、

α_2 は検索半径 R_{rs} にある点 n_2 と近傍点との法線ベクトルの夾角であり、

α_2' は検索半径 R_{rs}' にある点 n_2 と近傍点との法線ベクトルの夾角である。

20

【0094】

このようにして、クラスター拘束条件を追加し、点と現在近傍点との法線ベクトルの夾角が角度閾値 $thres$ よりも小さいことを満たしているときの点に対して、検索半径 R_{rs} を縮小してからさらにもう一度クラスター拘束を実行し、その点の検索半径 R_{rs} と検索半径 R_{rs}' における近傍点との法線ベクトルの夾角が角度閾値 $thres_2$ よりも小さい場合にのみそれを爪曲面点群 $Cloud_{Nail-s}$ 中の点であると判定し、且つ現在エリアリスト $\{R_c\}$ にまとめ、当該条件を満たさない点は除去することにより、爪曲面点群 $Cloud_{Nail-s}$ の誤った拡張が効果的に低減され、且つ図12に示す通り、指点群 $Cloud_{Finger}$ 中から爪曲面点群 $Cloud_{Nail-s}$ が正確に抽出・分割されるという分割効果が得られる。

30

【0095】

ここで、検索半径 R_{rs}' は検索半径 R_{rs} よりも格段に小さく、具体的には、検索半径 R_{rs}' は検索半径 R_{rs} の1%~80%に設定することができ、より好適には、検索半径 R_{rs}' は検索半径 R_{rs} の10%~30%である。このようにして、PCL点群ライブラリ3中の作業量を合理的に増やし、求める分割精度が満たされる状況下で指点群 $Cloud_{Finger}$ 中の点群に対して分割・拡張を実行し、そこからより正確な爪曲面点群 $Cloud_{Nail-s}$ を分割して取得することができる。

【0096】

より詳細には、分割モジュール4が前述の点群分割サブ工程及びクラスター拘束サブ工程の動作を実行し、爪曲面点群 $Cloud_{Nail-s}$ のエリア拡張を行うとき、リスト $\{A\}$ の残りの点が空になるまで、リスト $\{A\}$ 中の曲率が最も小さい点を繰り返し抽出してシード点 S_0 として拡張プロセスを開始する。これにより、入力指点群 $Cloud_{Finger}$ のすべてのエリアにおいてすべての拡張が完了し、入力指点群 $Cloud_{Finger}$ から爪曲面点群 $Cloud_{Nail-s}$ が取得される。

40

【0097】

但し、上述の工程中の分割モジュール4は爪曲面点群 $Cloud_{Nail-s}$ を取得できるものの、リスト $\{A\}$ 中の曲率が最も小さい点を繰り返し抽出してシード点 S_0 として拡張プロセスを開始することになり、これらのプロセスは指点群中の不必要な皮膚点群 $Cloud_{skin}$ も取得してしまううえに、拡張プロセスが繰り返されて冗長になり、爪曲

50

面点群 $Cloud_{Nail-s}$ の拡張効率が極めて低くなる。

【0098】

これについては、図13と図14に示すように、第3実施形態は、爪点群取得工程は、点群分割サブ工程前において、シード点 S_0 選択サブ工程をさらに含み、爪輪郭の抽出システムは分割モジュール4と通信接続される選択モジュール9をさらに含む。シード点 S_0 選択サブ工程において、選択モジュール9を用いて先に爪の爪甲面エリア $Nrange(x, y)$ を判定して取得した上で、その爪甲面エリア $Nrange(x, y)$ から1点をシード点 S_0 として選択し、分割モジュール4がそのシード点 S_0 によって拡張プロセスを開始する。

【0099】

このようにして、拡張プロセスが爪曲面点群 $Cloud_{Nail-s}$ の爪甲面エリア $Nrange(x, y)$ から開始して実行され、1回の拡張プロセスだけで拡張プロセスが確実に爪曲面点群 $Cloud_{Nail-s}$ を取得できるようになり、拡張アルゴリズムにかかる時間が減り、アルゴリズムの効率が向上する。

【0100】

より具体的には、爪甲面エリア $Nrange(x, y)$ の決定は、前述の入力指点群を tf で回転変換させた後に取得される座標軸と直交する出力指点群 $Cloud_{Finger-o}$ に基づき行われる。図15に示す通り、陰影部分は判定された爪甲面エリア $Nrange(x, y)$ であり、それは x 軸と y 軸の1つの範囲値が囲む1つの矩形範囲であり、その式は以下の通りである。

$$Nrange(x, y) = \{x \text{ (} x_{lower \ limit}, x_{upper \ limit} \text{)}, y \text{ (} y_{lower \ limit}, y_{upper \ limit} \text{)}\}$$

ここで、

- x_{min} は爪曲面点群 $Cloud_{Nail-s}$ の x 軸上の最小値であり、
- x_{max} は爪曲面点群 $Cloud_{Nail-s}$ の x 軸上の最大値であり、
- y_{min} は爪曲面点群 $Cloud_{Nail-s}$ の y 軸上の最小値であり、
- y_{max} は爪曲面点群 $Cloud_{Nail-s}$ の y 軸上の最大値である。

【0101】

この矩形範囲は、爪曲面点群 $Cloud_{Nail-s}$ に基づき、その x 軸と y 軸における幅の設定比率を選択して決定することができ、好適には x 軸の両側、 y 軸の両側で対称になるように設定される。本実施形態の好適な方式では、爪甲面エリア $Nrange(x, y)$ 全体における y 軸幅の $1/2$ 、 x 軸幅の $3/4$ 部分を矩形の爪甲面エリア $Nrange(x, y)$ の範囲として確定する。

即ち、

$$x_{lower \ limit} = (x_{min} \times (4 - (3 - a)) + x_{max} \times (3 - a)) / 4$$

$$x_{upper \ limit} = (x_{min} \times (4 - (3 + a)) + x_{max} \times (3 + a)) / 4$$

$$y_{lower \ limit} = (y_{min} \times (2 - (1 - b)) + y_{max} \times (1 - b)) / 2$$

$$y_{upper \ limit} = (y_{min} \times (2 - (1 + b)) + y_{max} \times (1 + b)) / 2$$

であり、

ここで、

a は x 軸方向の矩形幅半径であり、

b は y 軸方向の矩形幅半径である。

【0102】

このようにして、爪甲面エリア $Nrange(x, y)$ の決定後、PCL点群ライブラリ3中の分割モジュール4がその爪甲面エリア $Nrange(x, y)$ の点を分離したうえで、その中から随意に1つの点を最初に拡張するシード点 S_0 として選ぶことにより、この1回のみ拡張が確実に爪甲面エリア $Nrange(x, y)$ 内に発生するように保証することができ、爪曲面点群 $Cloud_{Nail-s}$ の拡張プロセスが簡単なうえに1回のみとなり、爪曲面点群 $Cloud_{Nail-s}$ の拡張効率が大幅に向上される。

【0103】

10

20

30

40

50

第4実施形態

本願の第4実施形態は、爪輪郭の抽出方法及びそのシステムを提供する。当該抽出方法及びそのシステムは、いずれも第1～第3実施形態中の爪輪郭の抽出方法及びそのシステムの設計及び実施に基づくものであり、実際の工程又は構造は似ているものの、異なる点も幾らかあり、以下では第1～第3実施形態と異なる点について重点的に説明を行う。異なる点以外の詳細な説明については、いずれも第1～第3実施形態中と同じであると見なすことができ、詳述しない。

【0104】

爪点群平坦化工程において、平坦化モジュール5を用いて爪曲面点群 $Cloud_{Nail-s}$ を表面積を変えずに平坦化して爪平面点群 $Cloud_{Nail-p}$ とするとき、爪曲面点群 $Cloud_{Nail-s}$ が弧度を有するため、爪平面点群 $Cloud_{Nail-p}$ を成型型として作られたネイルチップは、完璧には爪に貼付けることができない。

10

【0105】

この点を改善するために、第4実施形態の爪点群平坦化工程では、図16に示す通り、先に平坦化モジュール5を用いて爪曲面点群 $Cloud_{Nail-s}$ をx軸に沿って横向きに分割してn層の点群 L_i とし、次に各層の点群 L_i を全体として等しい長さに平坦化し、これによって平面状態における爪平面点群 $Cloud_{Nail-p}$ を取得する。

【0106】

各層の点群 L_i の点群については、

$$L_i = \{ p \in Cloud_{Nail-s} \mid p_x \leq limitL_i \ \& \ p_x \geq limitR_i \}$$

20

ここで、

L_i はx軸に沿って横向きに分割した後の現在層の点群 L_i であり、

p は現在層の点群 L_i 中の点であり、

p_x は点 p のx軸座標値であり、

$limitL_i$ は現在層の点群 L_i のx軸最小値であり、

$limitR_i$ は現在層の点群 L_i のx軸最大値である。

【0107】

爪曲面点群 $Cloud_{Nail-s}$ 中の各層の点群 L_i におけるすべての点を一つ一つチェックし、点 p のx軸座標値 p_x が $limitL_i \leq p_x \leq limitR_i$ という条件を満たす限り、その点 p を現在層 L_i に繰り入れることにより、現在エリアリスト $\{R_c\}$ にまとめる。

30

【0108】

より具体的には、各層の点群 L_i 中の点のx軸範囲は、以下を有する。

【0109】

【数8】

$$limitL_i = \min x + i \times ld$$

$$limitR_i = \min x + (i+1) \times ld$$

$$ld = (\max x - \min x) / n$$

40

ここで、

$\min x$ は現在層の点群 L_i のx軸極小値であり、

$\max x$ は現在層の点群 L_i のx軸極大値であり、

ld は各層の点群 L_i の幅であり、

n は爪曲面点群 $Cloud_{Nail-s}$ のx軸に沿って横向きに分割する層数である。

【0110】

ここで、層数 n は1つの固定された不変の数字ではなく、爪曲面点群 $Cloud_{Nail-s}$ のx軸上の長さ及び入力する点群の疎密度によって決定される。

【0111】

50

もちろん、爪曲面点群 $Cloud_{Nail-s}$ の x 軸上の長さ及び入力する層数 n によって点群の疎密度を取得することもできる。

【0112】

このようにして、爪の成長方向、即ち x 軸の正の向きに沿って爪曲面点群 $Cloud_{Nail-s}$ に対して層分割を行った上で、各層の点群 L_i に対して等しい長さの平坦化を行い、各層の点群 L_i の平坦化前後の y 軸上における長さは不変とし、これによって取得される爪平面点群 $Cloud_{Nail-p}$ を爪の幅方向、即ち y 軸方向において完璧にマッチングさせることができる。また、爪曲面点群 $Cloud_{Nail-s}$ に対して x 軸の正の向きにおいてのみ平坦化を行うことで、爪の成長方向及び爪の成長方向に沿って相対的に対称な特性に適合させ、曲面を平面に変換することで生じる誤差を低減し、展開時の爪平面点群 $Cloud_{Nail-p}$ の爪側面寸法に対する形状と長さをできる限り変えずに保持することができる。この爪平面点群 $Cloud_{Nail-p}$ の輪郭点群 $Cloud_{Contour}$ に基づきレンダリングして得る形状輪郭は、爪の側面寸法においてより一層爪にフィットすることができ、これに基づき製造されるネイルチップは、爪上に貼付ける際に完全に貼付き、貼付けが不完全になったり、はみ出したりするなどの問題が生じない。

10

【0113】

しかしながら、図17に示すように、爪点群平坦化工程において、各層の点群 L_i 中における各点 p_j の x 軸座標値を統一し、且つその y 軸及び z 軸座標値は変えずに保持し、1つの略円弧形の点群集合 $stand_L_i$ を取得する。

【0114】

簡単に説明すると、次の平坦化計算の便宜のため、異なる層の点群 L_i における各点 p_j をすべて同一層上に統一する。具体的には、各層の点群 L_i をマッピングによって平面 X_{L_i} 上にマッピングし、各層の点群 L_i から前述の略円弧形の点群集合 $stand_L_i$ を得る。

20

$$stand_L_i = f(L_i)$$

$$f: p_{L_i} \rightarrow p_{X_{L_i}}$$

【0115】

次に、図18に示すように、各層の点群集合 $stand_L_i$ 中におけるすべての点 p を爪曲面点群 $Cloud_{Nail-s}$ 中の爪の中心線の y 座標 $Center_y$ に従って2つの部分に分け、 L_part1_i 、 L_part2_i とし、ここで、

$$L_part1_i = \{p \mid p_y \leq Center_y\}$$

$$L_part2_i = \{p \mid p_y > Center_y\}$$

30

ここで、

$stand_L_i$ は各層の点群 L_i 中における各点 p_j の x 軸座標値を統一し、且つその y 軸及び z 軸座標値は変えずに保持した後に取得される略円弧形の点群集合であり、

f はマッピングであり、

p は現在層の点群集合 $stand_L_i$ 中の点であり、

p_x は点 p の x 軸座標値であり、

X_{L_i} は平面 X_{L_i} の x 軸座標値であり、

L_part1_i は現在層の点群集合 $stand_L_i$ 中の爪中心線の片側の円弧形点群集合であり、

40

L_part2_i は現在層の点群集合 $stand_L_i$ 中の爪中心線の他方側の円弧形点群集合であり、

p_y は点 p の y 軸座標値であり、

$Center_y$ は爪曲面点群 $Cloud_{Nail-s}$ 中の爪中心線の y 軸値である。

【0116】

$stand_L_i$ を L_part1_i 、 L_part2_i の2つの部分に分けた上で、さらに各部分の円弧形点群 L_part1_i 、 L_part2_i を爪の中心線に従って近くから遠くまで順序通り配列してから、点群 $L_ordered_part_i$ を得る。

【0117】

50

その具体的なアルゴリズムは以下の通りである。

入力：

点群 $L_part\ i$ を入力する

半径閾値 R_thres を入力する

出力：

整理して配列した後の点群 $L_ordered_part\ i$ を出力する

初期化：

配列後の点群 $L_ordered_part\ i$ を初期化する

点リスト $\{P\}$ を初期化し、点群 $L_part\ i$ のすべての点をリスト $\{P\}$ にプッシュする

入力点群の各点の法線ベクトル $\{N\}$ を計算する

入力点群の各点の曲率 $\{c\}$ を計算する

アルゴリズムループ：

リスト $\{P\}$ のすべての点を一つ一つチェックし、 $\{P_0\}$ の y 軸座標値が指の中心線の y 座標 $Center\ y$ から最も近いことを満たす $\{P_0\}$ を見つける

さらに $\{P_0\}$ をリスト $\{P\}$ から除去する $\{p\} = \{P\} \setminus P_0$

同時に $\{P_0\}$ を点群 $L_ordered_part\ i$ にプッシュする

$L_ordered_part\ i = L_ordered_part\ i \cup P_0$

リスト $\{P\}$ が空でない限り：

点群 $L_ordered_part\ i$ 中で現在のシーケンスが最大の点 p_{end} を見つける

リスト $\{A\}$ 中で現在点 p_{end} から直線距離が最小である点 P_{min} を見つけ、且つ点 P_{min} をリスト $\{A\}$ から除去する $\{A\} = \{A\} \setminus P_{min}$

同時に P_{min} を点群 $L_ordered_part\ i$ にプッシュする

$L_ordered_part\ i = L_ordered_part\ i \cup P_{min}$

リスト $\{P\}$ が空であるとき、配列後の点群 $L_ordered_part\ i$ を出力する

【0118】

次に、順序の処理を終えた円弧形点群 $L_ordered_part\ i$ を一本の直線点群 $L_expand_part\ i$ に展開し、ここで、

$d_curve = d_straight$ であり、

且つ d_curve は点 P_1 、 P_2 の直線距離に概ね等しく、

$d_curve = \sqrt{(P_{1y} - P_{2y})^2 + (P_{1z} - P_{2z})^2}$ であり、

展開後の対応する点 P_1' 、 P_2' は、

【0119】

【数9】

$$P_1' = P_1 + \vec{v}, \quad \vec{v} = (0 \quad 0 \quad 0 - P_{1z})$$

$$P_2' = P_1' + \vec{u}, \quad \vec{u} = (0 \quad d_curve \quad 0)$$

であり、

すべての点の $L_expand_part\ i$ に対応する点を順次算出する。

【0120】

最後に、すべての直線点群 $L_expand_part\ i$ を新しい $Cloud_expand\ Nail$ に統合し、

$Cloud_expand\ Nail = L_expand_part\ i$

この統合した新しい $Cloud_expand\ Nail$ が即ち爪平面点群 $Cloud\ Nail - p$ である。

【0121】

10

20

30

40

50

第5実施形態

本願の第5実施形態は、爪輪郭の抽出方法及びそのシステムを提供する。当該抽出方法及びそのシステムは、いずれも第1～第4実施形態中の爪輪郭の抽出方法及びそのシステムの設計及び実施に基づくものであり、第1～第4実施形態と実際の工程又は構造は似ているものの、異なる点も幾らかあり、以下では異なる点について重点的に説明を行う。異なる点以外の詳細な説明については、いずれも第1～第4実施形態中と同じであると見なすことができ、詳述しない。

【0122】

図19と図20に示すように、第5実施形態中、爪輪郭のレンダリングの前において、爪輪郭の抽出方法は境界抽出工程をさらに有しており、爪輪郭の抽出システムは境界抽出モジュール10をさらに含み、境界抽出モジュール10は平坦化モジュール5及びレンダリングモジュール6のいずれとも通信接続される。境界抽出工程において、境界抽出モジュール10を用いて爪平面点群CloudNail-pの外周エッジの輪郭点群CloudContourを抽出した後、レンダリングモジュール6を用いて輪郭点群CloudContourの形状を描画し、レンダリングして爪の爪甲型を取得する。

【0123】

境界抽出モジュール10によって、爪平面点群CloudNail-pの輪郭点群CloudContourを抽出する。一方、もしも平坦化後の爪平面点群CloudNail-pを直接抽出した場合、元の点群のノイズ及び爪分割時の誤差によって抽出したエッジに顕著なギザギザ形状が生じ、最外層の輪郭点群CloudContourの平滑性不足によって、レンダリングして取得した爪輪郭の美観性に影響が出てしまう。そのため、爪平面点群CloudNail-pのエッジに対して平滑化処理を行い、爪が元々有するエッジ特徴をできる限り保持するという前提の下で、爪平面点群CloudNail-pのエッジの輪郭点群CloudContourをより平滑で滑らかにできるようにしている。

【0124】

境界抽出モジュール10は平坦化モジュール5及びレンダリングモジュール6のいずれとも通信接続され、境界抽出モジュール10は、独立して設置してPCL点群ライブラリ3と通信接続してもよいし、PCL点群ライブラリ3内に統合するか、又はPCL点群ライブラリ3の一部であってもよい。

【0125】

このようにして、爪輪郭レンダリング工程中、境界抽出モジュール10を用いて爪平面点群CloudNail-p中の輪郭点群CloudContourを抽出する。

【0126】

次に、爪点群平坦化工程は、境界抽出工程後において、エッジ処理工程をさらに含み、爪輪郭の抽出システムはエッジ処理モジュール11をさらに含み、エッジ処理モジュール11は、PCL点群ライブラリ3中にあり、且つ境界抽出モジュール10及びレンダリングモジュール6のいずれとも通信接続される。境界抽出工程において、エッジ処理モジュール11を用いて取得した輪郭点群CloudContourを平滑化処理し、エッジを平滑化する。

【0127】

エッジ処理モジュール11は境界抽出モジュール10及びレンダリングモジュール6のいずれとも通信接続される。エッジ処理モジュール11は、独立して設置してPCL点群ライブラリ3と通信接続してもよいし、PCL点群ライブラリ3内に統合するか、又はPCL点群ライブラリ3の一部であってもよい。

【0128】

具体的には、エッジ処理工程中、エッジ処理モジュール11はカーブフィッティングの方式を用い、境界抽出工程中で境界抽出モジュール10を用いて抽出した平滑性不足の輪郭点群（以下では入力輪郭点群CloudContour-iとも呼ぶ）のエッジにカーブフィッティングを行って、平滑な輪郭点群（以下では出力輪郭点群CloudContourとも呼ぶ）を取得することにより、平滑な爪曲線を得る。エッジ処理工程の具体的な工

程は以下の通りである。

(1) PCL点群ライブラリ3が提供する境界抽出モジュール10によって爪平面点群 $Cloud_{Nail-p}$ の入力輪郭点群 $Cloud_{Contour-i}$ を抽出する。

(2) 抽出した入力輪郭点群 $Cloud_{Contour-i}$ にカーブフィッティングを行い、曲線 $Contour_hull_1$ を得る。

(3) x軸方向に沿って入力輪郭点群 $Cloud_{Contour-i}$ の最も広い箇所のx軸座標 x_1 を見つける。

(4) 入力輪郭点群 $Cloud_{Contour-i}$ のすべての点に対し、座標 x_1 よりも大きい点を取り、集合 $hull_1$ を得る。

(5) フィッティング曲線 $Contour_hull_1$ のすべての点に対し、座標 x_1 よりも小さい点を取り、集合 $hull_2$ を得る。

(6) 取得した集合 $hull_1$ と集合 $hull_2$ を統合して最終的な輪郭の輪郭点群 $Cloud_{Contour}$ を得る。

【0129】

このようにして取得した輪郭の輪郭点群 $Cloud_{Contour}$ は、爪の弧度及びエッジ輪郭のマッチングを保証するという前提の下で、爪の実際のエッジ輪郭により一層適合するようになり、レンダリングして得られる爪輪郭の美観性も向上する。これにより、作製されたネイルチップの成形型によって生産されて得られるネイルチップの美観性を向上させることができる。

【0130】

当業者であれば、本明細書が開示する実施形態に関連して説明した各実施形態のモジュールやアルゴリズムステップは、電子ハードウェア、コンピュータソフトウェア、あるいは双方を組み合わせたものとして実現し得ることを認識するだろう。また、ソフトウェアモジュールは任意の形式のコンピュータ記憶媒体中に配置することができる。ハードウェアとソフトウェアの互換性を示すために、上述の説明において機能に従い各実施形態の構成及び工程について一般的な説明を行った。それらの機能をハードウェアとソフトウェアのどちらによって実行するかは、技術案の特定の用途及び設計における制約条件に依存する。当業者は、説明した機能を特定の用途毎に異なる方法で実現し得るが、それらの実現を本発明の範囲から逸脱したものを見なすべきではない。

【0131】

上述が開示した各実施形態は様々な異なる形式で実現することができ、特許請求の範囲が本明細書に記載の実施形態に限定されると解釈すべきではないことを理解されたい。具体的には、これらの実施形態は、本開示を完全且つ包括的なものとし、本発明の実施形態の概念が当業者に十分伝わるように提供される。本開示は、事実上、添付の特許請求の範囲が限定する本開示の精神及び範囲内における実施形態のすべての代替案、修正及び均等物を網羅することを意図している。また、本開示の実施形態に対する詳細な説明では、完全な理解を可能にするために詳細を大量且つ具体的に示した。しかしながら、本開示の実施形態はそれらの具体的な詳述がない状況下であっても実践し得ることは当業者には明白であろう。

【符号の説明】

【0132】

- 1 データ収集装置
- 2 データ変換装置
- 3 PCL点群ライブラリ
- 4 分割モジュール
- 5 平坦化モジュール
- 6 レンダリングモジュール
- 7 矯正モジュール
- 8 チェックモジュール
- 9 選択モジュール

10

20

30

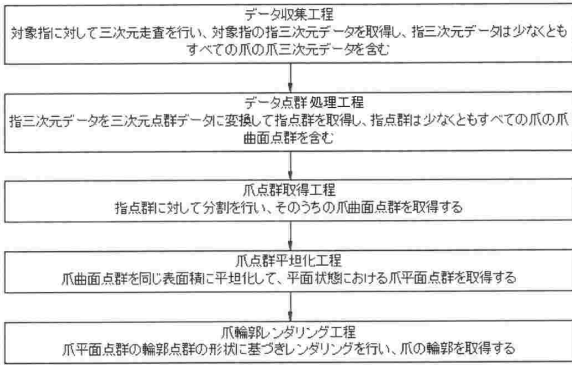
40

50

10 境界抽出モジュール
11 エッジ処理モジュール

【図面】

【図1】



【図】

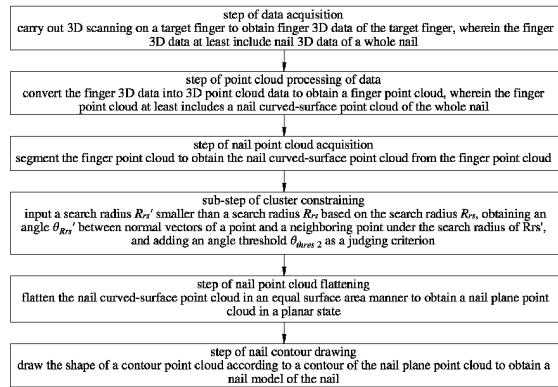
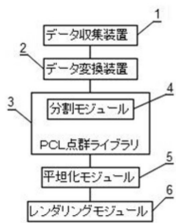
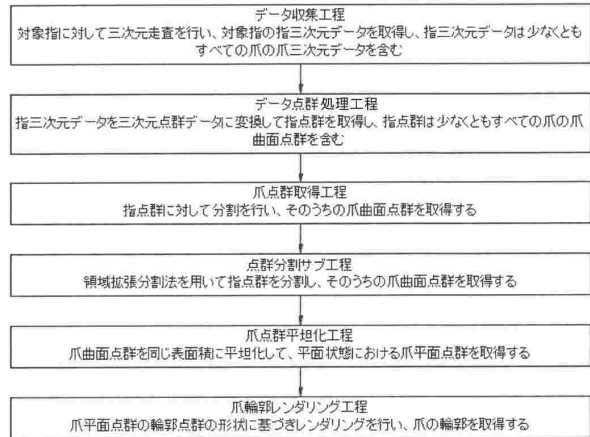


FIG. 1

【図2】



【図3】



10

20

30

40

50

【 図 3 】

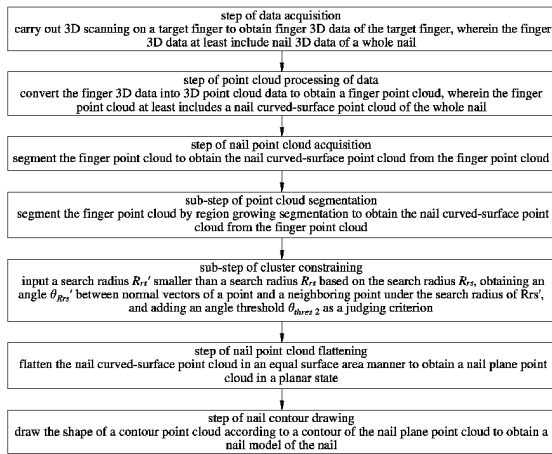


FIG 3

【 図 4 】

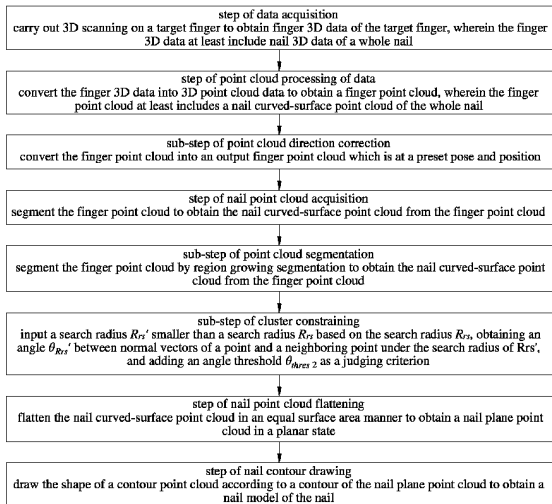
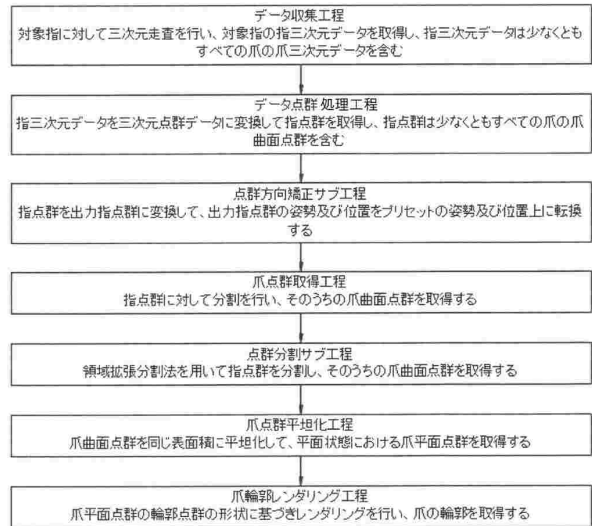


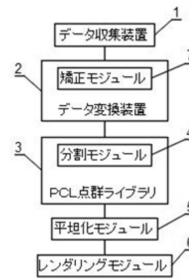
FIG 4

【 図 4 】



10

【 図 5 】



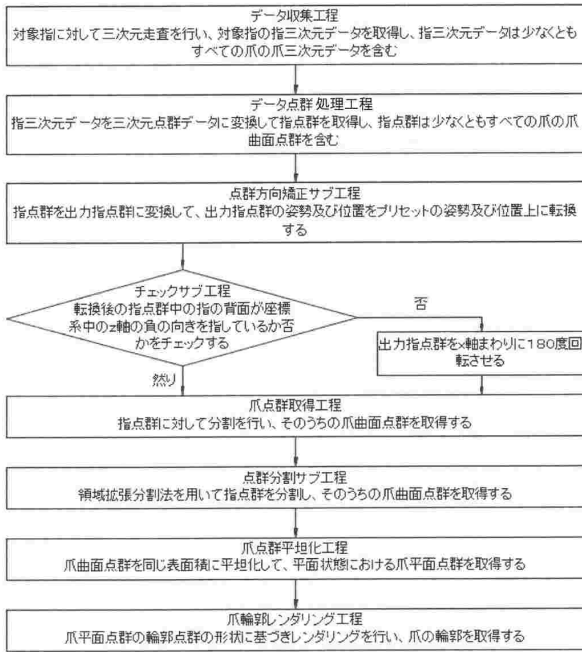
20

30

40

50

【 図 6 】



【 図 8 】

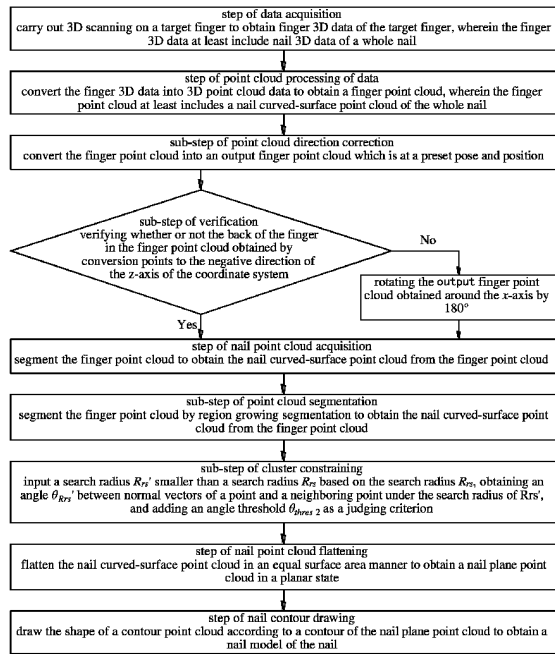
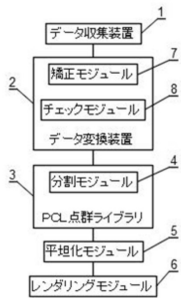
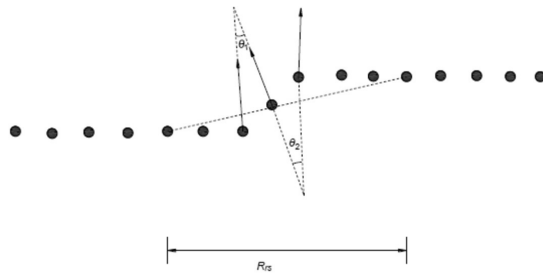


FIG 6

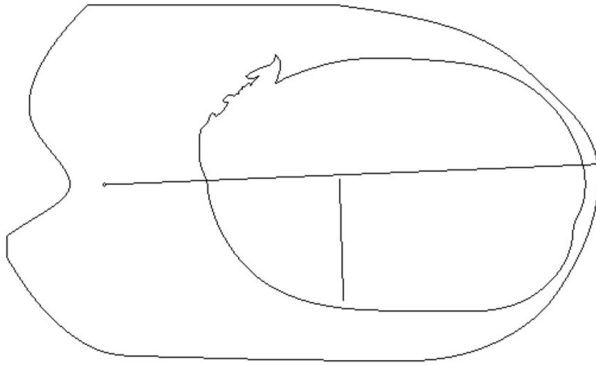
【 図 7 】



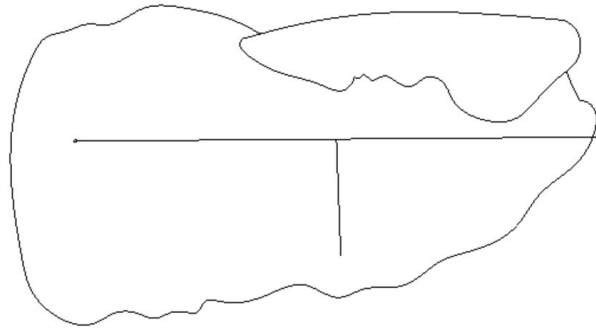
【 図 8 】



【図 9 a】

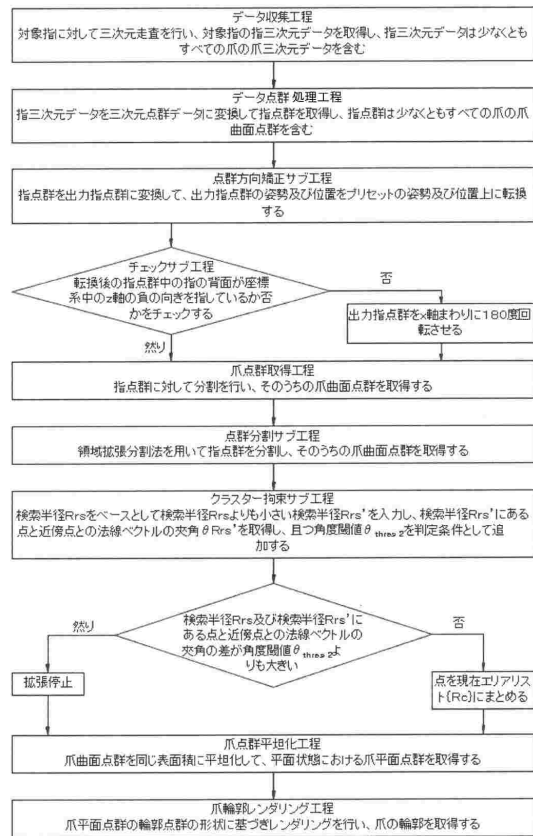


【図 9 b】

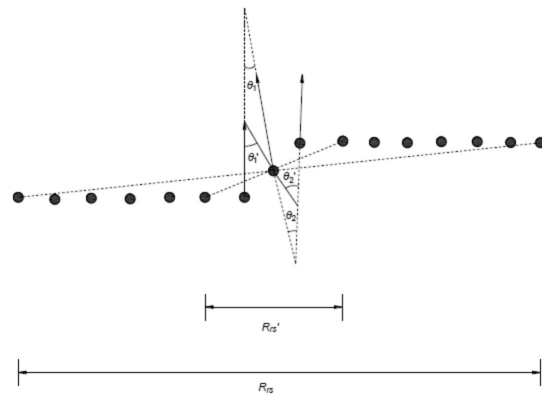


10

【図 10】



【図 11】



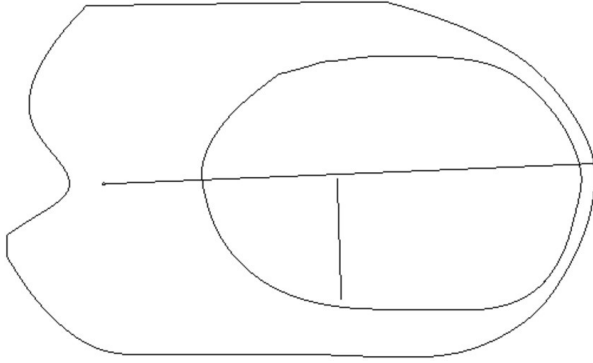
20

30

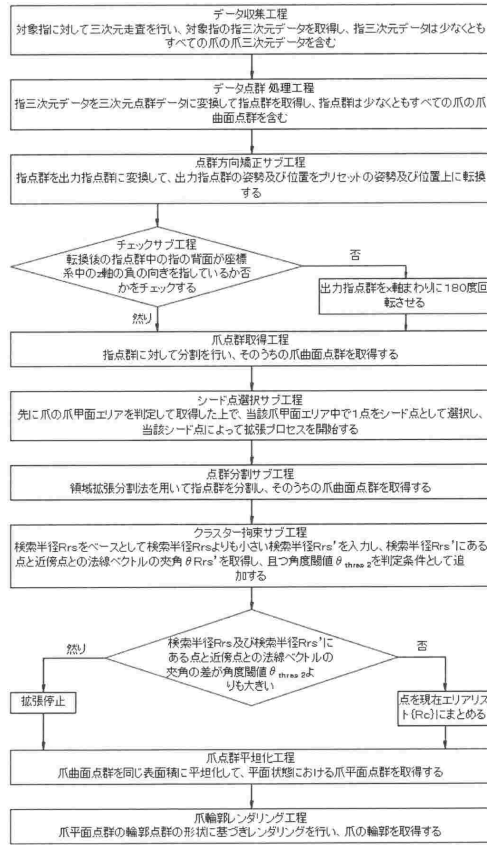
40

50

【 図 1 2 】



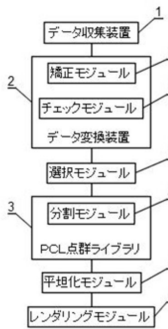
【 図 1 3 】



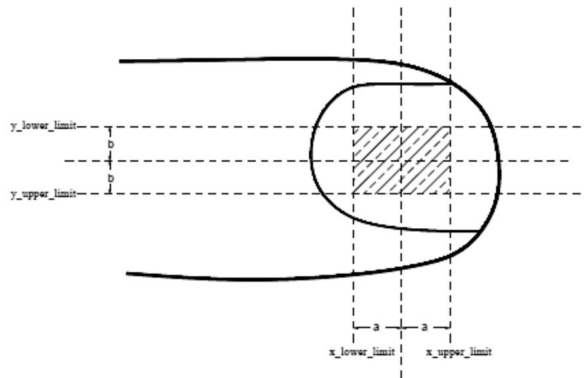
10

20

【 図 1 4 】



【 図 1 5 】

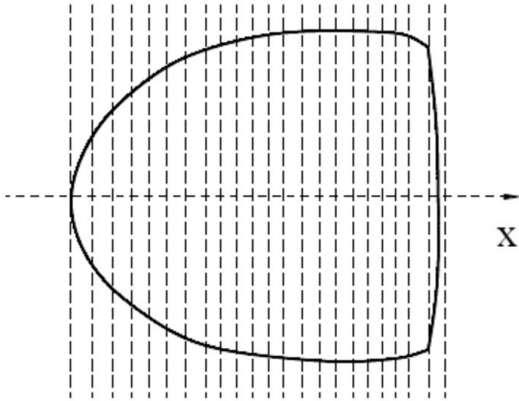


30

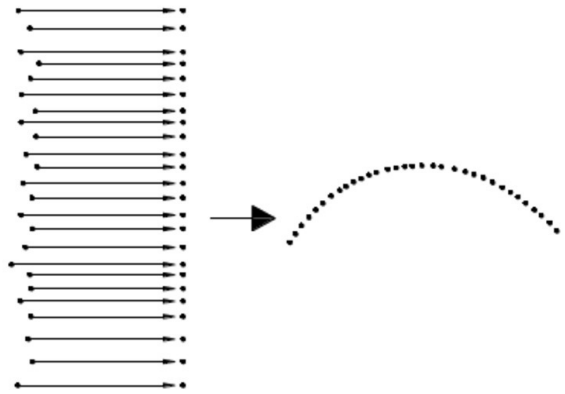
40

50

【 図 1 6 】

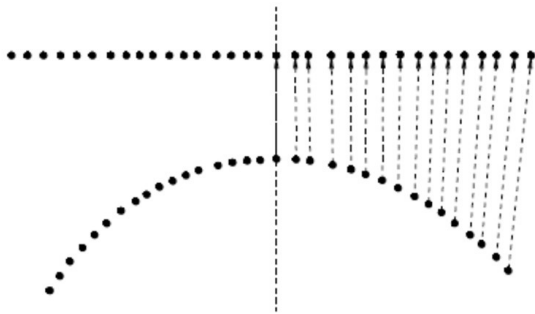


【 図 1 7 】

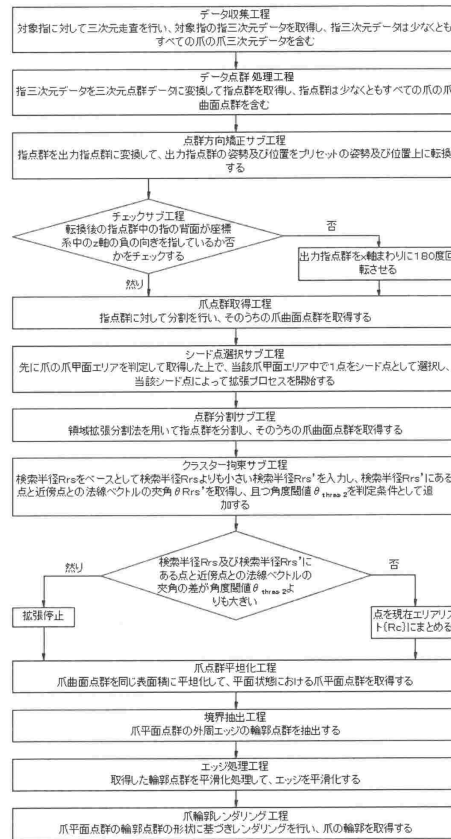


10

【 図 1 8 】



【 図 1 9 】



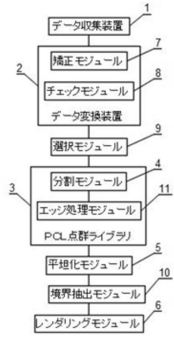
20

30

40

50

【図 20】



10

20

30

40

50

フロントページの続き

山路 8 8 9 号 4 幢 8 層 C 0 2 座

審査官 菊池 伸郎

- (56)参考文献 特表 2 0 0 6 - 5 1 9 0 3 0 (J P , A)
特開 2 0 2 0 - 1 7 7 5 7 8 (J P , A)
米国特許出願公開第 2 0 1 9 / 0 0 9 5 7 4 7 (U S , A 1)
- (58)調査した分野 (Int.Cl. , D B 名)
- G 0 6 T 1 / 0 0 - 7 / 9 0
G 0 6 V 1 0 / 0 0 - 4 0 / 7 0