

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7603006号
(P7603006)

(45)発行日 令和6年12月19日(2024.12.19)

(24)登録日 令和6年12月11日(2024.12.11)

(51)国際特許分類

G 0 6 T 13/40 (2011.01)

F I

G 0 6 T 13/40

請求項の数 14 (全33頁)

(21)出願番号	特願2021-523745(P2021-523745)	(73)特許権者	519327490
(86)(22)出願日	令和1年10月31日(2019.10.31)		ソウル マシーンズ リミティド
(65)公表番号	特表2022-509916(P2022-509916 A)		ニュージーランド国, 1 0 1 0, オークランド, カスタムズ ストリート ウェスト 1 0 6, レベル 1
(43)公表日	令和4年1月25日(2022.1.25)	(74)代理人	110002837
(86)国際出願番号	PCT/IB2019/059327		弁理士法人アスフィ国際特許事務所
(87)国際公開番号	WO2020/089817	(72)発明者	ウー, ティム
(87)国際公開日	令和2年5月7日(2020.5.7)		ニュージーランド国 0 6 1 2 オークランド, ヘンダーソン, フィッツウォーター プレイス 2 3
審査請求日	令和4年10月26日(2022.10.26)	(72)発明者	スミヤツ, バベル
(31)優先権主張番号	747627		ニュージーランド国 1 0 2 3 オークランド, ブロードウェイ 1 7 6, ビー 5 1 9
(32)優先日	平成30年10月31日(2018.10.31)		
(33)優先権主張国・地域又は機関	ニュージーランド(NZ)		

最終頁に続く

(54)【発明の名称】 アニメーションシステムのコンピュータ実装方法、データプロセッシング装置、アニメーションシステム、コンピュータプログラム、又はコンピュータ読み取り可能な媒体

(57)【特許請求の範囲】

【請求項1】

コンピュータグラフィックスアニメーションを実行するための、2つ以上のモーフターゲットシェイプの間の重み付き補間を生成するアニメーションシステムのコンピュータ実装方法であって、

前記重み付き補間を生成するために、モーフターゲットシェイプに対する入力ウェイトの組を受け取るステップと、

連続多変量補間関数を使用するステップであって、前記連続多変量補間関数は、モーフターゲットシェイプおよびモーフターゲットシェイプの重み付き組み合わせを表す制約シェイプを含む形状の組が与えられる場合、対応するウェイトがモーフターゲットシェイプに対する入力ウェイトとして当該連続多変量補間関数に提供される時にモーフターゲットシェイプおよび制約シェイプを再現するように構成され、

少なくとも1つの制約シェイプは、前記モーフターゲットシェイプのうちの1つまたは複数のモーフターゲットシェイプの部分的ウェイトに対応するインクリメンタルシェイプであり、

前記連続多変量補間関数は、ウェイトとモーフターゲットシェイプの間の非線形関係を表す、ステップと、

前記連続多変量補間関数への引数として前記入力ウェイトを用いることによって、前記重み付き補間を生成するステップと、

を備える、コンピュータ実装方法。

【請求項 2】

コンピュータグラフィックスアニメーションを実行するための、ベースシェイプに対して2つ以上のモーフターゲットシェイプの間の重み付き補間を生成するアニメーションシステムのコンピュータ実装方法であって、当該各モーフターゲットシェイプは、複数のトポロジ的に一貫した頂点座標を含み、

前記モーフターゲットシェイプの頂点座標とトポロジ的に一貫する複数の頂点座標を含む複数の入力制約シェイプを受け取るステップであって、当該入力制約シェイプはそれぞれ、前記モーフターゲットシェイプのうちの1つまたは複数のモーフターゲットシェイプに対するゼロ以外のウェイトに関連付けられ、少なくとも1つの制約シェイプは、前記モーフターゲットシェイプのうちの1つまたは複数のモーフターゲットシェイプの部分的ウェイトに対応するインクリメンタルシェイプであるステップと、

10

前記入力制約シェイプを用いて、モーフターゲットシェイプに対する複数の新しいウェイトのための追加制約シェイプを生成し、当該追加制約シェイプをそれらの新しいウェイトのそれぞれに関連付け、前記追加制約シェイプは、 n 次元立方体を完成または拡張するように新しいウェイトで生成され、モーフターゲットシェイプが、原点としてベースシェイプを有し、モーフターゲットシェイプに対するウェイトとして n 次元立方体のエッジを有する n 次元立方体の寸法であるステップと、

前記2つ以上のモーフターゲットシェイプのそれぞれに対する補間ウェイトを受け取るステップと、

前記ベースシェイプ、入力制約シェイプ、および追加制約シェイプを用いて、前記2つ以上のモーフターゲットシェイプの間を補間するための補間関数を生成し、前記補間関数は、ウェイトとモーフターゲットシェイプとの間の非線形関係を表すステップと、

20

前記補間関数に対する引数として補間ウェイトを用いて、前記2つ以上のモーフターゲットシェイプの間の前記重み付き補間に対応する頂点座標を生成するステップと、を含むコンピュータ実装方法。

【請求項 3】

少なくとも1つの入力制約シェイプが、単位ウェイトを有する前記2つ以上のモーフターゲットシェイプの組み合わせに対応するコンビネーションシェイプである、請求項2に記載の方法。

【請求項 4】

前記追加制約シェイプが、メッシュレス補間を用いて生成される、請求項2に記載の方法。

30

【請求項 5】

前記追加制約シェイプが、放射基底補間を用いて生成される、請求項4に記載の方法。

【請求項 6】

前記 n 次元立方体を低次元空間に分割し、各低次元空間に対して追加制約シェイプを生成するステップを含む、請求項2～5のいずれか一項に記載の方法。

【請求項 7】

コンピュータグラフィックスアニメーションを実行するための、ベースシェイプ B_0 に対して複数 n のモーフターゲットシェイプ $B_1 \dots B_n$ の間の重み付き補間を生成するアニメーションシステムのコンピュータ実装方法であって、

40

前記モーフターゲットシェイプ $B_1 \dots B_n$ のうちの各モーフターゲットシェイプ B_k に対して、当該モーフターゲットシェイプ B_k に適用されるウェイト w_k を含む、ウェイト W の組を受け取るステップと、

複数 m の制約シェイプ $C_1 \dots C_m$ を受け取るステップであって、制約シェイプはそれぞれ、前記モーフターゲットシェイプ $B_1 \dots B_n$ のうちの1つまたは複数のモーフターゲットシェイプ（関連付けシェイプ）に対するゼロ以外のウェイト（関連付けウェイト）に関連付けられ、少なくとも1つの制約シェイプ $C_1 \dots C_m$ は、1つ以上のモーフターゲットシェイプ $B_1 \dots B_n$ の部分的ウェイトに対応するインクリメンタルシェイプであるステップと、

50

連続多変量補間関数を生成するステップであって、当該連続多変量補間関数は、各モーフターゲットシェイプまたは関連付けシェイプに対する制約シェイプの関連付けウェイトが、当該連続多変量補間関数に対する引数として提供される時に、各モーフターゲットシェイプおよび各制約シェイプを再現するように構成されているステップと、

前記連続多変量補間関数の引数としてモーフターゲットシェイプに適用される前記ウェイトWを用いて前記重み付き補間を生成するステップと、を含み、

前記連続多変量補間関数は、ウェイトとモーフターゲットシェイプとの間の非線形関係を表す、コンピュータ実装方法。

【請求項 8】

前記連続多変量補間関数は、高次元のモデル表現である、請求項 1 または 7 に記載の方法。

10

【請求項 9】

前記連続多変量補間関数は、モーフターゲットシェイプおよび制約シェイプに適用されるモディファイアの線形加重和として線形化されている、請求項 1 または 7 に記載の方法。

【請求項 10】

前記線形加重和は、近似テイラー級数展開に基づいている、請求項 9 に記載の方法。

【請求項 11】

メッシュ補間またはメッシュレス補間を用いてインクリメンタルシェイプの間を補間することにより、補間精度がさらに向上する、請求項 10 に記載の方法。

【請求項 12】

20

前記連続多変量補間関数は、以下の式を有し、

【数 1】

$$f(W) = B_0 + \sum_{k=1}^n (w_k \Delta B_k) + \sum_{i=1}^m (\beta_i \Delta C_i)$$

式中、 ΔB_k は、修正モーフターゲットシェイプを表し；

30

ΔC_i は、修正制約シェイプを表し；

β_i は、各制約シェイプ C_i に適用されるモディファイアを表し、

前記連続多変量補間関数は、全ての制約シェイプ $C_1 \dots C_m$ について成り立つように構成されている、請求項 7 に記載の方法。

【請求項 13】

請求項 1 ～ 12 のいずれかに記載の方法を実行する手段を備えるデータプロセッシング装置又はアニメーションシステム。

【請求項 14】

コンピュータプログラムがコンピュータで実行される場合、コンピュータプログラムが請求項 1 ～ 12 のいずれかに記載の方法をコンピュータに実行させる指示を含む、コンピュータプログラムまたはコンピュータプログラムを保存してなるコンピュータ読み取り可能な媒体。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像データ処理／生成／アニメーションに関する。より具体的には、以下に限定するものではないが、本発明は、モーフターゲットアニメーションに関する。

【背景技術】

【0002】

コンピュータグラフィックスのキャラクタの動きは、制御パラメータの組を用いてアニ

50

メータによって制御される。これらのパラメータが与えられると、アニメーションソフトウェアはこのキャラクタの形状を計算する。コンピュータグラフィックスのキャラクタは、メッシュを形成する頂点の集合として表されるため、この問題はメッシュ変形問題とも呼ばれる。この問題の主な課題は、生成された形状を使用に好適な望ましい状態に保ちながら、制御パラメータを用いてアニメーション化できる技術を創出することである。

【0003】

物理ベースの変形アプローチでは、物理シミュレーションを用いてキャラクタの形状を生成する。これには通常、基礎となる筋肉または骨の運動力学が含まれる。この場合、制御パラメータを用いるメッシュ変形は、キャラクタの変形を使用に好適な現実的な形状に制限する力学法則に従う。物理ベースのシミュレーションには、計算コストの高い高度な数値手法に加えて、力学の知識が必要である。さらに、多くのコンピュータグラフィックスのキャラクタは想像上のキャラクタ（エイリアン、ドラゴンなど）であり、物理シミュレーションで特定の望ましい変形を実現するには、多くの試行錯誤を繰り返す必要がある。

【0004】

例ベースの方法では、アニメーションで参照状態として用いられる様々なキャラクタポーズを作成する。これらの例形状で、所望のキャラクタ変形を指定できる一方で、物理シミュレーションは間接的なプロセスである。つまり、任意のキャラクタ形状変形が、運動力学から計算され得る。線形ブレンドシェイプの例ベースのモデルが、アニメーションで広く用いられている（例えば、Lewis, J. P., Anjyo, K., Rhee, T., Zhang, M., Pighin, F. H., & Deng, Z. (2014). Practice and Theory of Blendshape Facial Models. Eurographics (State of the Art Reports), 1 (8)）。従来のアプローチには、線形にブレンドされるモーフターゲットシェイプが含まれるが、線形ブレンドでは、多くの場合、様々な組織タイプと解剖学的構造の非線形相互作用だけでなく、非線形軟組織変形も再現されない。これを回避するために、ウェイトの大きさに対するアルゴリズム的制約が、制約アルゴリズムを用いて導入されることがある。

【発明の概要】

【発明が解決しようとする課題】

【0005】

本発明の目的は、モーフターゲットアニメーションを改善すること、または少なくとも公衆または産業に有用な選択肢を提供することである。

【課題を解決するための手段】

【0006】

一実施形態によると、2つ以上のモーフターゲットシェイプの間の重み付き補間を生成するためのアニメーションシステムであって、当該アニメーションシステムは、前記重み付き補間を生成するために、モーフターゲットシェイプに対する入力ウェイトの組を受け取る入力手段と、多変量連続補間関数であって、モーフターゲットシェイプおよびモーフターゲットシェイプの重み付き組み合わせを表す制約シェイプを含む形状の組が与えられた場合、対応するウェイトがモーフターゲットシェイプに対する入力ウェイトとして当該多変量連続関数に提供される時にモーフターゲットシェイプおよび制約シェイプを再現するように構成された多変量連続補間関数と、前記多変量連続補間関数への引数として前記入力ウェイトを用いることによって、前記重み付き補間を生成する手段と、を備える。

【0007】

別の実施形態によると、ベースシェイプに対して2つ以上のモーフターゲットシェイプの間の重み付き補間を生成するための方法であって、当該各モーフターゲットシェイプは、複数のトポロジ的に一貫した頂点座標を含み、当該方法は、前記モーフターゲットシェイプの頂点座標とトポロジ的に一貫する複数の頂点座標を含む複数の入力制約シェイプを受け取るステップであって、当該各入力制約シェイプは、前記モーフターゲットシェイプのうちの1つまたは複数のモーフターゲットシェイプに対するゼロ以外のウェイトに

関連付けられている、ステップと、前記入力制約シェイプを用いて、モーフターゲットシェイプに対する複数の新しいウェイトのための追加制約シェイプを生成し、当該追加制約シェイプをそれらの新しいウェイトのそれぞれに関連付けるステップと、前記2つ以上のモーフターゲットシェイプのそれぞれに対する補間ウェイトを受け取るステップと前記ベースシェイプ、入力制約シェイプ、および追加制約シェイプを用いて、前記2つ以上のモーフターゲットシェイプの間を補間するための補間関数を生成するステップと、前記補間関数に対する引数として補間ウェイトを用いて、前記2つ以上のモーフターゲットシェイプの間の前記重み付き補間に対応する頂点座標を生成するステップと、を含む。

【0008】

別の実施形態によると、ベースシェイプ B_0 に対して複数 n のモーフターゲットシェイプ $B_1 \dots B_n$ の間の重み付き補間を生成する方法であって、当該方法は、前記モーフターゲットシェイプ $B_1 \dots B_n$ のうちの各モーフターゲットシェイプ B_k に対して、当該モーフターゲットシェイプ B_k に適用されるウェイト w_k を含む、ウェイト W の組を受け取るステップと、複数 m の制約シェイプ $C_1 \dots C_m$ を受け取るステップであって、各制約シェイプは、前記モーフターゲットシェイプ $B_1 \dots B_n$ のうちの1つまたは複数のモーフターゲットシェイプ（関連付けシェイプ）に対するゼロ以外のウェイト（関連付けウェイト）に関連付けられている、ステップと、連続多変量補間関数を生成するステップであって、当該連続多変量補間関数は、各モーフターゲットシェイプまたは関連付けシェイプに対する制約シェイプの関連付けウェイトが、当該補間関数に対する引数として提供される時に、各モーフターゲットシェイプおよび各制約シェイプを再現するように構成されている、ステップと、前記補間関数の引数としてモーフターゲットシェイプに適用される前記ウェイト W を用いて前記重み付き補間を生成するステップと、を含む。

【0009】

別の実施形態によると、コンプレックスシェイプの基礎となるコンポーネントシェイプウェイトを推定する方法であって、当該方法は、1つまたは複数の提案コンポーネントシェイプを受け取るステップと、提案コンポーネントシェイプに関連付けられた前記ウェイトが確実にゼロ以外になるようにペナルティとソリューション境界が適用される最小二乗問題を解決することで、コンポーネントシェイプウェイトを取得するステップと、を含む。

【0010】

2つ以上のコントロールシェイプの間の重み付き補間を生成するための方法であって、前記コントロールシェイプの少なくとも1つは、複数のコンポーネントシェイプの重み付き組み合わせからなるコンプレックスシェイプであり、当該方法は、前記コントロールシェイプをそれらの構成コンポーネントシェイプおよび当該構成コンポーネントシェイプのそれぞれに対する関連付けウェイトにマッピングして、重み付きターゲットコンポーネントシェイプの組を形成するステップと、前記重み付きターゲットコンポーネントシェイプの組をインターポレータに提供するステップと、前記インターポレータを用いて、前記重み付きターゲットコンポーネントシェイプの組の間を補間し、前記2つ以上のコントロールシェイプの間の前記重み付き補間を生成するステップと、を含む。

【図面の簡単な説明】

【0011】

【図1】線形モーフターゲットシェイプの例ベースのモデルの概略図を示す。

【図2】線形モーフターゲットシェイプの例ベースのモデルが望ましくない形状を生成する方法の概略図を示す。

【図3】凸包制約をモーフターゲットシェイプに適用する概略例を示す。

【図4】一実施形態に係る非線形補間関数を事前計算する方法のフロー図を示す。

【図5】一実施形態に係る補間の方法のフロー図を示す。

【図6】2Dラティスの例を示す。

【図7】3次元ハイパーキューブモデルを示す。

【図8】モーフターゲットシェイプウェイトを制御するためのユーザインターフェイスのスクリーンショットを示す。

10

20

30

40

50

【図 9】ベースシェイプを示す。

【図 10】複数のモーフターゲットシェイプを示す。

【図 11】左右対称の表情の左右のモーフターゲットシェイプを示す。

【図 12】コンビネーションシェイプである制約シェイプを示す。

【図 13】ビセム (viseme) を示す。

【図 14】インクリメンタルシェイプである制約シェイプを示す。

【図 15】コンビネーションインクリメンタルシェイプである制約シェイプを示す。

【図 16】コンポーネントシェイプウェイトを推定する方法のフロー図を示す。

【図 17】アニメーションフローを示す。

【図 18】アニメーションシステムを示す。

10

【発明を実施するための形態】

【0012】

モーフターゲットシェイプ補間

モーフターゲットアニメーションでは、モーフターゲットシェイプが、ブレンドシェイプ、基礎シェイプ、ブレンドシェイプターゲット、またはモーフターゲットとしても知られていて、頂点座標（または位置）の組として保存され得る、ベースシェイプ（または参照シェイプ）の「変形」バージョンを表す。モーフターゲットシェイプを用いるベースシェイプの変形は、制御パラメータの組を用いてアニメータによって制御され得る。これらの制御パラメータが与えられると、モーフターゲットアニメーションモデルが、変形の形状を計算する。コンピュータグラフィックスとアニメーション技術の重要な課題は、生成された形状を使用しに好適な望ましい状態に保ちながら、制御パラメータの下でアニメーション化できるようにすることである。

20

【0013】

モーフターゲットアニメーションモデルは、それぞれが異なる幾何学的形状を持つ数個のモーフターゲットシェイプを含んでもよい。モーフターゲットシェイプは、例えば、トポロジ的に一貫した頂点座標（幾何学的位置、または点）の組で表されてもよく、これによって、アニメータは、1つ以上のモーフターゲットシェイプを用いてベースシェイプを変形させて新しい形状を作成できる。顔のアニメーションでは、従来のモーフターゲットアニメーションモデル／モーフターゲットシェイプインターポレータが、顔のポーズを、多数の表情、つまり、モーフターゲットシェイプ（「ターゲット」）の線形結合として生成する。図 10 は、フェイシャルアクションコーディングシステム (FACS) シェイプを表す 4 つのモーフターゲットシェイプの例、「チークレイザー (cheek raiser)」A (AU06)、「リップパッカー (lip pucker)」B (AU18)、「ジョーオープナー (jaw opener)」C (AU26)、「リップサック (lip suck)」D (AU28) を示す。

30

【0014】

図 1 は、線形の例ベースのモーフターゲットアニメーションモデルの概略図を示す。示した人間の顔の線形モーフターゲットアニメーションモデルには、3 つの提供された形状、つまり、 B_0 、 B_1 、 B_2 が含まれている。これらの 3 つの例のみを用いて、モーフターゲットアニメーションモデルは、与えられた表情形状例を線形結合することにより、様々なブレンドされた表情を生成できる。ベースシェイプ B_0 は、顔の中立状態または安静状態を表す。モーフターゲットシェイプ B_1 は、顔の顎を開ける変形を表し、 B_2 は、目が開くように変形した顔を表す。補間（ブレンディングまたはモーフィングとも呼ばれる）とは、何らかの方法で複数の形状間を補間することを指す。

40

【0015】

図 1 は、 B_0 と B_1 の中間の補間を、半分開いた顎 (B_1 に 0.5 の係数を掛けることで得られる) として描写され得る $0.5 B_1$ として示し、 B_0 と B_2 の中間の補間を、半分開いた目 (B_2 に 0.5 の係数を掛けることで得られる) として説明され得る $0.5 B_2$ として示す。

【0016】

50

ウェイトをモーフトargetシェイプに適用して、インクリメンタルシェイプ（ベースシェイプとモーフトargetシェイプとの中間の形状、「中間」形状と呼ぶこともある）を作成できる。個々のモーフトargetシェイプに適用されるウェイトは、図8に示すように、スライダーを用いて、ユーザインターフェイス上に設定されてもよい。本明細書では、単一のモーフトargetシェイプに対するウェイト $w: 0 < w < 1$ を有する形状を、インクリメンタルシェイプと呼ぶ。

【0017】

モーフトargetアニメーションモデルが、単位ウェイト: $w = 1$ を有する複数のモーフトargetシェイプの組み合わせをアニメーション化するのに用いられてもよい。例えば、 $B_1 + B_2$ は、 B_1 と B_2 の組み合わせを示す。単位ウェイトを有する2つ以上のモーフトargetシェイプの組み合わせである任意の形状を、コンビネーションシェイプと呼ぶ。

【0018】

コンビネーションインクリメンタルシェイプは、2つ以上のモーフトargetシェイプの組み合わせである形状であり、それによって、その組み合わせを形成するモーフトargetシェイプの少なくとも1つは、重み付けられる（つまり、非単位ウェイトを有する）。図1では、 $0.5 B_1 + 0.5 B_2$ は、50%のウェイトを有する B_1 と50%のウェイトを有する B_2 との組み合わせであるコンビネーションインクリメンタルシェイプを示している。よって、図1に示すモーフトargetアニメーションモデルは、 B_1 と B_2 の様々な線形結合によって、理論的に無限の可能な表情の組を提供する。線形モーフトargetアニメーションモデルは、任意の数のモーフトargetシェイプ B_k を含んでもよく、モーフトargetシェイプ B_k は、下記数式1を用いて線形にブレンドされる。ここで、 B_0 は、静止している幾何学的形状（ベースシェイプ）における形状の頂点を表し、 ΔB_k は、ベースシェイプに対する k 番目のモーフトargetシェイプの変位を表すベクトル（ $\Delta B_k = B_k - B_0$ ）であり、 w_k は、基礎シェイプ B_k に適用される対応するウェイトであり、 f は、合形状である。ここで、用語「線形」は、合形状 f が基礎ウェイト w_k に対して線形に依存することを意味する。

【0019】

【数1】

$$f = B_0 + \sum_{k=1}^K (w_k \Delta B_k)$$

【0020】

モーフトargetアニメーションは、形状が増大および結合され得る方法の直感的な理解を、アニメータに提供する。しかし、図1を参照して上記で説明したような単純な線形モデルの使用には、いくつかの問題がある。

【0021】

第一に、モデル化されている対象が線形的に変形しない場合、形状補間は、使用に適さない動きを生成する可能性がある。これは、生物学をモデル化するときに特定の問題を引き起こす。例えば、人間の顔をモデル化する場合、線形方式では、実世界で発生する非線形の軟組織の変形を再現できない。しかし、表情が線形モーフトargetアニメーションモデルでモデル化される場合、ブレンドウェイトと変形した顔の幾何学的形状の間の線形関係を仮定することで、表情は線形化され、非現実的なアニメーションになる。

【0022】

第二に、図1の単純な線形モデルは、様々な組織タイプと解剖学的構造の非線形相互作用を表すことができない。図2は、アニメータが B_1 と B_2 の組み合わせのより現実的な

表現として表情Dを取得したい例を示している。この例示の目的のために、人間の顔の生物学を、目と顎を同時に開くと、顎も開いていない場合よりも目が開くことができず、そして、顎と一緒に目を開くと、顎はわずかに異なる様式で開く（楕円ではなく正方形として表される）ように想定する。単純な線形方式を用いると、 $B_1 + B_2$ が生成されるが、ユーザが表情Dを取得することを望んでいるため、このようなモデリングの結果は使用に適さない。換言すれば、モーフトargetシェイプの組み合わせは、必ずしも、個々のモーフトargetシェイプの外観変化の合計ではなく、実際、組み合わせによっては、外観に明らかに異なる変化が生じる場合がある。モーフトargetシェイプとしてのFACSユニットに基づいてモーフトargetアニメーションモデルを作成する際、モデラーが、モーフトargetシェイプの組み合わせをより許容できるように見せるために、個々のアクションユニット（AU）シェイプを微調整できる場合があるが、これは、個々のAUの表現力を犠牲にする可能性がある。

【0023】

複数の形状変形は同じ頂点を共有する可能性があり、これらの形状をブレンドすると頂点の変位が重なる。その結果、予測できないブレンド／補間された形状となるため、多くの形状の結合は、使用に適さない形状を生じる可能性がある。これを回避するために、ウェイトの大きさに対する制約が導入されてもよい。基礎ウェイトに対する制約を適用するいくつかの方法が適用されてもよく、例えば、制約として数式2を適用すると、モーフトargetモデルがモーフトargetの凸包内にあることが保証される。

【0024】

【数2】

$$\sum_{k=1}^n w_k = 1$$

【0025】

ウェイトを制約に違反せずに独立してモデル化することができないので、ウェイトに制約がある方法には問題がある。また、上記の制約式は、合形状が凸包内にあることを確実にするだけであり、変形が現実的であることを保証するものではない。実際には、この制約は、ブレンドされた頂点変位の大きさを低減することによって破壊を防ぐ。例えば、図3に示すように、凸包制約を基礎に適用することは、所望の結果が604に示されるものであるのに対し、モーフトargetシェイプ601および602の組み合わせは603に示すように見える可能性があることを、意味し得る。

【0026】

使用に好適なモーフトargetシェイプをもたらすウェイトを定義する別の方法は、下記数式3などの最小二乗問題を解くことである。ここで、 α_k は、解かれる新しいウェイトであり、 μ は、正則化パラメータである。このアプローチの欠点は、使用に好適なモーフトargetシェイプを得るためにパラメータ μ を選択することである。また、このアプローチでは、 w_k が更新されるたびにシステムが最小化問題を解く必要があるため、計算コストが高くなる。

【0027】

【数3】

$$\min \left\| \sum_{k=1}^n (W_k - \alpha_k) \triangle B_k \right\|_n + \mu \|\alpha_k\|$$

【0028】

別のアプローチには、あるヒューリスティックアルゴリズムに従って、指定されたモーフトargetシェイプを変更して、モデルを使用に好適な合成モーフトargetシェイプ

に制約することが含まれる。その結果、モーフターゲットモデルは、下記数式 4 の形式を有する非線形になる。ここで、新しいウェイト $\alpha_k = h(w_k)$ は、基礎ウェイトの関数である。ここでの重要な特徴は、関数 h の案出である。これは、多くの場合、幅広い実験と専門知識を要する。

【0029】

【数 4】

$$f = B_0 + \sum_{k=1}^n h(w_k) \Delta B_k,$$

10

【0030】

上記のアプローチでは、アルゴリズムベースの制約が導入される。言い換えると、制約アルゴリズムが、モデルパラメータ全体にこのアルゴリズムを適用することにより、基礎ウェイトを変更するという方法で導入される。しかし、実際には、これは、制約に違反せずにブレンドウェイトを独立してモデル化できないことを意味する。

【0031】

例えば、ある特定の場合では、ユーザは、 $w_1 = 1$ および $w_2 = 1$ を指定し、ブレンドシェイプ $f = B_0 + \Delta B_1 + \Delta B_2$ を想定できるが、制約を適用した後のアニメーションシステムは、 w_2 を再計算して、 $w_2 = \alpha \neq 1$ 、 $f = B_0 + \Delta B_1 + \alpha \Delta B_2$ となり、ユーザ側にとって望ましくない結果につながる（言い換えると、生成された形状 f は、ユーザがウェイトを指定して予期したものではない）。

20

【0032】

これは、いくつかのモーフターゲットシェイプ補間および／またはモーフターゲットシェイプの結合の非線形性を考慮に入れて、現実的かつ予測可能な様式でモーフターゲットシェイプを計算的に結合する方法を含む技術的な問題につながる。多くのモーフターゲットシェイプの間の補間を計算することは計算コストが高く、放射基底関数（RBF）補間などの方法は、多数のモーフターゲットシェイプ（次元）では非常に遅くなる。

【0033】

本発明の詳細な説明および実施形態

30

モーフターゲットアニメーションモデルは、モーフターゲットシェイプを、現実的で、一貫性があり、そして制御しやすい様式で、増大および結合することを可能にする。制約シェイプが、ベースシェイプと複数のモーフターゲットシェイプの間の補間に用いられる補間関数を作成するために提供される。制約シェイプは、これらが表現する個々のモーフターゲットシェイプに対するウェイトにマッピングされるモーフターゲットシェイプの結合および／または増大の現実的な例である。非線形補間関数に制約シェイプを組み込むことは、望ましくない形状が生成されるのを防ぐのに役立つ。これは、制約シェイプを「通過」する多変量連続関数の案出とみなすことができる。この補間関数は、ウェイトをモーフターゲットシェイプに適用することにより、ユーザ制御下でモーフターゲットシェイプの組み合わせである任意の望ましいポーズを生成するのに、用いられ得る。

40

【0034】

新しいウェイトを表す追加制約シェイプが、モーフターゲットシェイプと入力制約シェイプを用いて、非線形の様式で自動的に計算されてもよい。補間された合成ポーズは、制約シェイプ（入力制約シェイプと追加制約シェイプの両方）を介した実世界の非線形の影響を考慮に入れた、関連するすべてのモーフターゲットシェイプの組み合わせになる。非線形性が、事前計算段階で導入されるが、頂点の実際の補間は線形である。これにより、高速でリアルタイムの補間が可能になる。

【0035】

すべての形状（モーフターゲットシェイプおよび制約シェイプ）は、補間関数を介して

50

モーフトargetアニメーションモデルで説明できる。補間関数は、対応するウェイトが提供される場合にモーフトargetシェイプおよび制約シェイプを再現するように構成されている。すべての形状 S （モーフトargetシェイプと制約シェイプの両方を含む）の組が与えられると、補間関数 $f(W)$ が、 $f(W) = S$ となるように生成される。ここで、 W は、与えられたすべての補間ウェイトの組である。

【0036】

モーフトargetシェイプ空間の次元数（モデル中のモーフトargetシェイプの数）を拡張するのではなく、このシェイプ空間は、モーフトargetシェイプモデルへの非線形寄与としてモデル化される。つまり、モデルに必要なモーフトargetシェイプの数は、例えば、新しいユニットを必要とする代わりにユニットの混合を組み込むことによって、低減され得る。これは、非線形結合の数式を利用する高次元モデル表現（HDMR）を用いることで実現され得る。言い換えると、シェイプ空間は、モーフトargetアニメーションモデルへの非線形寄与の組み合わせとしてモデル化され得る。

【0037】

複数のモーフトargetシェイプ n が、表現空間または変形空間を形成する n 次元の立方体の頂点と見なされてもよい。各次元は、モーフトargetシェイプに関連付けられ、ベースシェイプは、表現空間の原点にある。このハイパーキューブのエッジは、対応するモーフトargetシェイプに対するウェイトを表す。図7は、3次元の立方体を示す。原点 B_0 は、ベースシェイプを表し、3つのモーフトargetシェイプ B_i 、 B_j 、および B_k に接続されている。 B_{jk} 、 B_{ik} 、および B_{ij} は、コンビネーションシェイプであり、例えば、 B_{jk} は、 B_j と B_k の組み合わせを表す。 B_{ijk} は、3つのモーフトargetシェイプ B_i 、 B_j 、および B_k を結合したコンビネーションシェイプである。シェイプ x は、単位ウェイト（1）を有する B_k と、0.5のウェイトを有する B_i と、0.8のウェイトを有する B_j との組み合わせである。

【0038】

補間関数は、 n 次元の立方体上に構成され、当該 n 次元の立方体上の点にある制約シェイプを通過する。一実施形態では、補間関数は、次の形式を有してもよい。

【0039】

【数5】

$$f(W) = B_0 + \sum_{k=1}^n (w_k \Delta B_k) + \sum_{i=1}^m (\beta_i \Delta C_i)$$

【0040】

ここで、 B_0 は、ベースシェイプ（静止している幾何学的形状）であり、 ΔB_k は、ベースシェイプ B_0 に対する k 番目のモーフトargetシェイプ変形であり、 w_k は、補間される対応するモーフトargetシェイプウェイトである。よって、この式の最初の部分は、単なるモーフトargetシェイプウェイトの線形結合である。 ΔC_i は、ベースシェイプを含む、非線形補間関数で考慮される任意の低次元形状に対する i 番目の制約シェイプ変形である。言い換えると、 ΔC_i は、 i 番目の制約シェイプと前の補間レイヤーによって生成された形状との差である。 β_i は、 ΔC_i に適用される対応するモディファイア（係数、モディファイアの項、または制約ウェイトに類似）であり、補間関数が基準 $f(W) = S$ を満たすように算出される。

【0041】

よって、下記数式6で現される項は、すべてのモーフトargetシェイプの変位寄与を合計し、下記数式7で現される項は、単一のモーフトargetシェイプに基づくインクリメンタルシェイプ、コンビネーションシェイプ、およびコンビネーションインクリメンタルシェイプである制約シェイプを含む、全ての制約シェイプの変位寄与を合計する。

【0042】

【数6】

$$\sum_{k=1}^n (w_k \Delta B_k)$$

【0043】

【数7】

$$\sum_{i=1}^m (\beta_i \Delta C_i)$$

10

【0044】

補間関数を生成するために、モデル化されている対象に対応するモーフターゲットシェイプ $B_1 \dots n$ の組および制約シェイプ $C_1 \dots n$ の組が提供され、制約シェイプ $C_1 \dots n$ のそれぞれは、その構成モーフターゲットシェイプおよび制約シェイプが表すこれらモーフターゲットシェイプに対するウェイトに対応するメタデータに関連付けられる。

20

【0045】

例えば、図2では、形状Dは、ウェイトの組み合わせ $w_1 = 1$ 、 $w_2 = 1$ に対応するコンビネーションシェイプである制約シェイプである。図14は、図10Dに示す、リップサック (AU28) を表すモーフターゲットシェイプに対する0.5のウェイトに対応するインクリメンタルシェイプである制約シェイプである。図15は、100%のモーフターゲットシェイプAU06 (チークレイザー) と、80%のモーフターゲットシェイプAU12 (リップコーナープラー (lip corner puller)) と、60%のモーフターゲットシェイプAU26 (ジョーオープナー) との重み付き組み合わせに対応する、コンビネーションインクリメンタルシェイプである、制約シェイプである。よって、図15のシェイプが非線形補間関数に組み込まれる場合、 $W = \{100\% AU06, 80\% AU12, 60\% AU26\}$ を入力すると、図15のシェイプと同一の補間された形状が出力される。仮に、代わりに $W = \{90\% AU06, 80\% AU12, 60\% AU26\}$ の補間 (制約シェイプとして提供されていない) が求められる場合、補間関数は、図15の制約シェイプを含む関連する制約シェイプによって提供された情報を考慮して、図15の制約シェイプに比べて微妙に頬が隆起した、新たな補間された形状を出力する。

30

【0046】

要約すると、非線形インターポレータは、システムで構成されるすべての形状Sについて、非線形補間関数が成り立つ、つまり、 $f(W) = S$ となるように、与えられた補間ウェイト $w_1 \dots w_n$ に対して、変形 ΔB_k と ΔC_i およびモディファイア β_i を生成する。例えば、非線形関数に制約シェイプ C_u が導入される場合、 $w_1 = 1$ 、 $w_2 = 1$ のモーフターゲットシェイプ B_1 および B_2 に対応して、非線形補間式における制約シェイプ C_u に適用されるモディファイア β_u が、制約基準: $B_0 + \Delta B_1 + \Delta B_2 + \beta_u \Delta C_u = C_u$ を満たすように算出される。このようにして、所望の合形状が、さらなる制約シェイプの導入によって、常に再現可能である。

40

【0047】

非線形補間関数を用いるために、補間ウェイトW、つまり、補間される各モーフターゲットシェイプに対する補間ウェイトW、の組が、非線形補間関数への入力として受け取ら

50

れ ($w_1 \dots w_n$)、補間された形状が出力される。

【0048】

ウェイト $w_1 \dots w_n$ ($0 < w_k < 1$) の組は、何らかの方法で事前に指定されてもよい。例えば、アニメータが、図8に示すように、スライダーを介して各モーフターゲットシェイプに対して補間ウェイトを指定する。他の実施形態では、補間ウェイトは、キーフレームをアニメーション化するアニメーションシステムを介して自動的に提供されてもよく、ランダムに生成されてもよく、または他の方法で指定されてもよい。

【0049】

モーフターゲットシェイプおよび／または制約シェイプが、実在の対象からスキャンされたデータ、技術者の修正、または物理的にシミュレートされたデータ、または方法の組み合わせを含む、任意の適切な方法で作成されてもよい。

【0050】

補間方法は、モーフターゲットシェイプ／ブレンドシェイプまたは形状プリミティブに基づく変形可能な幾何学的形状コントロールに適用され得る。モーフターゲットシェイプは、機械、ボディ、ボディのコンポーネントなど、任意の変形可能な対象を表してもよい。例としては、人間または動物の体、顔、手、手足、または空想上の生物やキャラクタなどの人間以外のキャラクタが含まれるが、これらに限定されない。本発明の実施を、顔、特に人間の顔を表すモーフターゲットシェイプを参照して説明する。

【0051】

他の実施形態では、この方法は、顔の個性または解剖学的変化をブレンドするために用いられてもよい。同じ対象の表情をブレンドするのではなく、システムは、対象間の異なる個性または解剖学的変化を有する対象をブレンドするために、用いられ得る。例えば、技術者が、猫の顔、犬の顔の表情、およびコンビネーションシェイプとして技術者が考える犬と猫の交雑種の表情を作成してもよく、これと、技術者によって提供された制約シェイプを用いて、犬と猫の顔の間に重み付きブレンドを作成するために、補間システムは用いられてもよい。

【0052】

モーフターゲットシェイプは、特定の表情または微妙な表情の顔を表してもよい。人間の顔は、1つまたは複数の顔の筋肉を様々な程度に収縮させた結果、ほぼ無限の表情を形成できる。複雑な表情または動きが、フェイシャルアクションコーディングシステム (FACS) 分類法で定義された動きなどの基本的な動きに「分解」され得る。FACSは、顔の外観によって人間の顔の動きを分類する。表情が、特定のアクションユニット (AU) の組に分解することにより、FACSを用いてコード化され得る。よって、FACSは、特定の筋肉の動きによる顔の変化を記述するための信頼できる正確な方法を提供する。特定の感情表現を表す表情やビセムなどのコンプレックスシェイプが、重み付きAUの組に換算して記述されてもよい。他の実施形態では、モーフターゲットシェイプは、特定の音に対応する一般的な顔の画像であるビセムを表してもよく、これは、話された言語の音素または音の単位の視覚的同等物と見なされ得る。

【0053】

シェイプは、多次元空間内のメッシュまたはその他の点または点の組である可能性がある。例えば、シェイプは、幾何学的形状の状態を表す頂点 (点) の集合として定義されてもよい。各シェイプの点の集合は、トポロジ的に互いに一貫している。シェイプ入力は、形状の名前やカテゴリなどのメタデータに沿った頂点の行列として表されてもよい (モーフターゲットシェイプ、ベースシェイプ、制約シェイプ、コンビネーションシェイプなどであるかどうか)。いくつかの実施形態では、適切な幾何学的形状定義ファイル形式が、3Dモデル形式OBJ (.OBJ) ファイル形式などの形状頂点を入力するのに、用いられてもよい。シェイプファイルに名前を付けるための命名規則が、シェイプに関するメタデータを提供するのに、用いられてもよい。可能な命名規則の一例は次のとおりである: `Name1_weight1_Name2_weight2...`、ここで、`weighti = wi` は、 $0 < w_i \leq 1$ となる数値である。 $w_i = 1$ の場合、ウェイトは省略されて

10

20

30

40

50

もよい。すべて $w_i = 0$ のシェイプは、ベースシェイプ（または参照形状）であり、任意の名前を付けることができる。インターポレータは、補間されたシェイプを出力してもよく、これは、行列として、または他の任意の適切な方法で表されてもよい。

【0054】

非線形補間関数の計算：

図4は、一実施形態に係る非線形補間関数を事前計算するフロー図を示す。ステップ401で、シェイプが、モーフターゲットシェイプ、インクリメンタルシェイプ、またはコンビネーションシェイプとして分類される。

【0055】

ステップ402で、デルタが、モーフターゲットシェイプおよび制約シェイプに対して計算される。モーフターゲットシェイプおよび制約シェイプの絶対座標を格納する代わりに、デルタは、ベースシェイプおよび非線形補間関数においてすでに考慮されているその他の形状に対する変位ベクトルを格納する。

【0056】

ステップ403で、点のラティスが定義／計算される。モーフターゲットシェイプモデルは、モーフターゲットシェイプ（ターゲット）を n 次元ハイパーキューブの（少なくとも）いくつかの頂点に配置することと見なされ得て、原点をベースシェイプとし、ハイパーキューブエッジが対応するモーフターゲットシェイプに対するウェイトを表す。コンビネーションシェイプは、ハイパーキューブの頂点に配置されたラティスノードである。インクリメンタルシェイプは、ベースシェイプ（原点）とモーフターゲットシェイプターゲットの間のエッジに沿ったラティスノードである。コンビネーションインクリメンタルシェイプは、各構成モーフターゲットシェイプに対するウェイトに対応するラティス内の位置にあるラティスノードである。入力シェイプは、必ずしも完全なラティスを形成する必要はない。追加制約シェイプが、ラティスを完成または少なくとも拡張するために生成されてもよい。追加制約シェイプは、インクリメンタルシェイプ、コンビネーションシェイプ、および／またはコンビネーションインクリメンタルシェイプであってもよい。

【0057】

図6は、2Dラティスの例を示し、入力シェイプ（モーフターゲットシェイプおよび制約シェイプを含む）がドットで示され、生成される未知の点にクロスマークが付されている。言い換えると、生成される追加制約シェイプが、クロスマークで示されるようにウェイトを有して提供される。放射基底関数補間（RBF）、または別のメッシュレス補間法など、任意の適切な方法が、追加制約シェイプを生成するのに、用いられてもよい。この文脈においてメッシュレスは、補間に通常のメッシュが必要ないことを意味する。

【0058】

低次元サブ空間への分割

高次元の補間は計算コストがかかる可能性があるため、補間を容易にするために低次元空間への分割が行われてもよい。低次元空間に分割すると、追加制約シェイプの生成が容易になる場合がある。 d 次元ハイパーキューブの間に補間を適用する代わりに、当該ハイパーキューブはグリッドの配列に分割される。例えば、制約シェイプの各グループが、その増分次元に分割されてもよく、これは、組み合わせられているモーフターゲットシェイプの数を意味する。例えば、増分次元には次のものがある：下記数式8で現される項は1の増分次元 d を有し；下記数式9で現される項は2の増分次元を有し；下記数式10で現される項は3の増分次元を有する、など。これに続いて、その増分次元に従った各グループは、形成され得る次元 d のユークリッド空間の数に基づいて、さらにサブグループに分割されてもよい。各サブグループの完全なラティスが作成され、取得されたグリッドに欠落している関数値が、RBFなどの適切なメッシュレス補間法を用いて計算される。その結果、グリッドの配列が各増分次元に関連付けられる。

【0059】

【数8】

10

20

30

40

50

$$C_{w_i}^i$$

【 0 0 6 0 】

【 数 9 】

$$C_{w_i w_j}^{ij}$$

【 0 0 6 1 】

【 数 1 0 】

$$C_{w_i w_j w_k}^{ijk}$$

【 0 0 6 2 】

一実施形態に係る補間関数の生成の詳細な説明

簡単に言えば、補間は、各シェイプタイプを考慮し、補間精度を向上させることによって行われ得る。一例を以下に詳細に説明する。

【 0 0 6 3 】

モーフターゲットシェイプのみを用いる補間

任意の数のモーフターゲットシェイプ B_k が、次の式を用いて線形にブレンドされる。

【 0 0 6 4 】

【 数 1 1 】

$$f = B_0 + \sum_{k=1}^k (w_k \Delta B_k)$$

【 0 0 6 5 】

ここで、 B_0 は静止している幾何学的形状（ベースシェイプ）における形状の頂点を表し、 ΔB_k は、当該ベースシェイプに対する k 番目の基礎シェイプの変位を表すベクトルである（ $\Delta B_k = B_k - B_0$ ）。

【 0 0 6 6 】

コンビネーションシェイプを用いる補間

補間関数は、近似テイラー級数展開または他の適切な多変量補間システム、例えば、多変量ラグランジュ補間を用いて、コンビネーションシェイプ間を補間してもよい。抽象的に言えば、モーフターゲットシェイプは、 m 次元の周囲空間に埋め込まれた n 次元のシェイプ「集合体」のベースシェイプの接空間と見なされ得る。ゼロ以外のウェイト $w_i = w_j = 1$ に関連付けられたシェイプが、 B_{ij} として示され、 $0 < w_i < 1$ 、 $0 < w_j < 1$ に関連付けられたシェイプが、下記数式 12 で現される項として示される。すべてのシェイプ S の組が、重複しない 3つのサブセット、モーフターゲットシェイプ B_i 、コンビネーションシェイプ $B_{ij} \dots k$ 、および下記数式 13 で現されるインクリメンタルシェイプに分割される。ベースシェイプ B_0 を原点つまり参照形状としてとると、 $f(W) = S$ となるように補間関数 $f(w)$ が、作成される。ここで、 W は与えられたすべての補間ウェイトの組を表す。

【 0 0 6 7 】

【数 1 2】

$$C_{w_i w_j}^{ij}$$

【 0 0 6 8 】

【数 1 3】

$$C_{w_i w_j \dots w_k}^{ij \dots k}$$

10

【 0 0 6 9 】

テイラー展開に基づいて、点（ $w = 0$ ）の近傍にあるブレンドシェイプ関数 $f(w)$ は、次のように展開され得る。

【 0 0 7 0 】

【数 1 4】

$$f(w) = f(0) + \sum_{i=1}^N \frac{\partial f(0)}{\partial w_i} w_i + \sum_{i,j=1}^N \frac{1}{2!} \frac{\partial^2 f(0)}{\partial w_i \partial w_j} w_i w_j + \sum_{i,j,k=1}^N \frac{1}{3!} \frac{\partial^3 f(0)}{\partial w_i \partial w_j \partial w_k} w_i w_j w_k + \dots \quad (1)$$

20

or

$$f(w) = f(0) + \sum_{i=1}^N \frac{\partial f(0)}{\partial w_i} w_i + \sum_{i=1}^N \frac{1}{2!} \frac{\partial^2 f(0)}{\partial w_i^2} w_i^2 + \sum_{1 \leq i < j \leq N} \frac{\partial^2 f(0)}{\partial w_i \partial w_j} w_i w_j + \sum_{i=1}^N \frac{1}{3!} \frac{\partial^3 f(0)}{\partial w_i^3} w_i^3 + \sum_{1 \leq i < j \leq N} \frac{1}{2!} \frac{\partial^3 f(0)}{\partial w_i^2 \partial w_j} w_i^2 w_j + \sum_{1 \leq i < j < k \leq N} \frac{\partial^3 f(0)}{\partial w_i \partial w_j \partial w_k} w_i w_j w_k + \dots (2)$$

30

【 0 0 7 1 】

展開式（2）において特定の項を保持すると、異なるレベルの精度で近似が生成される。結果として各シェイプタイプを考慮し、補間精度を向上させることにより、補間が行われ得る。モーフターゲットシェイプのみが与えられた場合、以下ようになる。

【 0 0 7 2 】

【数 1 5】

$$f(w) = f(0) + \sum_{i=1}^N \frac{\partial f(0)}{\partial w_i} w_i \quad (3)$$

40

【 0 0 7 3 】

以下の導関数に対して単位ステップサイズ（つまり、 $\Delta w_j = 1$ ）の有限差分を用いる。

【 0 0 7 4 】

【数 1 6】

50

$$f(0) = B_0, \quad (4)$$

$$\frac{\partial f(0)}{\partial w_i} = B_i - B_0, \quad \rightarrow$$

$$f(w) = B_0 + \sum_{i=1}^N (B_i - B_0) w_i \quad (5)$$

10

【0075】

コンビネーションシェイプ $B_{ijk} \dots$ が与えられると、混合偏導関数が、有限差分近似で (2) に保持され得る。

【0076】

【数17】

$$\frac{\partial^2 f(0)}{\partial w_i \partial w_j} = B_{ij} - B_i - B_j + B_0, \quad (6)$$

$$\frac{\partial^3 f(0)}{\partial w_i \partial w_j \partial w_k} = B_{ijk} - B_{ij} - B_{ik} - B_{jk} + B_i + B_j + B_k - B_0$$

20

【0077】

高次導関数である下記数式18が0であると、次の数式19の形式の線形（個々の座標に対する）モデルが生成される。

【0078】

【数18】

$$\frac{\partial^m f(0)}{\partial w_i^m}$$

30

【0079】

【数19】

$$f(w) = B_0 + \sum_{i=1}^N (B_i - B_0) w_i + \sum_{1 \leq i < j \leq N_2} (B_{ij} - B_i - B_j + B_0) w_i w_j \quad (7)$$

$$+ \sum_{1 \leq i < j < k \leq N_3} (B_{ijk} - B_{ij} - B_{ik} - B_{jk} + B_i + B_j + B_k - B_0) w_i w_j w_k + \dots$$

40

【0080】

または、次のコンパクトな形式で生成される。

【0081】

【数20】

$$f(w) = B_0 + \sum_{i=1}^N \Delta B_i w_i + \sum_{1 \leq i < j \leq N_2} \Delta B_{ij} w_i w_j + \sum_{1 \leq i < j < k \leq N_3} \Delta B_{ijk} w_i w_j w_k + \dots \quad (8)$$

50

【0082】

対応するコンビネーションシェイプが与えられる場合のみ、追加の項が（8）に存在する。与えられた値（コンビネーションシェイプ値）を補間点にとって、次のようになるように補間関数を作成することによって、係数 ΔB_{ijk} が、取得され得る。

【0083】

【数21】

$$f(0,0,\dots, w_i = 1, w_j = 1, w_k = 1, 0, \dots) = B_{ijk} \quad (9)$$

10

【0084】

その結果、次のようになる。

【0085】

【数22】

$$\Delta B_i = B_i - B_0 \quad (10)$$

$$\Delta B_{ij} = B_{ij} - \left(B_0 + \sum_{p=\{i,j\}} \Delta B_p \right)$$

20

$$\Delta B_{ijk} = B_{ijk} - \left(B_0 + \sum_{p=\{i,j,k\}} \Delta B_p + \sum_{p,q=\{i,j,k\}} \Delta B_{pq} \right)$$

【0086】

30

インクリメンタルシェイプを用いる補間

補間関数は、テンソルスプライン補間、三角多項式補間ネットワーク、または薄板スプライン補間を含むがこれらに限定されない、任意の適切な規則的または不規則なグリッド補間システムを用いて、インクリメンタルシェイプの間を補間してもよい。ここで使用できる補間手法は、ハイパーキューブの完全性に依存するものであってもよい。ラティスが完全なラティスである場合、テンソルスプライン補間が好ましい場合がある。ラティスが不完全／部分的なラティスである場合、三角多項式補間ネットワークまたは薄板スプライン補間が用いられてもよい。別の実施形態では、キュービックエルミートスプラインが、インクリメンタルシェイプを組み込むのに、用いられてもよい。エルミート多項式のパラメータが、補間オーバーシュートを最小限に抑えるための制約とともに、近くのノードから自動的に推定されてもよく、その場合、非線形インクリメンタルネットワークを設定する必要がなくなる。

40

【0087】

インクリメンタルシェイプを用いる補間の次の例では、テンソルスプライン補間が用いられる。モデル（8）は準線形である。インクリメンタルシェイプを考慮すると、非線形性が追加される。下記数式23で現されるインクリメンタルシェイプが、対応するインクリメンタルシェイプが存在する場合にゼロ以外のモディファイア項を用いて、組み込まれる。

【0088】

【数23】

50

$$C_{w_i w_j w_k}^{ijk}$$

【 0 0 8 9 】

【 数 2 4 】

$$f(w) = B_0 + \sum_{i=1}^N \Delta B_i w_i + \sum_{1 \leq i \leq M_1} h_i (\Delta C_{w_i}^i, w_i) + \sum_{1 \leq i < j \leq N} \Delta B_{ij} w_i w_j + \sum_{1 \leq i < j \leq M_2} h_{ij} (\Delta C_{w_i w_j}^{ij}, w_i, w_j) \\ + \sum_{1 \leq i < j < k \leq N} \Delta B_{ijk} w_i w_j w_k + \sum_{1 \leq i < j < k \leq M_3} h_{ijk} (\Delta C_{w_i w_j w_k}^{ijk}, w_i, w_j, w_k) + \cdots (11)$$

10

【 0 0 9 0 】

ここで、下記数式 2 5 で現される関数は、次の数式 2 6 ように案出される。

【 0 0 9 1 】

【 数 2 5 】

$$h_{ijk} (\Delta C_{w_i w_j w_k}^{ijk}, w_i, w_j, w_k)$$

20

【 0 0 9 2 】

【 数 2 6 】

$$h_{ijk} (\Delta C_{w_i w_j w_k}^{ijk}, 0, 0, 0) = 0 \quad (12)$$

$$h_{ijk} (\Delta C_{w_i w_j w_k}^{ijk}, 1, 1, 1) = 0$$

30

$$h_{ijk} (\Delta C_{w_i w_j w_k}^{ijk}, w_i, w_j, w_k) = \Delta C_{w_i w_j w_k}^{ijk}$$

40

【 0 0 9 3 】

関数 h は、結合点のみを用いて実行された補間の上に、インクリメンタルシェイプの点の間を補間し、テンソルスプライン補間が、関数 h を実施するために用いられる。補間表現 (11) は、与えられた形状値を正確に補間点に取り、補間される点に非線形近似する。関数 h は、 M 次元空間でテンソル積スプライン法を用いて定義される。 $n = 2^M$ の頂点 $\{V_m\}$ 、 $m = 1, \dots, n$ および対応する関数値 $\{F_m\}$ 、 $m = 1, \dots, n$ を有するハイパーキューブを想定する。すべての点 V_m は、次元ごとに 2 つの異なる値 (v_i^a 、 v_i^b) を有するため、このハイパーキューブ内の任意の点 x に対して、 $x_i \in [v_i^a, v_i^b]$ となる。次に、 $x_i \in [v_i^a, v_i^b]$ の場合、次のような線分が、

【 0 0 9 4 】

50

【数 2 7】

$$h(x) = y^a \left(1 - \frac{x_i - v_i^a}{v_i^b - v_i^a} \right) + y^b \frac{x_i - v_i^a}{v_i^b - v_i^a} \quad (15)$$

【0 0 9 5】

2つの値 y^a と y^b を接続する。そして、これらの値は、 x_j の関数でもある。ここで、関数 $y^a(x_j)$ と $y^b(x_j)$ が、区間 $[v_j^a, v_j^b]$ の x_j に対して定義される。

10

【0 0 9 6】

【数 2 8】

$$y^a(x_j) = y^{aa} \left(1 - \frac{x_j - v_j^a}{v_j^b - v_j^a} \right) + y^{ab} \frac{x_j - v_j^a}{v_j^b - v_j^a} \quad (16)$$

$$y^b(x_j) = y^{ba} \left(1 - \frac{x_j - v_j^a}{v_j^b - v_j^a} \right) + y^{bb} \frac{x_j - v_j^a}{v_j^b - v_j^a}$$

20

【0 0 9 7】

その結果、ハイパーキューブ内で補間を行う関数 h を計算するために反復シーケンスが取得される。下記数式 2 9 で現されるインクリメンタルシェイプのグリッドデータの組が得られ、補間する点 (x_i, x_j, x_k) が与えられると、点 (x_i, x_j, x_k) が属する対応するハイパーキューブが求められた後、(15)、(16) を用いて補間が行われる。

【0 0 9 8】

30

【数 2 9】

$$\left\{ (v_i, v_j, v_k), \Delta C_{v_i v_j v_k}^{ijk} \right\}$$

【0 0 9 9】

記載された実施形態によれば、補間アルゴリズムは、2つのステージを含んでもよい。

【0 1 0 0】

事前計算ステージ

40

・すべての形状を、3つのサブセット、つまり、基礎シェイプ、コンビネーションシェイプ、インクリメンタルシェイプ、に分割する。

・(10) と (14) から ΔB と ΔC を計算する。

・ ΔC を下記数式 30 ～ 32 で現される増分次元に基づいてグループに分割する。

・増分次元 d の ΔC の各グループは、形成され得る次元 d のユークリッド空間の数に基づいてサブグループに分割される。各サブグループの完全なラティスが構築され、取得されたグリッドに欠落している関数値が RBT 法から計算される。その結果、グリッド g_{ijk} の配列が下記数式 33 で現されるグループに関連付けられる。

【0 1 0 1】

【数 3 0】

50

$$\Delta C_{w_i}^i,$$

【0102】

【数31】

$$\Delta C_{w_i w_j}^{ij},$$

【0103】

【数32】

$$\Delta C_{w_i w_j w_k}^{ijk}$$

【0104】

【数33】

$$\Delta C_{w_i w_j w_k}^{ijk} \dots$$

【0105】

補間実行ステージ

・補間される点 $w = (w_1, w_2, \dots, w_N)$ が与えられた場合、(8) から $f(w)$ を計算する。

・すべてのグリッド g_{ijk} に対して反復適用して、点 $(w_i, w_j, w_k, \dots)$ が属するグリッドセルを求め、(15) ~ (16) から関数 h_{ijk} を計算する。

・最終結果は、(8) からの $f(w)$ の合計と h_{ijk} のすべての値となる。

【0106】

利点

非線形で制約シェイプを結合してコンピュータグラフィックスアニメーションを実行することで、対象のリアルな制約シェイプを提供することによって、望ましくない形状の生成が防がれる。インターポレータは、モーフターゲットシェイプのブレンドを改善する。モーフターゲットシェイプの組み合わせおよび／またはウェイトであるシェイプは、制約シェイプとしてモデルに入力されるため、再現可能である。制約シェイプは、非線形補間関数を介して補間モデルで考慮される。よって、インターポレータは、予測可能で直感的な様式で機能する。

【0107】

非線形変形はインターポレータで自動的に考慮されるため、デジタルモデラーは従来の例ベースの方法の場合のように多くの例を作成する必要がないため、現実的なモデルを開発するために必要な時間、労力、およびリソースが削減される。アニメーターが処理する必要があるものの次元数が低減されるため、インターポレータは使いやすくなる。制約シェイプは、追加の次元ではなく、ベースモーフターゲットシェイプで非線形の項として考慮される。例えば、スライダーベースのアニメーションシステムでは、実際には、アニメーターが210以上のスライダーよりも60のスライダーを制御の方が簡単である。コンビネーションシェイプおよび／またはインクリメンタルは、手動で制御または具体的に考慮される必要のある追加の形状ではなくなった。制御はよりスムーズで使いやすくなってい

10

20

30

40

50

る。

【0108】

リアルタイム補間はインターポレータを用いる線形プロセスであるため、スペックの低いマシンでも高速でリアルタイムの補間が実現され得る。モーフターゲットシェイプの数、制約されるコンポーネントの数は、インターポレータの性能に影響を与える可能性があるが、リアルタイム補間のコストが固定されているため、主に事前計算時間に影響する。

【0109】

テクスチャが、同様の方法でブレンドされてもよい。頂点の位置をブレンドするのではなく、色と透明度のチャンネルがブレンドされてもよい（例えば、R、G、B、A）。同様に、インクリメンタルテクスチャとコンビネーションテクスチャの例が作成され、ここで説明する補間方法に従ってチャンネルがブレンドされる。

【0110】

モーフターゲットシェイプマッパー

モーフターゲットシェイプは、コンポーネントシェイプであってもよく、「基礎シェイプ」とも呼ばれ、言い換えれば、静止状態からの独自の基本的な動きを表す形状であり、さらなる形状を作成するための基本的な構築用ブロックと見なされてもよい。言い換えれば、それらは形状の基本的な変形である。FACSアクションユニットは、複雑な表情がAUの混合または組み合わせとして解釈され得るため、コンポーネントシェイプつまり「基礎シェイプ」の良い例である。顔のアニメーションの設定により、アニメータはAUを混合することによって複雑な表情を構築して自然な変形を生成することが可能となる。ピセムは、コンポーネントシェイプ（「基礎シェイプ」）と見なされてもよく、見なされなくてもよい。図13は、ピセム「AI」A、ピセム「I」B、ピセム「O」C、ピセム「TS」D、ピセム「V」Eを含むピセムの例を示す。

【0111】

マッパーは、ユーザがモーフターゲットシェイプ補間方法を用いずに幾何学的な点の集合に換算して望ましい例の形状を作成した場合など、重み付き組み合わせが演繹的に分からない場合に、どのコンポーネントシェイプの重み付き組み合わせが特定の形状を反映するかを把握する問題に対処する。言い換えると、マッパーは、未知のウェイト（未解決の形状）を用いて、付与された形状を推定するコンポーネントシェイプの組み合わせを算出する。例えば、アニメータは、笑いの表情を見せけている人物のスキャンされた3Dモデルをモーフターゲットシェイプとして使用したい場合があるが、対応するコンポーネントシェイプおよび／またはコンポーネントシェイプウェイトにこの表情を手動でマッピングするのは難しい場合がある。さらに、そのコンポーネントシェイプのうちの1つの幾何学的形状が修正されると、マッピング結果が古くなる可能性がある。したがって、アニメータでマップされたシーケンスは、絶えず進展する開発環境においてエラーが発生しやすくなる。

【0112】

マッパーによって、例の形状頂点をコンポーネントシェイプ頂点にマッピングして、これらのウェイトを推定できる。一実施形態では、マッパーは、未解決シェイプの一部であることが分かっているモーフターゲットシェイプの数（または、結果に含まれることになる特定の形状の数）に応じたペナルティ項を有する最小二乗問題を用いて、未解決シェイプの基礎となるコンポーネントシェイプと各ウェイトを推定する。

【0113】

【数34】

$$\|T - f(w)\|^2 + k \sum_i \|\lambda_i w_i\|$$

【0114】

ソリューション境界、 $0 \leq w_i \leq 1$ に従い、ここでは、 T はターゲットシェイプであり、 $f(w)$ は形状組み合わせモデルであり、 λ_i はペナルティである。マッパーは、名前付きの形状データセットと推定される未解決シェイプの頂点を入力として受け取る。

【0115】

図16は、未解決シェイプに対応する、ユーザ指定のコンポーネントシェイプウェイトの値を推定する1つの方法のフロー図を示している。ステップ1601で、次元空間を構築し、すべてのコンポーネントシェイプをこのサブ空間に投影することによって、次元削減が行われる。ステップ1602で、未解決シェイプをサブ次元空間に投影することにより、未解決シェイプの次元数が削減される。ステップ1603で、最小二乗ソルバーが生成される。最小二乗ソルバーは、次の形式を有してもよい。

【0116】

【数35】

$$\min \left\| \sum_{k=1}^n w_k \Delta T_k - \Delta F \right\| + \mu \sum_{k=1}^n \left\| \lambda_k w_k \right\|.$$

【0117】

ここで、 ΔT_k と ΔF は、それぞれ次元削減された基礎シェイプと例の形状変位であり、係数 λ_i は、ユーザ指定のコンポーネントシェイプウェイトのペナルティを反映している。ステップ1604で、コンポーネントシェイプウェイトが、最小二乗問題を解くことにより、生成される。

【0118】

コンポーネントシェイプの提案

ユーザは、マッパーがマッピングするコンポーネントシェイプについて提案を行うことができる。これを可能にするために、提案されたコンポーネント λ_i のペナルティ係数が低減され、これにより、これらの提案されたコンポーネントシェイプがマッピングプロセス中に用いられる可能性が高まる。さらに、ソリューション境界は、提案されたコンポーネントシェイプに対して、 $0 \leq w_i \leq 1$ に変更される。

【0119】

別の実施形態では、コンポーネントシェイプとそれらの関連するウェイトの両方が、提案され実現され得る。この場合、ウェイトの正確な値は、最小二乗ソルバーにおいて、提案されたコンポーネントシェイプに対して定義される。

【0120】

次元削減

形状メッシュには多数の頂点が含まれる場合があるため、最小二乗問題の計算は非効率的で時間がかかる場合がある。したがって、最小二乗問題を解く前に、各形状の次元数が削減されてもよい。次元削減が、特異値分解と、主要な特異ベクトルの基礎を有する低次元空間へのデータの投影、またはその他の適切な方法によって行われてもよい。数値シミュレーション実験は、6つの主要な特異ベクトルを取ることで、顔モデルの信頼できるウェイト推定を取得できることを示しているが、他の適切な数の主要な特異ベクトルが用いられてもよい。

【0121】

シェイプの命名規則が、アニメーションを容易にするために用いられてもよい。例えば、2つのアンダーライン「__ __」が、補間形状から未知の形状名を示すために用いられてもよい。

・EXsmile__AU06__AU12__AU26__25__は、事前定義された組み合わせノードとAU26の既知のウェイトを有する笑顔の形状である（AU06とAU12

10

20

30

40

50

も提案されるが、それらのウェイトは推定する必要がある)。

・PhBut___は、組み合わせノードを推定する必要がある音素「しかし」の形状である。PhBut___の異形は、叫ぶ音素(PHbuts shout)またはサイレント音素(PHbutsilent)であってもよく、各異形は、マッパーを介してFACS空間に個別にマッピングされる。

【0122】

アニメーションシステム

アニメーションフレームワーク14が、2つ以上のコンビネーションシェイプを含むコンプレックスシェイプの、アニメーションターゲットまたはコントロールシェイプとしての使用を可能にするために提供されてもよい。複数の基礎となるコンポーネントシェイプからなる複雑な表現などのコンプレックスシェイプは、アニメータが数個のコンポーネントシェイプに対して個別にウェイトを設定する必要があるため、アニメーション化するのが難しく、時間がかかる場合がある。単一のコントロールで、コンプレックスシェイプをアニメーション化できるようにすることが望ましい。しかし、複雑な形状のコンポーネントは線形に変形しない場合があるため、コンプレックスシェイプとベースシェイプの間を単に補間しても、リアルな感じに見えない場合がある。また、コンポーネントシェイプの組み合わせも、それらの一部の合計と等しくならない様式で補間する場合がある。もう1つの問題は、基礎となるコンポーネントシェイプが不明であり手動で確認するのが難しい場合がある特徴的な形状のアニメーション化をどのように可能にするかである。例えば、顔のアニメーションの関係において、一部の俳優は、経験豊富なアニメータでさえ、基礎となるコンポーネントの動き(FACS AUなど)にマッピングするのに苦勞する可能性がある、独特の複雑な表情をする。

【0123】

アニメーションフレームワーク14は、コンプレックスシェイプ(コンプレックスモーフターゲットシェイプ)をそれらの構成コンポーネントシェイプにマッピングして、ターゲットコンポーネントシェイプの組を形成する。構成コンポーネントシェイプおよび/またはウェイトが不明な場合、マッパーは基礎となるコンポーネントシェイプとそれらの各ウェイトを把握する。インターポレータは、重み付き基礎となるターゲットコンポーネントシェイプの間を補間する。

【0124】

詳細な説明と実施形態

モーフターゲットシェイプアニメーションフレームワーク14は、ユーザへの柔軟性を付加しつつ、非線形モーフターゲットシェイプ補間システムとモーフターゲットシェイプマッパーを1つのシステムに組み込むように設計されている。エンドユーザは、形状を、コンポーネントシェイプ(基礎シェイプ)および例の形状に分割せずに、アニメーションとして見なして、操作および作成することができる。アニメーション形状は、アニメーションウェイトを介して制御され得て、アニメーションシステムは、すべてのウェイトをコンポーネントシェイプウェイトにマッピングし、モーフターゲットシェイプ補間を行うことができる。

【0125】

図17は、アニメーション方法の概略フロー図を示している。ソーター11が、複数のコントロールシェイプを受け取り、各コントロールシェイプを処理する。例えば、ソーター11は、シェイプ{AU12, AU26, AU12__AU26, Smile__AU12__AU26}を受け取る。コントロールシェイプが、その基礎となるコンポーネントシェイプが不明なコンプレックスシェイプである場合、当該コントロールシェイプは、マッパー12に入力され、マッパー12は、基礎となるコンポーネントシェイプとそれらの各ウェイトを出力する。次に、この結果がアニメーションフレームワーク14に渡され、アニメーションフレームワーク14は、ユーザ提供のコントロールウェイトをコンポーネントシェイプウェイトにマッピングする。インターポレータ13は、各コントロールシェイプの基礎となるコンポーネントシェイプの間を補間することにより、コントロールシェイ

プの間を補間する。

【0126】

ウェイトが、行列を介してコンポーネントシェイプウェイトにマッピングされるコントロールシェイプに対して提供され、次に、インターポレータは、コンポーネントシェイプウェイトを入力として用いて、補間された形状を出力する。コントロールシェイプは、コンポーネントシェイプ、コンプレックスシェイプ、または、コンポーネントシェイプとコンプレックスシェイプの組み合わせであってもよい。コントロールシェイプに、コントロールシェイプのコンポーネントであることが分かっている任意のコンポーネントシェイプを記述するメタデータが含まれていてもよい。

【0127】

いくつかの例は次のとおりである。

- ・AU12：コンポーネントシェイプ
- ・AU26：コンポーネントシェイプ
- ・AU12__AU26：2つのコンポーネントシェイプからなるコンプレックスシェイプ：存在するコンポーネントシェイプは、すでに既知であり、命名規則から明らかである。
- ・Smile__AU12__AU26.05：そのコンポーネントがAU12__AU26.05であることが分かっているコントロールシェイプ。したがって、このコントロールシェイプをマッパーに渡す必要はなく、アニメーションフレームワークに直接渡すことができ、アニメーションフレームワークは、基礎となるコンポーネントシェイプとウェイトをインターポレータに渡す。
- ・Laugh：そのコンポーネントが不明であるコントロールシェイプ。このコントロールシェイプをマッパーに渡すことができ、マッパーは、構成コンポーネントシェイプとそれらの各ウェイトを算出する。
- ・Happy__Laugh__Smile：2つの他のコントロールシェイプ（コンプレックスシェイプである）からなるコントロールシェイプ。

【0128】

図18は、アニメーションフレームワーク14を用いるアニメーションの概略図を示している。コントロールシェイプに対するアニメーションコントロールウェイト1801が、アニメーションフレームワーク14に提供される。アニメーションフレームワーク14は、コントロールシェイプ1803と基礎となるコンポーネントシェイプ1802の間のマッピングを提供する。アニメータに、コンポーネントシェイプに対するウェイトを提供する（数個のコンポーネントシェイプをコントロールシェイプとして用いる）か、対応する事前定義されたコンプレックスシェイプをコントロールシェイプとして用いるかのいずれかによって、コンプレックスシェイプをアニメーション化する選択肢があってもよい。図18に示すように、ユーザは、AU01LとAU01R、つまりコントロールシェイプのそれぞれに、1のウェイトを指定するか、AU01（アニメーションフレームワーク14を介してAU01LとAU01Rに内部的にマッピングされる）に1のウェイトを指定して、等価の合成形状を作成してもよい。インクリメンタルシェイプAU01__50が、内部的にAU12L__50とAU01R__50にマッピングされ、インターポレータは、AU12L__50とAU01R__50の組み合わせを補間する。よって、AU01は、1つのスライダーを切り替えるだけでよい。アニメータに便利さを提供するが、AU01は、別のコンポーネントシェイプ／モーフターゲットシェイプとして追加されないため、インターポレータに計算の複雑さを付加しない。コンポーネントシェイプがインターポレータ13に渡され、インターポレータ13は、マッピングされたウェイトを用いてコンポーネントシェイプの間を補間し、補間1806の結果がアニメーションに用いられる。

【0129】

コンプレックスシェイプ

コンプレックスシェイプは、複数のコンポーネントシェイプで構成される。例えば、コンプレックスシェイプは、次の表に示すように、複数のAUからなる感情を表してもよい。

【0130】

10

20

30

40

50

【表 1】

感情	アクションユニット
幸福	6 + 1 2
悲しみ	1 + 4 + 1 5
驚き	1 + 2 + 5 B + 2 6
恐怖	1 + 2 + 4 + 5 + 7 + 2 0 + 2 6

【0131】

コンポーネントシェイプは、モーフターゲットシェイプのサブカテゴリであり、そのウェイトは、補間関数の入力値の空間において基礎ベクトルを形成する。コンポーネントシェイプは、静止状態からの独自の基本的な動きを表す。FACS AUは、複雑な表情がAUの混合または組み合わせとして解釈され得るため、コンポーネントシェイプの良い例である。顔のアニメーションの設定により、アニメータはAUを混合することによって複雑な表情を構築して自然な変形を生成することが可能となる。ピセムは、スピーチに対する基本的なコンポーネントシェイプと見なされ得るが、それら自体はAUで構成されている。よって、コンポーネントシェイプつまり「基礎シェイプ」としての分類は抽象化の程度に依存することが分かる。

【0132】

左右対称形状は、コンポーネントシェイプではなく、コンポーネントシェイプの組み合わせであるコンプレックスシェイプと見なされてもよい。図11は、左右対称なFACS AU12に対応する、左右対称な表情、「左唇角プラー（left-lip-corner-puller）」Aと「右唇角プラー（right-lip-corner-puller）」Bの左右のモーフターゲットシェイプを示している。コンプレックスシェイプ（AU12__AU12L__AU12R）は、唇の両角を引っ張る表情（笑顔）のためのコントロールシェイプとして、用いられ得る。アニメーションフレームワーク14は、コントロールシェイプを、そのコンポーネントシェイプAU12LおよびAU01Rにマッピングし、インターポレータ13は、コンポーネントシェイプAU12LおよびAU01Rを補間して、笑顔を作成する。

【0133】

利点

アニメーションフレームワークによって、単一のコントロールで、基礎となるコンポーネントシェイプの相互作用と補間の複雑さを考慮に入れて、コンプレックスシェイプをリアルな感じにアニメーション化することが可能となる。ピセム、感情、その他の表情などのコンプレックスシェイプが、アニメーションコントロールとして定義され得るため、補間システムの次元数を増やすことはない。アニメータは、シェイプの基礎となるコンポーネントシェイプを簡単に確認できない場合でも、コンプレックスシェイプを用いて、アニメーション化できる。

【0134】

解釈

説明した方法およびシステムは、任意の適切な電子計算システムで利用されてもよい。以下に説明する実施形態によれば、電子計算システムが、様々なモジュールおよびエンジンを用いて本発明の手法を利用する。

【0135】

電子計算システムは、少なくとも1つのプロセッサと、1つまたは複数のメモリ装置または1つまたは複数のメモリ装置に接続するためのインターフェースと、システムが一人または複数のユーザまたは1つまたは複数の外部システムから命令を受け取り当該命令に従って動作できるようにするために外部装置に接続するための入力および出力インターフ

エースと、様々なコンポーネント間の内部および外部通信用のデータバスと、適切な電源とを備える。さらに、電子計算システムは、外部および内部装置と通信するための1つまたは複数の通信装置（有線または無線）と、ディスプレイ、ポインティングデバイス、キーボードまたは印刷装置などの1つまたは複数の入力／出力装置とを備えてもよい。

【0136】

プロセッサは、メモリ装置内にプログラム命令として格納されたプログラムのステップを実行するように構成されている。プログラム命令は、本明細書に記載された発明を実行する様々な方法を実行することを可能にする。プログラム命令は、例えば、Cベースの言語およびコンパイラなどの任意の適切なソフトウェアプログラミング言語およびツールキットを用いて開発または実装されてもよい。さらに、プログラム命令は、それらがメモリ装置に転送されるか、または例えばコンピュータ読み取り可能な媒体に格納されるなど、プロセッサによって読み取られることができるように、任意の適切な方法で格納されてもよい。コンピュータ読み取り可能な媒体は、例えば、ソリッドステートメモリ、磁気テープ、コンパクトディスク（CD-ROMまたはCD-R/W）、メモリカード、フラッシュメモリ、光ディスク、磁気ディスク、またはその他の適切なコンピュータ読み取り可能な媒体などの、プログラム命令を有形に格納するための任意の適切な媒体であってもよい。

【0137】

電子計算システムは、関連するデータを検索するために、データ記憶システムまたは装置（例えば、外部データ記憶システムまたは装置）と通信するように構成されている。

【0138】

本明細書に記載のシステムは、本明細書に記載の様々な機能および方法を実行するように配置された1つまたは複数の要素を備えることが理解されよう。本明細書で説明する実施形態は、機能を実装できるようにするために、システムの要素を構成する様々なモジュールおよび／またはエンジンを相互接続する方法の例を読者に提供することを目的としている。さらに、記載の実施形態は、システムに関連する詳細において、本明細書に記載の方法のステップを実行できる方法を説明している。概念図は、様々なデータ要素が様々な異なるモジュールおよび／またはエンジンによって様々な段階で処理される方法を読者に示すために提供されている。

【0139】

モジュールまたはエンジンの配置および構造は、システムおよびユーザの要件に応じて適宜適合されてもよく、その結果、様々な機能が、本明細書に記載されるものとは異なるモジュールまたはエンジンによって実行されてもよく、特定のモジュールまたはエンジンが組み合わされて単一のモジュールまたはエンジンとなってもよいことが理解される。

【0140】

記載のモジュールおよび／またはエンジンは、任意の適切な形態の技術を用いて、命令とともに実装、提供されてもよい。例えば、モジュールまたはエンジンは、任意の適切な言語で書かれた任意の適切なソフトウェアコードを用いて実装または作成されてもよく、そして、当該コードは、任意の適切な計算システム上で実行され得る実行可能プログラムを生成するためにコンパイルされる。代替的に、または実行可能プログラムと共に、モジュールまたはエンジンは、ハードウェア、ファームウェア、およびソフトウェアの任意の適切な組み合わせを用いて、実装されてもよい。例えば、モジュールの一部は、特定用途向け集積回路（ASIC）、システム・オン・チップ（SoC）、フィールドプログラマブルゲートアレイ（FPGA）、または任意の他の適切な適応可能またはプログラム可能な処理装置を用いて実装されてもよい。

【0141】

本明細書に記載の方法は、記載のステップを実行するように特別にプログラムされた汎用計算システムを用いて、実行されてもよい。代替的に、本明細書に記載の方法は、データ分類および視覚化コンピュータ、データベースクエリコンピュータ、グラフィカル分析コンピュータ、データ分析コンピュータ、製造データ分析コンピュータ、ビジネスインテリジェンスコンピュータ、人工知能コンピュータシステムなどの特定の電子コンピュータ

10

20

30

40

50

システムを用いて、実行されてもよく、当該コンピュータは、特定の分野に関連する環境から取り込まれた特定のデータに対して記載のステップを実行するように特別に適合されたものである。

【符号の説明】

【0142】

- 1 形状
- 2 モーフターゲットシェイプ
- 3 ベースシェイプ
- 4 制約シェイプ
- 5 インクリメンタルシェイプ
- 6 コンビネーションシェイプ
- 7 コンビネーション－インクリメンタルシェイプ
- 8 モディファイア
- 9 コンポーネントシェイプ
- 10 コンプレックスシェイプ
- 11 ソーター
- 12 マッパー
- 13 インターポレータ
- 14 アニメーションフレームワーク
- 15 コントロールシェイプ
- 16 未解決シェイプ
- 17 入力
- 18 付加
- 19 ラティス
- 20 モーフターゲットアニメーションモデル
- 21 補間関数

10

20

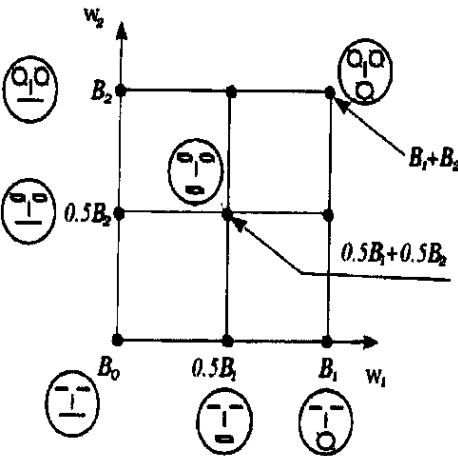
30

40

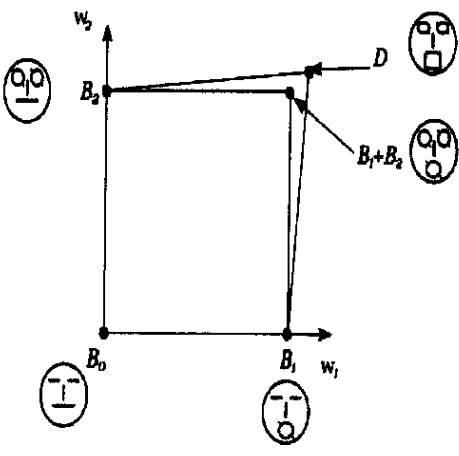
50

【図面】

【図 1】

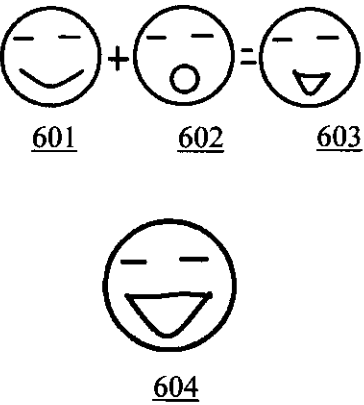


【図 2】

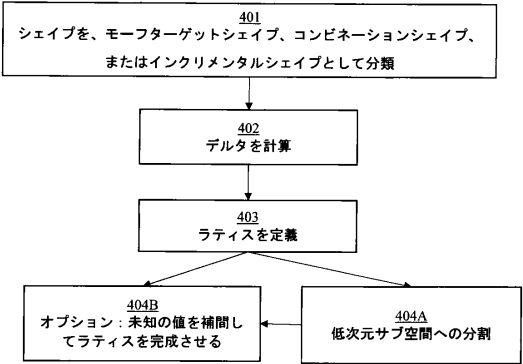


10

【図 3】



【図 4】



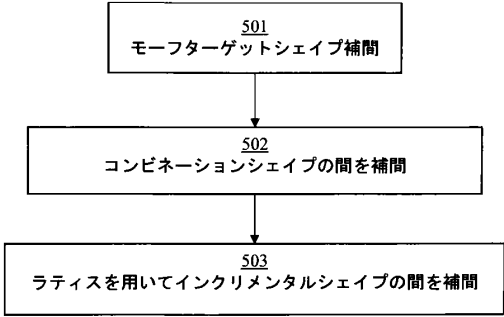
20

30

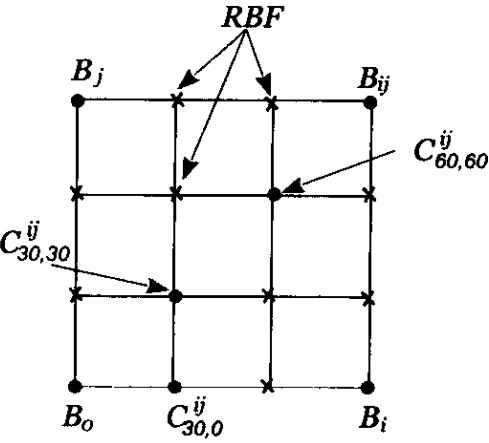
40

50

【図 5】

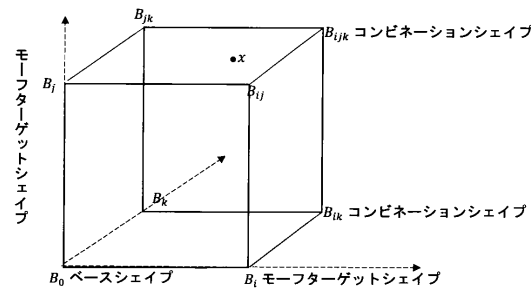


【図 6】

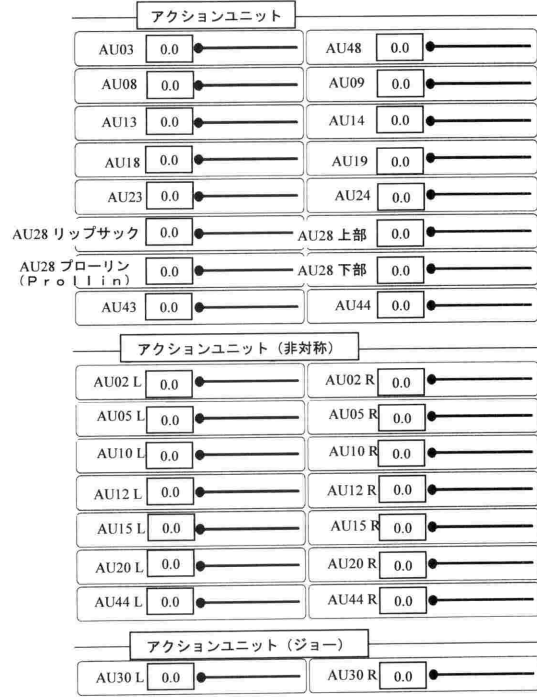


10

【図 7】



【図 8】



20

30

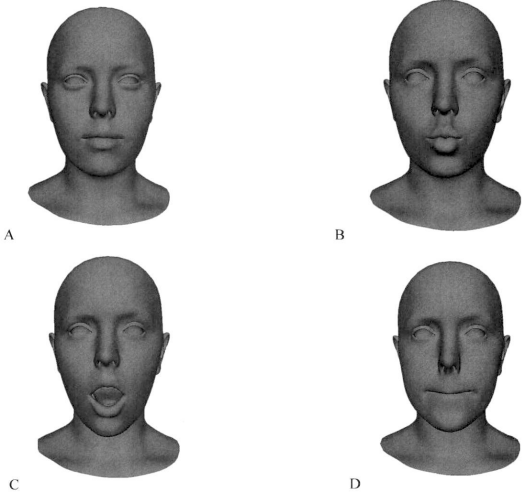
40

50

【図 9】

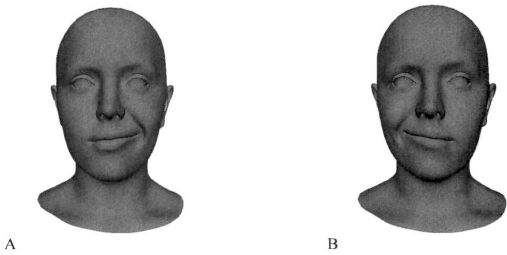


【図 10】

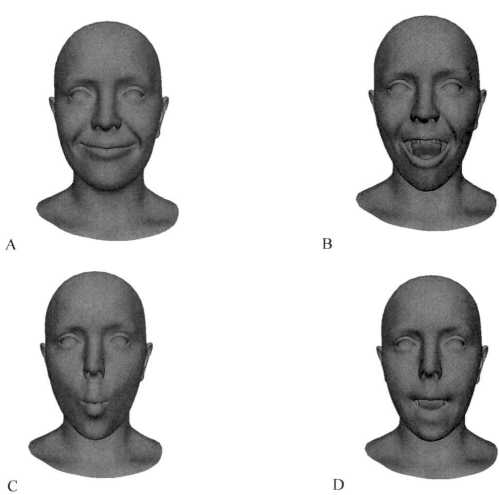


10

【図 11】



【図 12】



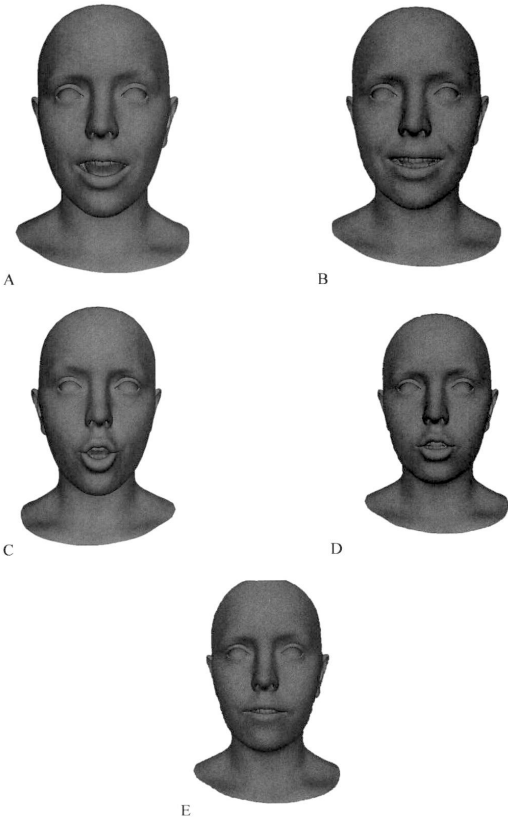
20

30

40

50

【図 1 3】



【図 1 4】



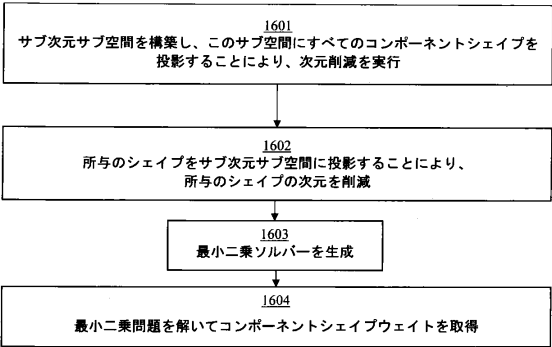
10

20

【図 1 5】



【図 1 6】

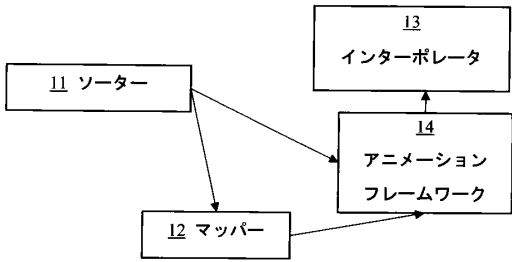


30

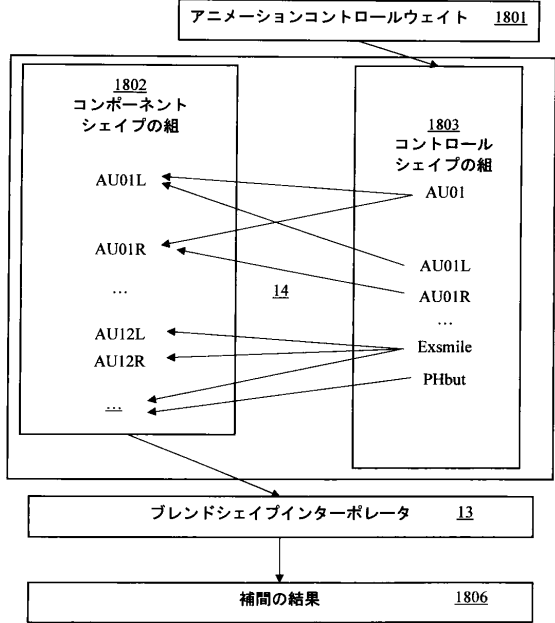
40

50

【図 1 7】



【図 1 8】



10

20

30

40

50

フロントページの続き

(72)発明者 バリバント, デイビッド

ニュージーランド国 5572 レビン, アール12, パエロア ロード 25

審査官 三沢 岳志

(56)参考文献

米国特許出願公開第2018/0130256 (US, A1)

米国特許出願公開第2018/0033190 (US, A1)

特表2014-522018 (JP, A)

特表2009-537927 (JP, A)

国際公開第2017/044499 (WO, A1)

米国特許出願公開第2015/0029198 (US, A1)

米国特許出願公開第2003/0164829 (US, A1)

米国特許出願公開第2008/0129730 (US, A1)

Song LI et al., "Deep 3D caricature face generation with identity and structure consistency", Neurocomputing, Vol. 454, 2021年09月, p.178-188, インターネット<URL:https://core.ac.uk/download/pdf/428341087.pdf>, [令和05年12月21日検索], DOI: 10.1016/j.neucom.2021.05.014

倉立尚明, "Talking Head Animation System Driven by Facial Motion Mapping and a 3D Face Database", 奈良先端科学技術大学院大学博士論文, 奈良先端科学技術大学院大学, 2004年06月29日, インターネット<URL:https://naist.repo.nii.ac.jp/record/10287/files/R003590.pdf>, [令和05年12月21日検索], DOI: 10.34413/dr.00435

(58)調査した分野 (Int.Cl., DB名)

G06T 13/40