

三角メッシュのへび

ステファン・ビショフ、トビアス・ウエヤンド、レイフ・コ
ベルト

Computer Science VIII 議長、RWTHアアヘン、52062アアヘンEメール:
{bischoff|kobbel|t}@informatik.rwth-aachen.de



抽象。この作品では、指定されたサーフェス（三角メッシュ）上にあるように拘束されているへびを表現し、追跡するための新しい方法を紹介します。新しい表現では、解像度を自動的にサーフェステクスセレーションに適合させることができ、（不安定な）逆投影操作は不要です。さらに、効率的で堅牢な衝突検出機能を備えており、へびのトポロジ的な振る舞いを完全に制御することができます。可能性のある用途としては、強化されたメッシュユザリング操作および表面の狭窄の検出が挙げられる。

ラダラント
2018-03-20 16:23:40

1 前書き

アクティブコンターモデル（スネーク）は、モーションキャプチャから画像セグメンテーションに至るまでのコンピュータビジョンおよび画像解析における幅広い用途で使用されてきた。伝統的にへびは一律な設定、すなわち画像をセグメント化するための曲線とボリウムデータをセグメント化するための表面にのみ適用されます。ただし、より一般的な設定から恩恵を受けるアプリケーションはたくさんあります。この作品では、特に任意の面に埋め込まれた曲線の場合を考えます。この設定は、埋め込み面を周囲の3空間で自由に曲げることができるため、単純な画像セグメンテーション問題よりもはるかに複雑です。

一例として、（例えば血管の）狭窄部を正確に突き止めて測定するという問題を考えてみましょう[1]。この場合、包埋面は血管の表面にすぎません。最初に、使用者は血管の周りを走るこの表面上に曲線を置く。この曲線をその曲率に従って展開させると、それは局所的に最短の曲線（測地線）になり、その長さは血管の直径の尺度になります。他のアプリケーションシナリオには、自動メッシュ分割およびインタラクティブメッシュ編集が含まれます。

埋め込まれたへびをモデル化するための初期の試みは特定のアプリケーションに限られていて、メッシュ頂点へのスナキセルの制限[2]または拡張前線だけをサポートしたため[3]低精度に苦しんでいました。ごく最近になつて三角形メッシュの幾何学的へびを完全にサポートする試みがなされました[4, 5]。しかしながら、これらのアプローチは位相的な柔軟性を提供せず、さらに基礎となるメッシュの複雑な区分的パラメータ化に頼る。パラメータ化アーチファクトによる問題を回避するために、我々は我々のフレームワークを[6]に提示されているパラメータ化のないアクティブコンターモデルに基づいている。

2 貢献

特定のサーフェスに埋め込まれているヘビの新しい表現を紹介します。明確にするために、表面は三角形メッシュによつて表されるが、この方法はまた、任意の多角形メッシュにも容易に拡張される。この表現の主な機能は次のとおりです。

1. 適応性: ヘビのサンプリングは自動的にそして局所的に三角メッシュの解像度に適応します。特に、パラメータ化アーチファクトの影響を受けやすい、ヘビまたは曲面のような表面の固有の特性に基づく複雑なサンプリング戦略は不要である。
2. 衝突検出とトポロジー制御: ヘビのトポロジー挙動はアプリケーションの要求に合うように調整することができます。固定された接続性に加えて、私たちのモデルはヘビのマーჯや分割もサポートしています。衝突検出は効率的かつ確実に実行され得る。
3. 頑健性: 我々のモデルは、メッシュ上へのスナクセルの数値的に不安定な（逆）投影を回避する。特に、ヘビは常に正確にメッシュ上にあることが保証されています。

3 方法

3.1 表現

私たちは宇宙の（多分開いている）多角形によつて蛇を表します。しかし、ヘビが実際に下にある三角メッシュに埋め込まれることを保証する2つの一貫性制約を強制します（図1）。

1. ヘビの頂点（スナクセル）はメッシュエッジ上になければなりません。
2. 蛇の線分は三角形の内側になければなりません。

さらに、我々は、スナクセルが方向付けられていると仮定する、すなわち、各スナクセル s は、として表すことができる。

$$s = (1 - d) v_{\text{から}} + d v_{\text{に}}, \quad d \in [0, 1)$$

ここで、 $v_{\text{から}}$ と $v_{\text{に}}$ はサポートエッジの端点です。この方向が蛇に沿つて一貫していること、すなわちすべての蛇がすべての蛇を指すことが重要です。

蛇の同じ側。特に、2つの連続したヘビのセグメントが同じ側ではありません。そのような2つのセグメントに隣接するスナクセルはあり得ません。

そして、それぞれの隣り合うスナクセルをつなぐ清掃征服返し除去することができます。

このような区分的線形表現は、一般に、メッシュ上をすべて捉えるのに十分です。例えばスカラーフィールドを考慮してみましょう。

これはメッシュの頂点で評価されます。スカラー場は各三角形の上では、レベルセットは三角形上の単なるトです。これは、モデルによつて正確に表すことができます。

ラダラント

2018-03-20 08:49:16

連続した連続した

で定義された詳細

ラダラント

2018-03-20 08:55:55

水平集合: 関数値に定数を与え、変数集合を持たせる。

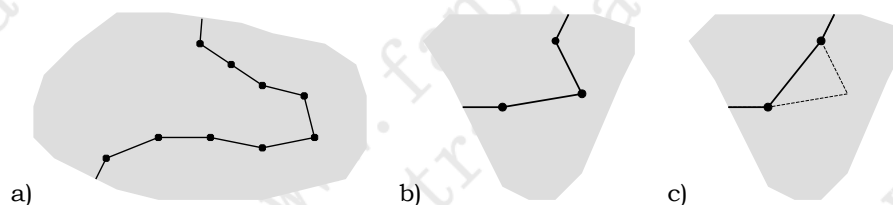


図1 スネーク表現：三角メッシュ上の有効な区分的線形スネーク (a) 無効なスナクセル (b) は清掃征服 (c) で除去されます。

3.2 進化

ヘビの進化は、通常それらの曲げエネルギーまたはその下にあるメッシュの曲率分布から派生する内力と外力によつて支配されます。一般性を失うことなく、力を各スナクセルにスカラー値の速度 v_s を割り当てるユーザー定義のブラックボックスとみなします。私たちはそれから、あらゆるヘビセルがその測地線法線に沿つて動くべきであると述べる Huygen の原理に従つてヘビ伝播をモデル化します。

下にあるサーフェスの接平面上の (空間的) 法線。2) ヘビの角二等分線としてのヘビセル s での測地法線を推定。スネクセルが辺上にあるのか頂点上にあるのかに応じて測地線測地線マップまたは指数マップを用いて局所的に構成を平面に平坦化した後の角度を α とする。次に、支持辺に沿つた s の射影速度 \hat{v}_s を次のように計算します。

$$\hat{v}_s = \frac{v_s}{\cos \alpha_s}$$

複数の連続するスナクセルがある場合、 \dots, s_k が頂点で一致すると、上記の計算は失敗し、代わりにスナクセル s_0, s_1, \dots, s_{k+1} から共通測地法線を推定するが、射影速度 \hat{v}_s を計算する。 \dots, \hat{v}_k を各スナクセルに対して個別に (その支持縁部に対する角度 α_i に応じて)。実際には、ヒンジまたは指数マップを明示的に計算する必要はありませんが、単にヘビセグメントとメッシュエッジ間の角度を合計して正規化するだけです。スネクセル速度を計算した後、オイラースキームを使用してスネクセルを時間的に前に移動します。

$$d_s \leftarrow d_s + \Delta t \hat{v}_s / \|e_s\|$$

ここで、 e_s は s のサポートエッジであり、スナクセルがメッシュの頂点と交差しないようにタイムステップ Δt を制限します。

$$\Delta t = \min_s (1 - d_s) / \hat{v}_s$$

スネクセルが価数 n の頂点にぶつかると、それは出ていく辺に置かれる $(n - 1)$ 個の新しいスナクセルに分割される。それらの距離値は 0 にリセットされるので、それらはすべて同じ空間位置にあります (サポートエッジは異なりますが)。この分割は一貫性違反をもたらすかもしれませんが、それは上述のように洗浄征服によつて容易に解決される。

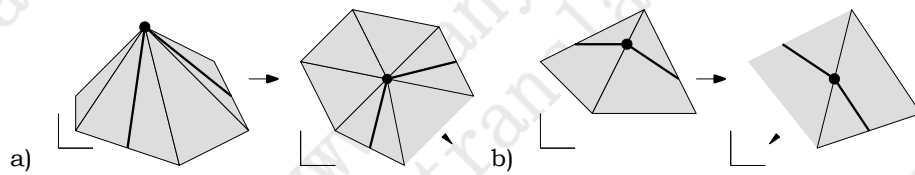


図2フラット化ヘビが表面に埋め込まれると、スナクセル速度は、スナクセルの周りの構成を平面に局所的に平らにし、次に測地法線のスナクセルの支持縁への投影に従って速度の大きさをスケーリングすることによって計算されなければならない。これは指数マップ (a) またはヒンジマップ (b) を使って行われます。

3.3 衝突検出とトポロジ制御

セクション3.1の一貫性制約は、2つのヘビ（または同じヘビの2つの部分）がヘビセルでしか衝突できないことを意味します。つまり、ヘビセルがヘビセグメントの内部を横切することは不可能です。したがって、衝突の検出は、各メッシュエッジについてそのエッジ上にあるスナクセルを格納し、各スナクセルが移動する前後にエッジ上のスナクセルの順序が変わったかどうかをテストすることによって効率的に実行することができる。もしそうであれば、我々は、スナクセル位置を直線的に内挿することによって接触点を決定する。衝突が発生した場合は、衝突している2つのヘビを結合するかどうかをアプリケーションに応じて選択できます。

- 結合の場合は、衝突しているスナクセルを削除し、それぞれの前身と後継者を接続することで、ヘビを再リンクします。その後、偽の無効なスナクセルを駆除するためのクリーニング征服を行います（図3）。
- 衝突が発生した場合は、衝突しているスナクセルを「固定」としてマークし、残りの更新手順から除外します。

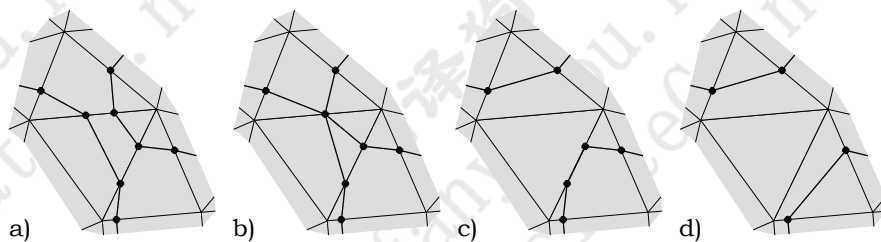


図3衝突検出とトポロジ制御衝突はメッシュエッジ (a) でのみ発生する可能性があるため、簡単に検出できます (b)。2つのヘビは、衝突しているスナクセルを取り除き、隣人を再リンクすることでマージできます (c)。偽の無効なスナクセルは、清掃征服 (d) で除去されます。

4 結果

新しい表現を実装し、さまざまな幾何学モデルでテストしました。実装は簡単で、標準的なPCでもインタラクティブな速度で実行されます。標準的なセグメンテーション問題に加えて、モデル上のくびれを検出する問題を検討しました。最初に、ユーザーは与えられた入力メッシュ上に一連の頂点を描きます。これらの頂点は、メッシュのエッジに沿って離散最短経路アルゴリズム (Dijkstra) を使用してリンクされます。結果は最初のへびであり、それを次にその曲率に従って進化させ、それゆえその長さを最小にします。図4最後のへびは局所的に最短経路であり、その長さはくびれの直径を表します。

へびは極小値しか検出できないので、血管の骨格に直交する断面を実行することによって、血管上に多数のへびを分布させることを計画しています。これらすべてのへびにそれらの長さを最小にさせ、最も短いものを選択させることによって、あらゆる狭窄を確実にそして全体的に検出することが可能であるはずである。私達はまた私達のへびの位相的な柔軟性を利用して全血管樹の構造を評価することを計画しています。ここでへびは木の枝に沿って走り、それぞれの分岐点で分割し、それによって木の構造を明らかにします。

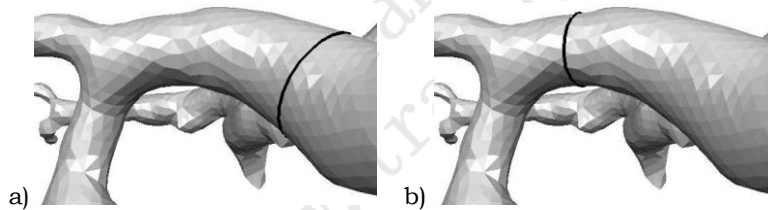


図4くびれの検出最初のへびを船の周りにインタラクティブに配置します (a)。数回繰り返した後、へびは血管の局所的に最も狭い部分の周りで収縮しました (b)。

参考文献

1. Hetroy, F., Attali, D.: 閉じた区分的測地線から閉じた多面体表面上のくびれまで。で: Pacific Graphics Proceedings. (2003) 394~398
2. Milroy, MJ, Bradley, C, Vickers, GW: ラップアラウンドモデルのセグメンテーション アクティブ輪郭を使用するコンピュータ支援設計29 (1997) 299-320
3. Lee, H, Kim, L, Meyer, M, Desbrun, M. で: Eurographics Work- コンピュータアニメーションとシミュレーションのショツプ。 (2001) 75-84
4. Lee, Y., Lee, S.: 三角メッシュの幾何学的なへび。コンピュータグラフィックスフォーラム 21 (2002) 229-238
5. Lee, Y, Lee, S, Shamir, A, Cohen-Or, D, Seidel, HP: インテリジェントメッシュユザリ ング 3Dへびを使うで: Pacific Graphics Proceedings. (2004) 279~287
6. Bischoff, S., Kobbelt, L.: パラメータ化されていないアクティブコンターモデル。 ビジュアルコンピュータ20 (2004) 217-228
7. Lee, AWF, Sweldens, W., Schroeder, P., Cowsar, L., Dobkin, D.: マツプ: Multires- 表面の解適応パラメータ化 IN SIGGRAPH 98 (1998) 95-104