Implicit Untangling: A Robust Solution for Modeling Layered Clothing

Thomas Buffet, Damien Rohmer, Loic Barthe, Laurence Boissieux,
Marie-Paule Cani

HAL Id: hal-02129156
https://hal.science/hal-02129156

Submitted on 16 May 2019

Implicit Untangling: A Robust Solution for Modeling Layered Clothing

THOMAS BUFFET, Inria Rhônes-Alpes, France and LIX, Ecole Polytechnique, CNRS, IP Paris, France
DAMIEN ROHMER, LIX, Ecole Polytechnique, CNRS, IP Paris, France
LOÏC BARTHE, University of Toulouse - IRIT - CNRS, France
LAURENCE BOISSIEUX, Inria Rhônes-Alpes, France
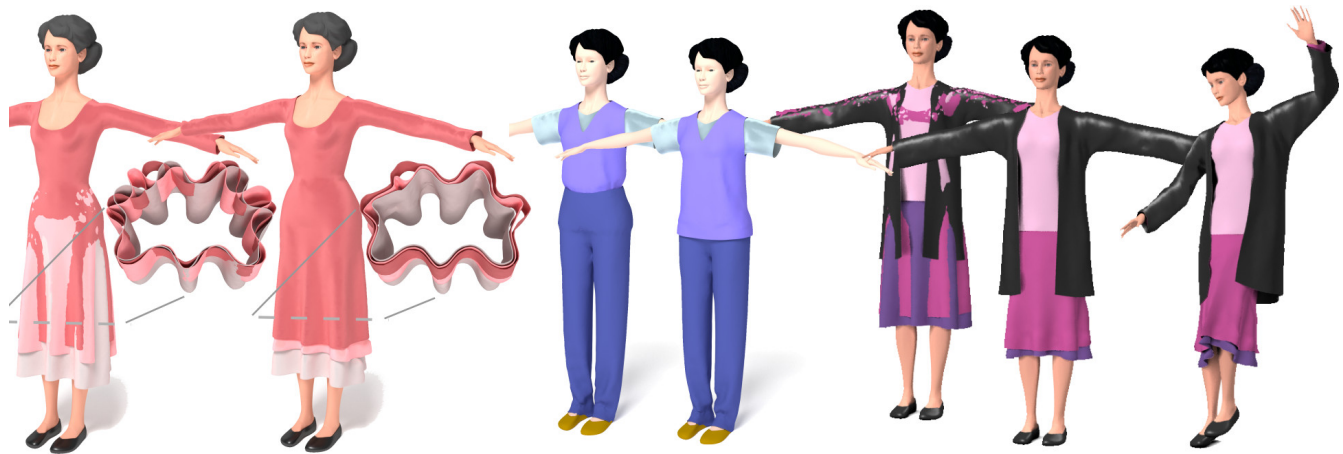MARIE-PAULE CANI, LIX, Ecole Polytechnique, CNRS, IP Paris, France

Fig. 1. Our method is able, from a set of garments possibly exhibiting deep interpenetration, to compute an untangled state, ie. a guaranteed intersection-free configuration, even while considering the thicknesses of cloth layers. Animations can then be launched.

We propose a robust method for untangling an arbitrary number of cloth layers, possibly exhibiting deep interpenetrations, to a collision-free state, ready for animation. Our method relies on an intermediate, implicit representation to solve the problem: the user selects a few garments stored in a library together with their implicit approximations, and places them over a mannequin while specifying the desired order between layers. The intersecting implicit surfaces are then combined using a new family of N-ary composition operators, specially designed for untangling layers. Garment meshes are finally projected to the deformed implicit surfaces in linear time, while best preserving triangles and avoiding loss of details.

Each of the untangling operators computes the target surface for a given garment in a single step, while accounting for the order between cloth layers and their individual thicknesses. As a group, they guarantee an intersection-free output configuration. Moreover, a weight can be associated with each layer to tune their relative influence during untangling, such as leather being less deformed than cloth. Results for each layer then reflect the combined effect of the other layers, enabling us to output a plausible configuration in contact regions. As our results show, our method can be used to generate plausible, new static shapes of garments when underwear has been added, as well as collision-free configurations enabling a user to safely launch animations of arbitrarily complex layered clothing.

CCS Concepts: • **Computing methodologies → Collision Detection**; **Volumetric Models**.

Additional Key Words and Phrases: Implicit surfaces, Cloth untangling, Collision processing, Cloth animation

1 INTRODUCTION

The efficient animation of dressed virtual characters is an essential step in 3D production pipelines. Tremendous efforts have been dedicated to achieve high quality dynamics for individual pieces of clothing, based on physically-based simulation models and efficient collision detection and response. Extending these animations to multiple layers of garments is however difficult, starting with the challenging problem of providing an initial collision-free state—which is mandatory for the animated cloth surfaces to behave as expected. In standard production, 3D garments are modeled on top of generic mannequin and poses. Reusing garments for slightly different mannequin, or launching the simulation in different poses, requires manual editing to set them in the expected collision free

Authors' addresses: Thomas Buffet, Inria Rhônes-Alpes , Montbonnot-Saint-Martin, France, 38330, 655 Avenue de l'Europe , LIX, Ecole Polytechnique, CNRS, IP Paris, France, 91120, Palaiseau, 1 rue Honoré d'Estienne d'Orves Alan Turing building; Damien Rohmer, LIX, Ecole Polytechnique, CNRS, IP Paris, France, 91120, Palaiseau, 1 rue Honoré d'Estienne d'Orves Alan Turing building; Loïc Barthe, University of Toulouse - IRIT - CNRS, France, Toulouse; Laurence Boissieux, Inria Rhônes-Alpes , Montbonnot-Saint-Martin, France, 38330, 655 Avenue de l'Europe; Marie-Paule Cani, LIX, Ecole Polytechnique, CNRS, IP Paris, France, 91120, Palaiseau, 1 rue Honoré d'Estienne d'Orves Alan Turing building.

state, and avoid a so-called *tangling* effect. This becomes very tedious when several layers of garments with extended contact surfaces are involved, since any deformation of a layer to avoid a collision could cause an intersection with another layer. For this reason, many clothing simulations, from video games to film production, still make use of a single garment layer. This results in both inaccurate rest states—since the deformation due to the underlying cloth layers are not accounted for, and wrong dynamics since the friction between neighboring cloth layers is neglected.

In this paper, we propose a robust solution to automate the tedious process of untangling intersecting garment layers. It can be used to quickly model plausible static shapes of garments given the underwear, and to safely initialize layered clothing animations (see Figure 1).

Our input is an arbitrary number of garment models. We precompute their implicit approximation as the intersection between an iso-surface in a scalar field and an influence zone in a co-variant field. When placed around a character, these garments typically exhibit various levels of interpenetration. We introduce an automatic method to compute a collision-free state, where contact is precisely modeled between initially colliding layers while taking into account the desired theoretical thickness of material (used to set up minimal distances between meshes). This is done using a set of closed-form N-ary composition operators, which deform the input fields while accounting for the user-specified order and thicknesses of the layers. In addition, the user can set weights in order to play with the relative stiffness of layers during deformation. After scalar field composition, garment meshes are projected in linear time with respect to their number of vertices, while tangentially relaxed to better preserve triangle shapes. This yields new shapes for top layers that take all the layers underneath into account, and provides interpenetration-free initial states for all garments, enabling a user to launch a simulation.

In summary, our contributions include:

- A new method for approximating a garment using a pair of scalar fields, given that garments are open surfaces that only span a part of the associated closed iso-surface.
- A closed-form solution for the N-ary operators enabling us to output untangled surfaces from a number of possibly intersecting layers.
- An adaptation of the projection/relaxation method proposed by Vaillant et al. [2014] for accurately projecting garment meshes within the scalar fields, yielding penetration-free configurations.

2 RELATED WORK

Our method is dedicated to the modeling of layered clothing, where inter-penetrations should be changed into contact regions with epsilon distance between meshes. We therefore first review contact modeling methods in cloth animation. Since our solution relies on implicit surfaces in scalar fields, we then review the implicit methods that have been used for modeling contact between non-nested volumetric bodies.

2.1 Contact modeling in cloth animation

While lots of methods are dedicated to efficient collision detection between rigid bodies [Fares and Hamam 2005] and deformable models [Teschner et al. 2004] , we focus here on approaches dedicated to cloth animation, and enabling not only the detection of collisions but also their resolution.

A first category of methods are dedicated to handling collision between a cloth surface and a volumetric body model [Guan et al. 2012], [Chen et al. 2013] ,[Sun et al. 2016] and [Wong et al. 2018]. They use some projection mechanism to locally deform the cloth and move it outside of the volume, therefore robustly handling cloth-to-body collision. However, they cannot directly extend to collisions between multiple cloth layers.

A second category of methods, first introduced by Bridson et al. [2002], and then further extended [Harmon et al. 2008; Selle et al. 2009], prevent collisions based on the history of deformations: penetrations detected at a given frame are corrected by looking backward in time to a past collision-free state. In particular, Harmon et al. [2008] achieve asynchronous correction leading to extremely robust and stable collision handling, although computationally costly. History-based correction have also been combined with penalty forces to increase stability [Tang et al. 2012], ray-tracing for accurate detection of intersection [Lehericey et al. 2015], and fast GPU computation [Tang et al. 2016, 2018]. Müller et al. [2015] propose an efficient unified approach dynamically handling collision detection and contact modeling by combining tetrahedrization of thin air layers between cloth surfaces and position based dynamics. Purely geometric approaches were also used to correct for existing collisions [Ye et al. 2017]. Most of these approaches are able to handle multiple layers of cloth and avoid self-intersection, however, they require an initial collision-free state as input. Therefore, they are not applicable to our problem.

The last category of approaches are dedicated untangling methods, able to correct existing penetrations without any knowledge of past states. In this case, a heuristic must be chosen in order to correct the ambiguous surface geometry. Previous methods have studied collisions between two cloth layers using repulsive force-based correction [Baraff et al. 2003], as well as geometric displacement in order to minimize the length of the collision contour [Volino and Magnenat-Thalmann 2006; Ye et al. 2017, 2015; Ye and Zhao 2012]. While repulsive force-based approaches can be expressed on arbitrary numbers of layers, they do not provide any guarantee of efficient convergence toward a collision-free state. Geometrical approaches provide this guarantee, but were restricted so far to collisions between two cloth layers only. A naive solution for generalizing to the multi-layers case consists in iteratively applying the untangling process to consecutive pairs of cloth layers, from the deepest layer in contact with the character's body to the most external one. In contrast with our N-ary approach, this strategy cannot, however, take the influence of upper layers into account when deforming a given layer.

2.2 Implicit surfaces

Implicit surfaces are well known for their ability to ease collision detection and model contact between non-nested volumetric bodies:
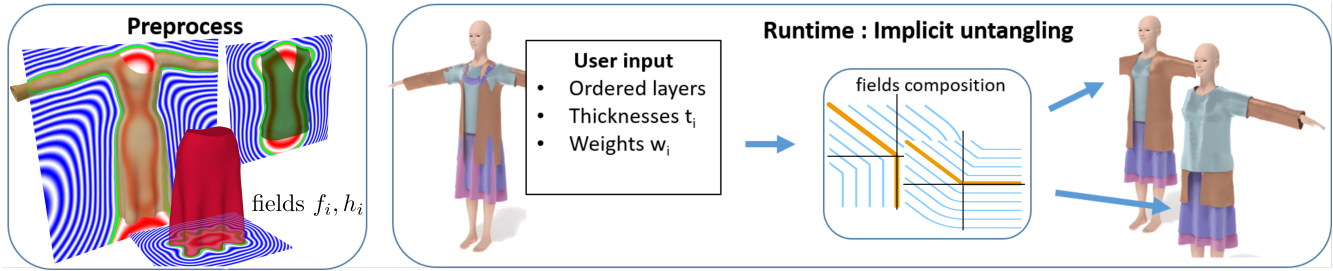
Fig. 2. Processing pipeline: Implicit reconstruction of all garments are computed as a preprocess (left). At run-time, the user selects a number of garments, sets their order, thicknesses and weights (center). The associated fields are composed using our new untangling operators, leading to a variety of possible results after mesh projection (right). For instance, the T-shirt (low weight during deformation) can either be worn below or above the thicker leather jacket.

A first remark is that collisions can be detected in linear time: Let f_1 and f_2 be two scalar fields and S_1, S_2 their associated 0-isosurfaces, the inside of the volume being defined by $V_i = \{p \in \mathbb{R}^3 / f_i(p) < 0\}$. Then for any point p on S_1, evaluating the sign of $f_2(p)$ provides a simple collision test with V_2, while the value of $|f_2(p)|$ gives some information about the penetration depth.

Another benefit of implicit surfaces is their ability to be composed into new surfaces [Bloomenthal 1997], using a variety of combination operators between fields. In addition to the well known blending operators, composition has been used to accurately model contact surfaces between colliding volumetric bodies. Cani [1993] linearly combines the input field functions in the region of interpenetration. Vaillant et al. [2014] introduce more complex binary operators for modelling contact between colliding skin parts of an animated character. Building on advanced implicit composition using field gradients [Gourmel et al. 2013], Angles et al. [2017] show that a variety of operators can be created on the fly from sketch-based user input, enabling, among other applications, the modeling of a variety of contact situations between volumes. Most of these works focus on binary operators numerically precomputed in a 2D or 3D discrete grid, which cannot be easily extended to n-ary composition. Moreover, none of them handles nested implicit volumes.

In addition to trivial operators such as maximum, minimum [Sabin 1968] and sum of field functions [Blinn 1982], n-ary composition operators include set-theoretic operations [Pasko et al. 1995], super-elliptic blends [Ricci 1973], extended convolution operators for topology control [Zanni et al. 2015] as well as various n-ary blends with range control [Barthe et al. 1998; de Groot et al. 2009; Hsu 2018; Hsu and Lee 2003]. The definition of n-ary operators can be tedious and their theory is not as deeply studied as the one of binary compositions. To our best knowledge, none are tackling our core challenge, namely contact modeling between formerly intersecting, nested implicit surfaces.

3 OVERVIEW

We aim at untangling an arbitrary number of intersecting garments. The key insight is to take benefits from the ability of volumetric representations to model contact through the composition of scalar fields. To achieve this, garments need to be converted to some implicit representation, while contact modeling by iso-surface composition needs to be extended to the untangling of an arbitrary number of nested surfaces.

3.1 Notations and input

In all this work, the implicit surfaces S_i we consider are closed surfaces defined as the 0-iso-surface of a scalar field f_i. By convention, we define the interior of the volume within S_i by $\{p \in \mathbb{R}^3 \mid f_i(p) < 0\}$. This is the convention used by HRBFs (Hermite Radial Basis Functions) [Wendland 2005] on which we will build on. They enable a user to reconstruct closed implicit surfaces from a set of sample points and the associated normal directions.

We consider a set of N predefined garment models given by their input meshes, which are already wrapped over an input character body. The user specifies the desired order between these garments, where layer 1 is the closest to the body and layer N is worn above the other layers.

Virtual *thickness* values t_i are specified for each layer, enabling the tuning of the minimal distance between the centered meshes representing each garment, while *weights* $w_i > 0$, acting as a stiffness parameter, enable the tuning of their relative influence when deformed during contact. Small values of w_i mimic very flexible material that tends to match the geometry of other surfaces in contact regions, while large values mimic stiffer layers that tend to keep their shape. Note that the body is handled as a fully rigid layer $w_i = +\infty$. The deformation we want to output for each garment is to account for the combined effect of all the other layers (above and under) both in terms of thicknesses and weights.

3.2 Processing pipeline and challenges

Our processing pipeline, depicted in fig. 2, includes a prepossessing step—namely implicit approximation of all garments in a library, which can be done independently from a specific set-up. At run time, the user selects layers, sets their parameters, and untangles them. More precisely, for each mesh vertex, gradient descent and tangential relaxation are interleaved to deform each garment towards a target implicit surface, computed by applying an untangling operator to the input fields. Let us detail the main problems to be solved.

Implicit representation of garments: While implicit surface reconstruction from sample points is a solved problem for closed surfaces such as the character's body, extending it to open surfaces with boundaries such as garments is a challenge. In particular, the new representation needs to generalize the notion of *inside* and *outside* to these open surfaces, to be able to detect the interpenetration regions between layers, in which corrections need to be applied through the field composition operator.

Therefore, our first contribution, presented in Section 4, is a general solution for approximating open surfaces with boundaries using a double implicit representation: we combine the field f_i which embeds the mesh among iso-surfaces with a *co-variant field* h_i, computed from the open borders of the mesh. This enables us to define the region where a garment layer could cause intersections, accounting for the fact that it does not cover the full 0-iso-surface of f_i.

Untangling operators for nested implicit surfaces: Once the user sets a new configuration with a character and some ordered layers of garments from the library, implicit composition is used to untangle the associated fields, generating contact surfaces through deformation in the regions where garments interpenetrate (see fig. 2-middle). Here, we are not looking for a single composition operator but for N operators O_i, parameterized by the layers weights w_i, and able to generate N new fields $\hat{f}_i$ whose respective 0-iso-surfaces coincide in the previously intersecting regions while remaining intersection-free elsewhere.

We introduce a closed-form solution for these operators, enabling us to apply them whatever the number of colliding layers and their relative rigidities. Section 5 first explains how these operators can be defined in the general case of N layers, before detailing it for the specific case of two and three layers.

Using the deformed fields for updating the garments: In our case, the untangling composition is not to be applied to some abstract, nested implicit surface, but to meshes representing garments. Similarly to Vailllant et al [2014], output meshes are computed by displacing vertices of the original input meshes along the gradient of their respective field $\nabla \hat{f}_i$ until reaching the zero value of $\hat{f}_i$.

Several changes however need to be made: first, the meshes should not lie on the same 0-iso-surface in contact regions, but at some offset distance from each other, to avoid visual artifacts and allow subsequent physically-based animation. Secondly, some tangential relaxation is required to ensure that the local mesh deformation remains minimal while well approximating the 0-iso-surface of $\hat{f}_i$. Indeed, over-elongated triangles could cause geometric intersection even while implicit layers do not intersect, and could lead to unstable subsequent simulations.

Section 6 explains our practical solution for applying the deformation to meshes, which involves an extended composition operator accounting for the required thicknesses t_i of each layer to guarantee nested target surfaces, and a new relaxation method that we interleave with gradient descent to reduce tangential distortions.

4 IMPLICIT APPROXIMATION FOR GARMENTS

While we use standard HRBFs to approximate the character's body, we need to extend implicit reconstruction to the case of open surfaces with boundaries for handling garments. The main challenge is to characterize regions where the garment might interpenetrate with others during field composition, used to test for interpenetration: indeed, using the full 0-iso-surface of f_i to approximate a cloth layer would not work, since extra collisions could then be wrongly detected, leading to unwanted deformations of the other layers, such as in the region below the jacket and t-shirt in Fig. 3. The key idea to achieve this is to use a second implicit volume, defined by a *co-variant field* h_i.
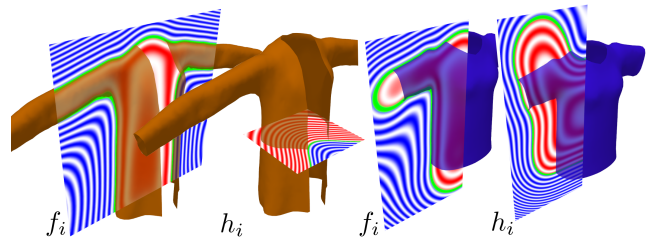

Fig. 3. From left to right: a jacket and a T-shirt, with the associated field f_i and co-variant field h_i. The iso-values of each field are depicted on a 2D plane set to intersect the model, where positive isovalues are in blue, negative ones in red, and the 0-iso-value in green. Note that the negative part of the covariant field (in red) enables us to characterize the part of space where a garment might interact with other layers. This enables to discard, for instance, the rounded part at the bottom of f_i's 0-iso-surface for the T-shirt.

More precisely, the input mesh is first sampled using Poisson dart-throwing. Samples are used to compute a HRBF approximation f_i ([Iske 2002] , [Macedo et al. 2011]), such that for each pair point/normal (p_s, n_s) in the sample, $f_i(p_s) = 0$ and $\nabla f_i(p_s) = n_s$. The implicit surface defined by $\{p \in \mathbb{R}^3 \mid f_i(p) = 0\}$ is then close to the initial mesh anywhere near one of its vertices. As in former works combining meshes and implicit modeling [Vaillant et al. 2013], each mesh vertex keeps track of its exact iso-value in the reconstructed field f_i, so that no detail is lost (eg. thicker parts such as seam-lines or pockets would be adequately reconstructed after deformation). Note that in practice, we pre-store the values of $f_i(p)$ and $\nabla f_i(p)$ in a grid. Evaluating f_i and its gradient at any spatial position is then efficiently approximated using a tri-linear interpolation.

Due to the continuity of the HRBF model, the 0-iso-surface of f_i is a closed surface. In order to be able to discard parts where interactions with the other garments should not be considered, we also compute a co-variant field h_i, aimed at capturing the region "inside borders". While the use of two scalar fields was already introduced in order to reconstruct open surfaces from incomplete point-set acquisition using evolving level-sets [Solem and Heyden 2004], we propose a new efficient formulation for the field h_i, based on HRBF, specifically adapted to take advantage of meshes with boundaries . More precisely, the HRBF is computed from the open-boundary samples $(p_{s'}, n_{s'})$, where the gradient $\nabla h_i(p_{s'}) = n_{s'}$ is

set to the unit vector both orthogonal to the open-boundary curve and in the tangent plane with respect to the surface (see Fig.3).

The region where $h_i(p) > 0$ is called the *zero-influence* region of the layer, meaning that it should not be considered for processing interactions with this layer. Conversely, the *influence region of a layer* is defined by $h_i(p) < 0$.

In consequence, the *inside* of the garment, which should be used to test for collision with outer cloth layers is defined as $\{p \in \mathbb{R}^3 \mid f_i(p) < 0 \ \& \ h_i(p) < 0\}$ while the *outside* of the garment, where collisions with inner cloth layers are detected, is given by $\{p \in \mathbb{R}^3 \mid f_i(p) > 0 \ \& \ h_i(p) < 0\}$.

5 UNTANGLING NESTED IMPLICIT SURFACES

In this section, we describe our implicit untangling operator in the general case of N implicit surfaces defined as the 0-iso-surfaces of fields $(f_i)_{i \in [1,N]}$. Without loss of generality, we consider in the following that i refers to their nesting order. The application to garments defined by both a field and a co-variant field is presented in Section 6. The objective is to define a set of closed-form composition operators enabling the deformation of each of the surfaces so that contact is modeled in regions where they previously interpenetrated.

Untangling is performed by replacing each field f_i by $\hat{f}_i$ such that $\forall p \in \mathbb{R}^3$, $\hat{f}_i(p) = O_i(f_1(p), .., f_N(p))$, where O_i is a composition operator i.e. a function from $\mathbb{R}^N$ to $\mathbb{R}$, and where each of the resulting isosurfaces $\{p \in \mathbb{R}^3 \mid \hat{f}_i(p) = 0\}$ does not intersect any of the other ones. To simplify notation, we consider the f_i, ie. functions applied to a position in space and returning $f_i(p) \in \mathbb{R}$, as a set of independent real variables. This enables us to define the N-dimensional vector $f = (f_1, \ldots, f_N) \in \mathbb{R}^N$, lying in the so-called *fields-space*, and to write the operator as $O_i(f)$.

Inspired from Angles *et al.* [2017], the operators are built in two steps: First, we build a *desired* zero-set Z_i in fields-space, i.e. in $\mathbb{R}^N$ in our case. Secondly, we build O_i from Z_i such that $O_i(f)$ is the signed Euclidean distance between f and Z_i, thus ensuring that Z_i is the zero-set of O_i.

Finally, applying this operator to a position p in 3D space, consists in evaluating $O_i(f(p))$, where f is now considered as a function of the input position p.

In the following, we present the construction of the zero sets Z_i, first by defining their general closed-form expression in the N-dimensional case (sec. 5.1), and then by explaining their intuitive construction in the special cases $N = 2$ (sec. 5.2) and $N = 3$ (sec. 5.3). The case $N = 4$ is detailed in the supplementary material associated with this paper.

5.1 General formulation in the N-dimensional case

Let us consider a given layer i. We define its zero-set Z_i as the union of $d = i(N - i + 1)$ sub-spaces $(H_i^{b,c})_{b \in [0, i-1], c \in [0, N-i]}$.

$$\forall i \in [1, N], \ Z_i = \bigcup_{b,c} H_i^{b,c} . \tag{1}$$

Each subspace $H_i^{b,c}$ is itself defined by a hyperplane with normal vector $n_i^{b,c} \in \mathbb{R}^N$, and $N - 1$ additional inequalities expressed as

linear combinations of the (f_i), formally described using a matrix $M_i^{b,c}$.

$$\forall i \in [1, N], \ \forall b \in [0, i-1], \ \forall c \in [0, N-i],$$
$$H_i^{b,c} = \left\{ f \in \mathbb{R}^N \mid n_i^{b,c} \cdot f = 0 \ \text{and} \ M_i^{b,c} f > 0 \right\} . \tag{2}$$

As further detailed in the next sections, $H_i^{b,c}$ corresponds to a subset of $\mathbb{R}^N$ that expresses the interactions between the layer i and all other layers within the interval $[i - b, i + c] \setminus i$. Intuitively, the equality constraint (related to the normal n) shifts the layers to be coincident rather than interpenetrating, and the inequality constraints (related to the matrix M) select whether this shift should be applied based on what interpenetrations are present.

The vector $n_i^{b,c}$ is defined as

$$\forall j \in [1, N], \ n_i^{b,c}[j] = \begin{cases} w_j & \text{if } j \in [i - b, i + c] \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where the w_j are the weights associated with each layer, and acting as stiffness parameters during the untangling process. Note that $n_i^{b,c}$ can be multiplied by a scalar value without changing the zero-set.

The matrix $M_i^{b,c}$ of size $(N - 1) \times N$ defining the inequality constraints on $H_i^{b,c}$ can be expressed using four squared triangular blocks, respectively, A of size $(i - b - 1)^2$, B of size b^2, C of size c^2, D of size $(N - i - c)^2$. These matrices have the following expression, when layer i is in collision with its b inner, and c above layers.

$$M_i^{b,c} = \begin{pmatrix} A & 0 & 0 & 0 & 0 \\ 0 & B & 0 & 0 & 0 \\ 0 & 0 & 0 & C & 0 \\ 0 & 0 & 0 & 0 & D \end{pmatrix}, \text{ with} \tag{4}$$

$$A = \begin{pmatrix} w_1 & w_2 & \cdots & w_{i-b-1} \\ 0 & w_2 & \cdots & w_{i-b-1} \\ & & \ddots & \vdots \\ 0 & \cdots & 0 & w_{i-b-1} \end{pmatrix} \quad B = \begin{pmatrix} -w_{i-b} & 0 & \cdots & 0 \\ \vdots & \ddots & & \\ -w_{i-b} & \cdots & -w_{i-2} & 0 \\ -w_{i-b} & \cdots & -w_{i-2} & -w_{i-1} \end{pmatrix}$$

$$C = \begin{pmatrix} w_{i+1} & w_{i+2} & \cdots & w_{i+c} \\ 0 & w_{i+2} & \cdots & w_{i+c} \\ & & \ddots & \vdots \\ 0 & \cdots & 0 & w_{i+c} \end{pmatrix} \quad D = \begin{pmatrix} -w_{i+c+1} & 0 & \cdots & 0 \\ \vdots & \ddots & & \\ -w_{i+c+1} & \cdots & -w_{N-1} & 0 \\ -w_{i+c+1} & \cdots & -w_{N-1} & -w_N \end{pmatrix}$$

Matrices B and C represent the conditions for layer i and its direct neighbors to be in collision, while matrices A and D represent the absence of collision between layer i and the other layers. Note that each line of these matrices is defined up to a positive scaling factor, since $M_i^{b,c}$ is used to express the N-1 conditions $M_i^{b,c} f > 0$.

5.2 Case of two layers (N = 2)

To give some intuition, let us explain in more details the construction of the zero-sets in the simple case of two layers $N = 2$, and show the correspondences with the variables previously defined.

Let f_1 and f_2 be the two fields representing the two implicit surfaces to be nested, where f_1 corresponds to the inner one and f_2 to the outer one. Let us detail the creation of the zero-set Z_1, which drives the creation of the operator O_1 (see Fig. 4 for a visual illustration in field-space). Let us consider a point $p \in \mathbb{R}^3$ on the inner implicit surface such that $f_1(p) = 0$. We can then consider two cases:

(1) $f_2(p) < 0$: p is inside the outer layer. This is a *legal position* in field-space, ie. an expected configuration compatible with the absence of collision. In such case, the iso-surface from layer 1 defined by $f_1 = 0$ should remain unchanged around p.

(2) $f_2(p) > 0$: p is outside the outer layer which is an illegal position, meaning that there is some intersection between the two layers. Intuitively, in this case we would want the inner iso-surface to be pushed inward by the outer one to solve such collision. We choose to correct the iso-surface by making it lie on $w_1 f_1 + w_2 f_2 = 0$. Indeed, this expression can be written as $f_1 = -\frac{w_2}{w_1} f_2 < 0$: following the conventions expressed in Section 3, the resulting surface is in fact placed inward with respect to $f_1 = 0$. As depicted in Fig.5, modifying the relative weights (w_1, w_2) moves the corrected iso-surface continuously from $f_1 = 0$ to $f_2 = 0$.

The zero-set Z_1 can therefore be expressed as the following subset of $\mathbb{R}^2$:

$$Z_1 = \left\{ f = (f_1, f_2) \in \mathbb{R}^2 \mid \begin{array}{l} (f_2 < 0 \ \& \ f_1 = 0) \ || \\ (f_2 > 0 \ \& \ w_1 f_1 + w_2 f_2 = 0) \end{array} \right\}. \quad (5)$$

This relation can be rewritten under the general form described previously as:

$$\begin{cases} Z_1 = H_1^{0,0} \bigcup H_1^{0,1} \\ H_1^{0,0} = \{ f \in \mathbb{R}^2 \mid n_1^{0,0} \cdot f = 0 \text{ and } M_1^{0,0} f > 0 \} \\ H_1^{0,1} = \{ f \in \mathbb{R}^2 \mid n_1^{0,1} \cdot f = 0 \text{ and } M_1^{0,1} f > 0 \}, \end{cases} \quad (6)$$

with $n_1^{0,0} = (1, 0)$, $n_1^{0,1} = (w_1, w_2)$, and $M_1^{0,0} = (0, -1)$, $M_1^{0,1} = (0, 1)$. As explained previously, vector n and the lines of matrix M can be multiplied by positive scalars, and can equivalently be defined to match with the general notation of the equations (3) and (4), as $n_1^{0,0} = (w_1, 0)$, $n_1^{0,1} = (w_1, w_2)$, and $M_1^{0,0} = (0, -w_2)$, $M_1^{0,1} = (0, w_2)$, respectively.

In these cases, $H_1^{0,0}$ and $H_1^{0,1}$ represent half-lines in the 2D fields-space. $M_1^{0,0} = (0, D)$, with $D = (-w_2)$ (no collision with the single outer layer), while the other matrices A, B, C are empty. Similarly, $M_1^{0,1} = (0, C)$ with $C = (w_2)$ (expressing collision with the single outer layer).

Z_2 is defined in a symmetric way. For a point p lying on the outer implicit surface, for which $f_2(p) = 0$, two similar cases can arise:

(1) $f_1(p) > 0$, denoting a legal position for which the second implicit surface should remain unchanged;

(2) $f_1(p) < 0$, denoting an illegal position for which we set the new iso-surface to lie on $w_1 f_1 + w_2 f_2 = 0$, which can be rewritten as $f_2 = -\frac{w_1}{w_2} f_1 > 0$. The corrected iso-surface then lies outward the input one, which models the action of the inner implicit surface pushing the outer one away.

This leads to the following definition for the zero-set:

$$\begin{cases} Z_2 = H_2^{0,0} \bigcup H_2^{1,0} \\ H_2^{0,0} = \{ f \in \mathbb{R}^2 \mid n_2^{0,0} \cdot f = 0 \text{ and } M_2^{0,0} f > 0 \} \\ H_2^{1,0} = \{ f \in \mathbb{R}^2 \mid n_2^{1,0} \cdot f = 0 \text{ and } M_2^{1,0} f > 0 \}, \end{cases} \quad (7)$$

with $n_2^{0,0} = (0, 1)$, $n_2^{1,0} = (w_1, w_2)$, $M_2^{0,0} = (1, 0)$ and $M_2^{1,0} = (-1, 0)$. Using the convention from Section 5.1, this can be re-expressed as $n_2^{0,0} = (0, w_2)$, $n_2^{1,0} = (w_1, w_2)$, $M_2^{0,0} = (w_1, 0)$ and $M_2^{1,0} = (-w_1, 0)$.
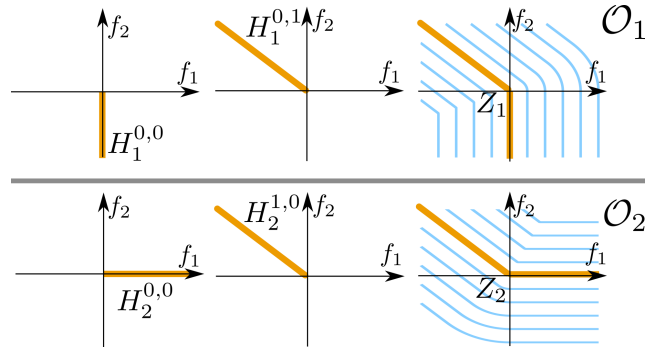

Fig. 4. N=2 with fixed w_1 and w_2: Left and middle: Visualisation of the subparts forming Z_i. Right: Visualisation of Z_i and some iso-lines of O_i, computed as the distance to Z_i. Notice how $H_1^{0,1}$ and $H_2^{1,0}$ are similar, leading eventually to the contact between the two corrected iso-surface.

As we can check, both operators push points formerly on one of the implicit surfaces, but in the intersection region, to the same iso-surface $w_1 f_1 + w_2 f_2 = 0$, which untangles the two volumes and creates a contact surface instead. Note that the weights w_1 and w_2 do not need to sum to one: only their relative value is important, since the contact surface they define can be re-written as $f_1 + \frac{w_1}{w_2} f_2 = 0$. Playing with their relative value enables us to tune the position of the contact surface anywhere between $f_1 = 0$ (layer 1 not changed, ie. behaving rigidly during untangling) and $f_2 = 0$ (layer 2 not changed, ie. behaving rigidly, as illustrated in Figure 5).
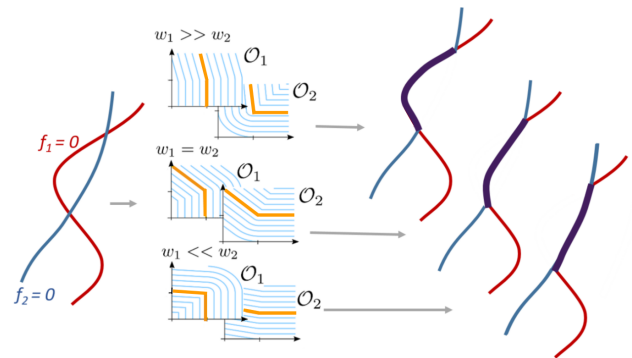

Fig. 5. Case of two layers. From left to right: two intersecting implicit surfaces in cross section, where the red one should be kept at the right; view of Z_i and iso-lines of the operators O_i in field-space ; untangled results. Modifying the orientation of the half-lines by tuning the ratio between w_1 and w_2 brings the contact surfaces after untangling from the former position of the blue layer to the one of the red layer (right, top to bottom)

5.3 Case of three layers ($N = 3$)

The case of three layers is slightly more complex. Indeed, a collision between layer 1 and layer 3 should not be handled directly without

taking into account the influence of layer 2 in-between. To solve such inter-dependence, we consider first the collision of layer 2 and 3 independently from layer 1, and set, in case of collision, an intermediate corrected iso-surface defined as $w_2 f_2 + w_3 f_3 = 0$. Then the collision is handled between the intermediate iso-surface and layer 1.

To define Z_1, we are left with three cases when classifying the possible interpenetration states at a point p on the first layer such that $f_1(p) = 0$:

(1) No intersection occurs if $f_2(p) < 0$ (layer 2 is above, see Fig. 6.a), and $w_2 f_2(p) + w_3 f_3(p) < 0$ (layer 3, after intermediate correction, is above, see Fig. 6.b). In this case, we keep layer 1 non-deformed by maintaining the zero-set on $f_1 = 0$.
(2) Only layer 2 is intersecting layer 1 if $f_2(p) > 0$ and $f_3(p) < 0$ (Fig. 6.c). In this case, similarly to $N = 2$, we chose to model coherently the corrected zero-set using $w_1 f_1 + w_2 f_2 = 0$.
(3) Layers 2 and 3 are both interacting with layer 1 when $f_3(p) > 0$ (p is outside layer 3) and $w_2 f_2(p) + w_3 f_3(p) > 0$. In this case, even if layer 2 is not intersecting layer 1 in its initial configuration, it is pushed by layer 3, and the resulting intermediate iso-surface ends up in collision with layer 1. (see Fig. 6 (e)). In this case, we set the corrected 0-set of layer 1 to $w_1 f_1 + w_2 f_2 + w_3 f_3 = 0$
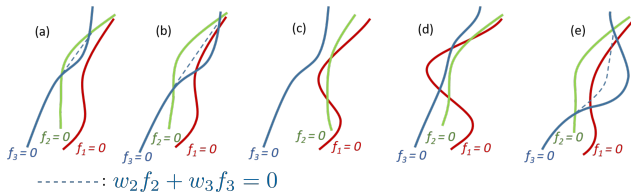

Fig. 6. Partial view of three implicit layers in cross section. The inner side is at the right of f_1, and the outer side is on the left of f_3. The dashed line represents, in case where it is needed, the intermediate surface profile between layer 2 and 3 represented by $w_2 f_2 + w_3 f_3 = 0$. (a) and (b): No interaction with layer 1; (c): layer 1 intersects only layer 2; (d) and (e): All three layers are interacting (note: the corrected state between 1 and the other layers is not represented).

These three cases turn into the 3 sub-spaces forming Z_1:

$$\begin{cases} Z_1 = H_1^{0,0} \cup H_1^{0,1} \cup H_1^{0,2} \\ H_1^{0,0} = \{f \in \mathbb{R}^3 \mid n_1^{0,0} \cdot f = 0 \text{ and } M_1^{0,0} f > 0\} \\ H_1^{0,1} = \{f \in \mathbb{R}^3 \mid n_1^{0,1} \cdot f = 0 \text{ and } M_1^{0,1} f > 0\} \\ H_1^{0,2} = \{f \in \mathbb{R}^3 \mid n_1^{0,2} \cdot f = 0 \text{ and } M_1^{0,2} f > 0\}, \end{cases} \quad (8)$$

with $n_1^{0,0} = (1,0,0)$, $n_1^{0,1} = (w_1, w_2, 0)$, $n_1^{0,2} = (w_1, w_2, w_3)$, and

$$M_1^{0,0} = \begin{pmatrix} 0 & -1 & 0 \\ 0 & -w_2 & -w_3 \end{pmatrix}, M_1^{0,1} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, M_1^{0,2} = \begin{pmatrix} 0 & w_2 & w_3 \\ 0 & 0 & 1 \end{pmatrix}.$$

Using the convention from Section 5.1 where 1 is replaced by the corresponding weight, this can be rewritten as:
$n_1^{0,0} = (w_1, 0, 0)$, $n_1^{0,1} = (w_1, w_2, 0)$, $n_1^{0,2} = (w_1, w_2, w_3)$, and

$$M_1^{0,0} = \begin{pmatrix} 0 & -w_2 & 0 \\ 0 & -w_2 & -w_3 \end{pmatrix}, M_1^{0,1} = \begin{pmatrix} 0 & w_2 & 0 \\ 0 & 0 & -w_3 \end{pmatrix}, M_1^{0,2} = \begin{pmatrix} 0 & w_2 & w_3 \\ 0 & 0 & w_3 \end{pmatrix}$$

In this case, we have $M_1^{0,0} = \begin{pmatrix} 0 & D \end{pmatrix}$ with $D = \begin{pmatrix} -w_2 & 0 \\ -w_2 & -w_3 \end{pmatrix}$, $M_1^{0,1} = \begin{pmatrix} 0 & C & 0 \\ 0 & 0 & D \end{pmatrix}$ with $C = (w_2)$ and $D = (-w_3)$, and $M_1^{0,2} = \begin{pmatrix} 0 & C \end{pmatrix}$ with $C = \begin{pmatrix} w_2 & w_3 \\ 0 & w_3 \end{pmatrix}$.

While the constraints associated to layer 3 can be derived symmetrically to the ones we just described for layer 1, the set of constraints to set up for the second layer are slightly different. Indeed, layer 1 and 3 can be directly compared to layer 2 without requiring the consideration of any intermediate corrected surface. We therefore identify four possibles cases:

(1) No collision: $f_1 > 0$ and $f_3 < 0$, leading to the isosurface $f_2 = 0$.
(2) Collision with layer 1 only: $f_1 < 0$ and $f_3 < 0$ leading to the isosurface $w_1 f_1 + w_2 f_2 = 0$.
(3) Collision with layer 3 only: $f_1 > 0$ and $f_3 > 0$ leading to the isosurface $w_2 f_2 + w_3 f_3 = 0$.
(4) Collision with layer 2 and 3: $f_1 < 0$ and $f_3 > 0$ leading to the isosurface $w_1 f_1 + w_2 f_2 + w_3 f_3 = 0$.

Using the previous formulations, vector $n_i^{b,c}$ and matrices $M_i^{b,c}$ are defined as follows:

$$n_2^{0,0} = (0, w_2, 0), \ n_2^{1,0} = (w_1, w_2, 0),$$
$$n_2^{0,1} = (0, w_2, w_3), \ n_2^{1,1} = (w_1, w_2, w_3),$$

$$M_2^{0,0} = \begin{pmatrix} w_1 & 0 & 0 \\ 0 & 0 & -w_3 \end{pmatrix}, M_2^{1,0} = \begin{pmatrix} -w_1 & 0 & 0 \\ 0 & 0 & -w_3 \end{pmatrix},$$

$$M_2^{0,1} = \begin{pmatrix} w_1 & 0 & 0 \\ 0 & 0 & w_3 \end{pmatrix}, M_2^{1,1} = \begin{pmatrix} -w_1 & 0 & 0 \\ 0 & 0 & w_3 \end{pmatrix}.$$

and

$$\begin{cases} Z_2 = H_2^{0,0} \cup H_2^{1,0} \cup H_2^{1,0} \cup H_2^{1,1} \\ H_2^{0,0} = \{f \in \mathbb{R}^3 \mid n_2^{0,0} \cdot f = 0 \text{ and } M_2^{0,0} f > 0\} \\ H_2^{1,0} = \{f \in \mathbb{R}^3 \mid n_2^{1,0} \cdot f = 0 \text{ and } M_2^{1,0} f > 0\} \\ H_2^{0,1} = \{f \in \mathbb{R}^3 \mid n_2^{0,1} \cdot f = 0 \text{ and } M_2^{0,1} f > 0\} \\ H_2^{1,1} = \{f \in \mathbb{R}^3 \mid n_2^{1,1} \cdot f = 0 \text{ and } M_2^{1,1} f > 0\}. \end{cases} \quad (9)$$

Again, note that the equalities $n_1^{0,1} = n_2^{1,0}$ and $n_1^{0,2} = n_2^{1,1}$ allow to model the same contact iso-surface between, respectively, layer 1 interacting with layer 2, and all layers 1, 2, 3 interacting.
As in the case of two layers, the relative values given to the weights can be used to tune the amount of deformation applied to each layer in interpenetration situations. For instance, in the deep intersection case when all three surfaces interact and end up on $w_1 f_1 + w_2 f_2 + w_3 f_3 = 0$, this contact surface can be pushed towards $f_1 = 0$ ($w_1 \gg w_2 \ \& \ w_3$), $f_2 = 0$ ($w_2 \gg w_1 \ \& \ w_3$), or $f_3 = 0$ ($w_3 \gg w_1 \ \& \ w_2$).

The case $N > 3$ can be further developed by considering more intermediate layers for higher values of N, to reach the formulation given in Section 5.1. To provide a better insight on the construction of the matrices in the general case, $N = 4$ is detailed in the supplementary material.

5.4 General evaluation of the operator

The operator O_i can be evaluated at any position $f \in \mathbb{R}^N$ as the signed Euclidean distance between f and Z_i. To evaluate this distance, we compute, as an intermediate step, the closest point $f_i^{b_0, c_0}$ such that

$$(b_0, c_0) = \min_{b, c} ||f - f_i^{b, c}||,$$

where $f_i^{b, c}$ is the closest point to f on $(H_i^{b, c})$.

The algorithm to compute efficiently $f_i^{b, c}$ for any (b, c) is provided as supplementary material. It consists of computing an appropriate orthogonal basis defined for each $H_i^{b, c}$ from the values of $n_i^{b, c}$ and all the constraint vectors given by the rows of $M_i^{b, c}$. (b_0, c_0) is eventually computed in $O(N^3)$ operations. The complexity is related to the use of a Gram-Schmidt orthogonalization iterated over every $H_i^{b, c}$.

The final signed distance is then computed as:

$$O_i(f) = \text{sign}(n_i^{b_0, c_0} \cdot f) \, ||f - f_i^{b_0, c_0}|| . \tag{10}$$

Once $f_i^{b_0, c_0}$ is known, we can note that computing the gradient of the operator $\nabla O_i = (\partial O_i / \partial f_j)_{j \in [1, N]}$ (which will be used in Section 6.3) is straightforward. Indeed, O_i is the Euclidean distance function, therefore its gradient at position f is given by:

$$\nabla O_i(f) = \text{sign}(n_i^{b_0, c_0} \cdot f) \, (f - f_i^{b_0, c_0}) / ||f - f_i^{b_0, c_0}||. \tag{11}$$

6 APPLICATION TO UNTANGLING GARMENT MESHES

In this Section, we present the extension of field composition to open surfaces using the co-variant fields introduced in Section 4, the modification of the operators to take cloth thickness into account, and the computation of the resulting deformation of garment meshes.

6.1 Using co-variant fields to detect active layers

In practice, garments on top of a mannequin are only locally nested. For instance a jacket and trousers may only be overlapping around the hips of the character. As a result, detected inter-penetrations between the associated implicit surfaces should be discarded, except within the overlapping region.

More precisely, we define the *active layers* at a point p in space as the set of garment layers j whose influence region includes p, ie. such that $h_j(p) < 0$ (see Section 4). This enables us to compute the corrected 0-iso-surface of a given layer i — giving the deformation to be applied to a mesh point p of layer i, while only considering the other locally-active layers j.

In field space F, canceling-out the effect of a layer j consists in not taking its corresponding field f_j into account within the computation of the operator, while keeping the nesting order unchanged. Thus, the operator must be applied at each point p on the subset of fields $(f_j)_{j \in \mathcal{J}(p)}$, with $\mathcal{J} = \{j \in [1, N] \mid h_j(p) < 0\}$. The corrected field for garment i at p is thus given by:

$$\hat{f}_i(p) = O_i \left((f_j(p))_{j \in \mathcal{J}(p)} \right). \tag{12}$$

Note that the closed-form solution for our operators enable us to seamlessly switch to this lower-dimensional field space. In the remainder of this section, for sake of simplicity, we re-number the M active layers at p within $[1...M]$.

6.2 Taking cloth thickness into account

Applying the operators on values $f = (f_1, \ldots, f_M)$ leads to 0-iso-surfaces that are in exact contact when inter-penetrations are corrected. While this property was desirable to design our untangling operator, we actually aim at modeling cloth layers that may have a non-negligible physical thickness, and that will anyway need to be located on different surfaces for launching an animation. In the following, we include the required void space between garment layers aimed at avoiding ill-conditioned simulation within the notion of "thickness" of a cloth layer.

A naive approach to handle thickness would be to leave some geometric gaps between garment meshes and their target implicit surfaces, during the process of meshes projection to the associated implicit surfaces. Unfortunately, this could result in new collisions with neighboring layers due to this extra thickness. In contrast, our approach relies on directly integrating thickness values $(t_i)_{i \in [1, N]}$ within the expression of the operators, allowing to seamlessly and robustly handle collisions between thick layers.

Let us consider that the cloth surface, and thus the iso-surfaces defined by $O_i(f) = 0$, is centered within the associated, thick cloth layer. As a result, layer i should remain at a minimal distance of $t_i/2 + t_{i+1}/2$ from layer $i+1$, at a minimal distance $t_i/2 + t_{i+1} + t_{i+2}/2$ from layer $i + 2$, etc.

We model this effect by applying offsets in field space which convert into adequate displacements of the iso-surfaces in the 3D space. More precisely, applying an offset to a field f_j leads to a geometric displacement of the layer j along the gradient of the field ∇f_j. In our case, f_j is computed as a HRBF having, by construction, a unit gradient norm on the sampled surface points, and thus, at first approximation, $\simeq 1$ in the neighborhood of the 0-isosurface. Therefore applying a small offset δ in field space leads to displacement of the layer j along its normal by a length $\simeq \delta$.

Thanks to this property, we take into account the offset on the M active layers with respect to the current layer i by applying a change of variables before applying the operator. More precisely, we replace $O_i(f)$ in Equation (12) by $O_i(\tilde{f})$, with $\tilde{f} = (\tilde{f}_1, \ldots, \tilde{f}_M)$ defined as:

$$\tilde{f}_j = \begin{cases} f_j & if \ i = j \\ f_j + \frac{t_i}{2} + \frac{t_j}{2} + \sum_{k=i+1}^{j-1} t_k & if \ j > i \\ f_j - \frac{t_i}{2} - \frac{t_j}{2} - \sum_{k=i-1}^{j+1} t_k & if \ j < i. \end{cases} \tag{13}$$

6.3 Projecting mesh vertices to their iso-surface

The modified operators we just presented are the ones used for deforming the meshes towards their corrected, untangled configuration. This is done by interleaving two relaxation processes at each vertex of the mesh: *gradient relaxation* consists in moving points along the gradient $\nabla \tilde{f}_i$, toward their original iso-value, while *tangential relaxation* tends to make them slide along the iso-surfaces of $\hat{f}_i$ so that the distortion of mesh triangles is minimized.
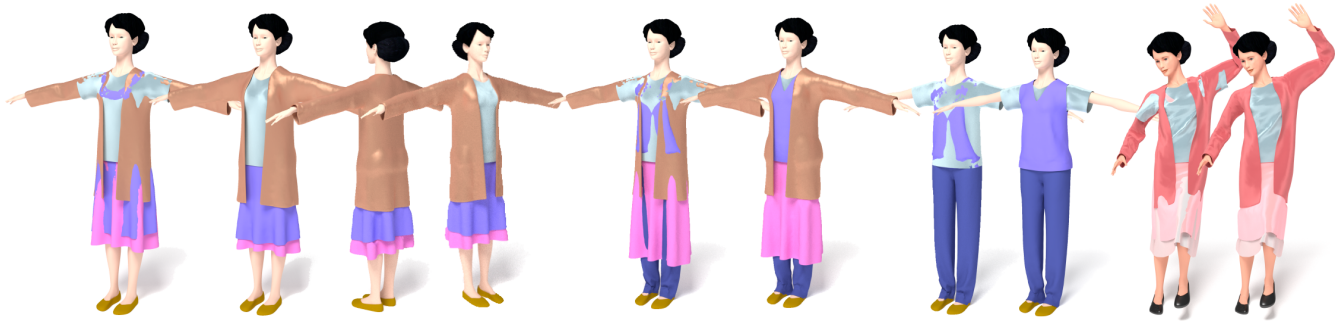
Fig. 7. Various untangled clothing on top of the same mannequin body (the initial colliding configuration is shown on the left of each example). Note that the resulting silhouette is strongly influenced by the worn layers. Right: application of our method on a different pose.

Gradient relaxation is computed using Newton iterations using the gradient value given by

$$\nabla \hat{f}_i = \sum_j \frac{\partial O_i}{\partial f_j} \, \nabla f_j \;\;, \tag{14}$$

where the ∇f_j are obtained using trilinear interpolation of field values pre-stored in a grid and $\frac{\partial O_i}{\partial f_j}$ is computed from the closest point $f_i^{b_0, c_0}$ as mentioned in Section 5. In practice, we use a step length $= 0.3$.

Tangential relaxation is inspired from As Rigid As Possible deformations (ARAP) [Sorkine and Alexa 2007]. The key adaptation to our case is to compute per-edge rotations and length changes: the rotation computed for each edge is the minimal rotation around its center that makes it tangential to the field, and the additional length change is a symmetric displacement of the vertices enabling the edge to restore its original length. These are used as target displacements for the two vertices defining the edge. At each tangential relaxation step, each mesh vertex applies an average of the displacements assigned the the adjacent edges.

Interleaving this second relaxation process with the more standard gradient descent enables-us to avoid over-elongated or inverted triangles, ensuring that the corrected iso-surface will be well approximated in deformed regions.

7 RESULTS AND DISCUSSION

7.1 Implementation

All times measured in this paper were taken on a standard laptop computer with an Intel quad Core i7 CPU, clocked at 3.1GHz with 32GB of RAM. Our software uses up to 1Gb of RAM memory at run-time for the presented examples (including the storage of all fields and their gradients stored in uniform grid of size 256×256×64) As our approach treat each vertex independently during mesh deformation described in Section 6, we use OpenMP to trivially parallelize our code. The rendered images were computed off-line using Blender and 3DSMax renderer for the animations. A real-time screen-capture of our software is shown in the accompanying video.

7.2 Qualitative results

Several results of our method are depicted in Fig. 1 and Fig. 7, showing it can be used to model a variety of layered clothing while ensuring collision-free states. We note that although aimed at providing a collision-free configuration for the garments, our method is also able to generate a quite plausible initial configuration, enabling the user to test the look of the virtual character even before launching animations.

Fig. 8-left illustrates a change of layer order between a rigid jacket and a flexible t-shirt. Note how the strong rigidity of the jacket influences the visible silhouette of the t-shirt when the latter is above. Another example of exchange of layer order is shown in Fig. 1-middle between a t-shirt and a trouser. Fig. 8-right shows the action of the mannequin body which is modeled as a layer of infinite rigidity and has visible action around the hips.

Fig. 8. Influence of the interior layers on the visible silhouette. Left: Exchange between a rigid leather jacket and a flexible t-shirt. Right: Result with and without the mannequin body.

We tested our approach on the extreme case of a character wearing 9 layers (including the body) and show the result in Fig. 9. This validates the robustness of the method in high dimensional fields space, with strongly tangled initial configuration, and deep inter-penetration. Vertical and top-to-bottom cuts are provided to illustrate the well behaved collision-free geometry of all internal layers, in comparison to the initial state.

As explained previously, user defined weights w_i can model the relative influence between layers when collision is corrected. Changing these weights allows us to tune the relative amount of deformation of the layers from fully rigid to fully flexible. Fig. 10 shows a

horizontal cut through the dresses of the model shown in Fig. 1-left when weights are modified.

Finally, as shown in Fig. 1-right, our untangled model can be directly plugged in as the initial condition of common cloth simulators to be animated without requiring any manual modification on the surface geometry. Note that the animated version of this model is provided in the accompanying video.

Fig. 9. Untangling an extreme configuration made of nine initially colliding layers. Cuts along layers are shown on both the initial input (left) and on the untangled result (right). A horizontal cut is shown across the dress in bottom, while the left-most and right-most cuts are performed on a 45° corner and zoomed-in to check that the resulting surfaces are fully exempt of collisions.
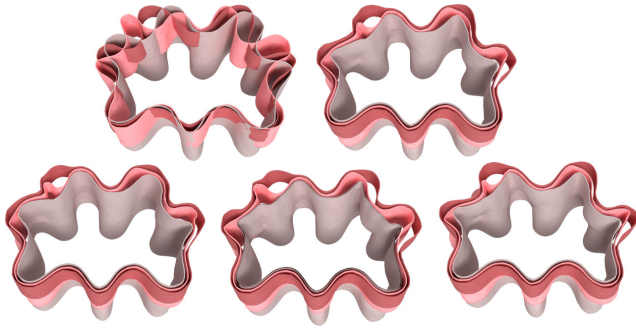

Fig. 10. Horizontal cut through the red dress layers shown in Fig. 1-left. Top-left: initial colliding configuration. Top-right: untangled configuration when all layers have the same weights. Bottom: layers weights are set, respectively from left to right, to $(w_1, w_2, w_3) = (2.5, 1, 1)$, $(w_1, w_2, w_3) = (1, 2.5, 1)$, $(w_1, w_2, w_3) = (1, 1, 2.5)$. Note how the most-rigid layer mostly-keeps its original shape and deforms the surrounding ones.

7.3 Quantitative results

The overall number of operations to untangle a model is $O(nKN^3)$, where N is the number of cloth layers, n is the number of vertices,

and K is the number of steps required in the iterative deformation. Indeed, the cubic complexity is brought by the field evaluation, while this evaluation has to be performed for every vertex until converging toward the 0-iso-surface. For all our examples we have $K \leq 15$, while N is at most 9. As a result for a constant number of layers and iterations, our untangling algorithm is linear with respect to the number of vertices. Time variation with respect to N in shown in Table 1. Table 2 shows our computation time for different cases and validates the roughly linear dependency of the computation time with respect to n. We can also note that, in practice, timings strongly depend on the number of vertices in collisions that need to be corrected. Therefore deep penetration of multiple layers are more computationally costly that correcting slight superficial ones. Table 3 provides a measure of the error in the user-defined layer thickness due to the approximation detailed in Section 6.2. Lastly, the precomputation of the fields for each garment takes less than 4 seconds for all our examples.

Table 1. Runtime in seconds with respect to the number of layers and of vertices, for two examples of increasing complexity. The 3 left (resp. right) columns correspond to the example depicted Fig. 1-middle (resp. Fig. 9) on which we added layers one by one. On the right, notice how deep penetrations involving the 3 first layers cause the runtime to drop for the entire example, while on the left the penetrations are superficial and are resolved in a few iterations.

N	#Vertices n	Time(s)	N	#Vertices n	Time(s)
3	6460	0.12	3	7751	1.04
4	9658	0.20	4	9200	1.29
5	15543	0.36	5	13673	2.35
6	19672	0.87	6	16871	3.45
7	18740	1.45	7	23385	4.54
			8	22453	6.80

Table 2. Each pair of lines correspond to the same example, on which we subdivided the meshes two times. Line one and two : example Fig. 1-middle, with one less layer. Line three and four : example Fig. 1-right. Line 5 and 6 : example Fig. 8-second picture. We can note that our method exhibits a linear complexity with respect to the number of vertices.

	N	#Vertices1	#Vertices2	#Vertices3
	3	5062	20062	79816
Time		0.22 s	0.76 s	2.71 s
	4	7502	29635	117785
Time		0.28 s	1.06 s	3.56 s
	5	10694	42315	168275
Time		0.57 s	2.1 s	7.15 s

7.4 Limitations

Although our method tends to generate plausible configurations in most cases — thanks to the proper handling of relative weights, and thicknesses — it stops considering the influence of a layer as soon as a point is out of the associated influence region. While this is not a problem with respect of getting correct collision-free configurations,

Table 3. Measures of error between user defined thickness value and computed one. Note that the error increases with the thickness value and local curvature of the surface.

Mesh	δ_{target}	$\delta_{obtained}$	Relative error
Jacket	0.20	0.1983	0.85%
PullV	0.09	0.0901	0.11%
TShirt	0.04	0.0406	1.5%
Petticoat	0.04	0.0421	5.3%

this results in an overly distorted shape for the underlying layers, such as we can see in Fig. 11 for the parts of the skirts immediately below the jacket. If our method is an initialization before animation this is not a problem, but if our results are to be directly used as an illustration, some local relaxation is required, as shown in Figure 11 (right).

Fig. 11. Left : limitations. The salient distortion that occurs near the border of the influence zone of a top layer (left) can be attenuated using relaxation (center), if our results are to be directly used as plausible shapes. Right : Even when launched in collision-free states, standard cloth simulators often fail to generate collision-free motion for layered garments. In future works, our approach could also be used to correct such collision happening in a dynamic context.

To achieve a better level of plausibility, our method would also need to consider the fact that garments deform isometrically with the associated 2D pattern: even is the cloth is slightly extensible, they tend to fold rather than compress. Consequently, folds should be added to the new shapes of inner layers, when they get compressed by a stiffer layer on top. This could be done by using inspiration from Rohmer et al. [2010], which makes use of an implicit model for folds. The latter could be integrated as an additional displacement within our operator.

In addition, our method suffers from a few failure cases. Indeed, it only applies when a well-defined nesting between different cloth layers can be defined. This is not the case for a single, self-intersecting garment (see Fig. 12-left). In such case, the self-intersecting cloth surface cannot be reconstructed as some zero-sets of implicit fields to be untangled, which prevent the use of our method. Enabling implicit untangling to be applied on local surface patches computed on the fly would be a nice extension of our method, since it could allow us to handle the challenging case of self-collision states [Ainsley et al. 2012].
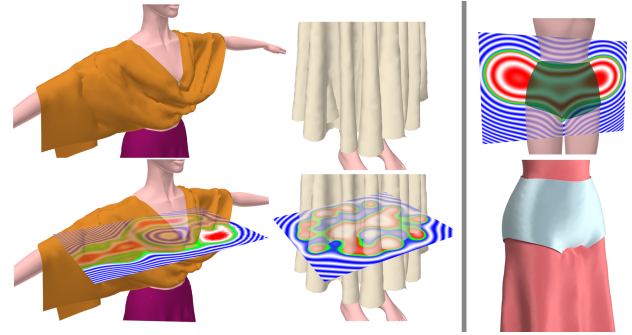

Fig. 12. Failure cases: on the left, a shawl folded back onto itself and a large skirt in deep intersection with itself cause the implicit reconstruction to fail: no clean 0-isosurface can be defined, which makes our approach inadequate for processing such self-collision case. On the right, the co-variant field h_i computed for a shorty does not capture the influence zone of the shorty between the legs, leading the shorty to be ignored as an outer layer when the vertices of the dress are processed in this zone.

Another possible failure case is illustrated in Fig. 12-right, where a shorty is worn on top of a long dress. In this case, the nesting order is not captured globally by the field h_i, which leads to a final result where collision still occurs. Extending the definition of h_i, possibly taking into account user indications, could also be handled as a future work.

Lastly, being able to apply our untangling method at each step of an animation would be highly desirable, since cloth simulation engines may not be able to maintain collision-free states during highly dynamic motion, even if we provide one to start with. While applying our current method is feasible, it would require reconstructing the field and co-variant field of each garment at each frame, leading to a few tenths of seconds of computational time. Taking temporal coherence into account to allow us a more efficient reconstruction over time would be an interesting direction for future work.

8 CONCLUSION

We described an implicit solution for untangling layered garments. Our method allows to robustly convert penetration states into collision-free contact regions, while taking into account both a thickness parameter for cloth layers and their relative rigidity. Key to our approach are the use of a pair of HRBFs, which are closed implicit surfaces, to approximate open garment surfaces, the introduction of closed-form N-ary untangling operators for layered implicit surfaces, and a new method for accurately projecting cloth meshes within the associated implicit field.

The configuration we generate for the dressed character can serve as valid initial state for launching simulations as layers are guarantee to not intersect, and are even separated by a user defined distance. Moreover, since relative cloth rigidity can be considered, this configuration remains close to the rest state in contact regions. This enables our method to be used for quickly generating plausible static shapes of garments that take any number of layers of underwear into account.

Our work opens up several new directions of research. First, the combination of the computed untangling deformations with the automatic generation of implicit folds could lead to more realistic shapes which would be useful to start simulation in a close to equilibrium state. Secondly, finding an efficient way to use implicit untangling at each animation step would enable robust multi-layer clothing animations, even for highly dynamic motions.

ACKNOWLEDGMENTS

REFERENCES

Samantha Ainsley, Etienne Vouga, Eitan Grinspu, and Rasmus Tamstorf. 2012. Speculative Parallel Asynchronous Contact Mechanics. *ACM Trans. Graph., Proc. ACM SIGGRAPH Asia* 31, 6 (2012).

Baptiste Angles, Marco Tarini, Brian Wyvill, Loïc Barthe, and Andrea Tagliasacchi. 2017. Sketch-based Implicit Blending. *ACM Trans. Graph.* (2017).

David Baraff, Andrew Witkin, and Michael Kass. 2003. Untangling Cloth. *ACM Trans. Graph., Proc. ACM SIGGRAPH* 22, 3 (2003), 862–870.

L. Barthe, V. Gaildrat, and R. Caubet. 1998. Combining implicit surfaces with soft blending in a CSG tree. In *Proc. of CSG Conference Series.* 17–31.

James F. Blinn. 1982. A Generalization of Algebraic Surface Drawing. *ACM Trans. Graph.* 1, 3 (1982), 235–256.

Jules Bloomenthal (Ed.). 1997. *Introduction to Implicit Surfaces.* Morgan Kaufmann.

Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Trans. Graph., Proc. ACM SIGGRAPH* 21, 3 (2002), 594–603.

Marie-Paule Cani. 1993. An implicit formulation for precise contact modeling between flexible solids. In *ACM SIGGRAPH.* 313–320.

Zhili Chen, Renguo Feng, and Huamin Wang. 2013. Modeling Friction and Air Effects Between Cloth and Deformable Bodies. *ACM Trans. Graph.* 32, 4 (2013).

Erwin de Groot, Brian Wyvill, and Huub van de Wetering. 2009. Locally restricted blending of Blobtrees. *Computers & Graphics* 33, 6 (2009), 690–697.

Charbel Fares and Ar Hamam. 2005. Collision detection for rigid bodies: A state of the art review. In *GraphiCon.*

Olivier Gourmel, Loic Barthe, Marie-Paule Cani, Brian Wyvill, Adrien Bernhardt, Mathias Paulin, and Herbert Grasberger. 2013. A Gradient-based Implicit Blend. *ACM Trans. Graph.* 32, 2 (2013).

Peng Guan, Loretta Reiss, David A. Hirshberg, Alexander Weiss, and Michael J. Black. 2012. DRAPE: DRessing Any PErson. *ACM Trans. Graph., Proc. ACM SIGGRAPH* 31, 4 (2012).

David Harmon, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. 2008. Robust Treatment of Simultaneous Collisions. *ACM Trans. Graph., Proc. ACM SIGGRAPH* 27, 3, Article 23 (2008).

P.-C. Hsu. 2018. K-ary Implicit Blends with Increasing or Decreasing Blend Ranges for Level Blend Surfaces. *Journal of Advances in Information Technology* (2018).

P. C. Hsu and C. Lee. 2003. Field Functions for Blending Range Controls on Soft Objects. *Proc. of Eurographics, Computer Graphics Forum* 22, 3 (2003), 233–242.

Armin Iske. 2002. Scattered Data Modelling Using Radial Basis Functions. *Tutorials on Multiresolution in Geometric Modelling. Mathematics and Visualization* (2002).

François Lehericey, Valérie Gouranton, and Bruno Arnaldi. 2015. GPU Ray-traced Collision Detection for Cloth Simulation. In *ACM Symp. on VRST.* 47–50.

Ives Macedo, Joao Paulo Gois, and Luiz Velho. 2011. Hermite Radial Basis Functions Implicits. *Comput. Graph. Forum* 30 (2011), 27–42.

Matthias Müller, Nuttapong Chentanez, Tae-Yong Kim, and Miles Macklin. 2015. Air Meshes for Robust Collision Handling. *ACM Trans. Graph.* 34, 4 (2015).

A. Pasko, V. Adzhiev, A. Sourin, and V. Savchenko. 1995. Function representation in geometric modeling: concepts, implementation and applications. *The Visual Computer* 11, 8 (1995), 429–446.

A. Ricci. 1973. Constructive Geometry for Computer Graphics. *Computer journal* 16, 2 (1973).

Damien Rohmer, Tiberiu Popa, Marie-Paule Cani, Stefanie Hahmann, and Sheffer Alla. 2010. Animation Wrinkling: Augmenting Coarse Cloth Simulations with Realistic-Looking Wrinkles. *ACM Transactions on Graphics, Proc. SIGGRAPH Asia* 29, 5 (2010), 157.

M-A Sabin. 1968. The Use of Potential Surfaces for Numerical Geometry. In *Tech. Report VTO/MS/153, British Aerospace Corp., Weybridge, U.K.*

Andrew Selle, Jonathan Su, Geoffrey Irving, and Ronald Fedkiw. 2009. Robust High-Resolution Cloth Using Parallelism, History-Based Collisions, and Accurate Friction. *IEEE TVCG* 15, 2 (2009), 12.

J. E. Solem and A. Heyden. 2004. Reconstructing open surfaces from unorganized data points. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, Vol. 2. II–II.

Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible Surface Modeling. In *Proc. Symposium on Geometry Processing.*

Liming Sun, Timo R. Nyberg, Gang Xiong, and Juntao Ye. 2016. Minimum Displacements for Cloth-obstacle Penetration Resolving. In *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: Short Papers (EG '16).* 53–56.

Min Tang, Dinesh Manocha, Miguel A. Otaduy, and Ruofeng Tong. 2012. Continuous Penalty Forces. *ACM Trans. Graph.* 31, 4 (2012).

Min Tang, Huamin Wang, Le Tang, Ruofeng Tong, and Dinesh Manocha. 2016. CAMA: Contact-Aware Matrix Assembly with Unified Collision Handling for GPU-based Cloth Simulation. *Computer Graphics Forum, Proc. Eurographics* 35, 2 (2016).

Min Tang, Tongtong Wang, Zhongyuan Liu, Ruofeng Tong, and Dinesh Manocha. 2018. I-Cloth: Incremental Collision Handling for GPU-Based Interactive Cloth Simulation. *ACM Trans. Graph. Proc. ACM SIGGRAPH Asia* 37, 6 (2018).

M. Teschner, S. Kimmerle, Gabriel Zachmann, B. Heidelberger, Laks Raghupathi, A. Fuhrmann, Marie-Paule Cani, François Faure, N. Magnenat-Thalmann, and W. Strasser. 2004. Collision Detection for Deformable Objects. In *Eurographics 2004, State-of-the-Art Report.* Eurographics Association, 119–135.

Rodolphe Vaillant, Loïc Barthe, Gaël Guennebaud, Marie-Paule Cani, Damien Rohmer, Brian Wyvill, Olivier Gourmel, and Mathias Paulin. 2013. Implicit Skinning: Real-time Skin Deformation with Contact Modeling. *ACM Trans. Graph.* 32, 4 (2013).

Rodolphe Vaillant, Gäel Guennebaud, Loïc Barthe, Brian Wyvill, and Marie-Paule Cani. 2014. Robust Iso-surface Tracking for Interactive Character Skinning. *ACM Trans. Graph.* 33, 6 (2014).

Pascal Volino and Nadia Magnenat-Thalmann. 2006. Resolving Surface Collisions Through Intersection Contour Minimization. In *ACM Trans. Graph., Proc. ACM SIGGRAPH.*

Holger Wendland. 2005. *Scattered Data Approximation, Cambridge University Press.*

Audrey Wong, David Eberle, and Theodore Kim. 2018. Clean Cloth Inputs: Removing Character Self-intersections with Volume Simulation. In *ACM SIGGRAPH Talks.*

Juntao Ye, Guanghui Ma, Liguo Jiang, Lan Chen, Jituo Li, Gang Xiong, Xiaopeng Zhang, and Min Tang. 2017. A Unified Cloth Untangling Framework Through Discrete Collision Detection. *Computer Graphics Forum* 36, 7 (2017).

J. Ye, T. R. Nyberg, and G. Xiong. 2015. Fast Discrete Intersection Detection for Cloth Penetration Resolution. In *IEEE Int. Conf. on Multimedia Big Data.* 352–357.

Juntao Ye and Jing Zhao. 2012. The Intersection Contour Minimization Method for Untangling Oriented Deformable Surfaces. In *Symposium on Computer Animation.*

C. Zanni, M. Gleicher, and M.-P. Cani. 2015. N-ary Implicit Blends with Topology Control. *Comput. Graph.* 46 (2015).