IMET - International Conference on Interactive Media, Smart Systems and Emerging Technologies (2023)
N. Pelechano, F. Liarokapis, D. Rohmer, and A. Asadipour (Editors)

Collision Free Simplification for 2D Multi-Layered Shapes

Xianjin Gong[1], Amal Dev Parakkat[2] and Damien Rohmer[1]

[1]LIX, Ecole Polytechnique/CNRS, IP Paris
[2]LTCI, Telecom Paris, IP Paris

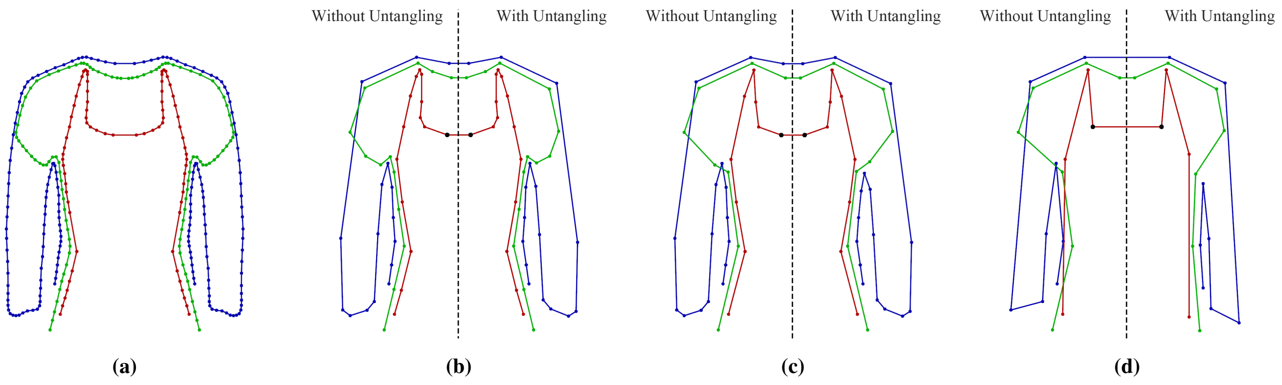

Figure 1: *Our 2D collision-free shape simplification is applicable to multiple stacked polylines. a) Original shapes depicting each layer with a different color. b,c,d) Simplification of 80%, 90%, and 95% of the initial vertices. The left side shows the result without our untangling approach, and the right side shows our results.*

Abstract

We propose a simplification-aware untangling algorithm for 2D layered shapes stacked on each other. While the shape undergoes simplification, our approach adjusts the vertex positions to prevent collision with other layers while simultaneously maintaining the correct relative ordering and offsets between the layers. The method features a field-based representation of the shapes and extends the concept of "implicit untangling" by incorporating interleaved shape preservation through a parameterized shape-matching technique. Our approach can be plugged on top of any existing vertex-decimation approach, leveraging its localized nature to accelerate the field evaluation. Furthermore, our method can seamlessly handle an arbitrary number of stacked layers, making it a versatile solution for stacked garment simplification.

CCS Concepts
• ***Computing methodologies*** $\rightarrow$ ***Mesh geometry models; Volumetric models;***

1. Introduction

Mesh simplification is widely used to adapt the level of details of 3D shapes. Since the seminal work of Garland et al. [GH97], multiple efficient approaches have been developed for efficient simplification revolving around the edge-collapse operation and can even adapt to dynamic levels of details for varying camera viewpoints or animation. However, some shapes still remain notoriously difficult to simplify. Among the most challenging ones are the multi-layered surfaces, when surface layers are tightly stacked on each other. Such a stacked configuration is commonly met in virtual character clothing, where each cloth layer is worn on top of the other. Applying a standard simplification scheme to such a layered structure would create tangled configurations that are visually perturbing and unsuited for simulation.

In this work, we propose a dedicated approach that handles layered shape simplification in a robust and efficient way. Our approach focuses on the 2D case, where the shape is described by a polyline that can typically represent a planar cut of a layered garment configuration. Our method considers an arbitrary simplification scheme applied on a set of layered 2D polylines and computes

DOI: 10.2312/imet.20231260

a geometric deformation, ensuring that the result is collision-free and preserves the local ordering of the polylines layers (see Fig. 1). To this end, we first associate the shape with an implicit form using Signed Distance Functions (SDF), allowing us to robustly express the notion of the interior and exterior of a shape. After the simplification process, we then apply a local iterative deformation scheme coupling an implicit-based deformer with a shape preservation approach to converge toward a collision-free state.

Our method builds upon the concept of implicit untangling proposed by Buffet et al. [BRB*19] but extends it along two technical contributions. Firstly, we extend the convergence process of the N-ary implicit-based deformer by coupling it with a shape-matching approach, which helps to limit non-rigid shape deformations. Secondly, we optimize the efficiency by restricting the implicit surface evaluation to a local area based on the simplification scheme and the shape topology.

2. Related Work

The main focus of this work is to perform collision-free simplification on 2D multi-layered shapes, specifically those that can be used on a planar cut of multi-layered clothes; we restrict our related works to Simplification and Clothes untangling.

2.1. Mesh Simplification

We categorize existing simplification algorithms into the three following groups:

Vertex Decimation approaches simplify models by removing vertices that minimize shape modifications [SL96, SZL97]. The empty space left by the vertex removal is then filled by re-triangulation. Decimation methods have been extensively studied in order to minimize various global error control, for instance, using simplification envelopes [CVM*96], or by re-evaluating the error after each iteration [CCMS96, BS96, KLS96, RR96, Gué96].

Vertex Clustering: Another way of simplification is to use one vertex as the replacement of a group of vertices. First, bounding boxes were used by Rossignac and Borrel [RB93] to cluster vertices, which is later improved by Kalvin and Taylor [KT96] using adaptive grid structures. Then, pair-wise vertex decimation was introduced by Garland [GH97] to reach higher efficiency, which groups vertices based on quadric error metrics.

Model Rebuilding: Simplification can also be done by rebuilding with new vertices sampled on the surface of the model [Tur92]. New vertices are sampled on maximal curvature locations, and their number is smaller than the model's old vertices.

While simplification approaches have been widely studied to preserve specific properties of a given surface, they do not take into account possible collisions with other surfaces resulting from the decimation. In contrast, our approach serves as a post-process, that rectifies a temporary state after a simplification step. This makes it compatible with any simplification method and leverages the localized nature of decimation to achieve efficient computation.

2.2. Collision avoidance

In the context of cloth simulation, the most common approaches are based on the collision handling pipeline introduced by Bridson et al. [BFA02]. This pipeline uses continuous collision detection (CCD) to detect and handle collisions between triangles of cloth meshes. Acceleration structures such as bounding volume hierarchies by Wong et al. [WLBY10] are proposed to increase the efficiency of CCD. While this methodology has achieved impressive results in complex simulation scenarios, it relies on the knowledge of a continuous displacement of the vertices with constant topology over the simulation step time. In our case, the simplification process can lead to an arbitrarily large displacement and change the shape's connectivity.

Untangling algorithms have been developed to resolve situations where shapes are tangled or colliding, without the need for prior knowledge of a mapping to a collision-free state. The seminal work of Baraff et al. [BWK03] introduced the use of Discrete Collision Handling (DCH) and Global Intersection Analysis to perform untangling. The original algorithm is limited to clothes without boundaries, and this limitation is removed by Wicke et al. [WLG06]. DCH method is further improved by Volino and Magnenat-Thalmann [VMT06] through the introduction of Intersection Contour Minimization. Recently, Cha and Ko [CK22] proposed a new DCH method with explicit convergence under certain conditions, which can guarantee the resolution of tanglements in finite time steps. However, because DCH is based on discrete intersection detection between pairs of meshes, each time, it can only handle at most two layers of clothes. For multi-layer clothes, it may happen that the solution of the first untangling will cause new tangling with other layers.

Another group of recent history-free cloth collision handling algorithms is based on volumetric models [GRH*12, CFW13, SNXY16, WEK18]. However, these works can only handle collision between a cloth mesh and a volumetric body. They use a projection mechanism to move penetrating cloth vertices out of the body volume and are not designed for untangling multi-layer clothes.

Buffet et al. [BRB*19] introduced a new algorithm for untangling by using scalar fields. The main idea is to combine scalar fields of tangled layers to find the scalar field of wanted untangled states. The advantage of this algorithm is that all corrections are done in one go. But because it uses iso-surface of scalar field to represent the untangled states of clothes, if the model is vertex-based, a method is required to move vertices to the corresponding iso-surface. The author proposed to use the gradient of the scalar field, but paths guided by the gradient are not evenly distributed and may intersect. This will not only modify the shape of the original clothes but it will also introduce new self-tangling. This paper improves this algorithm to make it compatible with vertex-based multi-layer models in 2D. Our method embeds the shape-matching algorithm proposed by Müller [MHTG05] to maintain the shape of clothes after untangling and to avoid self-tangling caused by gradient descent. Furthermore, because we assume that clothes are intersection-free before simplification and penetrations only happen at places of simplification, our method is localized to simplified areas, which largely improves efficiency.

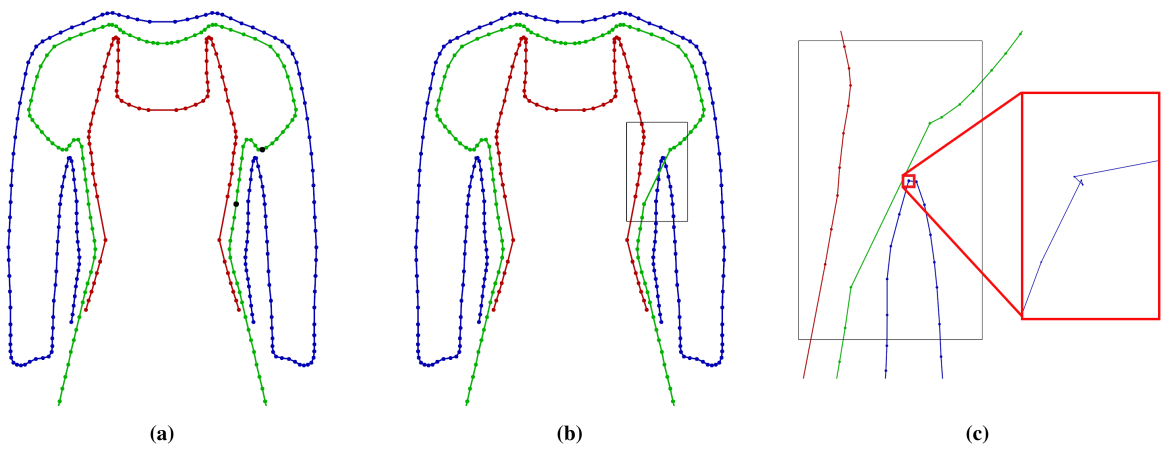

Figure 2: *Untangling without shape matching may lead to singularities and bad shape sampling. a) Initial models where the two black dots on the green layer indicate where the simplification will take place. b) Result obtained after simplification, before untangling. c) Two levels of zoom in the corrected area obtained after 150 steps of gradient descent. While the blue curve seems correctly untangled from far away, several vertices converged toward a similar position, leading to a local singularity with self-collisions.*

3. Shape preserving implicit untangling

This section describes our extended version of Implicit Untangling, providing a tradeoff between fast collision-free convergence and shape details preservation.

3.1. SDF-based implicit untangling

Suppose a set of shapes made of $N > 1$ locally stacked curves represented as 2D polylines. We assume that each curve $(c_i)_{i\in[1,N]}$ can be indexed from inside to outside. As such $i_1 < i_2 \Rightarrow c_{i_1}$ to be *under* the curve c_{i_2}. We associate to every curve c_i a Signed Distance Field (SDF) f_i such that c_i is the 0-isovalue of f_i, with

$$\forall \mathbf{x} \in \mathbb{R}^2, \left| \begin{array}{ll} f_i(\mathbf{x}) = 0, & \text{if } \mathbf{x} \in c_i \\ f_i(\mathbf{x}) > 0, & \text{if } \mathbf{x} \text{ is above } c_i \\ f_i(\mathbf{x}) < 0, & \text{if } \mathbf{x} \text{ is below } c_i \,. \end{array} \right. \tag{1}$$

As the 2D curves may represent a slide of a 3D cloth, they do not necessarily represent a closed contour. Thus, the notion of *above* and *below* have to be defined specifically. We rely on the approach described in [BRB*19] using a secondary *covariant* field defined from the border vertices, allowing us to robustly define the notion of interior/exterior from an open curve.

The Implicit Untangling operation relies on computing a new field $\hat{f}_i$ such that

$$\forall i \in [1,N],\ \hat{f}_i(\mathbf{x}) = \mathcal{O}_i(f_1(\mathbf{x}), \cdots, f_N(\mathbf{x}))\ , \tag{2}$$

where $\mathcal{O}_i$ is a composition operator defined in [BRB*19], and the zero-isovalue surface of $\hat{f}_i$ represents the untangled surface for the layer i. These operators can take into account per-layer weights ω_i and offsets o_i. These two parameters can relate to cloth stiffness, i.e. allowing some layers to be easier to deform than others, and cloth thickness, i.e. allowing a specific empty gap between two consecutive layers. Note that the Implicit Untangling approach only prevents collision between different layers, but cannot handle self-collision, which remains out of the scope of this work.

In our case, layers are initially represented as polylines; we can deform it to its untangled state by displacing the vertex positions onto the zero-set using an iterative gradient descent algorithm such that

$$\begin{aligned} \mathbf{x}^{n+1} &= \mathbf{x}^n + \Delta t\, \nabla \hat{f}_i(\mathbf{x}^n) \\ \hat{f}_i &= \sum_j \frac{\partial \mathcal{O}_i}{\partial f_j} \nabla f_j\ , \end{aligned} \tag{3}$$

where $\mathbf{x}^n$ (resp. $\mathbf{x}^{n+1}$) are the current (resp. next) positions, and Δt is the gradient descent step size.

While the original Implicit Untangling approach relied on the use of the Hermite Radial Basis function, we rather use SDF, providing a simpler mapping between distances measured in the field space and in the spatial domain as $\|\nabla f_i\| = 1$. Still, the use of the gradient descent algorithm may lead to artifacts and large shape deformation that we propose to tackle in the following part.

3.2. Shape preserving untangling

Applying a gradient descent within the field $\hat{f}_i$ on multiple vertices may lead to paths that can get arbitrarily close to each other or even cross when following discrete steps. This might result in undesirable outcomes like intersecting paths, different vertices ending up at the same final position, and self-tangling due to the change in its relative position (as shown in Figure 2). One solution suggested in [BRB*19] is to use tangential relaxation inspired by ARAP [SA07] that allows for matching constrained positions for a subset of vertices while preserving the appearance of its local neighborhood. However, ARAP only enforces rigid transformation, which leads to the apparition of sharp features. Our proposed method employs gradient descent steps interleaved with shape-matching [MHTG05] to preserve the shape's appearance. By leveraging shape matching, we gain finer control over the extent of deformation, allowing for the simulation of layer stiffness and quadratic deformation. This approach produces smoother and more desirable results, as discussed in Section 5.1.

Let us consider a local subset of N_v vertex positions $\mathbf{x}_i^0, i \in [1, N_v]$ before applying the gradient descent steps, and $\mathbf{x}_i$ their positions after a few steps of gradient descent from Eq. (3). We call $\mathbf{x}_{cm}$ (resp. $\mathbf{x}_{cm}^0$) the barycenter of these positions. We defined $\mathbf{q} = [q_x, q_y]^T = \mathbf{x}_i^0 - \mathbf{x}_{cm}^0$, $\mathbf{p} = [p_x, p_y]^T = \mathbf{x}_i - \mathbf{x}_{cm}$, and $\tilde{\mathbf{q}} = [q_x, q_y, q_x^2, q_y^2, q_x q_y]^T$. We then compute the optimal rotation $\mathbf{R}$ as

$$\begin{aligned} &\mathbf{R} = \mathbf{A}_{pq}\left(\sqrt{\mathbf{A}_{pq}^T \mathbf{A}_{pq}}\right)^{-1} \\ &\text{with } \mathbf{A}_{pq} = \left(\sum_i \mathbf{p}_i \mathbf{q}_i^T\right) \ , \end{aligned} \tag{4}$$

and transformation $\tilde{\mathbf{A}}$

$$\tilde{\mathbf{A}} = \left(\sum_i \mathbf{p}_i \tilde{\mathbf{q}}_i^T\right)\left(\sum_i \tilde{\mathbf{q}}_i \tilde{\mathbf{q}}_i^T\right)^{-1} \ . \tag{5}$$

By taking linear and quadratic terms formed by vertices' coordinates, the mode of transformation can be confined to shearing, stretching, bending, and twisting. Furthermore, the deformation and rotation parts can be adjusted using a weighted combination to compute the final vertex position $\bar{x}_i$ as

$$\bar{\mathbf{x}}_i = (1-\alpha)\mathbf{x}_i + \alpha\left[(\beta\tilde{\mathbf{A}} + (1-\beta)\tilde{\mathbf{R}})(\mathbf{x}_i^0 - \mathbf{x}_{cm}^0) + \mathbf{x}_{cm}\right] \ , \tag{6}$$

where the rotation matrix is extended to $\tilde{\mathbf{R}} \in \mathbb{R}^{2\times 5} = [\mathbf{R}\ \mathbf{0}\ \mathbf{0}]$ to make dimension consistent. The parameter $\alpha \in [0,1]$ is used to provide a tradeoff between the position obtained through gradient descent, and the one from shape-matching. High α values close to 1 fully preserved the original shape after deformation, but the vertices may deviate from the 0-isosurface. On the other hand, $\alpha = 0$ yields the result of gradient descent alone. The parameter β determines the desired level of deformation and stiffness in shape matching. A value of 0 simulates rigid deformation, while $\beta = 1$ incorporates all quadratic terms of the deformation.

Our full shape preserving untangling then works as follows. After performing a simplification, we then perform N_{gd} iterations of gradient descent from Eq. (3), followed by one iteration of weighted shape matching from Eq. (6). This interleaved process is iterated until converging to vertex positions that remain at fixed positions. We introduce the average displacement speed defined as

$$V = \frac{1}{N_v}\sum_{i=1}^{N_v}\frac{\|\bar{\mathbf{x}}_i - \mathbf{x}_i^0\|}{\Delta t} \ , \tag{7}$$

and set the convergence criteria to be $V \leq \varepsilon$. In our case, we typically considered N_{gd} around 20 to 30, and $\varepsilon = 0.1$.

4. Localization after Simplification

The calculation of SDF is the most computationally expensive part of the untangling algorithm. In 2D, calculating this SDF involves determining the shortest distance between a vertex under consideration to all line segments of the layer, resulting in a complexity of O(n), where n is the number of vertices in a layer.

Assuming that the input cloth models are free of intersection or a global untangling has been performed already, we do not require a global untangling procedure after simplification since the penetration only happens in the vicinity of the modified vertex. Based on this finding, only the vertices within a specific region around the simplified area have to be adjusted to untangle any potential intersections. We describe in the following a way to define this local region and propose in Algorithm 1 a summary of the overall process.

Algorithm 1 Localized Collision Free Simplification

```
while Simplification is not finished do
    Perform one simplification iteration
    Create a bounding box $B$ for modified vertices
    Extend sides of $B$ by double the total thickness of clothes
    Create empty cloth model $C'$
    for Each layer $L_i$ in original clothes model $C$ do
        Find the first and the last vertex contained in $B$
        if Vertices are found then
            Record their index as $start$ and $end$
            $start \leftarrow \max(start - 7,\ 0)$
            $end \leftarrow \min(end + 7,$ size of $L_i)$
            Append a new empty layer $L_i'$ to $C'$
            Append $V_{start}$ to $V_{end}$ to $L_i'$
        end if
    end for
    while Untangling does not converge do
        Perform untangling on $C'$
    end while
    Replace corresponding vertices in $C$ by $C'$
end while
```

The main criteria to define this local region is to ensure that it contains all the modified vertices from all layers. To achieve this, we initially find the minimum connected components that include all simplified vertices in every cloth layer. These connected components are then extracted as new cloth models, and the untangling algorithm is applied to them. The untangled components are then finally stitched back to the original model to complete our algorithm. A simple solution to defining the local region would be to use the minimum bounding box containing these connected components. However, stopping the curve abruptly at the bounding box border would lead to boundary artifacts when evaluating the field. Our approach consists of slightly extending the box and the vertices used to compute the field and the shape matching in order to avoid such artifacts.

Firstly, the Implicit Untangling models the cloth thickness using a prescribed offset in the SDF space. Here, the thickness of a cloth layer is given by the user to indicate the minimum gap between the current layer and its neighboring layers. In order to take into account such thickness in the spatial domain, we expand the size of the bounding box by twice the total thickness of all the layers, thus allowing enough space to replace the vertices at their correct offsets after deformation. Secondly, we aim to achieve a continuous field evaluation by integrating additional vertices outside the bounding box. For each connected component that intersects the bounding box, we include N_{out} extra contiguous vertices in the evaluation. These additional vertices serve two purposes: they contribute to the smoothness and well-defined behavior of the fields

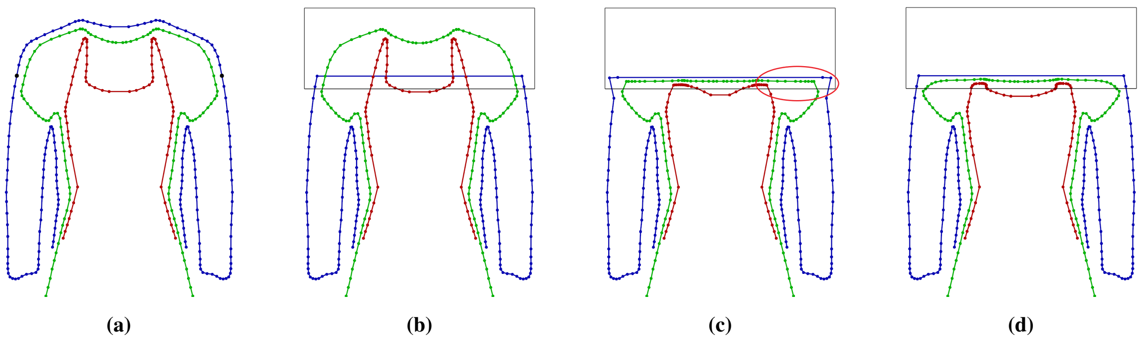

Figure 3: *Untangling localization: a) Original configuration of the model. b) Simplification applied on the blue model before untangling. c) Untangling applied within the bounding box without taking into account the N_{out} extra vertices. Note the sharp corners and flat surface on the green layer due to the sudden field discontinuity near the border. d) Considering N_{out} additional vertices outside of the box allow smooth and well-behaved untangled solutions for the green and red curves despite the extreme simplification of the blue layer.*

at the boundary, and they aid the shape-matching process in providing additional boundary conditions that prevent arbitrary vertex displacements near the border (as shown in Figure 3). We found that $N_{out} = 7$ suited our requirements, but this value can be adjusted case by case based on the sampling distance between vertices for different models.

5. Experiments and Results

We describe in the following the results and analysis obtained with our simplification aware untangling. All tests were performed on a PC Intel i7-10750H, 2.6GHz, using a C++ implementation.

Our result featuring the case of a cloth-looking scenario is illustrated in Fig. 1. Starting with a high-resolution initial configuration, we perform a series of simplifications at 80%, 90% and 95% of total vertex decimation. We compare at each step the result obtained without untangling and with our approach. We can note that our approach avoids collision between layers and ensures that each layer remains in its local order, while still preserving the global appearance of the initial cloth-looking model.

A key advantage of our result is its ability to handle an arbitrary number of layers. We highlight its robustness in Fig. 4 featuring 8-stacked layers on top of the others. The result of such a challenging case is obtained in 8s.

We then detail in the following a quantitative evaluation regarding the effectiveness of our approach in terms of shape preservation using shape-matching parameters, and speed-up from our local field evaluation.

Because this paper focuses on the untangling process, the simplification method used here is the simplest one, where random vertices are deleted, and their neighbors are connected. The influence of different choices of simplification remains as a future work.

5.1. Shape preservation parameter

We first propose the study of the influence of the parameters α and β used to define the shape preservation in Eq. (6).

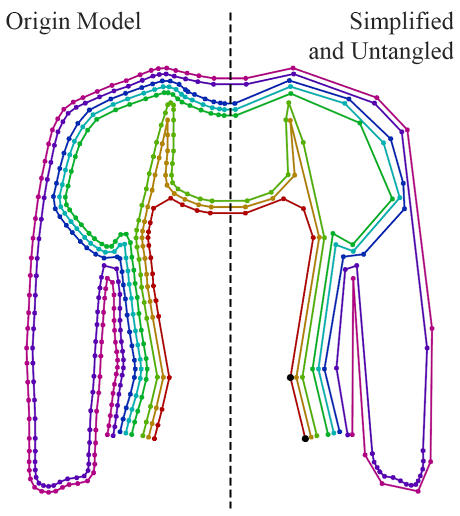

Figure 4: *Collision Free Simplification on 8-Layer Model*

Experiment Setup We consider a 5-layer model illustrated in Fig. 5. The red layer is the innermost layer, and the purple layer is the outermost layer. The third layer is perturbed, while others are curved and smooth. In the image (b), we set α and β to 0.99 and 0.9. Subsequently, in the image (c), we fixed the value of α while decreasing β to 0.2. In the image (d), we fixed the value of β and decreased α to 0.9. In image (e), the result is obtained using ARAP by replacing shape matching. In this study, twenty gradient descent iterations are performed before applying shape matching or ARAP.

Result and analysis In our algorithm, α controls how much shape matching is going to influence the result. The higher α is, the more two shapes are matched. β controls how much quadratic deformation is contained. The higher the value, the more deformation models can have. By alternatively decreasing two parameters, we can observe that the result meets the expectation by getting more and more rugged. Instead of being just pushed away

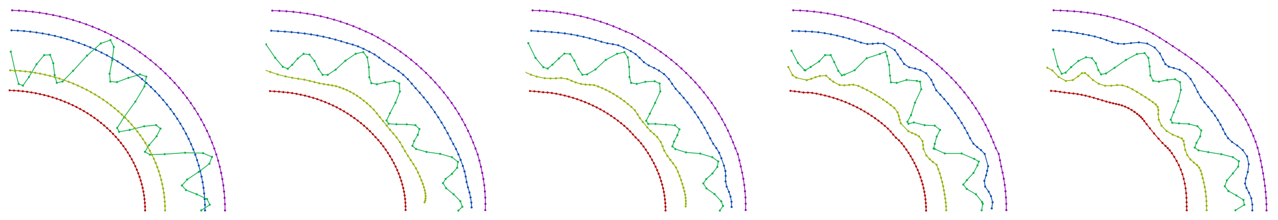
(a) *Origin Model*
(b) $\alpha = 0.99, \beta = 0.9$
(c) $\alpha = 0.99, \beta = 0.2$
(d) $\alpha = 0.9, \beta = 0.2$
(e) *ARAP*

Figure 5: *Results obtained in a test case when varying the shape preservation parameters α and β. a) The original model before untangling. b), c), d) Result obtained by using shape matching with different α and β as indicated in the captions. e) Result obtained by ARAP.*

by the middle green layer, the rest four layers deformed more to fit the shape of the middle layer when α and β are low.
Compared with ARAP, shape matching is better because we gain the ability to control how neighboring layers interact with each other through adjusting α and β. Furthermore, if we take a closer look at the result in the image(e), the innermost red layer is strangely influenced by the shape of the middle layer. This is because ARAP only matches the old and new positions of vertices, which are marked as constraints. The rest of the vertices are moved to best match the old and new Laplacian of the surface. Although layers produced by ARAP are the smoothest in terms of second-order derivative, large artificial bumps were created, and they can not be adjusted within the algorithm.

5.2. Localization of Untangling

We provide a second experiment quantifying the speed-up provided by our local untangling model.

Experiment Setup We consider the configuration shown in Fig. 2-b, with $N_{gd} = 30$, $\alpha = 0.8$ and $\beta = 0.9$. Firstly, we compute how much time is needed to perform 150 untangling iterations. Secondly, we compare the total number of iterations needed for convergence and the corresponding computational time.

Result and analysis For the first test, 150 steps of non-local untangling are computed in 9.7s, while the local version requires only 0.2s. For the second test, the non-local version reaches convergence in 300 iterations after 19.2s. The local version required 600 iterations that were computed in 0.8s.
These results reveal several key observations. Firstly, it is evident that the localized version significantly reduces the required time. As mentioned earlier, the main computational cost of the untangling process arises from the calculation of the Signed Distance Function (SDF), which is directly proportional to the number of vertices. Consequently, when the ratio of untangling region vertices to the total number of vertices is lower, the time saved by localization becomes more pronounced. Secondly, an interesting finding is that the global untangling approach necessitates fewer iterations to achieve convergence compared to the localized version. This discrepancy arises because, during global untangling, shape matching is performed on a global scale. As a result, the untangled vertices are heavily influenced by the fixed vertices, causing the cloth layer to become much stiffer with vertices remaining in their original positions. One potential solution to mitigate this issue is to reduce the value of α in Equation 6, although this may introduce the risk of self-tangling. Therefore, shape matching not only reduces processing time but also allows for an increase in α, thereby avoiding self-tangling and preserving fine details.

5.3. Weights in Implicit Untangling

We propose a final test checking the relation between the magnitude of deformation as a function of the layer weights ω_i. While not yet used in this work, knowing such a relation could help optimize the untangling weights ω_i for a prescribed *deformation budget*.

Experiment Setup Tests are performed based on the same configuration in the image (a) of Fig. 5. Here We considered $\alpha = \beta = 0.9$. We considered for the third dark-green layer a varying ω_3 on the interval $[0,1]$, while adapting the other weights such that $\sum_i \omega_i = 1$. The amount of deformation is computed as the summed distance between the vertices before, and after, the deformation.

Result and analysis The plot of the shape deformation as a function of the ω_3 weight is shown in Fig. (6). The graph shows a non-linear relation between the deformation and the weights, even if we consider $\mathcal{O}$ to be a linear composition. Indeed, while the fields $\hat{f}_i$ can be a linear function of the other fields, they are evaluated at different positions during the untangling iterations. As such, the displacement of the vertices relative to the other layers introduces a non-linear relationship. However, leveraging an approximation of such a relation for future optimization could be a future topic for our research.

6. Performance on More Complex Models

When clothes are stacked up layer by layer and flat, our algorithm works well. But when clothes are rolled up or folded, our algorithm can not be applied generally. This is because the untangling algorithm moves vertices based on whether vertices are on the outside or the inside of other layers. If some vertices need to be placed at the outside when the majority of the layer is on the inside, or if the definition of outside and inside is ambiguous, the algorithm will fail. We showed three examples in Fig. 7 to demonstrate.

In the image (a), the sleeves of the inner and the outer clothes are rolled up. For each vertex, it is straightforward to determine whether it is on the inside or the outside of the other layer. So, the

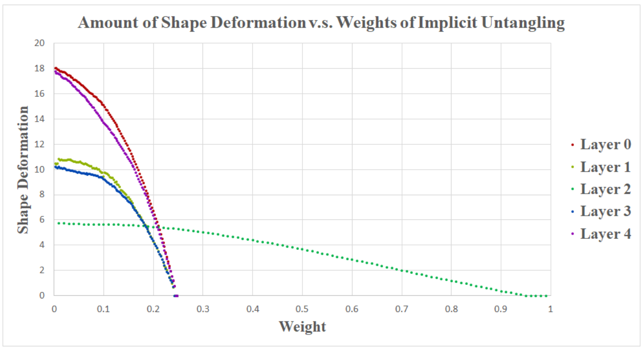

Figure 6: *Experiment results of different weights for implicit untangling.*

untangling algorithm successfully separated the rolled-up part to fulfil the gap requirement.

In the image (b), only the sleeves of the inner layer are rolled up. In this case, for rolled-up vertices, they are on the outside of the outer layer. So, the untangling algorithm fixed them by pushing them to the inside, which introduced new intersections. We have not found a good way to modify the untangling algorithm so that it can treat vertices in the same layer differently without introducing intersecting. This could be a possible direction for the future work.

In image (c), the case is more complicated. Sleeves of both layers are rolled up twice. For the vertices at the end of the inner (red) layer, they are on the outside of some segments of the cyan layer while on the inside of the other segments. Meanwhile, these segments are stacked closely to each other. The definition of inside and outside, in this case, is ambiguous. It can not perfectly guide the algorithm to move vertices to untangle the clothes. As a result, new intersections are introduced.

7. Conclusions and Future Work

We proposed a collision-free simplification algorithm for 2D multi-layer shapes that can be applied to cloth modeling. Our algorithm can be combined with any simplification algorithms but is able to take advantage of their locality to increase its efficiency. Our algorithm improved the *Implicit Untangling* algorithm by using a weighted shape matching to prevent self-untangling and preserve clothes' details. The apparent stiffness of clothes can be fined-tuned with both the parameters from implicit tangling ω_i and the shape matching (α_i, β_i). Our approach can seamlessly handle an arbitrary number of stacked layers without suffering from stability issues.

Our implementation is currently limited to 2D cases, but we plan to extend it to full 3D surfaces in future work. While the untangling and the shape-matching processes are independent of the embedding dimension, the localization speed-up may need to be adapted for 3D meshes in order to retrieve efficiently connected components. Our longer-term objectives include integrating shape-matching modification directly in the field estimation, thus improving efficiency. Finally, our preliminary study of the relation between the shape deformation and its untangling weights requires refinement. Being able to express a closed-form relation would open the possibility of finding optimal stiffness automatically in order to better preserve the shape appearance. We also consider the alternative use of a data-oriented approach. Incorporating a learning-based mechanism to establish the mapping without explicit formulation could alleviate the need for manual adjustments and potentially improve the overall performance even further.

References

[BFA02] Bridson R., Fedkiw R., Anderson J.: Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. 21*, 3 (jul 2002), 594–603. URL: https://doi.org/10.1145/566654.566623, doi:10.1145/566654.566623. 2

[BRB*19] Buffet T., Rohmer D., Barthe L., Boissieux L., Cani M.-P.: Implicit untangling: A robust solution for modeling layered clothing. *ACM Trans. Graph. 38*, 4 (jul 2019). URL: https://doi.org/10.1145/3306346.3323010, doi:10.1145/3306346.3323010. 2, 3

[BS96] Bajaj C. L., Schikore D. R.: Error-bounded reduction of triangle meshes with multivariate data. In *Visual Data Exploration and Analysis III* (1996), vol. 2656, SPIE, pp. 34–45. 2

[BWK03] Baraff D., Witkin A. P., Kass M.: Untangling cloth. *ACM SIGGRAPH* (2003). 2

[CCMS96] Ciampalini A., Cignoni P., Montani C., Scopigno R.: Multiresolution decimation based on global error. *The Visual Computer 13* (1996), 228–246. 2

[CFW13] Chen Z., Feng R., Wang H.: Modeling friction and air effects between cloth and deformable bodies. *ACM Trans. Graph. 32*, 4 (2013). URL: https://doi.org/10.1145/2461912.2461941, doi:10.1145/2461912.2461941. 2

[CK22] Cha I.-H., Ko H.-S.: Tanglement resolution in clothing simulation with explicit convergence. *IEEE Transactions on Visualization and Computer Graphics 28*, 7 (2022), 2764–2775. doi:10.1109/TVCG.2020.3039566. 2

[CVM*96] Cohen J., Varshney A., Manocha D., Turk G., Weber H., Agarwal P., Brooks F., Wright W.: Simplification envelopes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1996), SIGGRAPH '96, Association for Computing Machinery, p. 119–128. URL: https://doi.org/10.1145/237170.237220, doi:10.1145/237170.237220. 2

[GH97] Garland M., Heckbert P. S.: Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (1997), SIGGRAPH, p. 209–216. URL: https://doi.org/10.1145/258734.258849, doi:10.1145/258734.258849. 1, 2

[GRH*12] Guan P., Reiss L., Hirshberg D., Weiss A., Black M.: Drape: Dressing any person. *ACM Transactions on Graphics - TOG 31* (2012). doi:10.1145/2185520.2185531. 2

[Gué96] Guéziec A.: *Surface simplification inside a tolerance volume*. IBM TJ Watson Research Center New York, NY, USA, 1996. 2

[KLS96] Klein R., Liebich G., Strasser W.: Mesh reduction with error control. In *Proceedings of Seventh Annual IEEE Visualization* (1996), IEEE, pp. 311–318. 2

[KT96] Kalvin A., Taylor R.: Superfaces: polygonal mesh simplification with bounded error. *IEEE Computer Graphics and Applications 16*, 3 (1996), 64–77. doi:10.1109/38.491187. 2

[MHTG05] Müller M., Heidelberger B., Teschner M., Gross M.: Meshless deformations based on shape matching. *ACM Trans. Graph. 24*, 3 (jul 2005), 471–478. URL: https://doi.org/10.1145/1073204.1073216, doi:10.1145/1073204.1073216. 2, 3

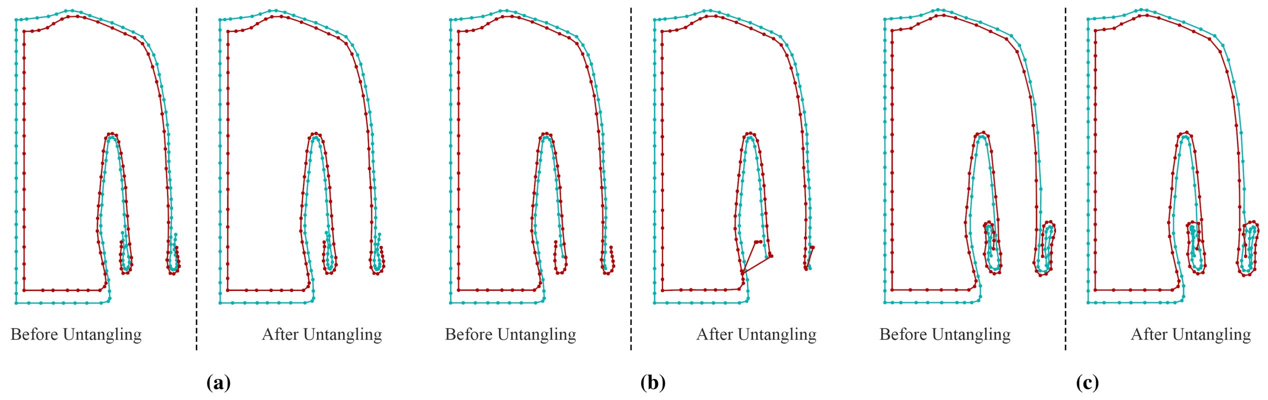

Figure 7: *a) Sleeves of both layers are rolled up once. The untangling algorithm successfully modified the model to fulfil the gap requirement. b) Only sleeved of the inner layer is rolled up once. The untangling algorithm failed because it can not handle the case when one layer is above and below another layer at different segments. c) Sleeved of both layer are rolled up twice. The untangling algorithm failed because vertices of the inner layer lie at certain regions, which can be considered simultaneously as the inside and the outside of the outer layer with respect to different segments.*

[RB93] ROSSIGNAC J., BORREL P.: Multi-resolution 3d approximation for rendering complex scenes. *IFIP Series on Computer Graphics* (01 1993), 455–465. doi:10.1007/978-3-642-78114-8_29. 2

[RR96] RONFARD R., ROSSIGNAC J.: Full-range approximation of triangulated polyhedra. In *Computer graphics forum* (1996), vol. 15, Wiley Online Library, pp. 67–76. 2

[SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (2007), SGP, Eurographics Association, p. 109–116. 3

[SL96] SOUCY M., LAURENDEAU D.: Multiresolution surface modeling based on hierarchical triangulation. *Comput. Vis. Image Underst. 63*, 1 (jan 1996), 1–14. URL: https://doi.org/10.1006/cviu.1996.0001, doi:10.1006/cviu.1996.0001. 2

[SNXY16] SUN L., NYBERG T. R., XIONG G., YE J.: Minimum displacements for cloth-obstacle penetration resolving. In *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: Short Papers* (2016), EG '16, Eurographics Association, p. 53–56. 2

[SZL97] SCHROEDER W., ZARGE J., LORENSEN W.: Decimation of triangle meshes. *SIGGRAPH Comput. Graph. 26* (06 1997), 65–70. doi:10.1145/133994.134010. 2

[Tur92] TURK G.: Re-tilingpolygonal surfaces. *Edwin E. Catmull, editor, ACM Computer Graphics (SIGGRAPH '92 Proceedings) 26* (jul 1992), 55–64. 2

[VMT06] VOLINO P., MAGNENAT-THALMANN N.: Resolving surface collisions through intersection contour minimization. *ACM Trans. Graph. 25*, 3 (jul 2006), 1154–1159. URL: https://doi.org/10.1145/1141911.1142007, doi:10.1145/1141911.1142007. 2

[WEK18] WONG A., EBERLE D., KIM T.: Clean cloth inputs: Removing character self-intersections with volume simulation. In *ACM SIGGRAPH 2018 Talks* (New York, NY, USA, 2018), SIGGRAPH '18, Association for Computing Machinery. URL: https://doi.org/10.1145/3214745.3214786, doi:10.1145/3214745.3214786. 2

[WLBY10] WONG S.-K., LIU C.-M., BACIU G., YEH C.-C.: Robust continuous collision detection for deformable objects. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2010), VRST '10, Association for Computing Machinery, p. 55–62. URL: https://doi.org/10.1145/1889863.1889873, doi:10.1145/1889863.1889873. 2

[WLG06] WICKE M., LANKER H., GROSS M.: Untangling cloth with boundaries. *Vision, Modeling and Visualization* (2006). 2