Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2014)
Vladlen Koltun and Eftychios Sifakis (Editors)

Strain Based Dynamics

Matthias Müller Nuttapong Chentanez Tae-Yong Kim Miles Macklin

NVIDIA

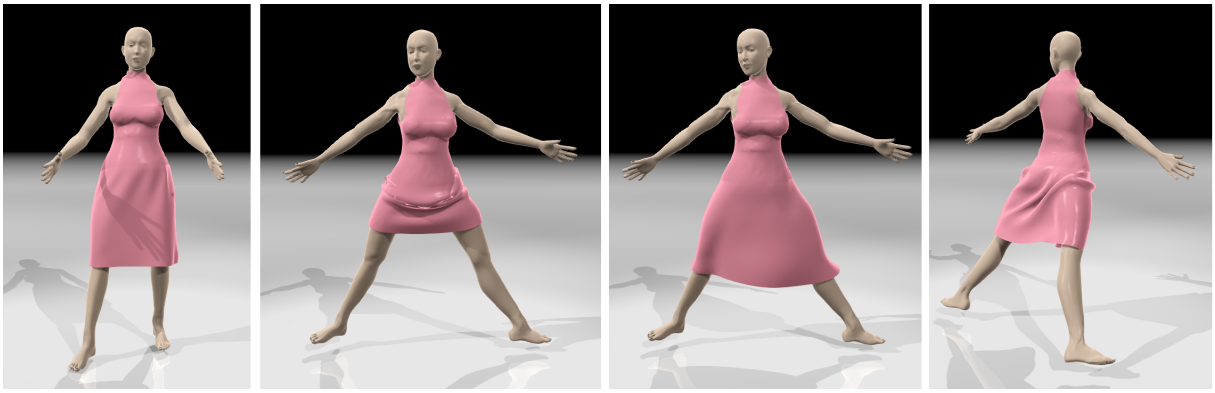

Figure 1: *From left to right: Tight skirt in the bind pose. Causing problems in poses that differ significantly from the bind pose. Lowering the overall stiffness results in sagging of the dress. Lowering the stiffness anisotropically along the horizontal direction solves the problem.*

Abstract

We propose a new set of constraints within the Position Based Dynamics (PBD) framework that allow the control of strain in directions that are independent of the edge directions of the simulation mesh. Instead of constraining distances between points, we constrain the entries of the Green - St Venant strain tensor. Varying the stiffness values corresponding to the individual strain coefficients lets us simulate anisotropic behavior.

By working with Green's rotation-independent, non-linear strain tensor directly we do not have to perform a polar decomposition of the deformation gradient as in most strain limiting approaches. In addition, we propose a modification of the constraints corresponding to the diagonal entries of the strain tensor such that they can be solved in a single step and a modification of the constraints corresponding to the off-diagonal entries to decouple stretch from shear resistance.

By formulating the constraints within the PBD framework, they can be used not only for strain limiting but to perform the actual simulation of the deformable object whereas traditional strain limiting methods have to be paired with a separate simulation method.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically Based Modeling I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation and Virtual Reality

1. Introduction

Position Based Dynamics [MHR06] has become a popular method for simulating cloth and soft bodies both in games and the film industry due to its simplicity, speed and robustness. Traditionally, meshes are simulated by constraining the distances along edges and bending angles. This approach controls strain only along the directions of the edges of the

DOI: 10.2312/sca.20141133

mesh. The same is true for mass spring systems, the corresponding force based method. In the force based framework the Finite Element (FEM) formulation removes this dependency and allows the specification of stiffness values in directions that are independent of the tesselation. Our goal was to come up with a set of constraints that achieve the same in the position based framework.

Instead of constraining edge lengths, we constrain the entries of the Green - St Venant strain tensor induced by the deformed particle positions. Unlike FEM which derives stresses from strains, integrates the deformation energy and derives forces, we work with the strain directly by deriving positional constraints for each entry of the strain tensor.

Being able to control strain along pre-defined directions independent of the triangulation is useful in a variety of scenarios. For instance, the stiffness of cloth typically varies along the warp and weft directions with relatively low shearing resistance. Often, artists provide animations with frames in which the clothing is more stretched than in the bind pose. Having the solver enforce the bind pose dimensions can cause collision issues and problematic configurations as shown in Figure 1. Reducing the stiffness an-isotropically yields stretchy cloth while reducing stiffness horizontally only solves the problem nicely. This behavior can be approximated by using edge constraints and regular triangulations that are aligned with the warp and weft directions. In practice, however, characters are often not designed with cloth simulation in mind. Moreover, diagonal edges in regular meshes still couple strain modes that should be independent.

Simulating a skin layer with a tetrahedral mesh is a second example in which mesh independent strain control is useful. Here skin sliding can be simulated by lowering the shear resistance along the surface of the character (see Fig. 5).

Locking is a third problem that can be addressed with our approach. It refers to the fact that perfectly constraining the edge lengths of a triangle mesh prevents it from being bent in general directions. English et al. [EB08] solve this problem by using non-conforming triangles. Since a non-conforming mesh is not suitable for collision handling, two meshes are necessary for simulation. Goldenthal et al. [GHF*07] only constrain the edges of a regular quad mesh to prevent locking. With out method, we achieve the same effect on arbitrary triangle meshes by reducing the shear stiffness only.

Our main contributions are

- Deriving PBD constraints from constraining the entries of the strain tensor to simulate deformable objects.
- Using Green's rotation-independent strain tensor instead of the Cauchy strain tensor and a polar decomposition of the deformation gradient.
- A modification of the diagonal constraints such that they can be solved in a single step.
- A modification of the off-diagonal constraints to decouple stretch from shear resistance.

2. Related Work

Since Müller et al. introduced Position Based Dynamics in [MHR06], various improvements have been proposed. Kubiak et al. [KPGF07] extended the method to simulate threads in surgical simulation. Müller [Mül08] introduced hierarchical PBD to reduce cost of simulating high resolution meshes. A new bending model for triangle meshes was devised by Kelager et al. [KNE10] and a new volume conservation constraint by Diziol et al. [DBB11]. Kim et al. [KCMF12] and Müller et al. [MKC12] studied the special case of one dimensional rods for hair and fur simulation and proposed fast ways to enforce inextensibility. For a survey on position based methods we refer the reader to [BMOT13].

The idea of using the strain components for cloth simulation was explored by Baraff and Witkin in [BW98] in a force based setting. We generalize this idea and apply it to cloth and tetrahedral objects in a position based framework.

Strain limiting in force based approaches is closely related to PBD. Here, the positions or the velocities of the vertices are manipulated directly after each force based simulation step. In contrast to our approach, strain limiting only kicks in when material is overstretched while we use the strain components to do the actual simulation of the material. On the other hand, our method could be used in a force based setting for strain limiting as well.

Provot [Pro95] and Bridson et al. [BMF03] proposed to constrain the length of springs to not stretch or compress beyond a given limit in the context of mass spring simulations. Hong et al. [HCJ*05] use an implicit formulation to allow for larger time steps. To increase the convergence rate Goldenthal et al. [GHF*07] used a global solver to constrain edge lengths of a regular quad mesh to upper limits.

In the context of Finite Element Method (FEM), Picinbono et al. [PDA03] limit strain anisotropically by adding an energy term to penalize strain in certain directions. Perez et al. [PCH*13] used Lagrange multipliers to constrain strain components isotropically. Hernandez at el. [HCPO13] improve this method to support anisotropic material.

The two most closely related methods are the ones proposed by Thomaszewski et al. [TPS09] and Wang et al. [WOR10]. Therefore we discuss the differences in a bit more detail. Both work on single elements and solve globally using either Gauss-Seidel or Jacobi iterations as we do.

Wang et al. [WOR10] limit strain isotropically. Their method is an extension of the strain limiting approach of Tsiknis at al. [Tsi06]. Wang et al. extract the principal strains from the deformation gradient of an element using a polar decomposition. Next, they clamp the principal strains and

compute a new clamped deformation gradient. This new gradient is then used to transform the initial shape of the element again to get the target shape. In contrast, we compute different position correction vectors for each individual shear and stretch mode which we combine using different scaling factors to simulate anisotropic behaviour.

Thomaszewski et al. [TPS09] work with Cauchy's linear strain tensor and extract the rotational part of the deformation gradient using a polar decomposition. With linear strain, the strain components depend linearly on the velocities of the adjacent vertices. Specifying the strain components yields a 6×6 linear system for the 3×2 velocity components of a triangle which has to be solved for each element in addition to performing a polar decomposition. This computation can be sped up by pre-computing the inverse of a 5×5 sub block which requires storing 25 floats per triangle, however. In the tetrahedral case, the linear system becomes 12×12 dimensional while in our case, processing tetrahedral elements is not much slower than handling triangles as our results show.

In both methods, attachments have to be handled separately. In the method of Thomaszewski et al. the linear system becomes over-constrained and has to be solved as a least squares problem. In contrast, in the PBD framework, vertices are simply attached by setting their inverse mass to zero.

3. Method

We will now derive the new strain-based position constraints that replace the traditional distance constraints. For the sake of completeness, we briefly recap the basic concepts of PBD first.

3.1. Basic PBD

Let us assume we have N particles with positions $\mathbf{x}_i$, velocities $\mathbf{v}_i$ and inverse masses w_i. The main PBD simulation loop has the following form:

```
initialize x_i and v_i
while simulating do
    v_i ← v_i + Δt f_i
    p_i ← x_i + Δt v_i
    p_i ← solve (p_i)
    v_i ← (p_i − x_i)/Δt
    x_i ← p_i
end
```

In the simulation loop, after updating the velocities, predicted positions $\mathbf{p}_i$ are computed using an explicit Euler step. These positions are modified by a solver to meet a set of positional constraints C_j. A positional constraint can be defined by a scalar function $C(\mathbf{p}_1, \dots \mathbf{p}_N)$ that is zero when the constraint is satisfied. The solver iterates multiple times over all constraints in a Gauss-Seidel type fashion solving the system of non-linear equations. For a single constraint C, the positional corrections $\Delta\mathbf{p}_i$ for point i is computed as

$$\Delta\mathbf{p}_i = -s\, k\, w_i \frac{\partial}{\partial \mathbf{p}_i} C(\mathbf{p}_1, \dots \mathbf{p}_N), \tag{1}$$

where

$$s = \frac{C(\mathbf{p}_1, \dots \mathbf{p}_N)}{\sum_j w_j \left| \frac{\partial}{\partial \mathbf{p}_i} C(\mathbf{p}_1, \dots \mathbf{p}_N) \right|^2} \tag{2}$$

and $k \in [0,1]$ stiffness parameter. The coefficient k is not a true physical stiffness coefficient because its effect is time step and iteration count dependent. However, in games where both are typically constant, k is an intuitive and easy to tune parameter.

The equations above result from a local linearization of the constraint function at the current particle configuration. What makes PBD so robust is the fact that the linearization is updated before each constraint projection and not held fixed as with global solvers. After the solve, the velocities and positions are updated based on the modified predicted positions. To give an example, $C(\mathbf{x}_1, \mathbf{x}_2) = |\mathbf{x}_1 - \mathbf{x}_2| - d$ constrains the distance between particles 1 and 2 to be d. Plugging this constraint function into Eq. (1) and Eq. (2) yields the intuitive corrections

$$\Delta\mathbf{p}_1 = -\frac{w_1}{w_1 + w_2} \left(|\mathbf{p}_1 - \mathbf{p}_2| - d\right) \frac{\mathbf{p}_1 - \mathbf{p}_2}{|\mathbf{p}_1 - \mathbf{p}_2|} \tag{3}$$

$$\Delta\mathbf{p}_2 = +\frac{w_2}{w_1 + w_2} \left(|\mathbf{p}_1 - \mathbf{p}_2| - d\right) \frac{\mathbf{p}_1 - \mathbf{p}_2}{|\mathbf{p}_1 - \mathbf{p}_2|} \tag{4}$$

3.2. Tetrahedral Constraints

We will now turn to our new constraints. Our basic idea applies to both triangle and tetrahedral meshes. Let us first consider a single tetrahedron. Instead of using constraints on the edges that control the pairwise distances of the particles, we derive constraints that involve all four particles and drive the configuration to a state in which the components of Green's strain tensor assume given values. To formulate these constraints we need the expressions of the strain components in terms of the positions of the four particles adjacent to the tetrahedron.

Let $\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ be their material positions and $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ the corresponding world positions. Since translation does not contribute to strain we can assume that $\mathbf{q}_0$ and $\mathbf{p}_0$ are zero. Then, with

$$\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3] \quad \text{and} \tag{5}$$

$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3] \tag{6}$$

we can express the deformation gradient as

$$\mathbf{F} = \mathbf{P}\mathbf{Q}^{-1} \tag{7}$$

and the Green - St Venant strain tensor as

$$\mathbf{G} = \mathbf{F}^T\mathbf{F} - \mathbf{I} \tag{8}$$

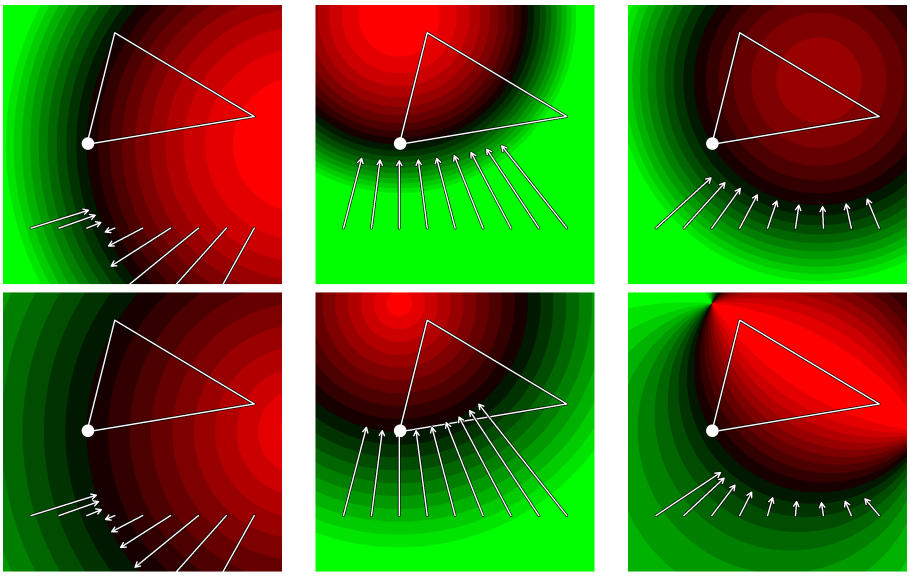

Figure 2: *Constraint values and projection vectors for the white vertex moving in the plane with the other two vertices are attached. The three columns show x-, y- and shear strain (green positive, red negative, clamped). On the top row we used the standard strain functions* $\mathbf{S}_{11}-1$, $\mathbf{S}_{22}-1$ *and* $\mathbf{S}_{12}$. *The bottom row depicts our modified constraints functions* $\sqrt{\mathbf{S}_{11}}-1$, $\sqrt{\mathbf{S}_{22}}-1$, *and* $\mathbf{S}_{12}/|\mathbf{f}_i||\mathbf{f}_j|$. *Note that the zero sets are unchanged. In contrast to the top row, the x- and y- constraints are solved with a single step because the gradient is constant along the projection direction. The modified shear function tilts the projection vectors to decouple shear from stretch resistance.*

where $\mathbf{I}$ is the identity matrix. We dropped the factor $\frac{1}{2}$ in the original definition of Green's strain tensor because it cancels out in our constraint formulation. The matrix $\mathbf{Q}^{-1}$ is constant and can be pre-computed.

The diagonal entries of $\mathbf{G}_{ii}$ represent stretch and the off-diagonal entries $\mathbf{G}_{ij}=\mathbf{G}_{ji}$ shear both with respect to the main axes in the material frame. We now introduce the three stretch and three shear constraint functions

$$C(\mathbf{p0},\mathbf{p1},\mathbf{p2},\mathbf{p3}) = \mathbf{S}_{ii} - s_i^2 \tag{9}$$

$$C(\mathbf{p0},\mathbf{p1},\mathbf{p2},\mathbf{p3}) = \mathbf{S}_{ij} \qquad i<j, \tag{10}$$

where $\mathbf{S}=\mathbf{F}^T\mathbf{F}$ and s_i rest stretches, typically equal to 1. These constraints pull the particles towards states of zero stretch and zero shear. Associating separate stiffness coefficients k_{ij} with each constraint lets us simulate anisotropic material. Note that no rotation matrices have to be estimated via polar decomposition as in most strain limiting methods because rotational independence is built into the definition of Green's strain tensor.

3.3. Triangle Constraints

For triangles the corresponding matrices $\mathbf{P}$ and $\mathbf{Q}$ are not square because the number of particles is reduced by one but the dimensionality of the particle positions stays the same so $\mathbf{Q}^{-1}$ is not defined. This problem can be solved by the natural choice of defining the rest state of the triangles via two dimensional texture coordinates on the triangle mesh. To properly simulate anisotropic cloth behavior the texture coordinates must be aligned with the weft and warp directions of the cloth. Now $\mathbf{Q}$ becomes a 2×2 dimensional matrix and $\mathbf{Q}^{-1}$ is well defined. The definition

$$\mathbf{S} = \mathbf{Q}^{-1^T}\mathbf{P}^T\mathbf{P}\mathbf{Q}^{-1} \tag{11}$$

is valid too with $\mathbf{S},\mathbf{Q}\in\mathbb{R}^{2\times 2}$ and $\mathbf{P}\in\mathbb{R}^{3\times 2}$. Note that the strain based constraints for tetrahedra and triangles are agnostic to reflections which will be considered in Section 3.7.

3.4. The Square Root of Strain

Although these are natural definitions of the constraint functions, there is a more stable way to formulate the stretch constraints in Eq. (9). To see this, let us have a look at a simple distance constraint with rest length d between two points. The two constraint functions

$$C(\mathbf{p1},\mathbf{p2}) = |\mathbf{p}_1-\mathbf{p}_2| - d \qquad \text{and} \tag{12}$$

$$C(\mathbf{p1},\mathbf{p2}) = |\mathbf{p}_1-\mathbf{p}_2|^2 - d^2 \tag{13}$$

are both valid. However, the first function is linear along $\mathbf{p}_1-\mathbf{p}_2$ while the second is not. This means that the linearizing constraint projection of PBD can solve the first constrain perfectly in one step. This is not true for the second constraint function which corresponds to our measure of stretch.

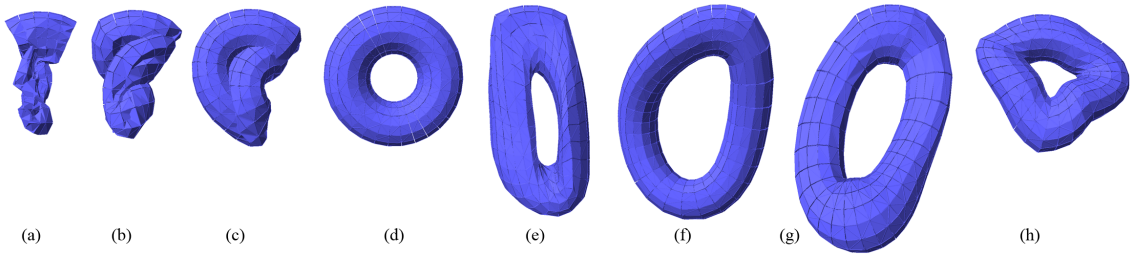

Figure 4: *Varying soft body stiffness parameters. Figures (a) - (d) show the recovery of a torus from a heavily entangled state by increasing the volume stiffness. For (e) we reduced all but the volume conservation stiffness values. As a result, the torus heavily deforms but its volume is conserved. Figure (f) shows the result of only softening the volume stiffness and the stiffness along the main axis of the torus. The result of high shear and low stretch resistance is shown in Figure (g) where angle distortion is small while the shape is stretched. Figure (h) shows the opposite configuration. Here, stretching is small while the torus bends heavily.*

The problem can easily be fixed by replacing Eq. (9) by

$$C(\mathbf{p0},\mathbf{p1},\mathbf{p2},\mathbf{p3}) = \sqrt{\mathbf{S}_{ii}} - s_i. \tag{14}$$

With this modification, the stretch constraint of an element is solved correctly with a single projection step because the gradient of the constrain function is constant along the projection direction as the bottom images of the first two columns of Fig. 2 show. For a derivation of the position corrections based on the proposed constraints see the Appendix. This modification reduces the relative remaining stretch with the same number of solver iterations by 25% on average. More importantly, it increases the stability of the simulation because it prevents overshooting.

3.5. Decoupling Shear from Stretch

The shear constraint function $\mathbf{S}_{ij}$ can be written as $\mathbf{S}_{ij} = \mathbf{f}_i \cdot \mathbf{f}_j$, where $\mathbf{f}_i$ and $\mathbf{f}_j$ are the i^{th} and j^{th} column vectors of $\mathbf{F}$. However, this function not only penalizes the angle between the axes of the deformed coordinate system, i.e. the dot product of the column vectors, but also the principal stretches, i.e. the magnitudes of the column vectors. To decouple strain from stretch we propose a modified shear constraint function

$$\bar{\mathbf{S}}_{ij} = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{|\mathbf{f}_i||\mathbf{f}_j|} = \frac{1}{|\mathbf{f}_i||\mathbf{f}_j|}\mathbf{S}_{ij}. \tag{15}$$

The modified constraint function and the corresponding PBD projections are shown in the last column of Fig. 2. Since the shear constraints are non-linear along the projection direction, overshooting is possible. To be on the safe side, one can reduce the shear coefficient as done in Fig. 2. We have not seen instabilities in our experiments due to the shear constraints though.

3.6. General Strain Orientation

Eq. (9) and Eq. (10) constrain strain along the global coordinate axes. In certain cases, this is not desirable. Let us assume we have a tetrahedral layer on the surface of a character for simulating skin and we want the skin to slide easily tangential to the surface but not normal to it. There is a simple and elegant solution to this problem. We do not even have to modify our formulation. All we need to do is to modify the rest shape of the tetrahedra. As a pre-computation step, a local frame is computed for each tetrahedron. In the example above, this frame would be spawned by the tangent and normal vector of the surface at the location of the tetrahedron. The constant rest positions $\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2$ and $\mathbf{q}_3$ are then simply stored with respect to this local frame.

3.7. Volume and Area Conservation

None of the constraints above controls the volume of tetrahedra or area of triangles on its own. Only if all constraints are satisfied at the same time, we have $\mathbf{G} = \mathbf{I}$ and, thus, $\det(\mathbf{F}) = 1$, i.e. conservation of volume. Often it is important to control volume conservation separately though. It allows the simulation of soft material with strong volume conservation for instance. Adding a separate volume/area conservation constraint is straight forward in the PBD framework. We simply define

$$C_{\text{volume}}(\mathbf{p}_0,\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3) = \det(\mathbf{F}) - 1 \text{ and} \tag{16}$$

$$C_{\text{area}}(\mathbf{p}_0,\mathbf{p}_1,\mathbf{p}_2) = \det(\mathbf{F}) - 1 \tag{17}$$

for 3D solids and 2D cloth respectively, where the first constraint corresponds to the volume conservation term proposed in [MHR06]. To be compatible with rest stretches

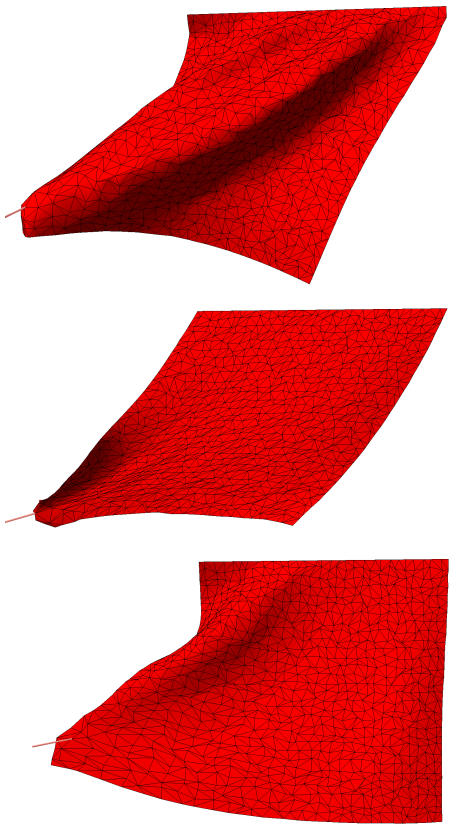

Figure 3: *Varying the cloth stiffness parameters of different strain components. From top to bottom the resistance to x-stretch, y-stretch and shear are: (high,high,high), (hight,high,low) and (low,high,high). Our method allows the control of these modes independently on triangle meshes with highly non-regular tessellations as the one used here.*

other than one these formulas need to be generalized to

$$C_{\text{volume}}(\mathbf{p}_0,\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3) = \det(\mathbf{F}) - s_1 s_2 s_3 \quad (18)$$

$$C_{\text{area}}(\mathbf{p}_0,\mathbf{p}_1,\mathbf{p}_2) = \det(\mathbf{F}) - s_1 s_2. \quad (19)$$

Another important feature of this constraint is that it handles element inversion for tetrahedra since

$$\det(\mathbf{F}) = 1 \quad (20)$$

$$\det(\mathbf{P})\det(\mathbf{Q}^{-1}) = 1 \quad (21)$$

$$\det(\mathbf{P}) = \det(\mathbf{Q}) \quad (22)$$

states that the *signed* volume of the tetrahedron must match its *signed* rest volume. In the Appendix we give the explicit formula for this constraint which shows that in case of volume inversion, vertices are projected across the base face to the correct side.

Element inversion is a general issue with FEM-based approaches because forces are in general derived from the strain tensor only. Various papers such as [ITF04] have concentrated on this problem alone. Our volume conservation term is a simple and effective solution as Fig. 4 shows.

Figure 5: *Skin sliding on an alien bull. The top image shows the skin deformation created by linear blend skinning which stretches the surface unevenly. Our method allows the simulation of a tetrahedral layer on the surface with low tangential shear resistance yielding correct skin sliding (bottom).*

However, adding an additional constraint for volume conservation can yield jittering due to the over-constrained system. This happens with extreme deformations when the stiffnesses of the strain and volume constraints are close to 1. We found in our experiments that softening the volume constraint by a small amount typically solves this problem.

3.8. Bending

In the case of cloth simulation, since strain is an intrinsic measure, the strain based constraints do not influence bending which is an extrinsic quantity. Therefore, bending needs to be handled separately as well.

To simulate bending resistance we constrain the dihedral angle of pairs of adjacent triangles as proposed in [MHR06]. However, we use simplified formulas for the derivatives of

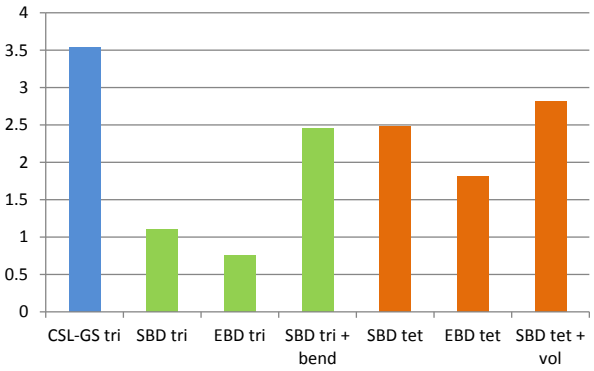

Figure 6: *Time in milliseconds for one iteration over 3600 elements, where CSL-GS stands for Continuum based Strain Limiting with Gauss-Seidel iterations, SBD for Strain Based Dynamics and EDB for Edge Based Dynamics.*

the dihedral angle w.r.t. the particle positions, namely the bending mode of a pair of triangles described by Bridson et al. [BMF03]. This mode corresponds one to one to the derivatives of the bending angle including scaling – a fact that is not mentioned in the paper. Only the sign has to be flipped depending on the orientation of the triangle pair as we show in the Appendix.

3.9. Damping

For damping, we use a general PBD formulation that specifically damps the relative velocities with respect to a constraint with positional correction $\Delta\mathbf{p}_1,\ldots,\Delta\mathbf{p}_N$ as

$$\mathbf{v}_i \leftarrow \mathbf{v}_i - k\left(\sum_{j=1}^{N}\mathbf{v}_j^T\mathbf{n}_j\right)\mathbf{n}_i \qquad (23)$$

where $k\in[0,1]$ is the corresponding stiffness and $\mathbf{n}_i=\frac{\Delta\mathbf{p}_i}{|\Delta\mathbf{p}_i|}$. Note that the term $\sum_{j=1}^{N}\mathbf{v}_j^T\mathbf{n}_j$ makes sure that damping is only applied to a given mode $\mathbf{n}_j$, without adding any artificial damping to rigid body motion as the sum cancels out when the mode itself does not change.

4. Results

In the experiment shown in Fig. 3, we used a rectangular piece of cloth with irregular triangulation. We modified the stiffness parameters with respect to x-stretch, y-stretch and shear individually. Even with a highly irregular mesh, the cloth shows the expected behavior.

Fig. 4 shows the tetrahedral case with a volumetric torus and locally aligned elements. To demonstrate the effectiveness of the volume term to correct inverted elements we started with the heavily entangled state in Fig. 4(a). As the first four images show, the shape recovers with only a few iterations. Playing with the stiffness parameters of the different modes lets us create interesting volumetric effects. In Fig. 4(c) we reduced all but the volume conservation stiffness. Reducing only the volume stiffness and the stiffness along the main axis creates the behavior shown in Fig. 4(f). With high shear and low stretch resistance in Fig. 4(g) the edge lengths are more deformed than the angles between edges. In the opposite configuration in Fig. 4(h), the torus is free to bend significantly while stretching is small.

Timings using a single thread on a Core i7 CPU at 3 Ghz are given in Fig. 6. It was not possible to extract performance numbers on strain limiting alone from the paper of Wang et al. [WOR10] because the authors only measured the time spent on the combination of strain limiting and simulation. From the performance numbers given by Thomaszewski et al. [TPS09] we were able to extract a time of 5.3 milliseconds for one iteration of strain limiting over 3600 triangles which we divided by a factor of 1.5 to compensate for the fact that they used a 2 Ghz CPU. Our method is more than three times faster than Continuum based Strain Limiting (CSL) and about 30 percent slower than Edge Based Dynamics (EBD). For tetrahedra, the slowdown of our approach w.r.t. EBD is slightly higher. In both approaches, the number of constraints increases from 3 to 6 but the number of vertices per constraint increases from 3 to 4 in SBD while it stays at 2 in EBD. Even though not analyzed by Thomaszewski et al. we expect CSL to be significantly slower for tetrahedra in which case the size of linear system per element increases to 12×12.

A common problem in clothing simulations is that the cloth fits well in the bind pose but causes problems in poses that differ significantly from it. Lowering the stiffness in general creates stretchy cloth. As the results in Figure 1 show, being able to lowering the stiffness horizontally only solves the problem.

We also used our approach for simulating a tissue layer on a character as shown in Figure 5. Making the shear constraints very soft lets us simulate the skin sliding effect in a simple way. In the given example, skin sliding significantly reduces the uneven stretching of linear blend skinning.

5. Conclusion

We have presented a new way to simulate deformable objects in the PBD framework. Instead of constraining the distances on edges, we derive sets of positional projections for the deformation modes corresponding to the entries of the Green - St Venant strain tensor. We proposed modifications of the constraints to make the projections more robust and to decouple shear from stretch. We also discussed volume conservation, bending and damping.

Our formulation is a step towards bridging the gap between PBD popular in games and continuum based FEM which is often considered to be too expensive for real-time

applications. With the relatively simple formulation of our framework and the explicit formulas for the position corrections given in the Appendix we hope that this work is practical enough to be widely adopted in the gaming and movie industry.

References

[BMF03] BRIDSON R., MARINO S., FEDKIW R.: Simulation of clothing with folds and wrinkles. In *ACM SIGGRAPH Symposium on Computer Animation* (2003), pp. 28–36. 2, 7, 9

[BMOT13] BENDER J., MÜLLER M., OTADUY M. A., TESCHNER M.: Position-based Methods for the Simulation of Solid Objects in Computer Graphics. pp. 1–22. 2

[BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 43–54. 2

[DBB11] DIZIOL R., BENDER J., BAYER D.: Robust real-time deformation of incompressible surface meshes. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2011), SCA '11, ACM, pp. 237–246. 2

[EB08] ENGLISH E., BRIDSON R.: Animating developable surfaces using nonconforming elements. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 66:1–66:5. 2

[GHF*07] GOLDENTHAL R., HARMON D., FATTAL R., BERCOVIER M., GRINSPUN E.: Efficient simulation of inextensible cloth. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007) 26*, 3 (2007), to appear. 2

[HCJ*05] HONG M., CHOI M.-H., JUNG S., WELCH S., TRAPP J.: Effective constrained dynamic simulation using implicit constraint enforcement. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (2005), pp. 4520–4525. 2

[HCPO13] HERNANDEZ F., CIRIO G., PEREZ A. G., OTADUY M. A.: Anisotropic strain limiting. In *Proc. of Congreso Español de Informática Gráfica* (2013). 2

[ITF04] IRVING G., TERAN J., FEDKIW R.: Invertible finite elements for robust simulation of large deformation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2004), SCA 04, Eurographics Association, pp. 131–140. 6

[KCMF12] KIM T.-Y., CHENTANEZ N., MÜLLER-FISCHER M.: Long range attachments - a method to simulate inextensible clothing in computer games. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2012), Eurographics Association, pp. 305–310. 2

[KNE10] KELAGER M., NIEBE S., ERLEBEN K.: A triangle bending constraint model for position-based dynamics. In *VRIPHYS* (2010), Erleben K., Bender J., Teschner M., (Eds.), Eurographics Association, pp. 31–37. 2

[KPGF07] KUBIAK B., PIETRONI N., GANOVELLI F., FRATARCANGELI M.: A robust method for real-time thread simulation. In *Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2007), VRST '07, ACM, pp. 85–88. 2

[MHR06] MÜLLER M., HENNIX B. H. M., RATCLIFF J.: Position based dynamics. *Proceedings of Virtual Reality Interactions and Physical Simulations* (2006), 71–80. 1, 2, 5, 6, 9

[MKC12] MÜLLER M., KIM T.-Y., CHENTANEZ N.: Fast simulation of inextensible hair and fur. In *VRIPHYS* (2012), Bender J., Kuijper A., Fellner D. W., Guérin E., (Eds.), Eurographics Association, pp. 39–44. 2

[Mül08] MÜLLER M.: Hierarchical position based dynamics. *Proceedings of Virtual Reality Interactions and Physical Simulations* (2008). 2

[PCH*13] PEREZ A. G., CIRIO G., HERNANDEZ F., GARRE C., OTADUY M. A.: Strain limiting for soft finger contact simulation. In *Proc. of World Haptics Conference* (April 2013), IEEE. 2

[PDA03] PICINBONO G., DELINGETTE H., AYACHE N.: Nonlinear anisotropic elasticity for real-time surgery simulation. *Graph. Models 65*, 5 (Sept. 2003), 305–321. 2

[Pro95] PROVOT X.: Deformation constraints in a mass-spring model to describe rigid cloth behavior. *Proceedings of Graphics Interface* (1995), 147–154. 2

[TPS09] THOMASZEWSKI B., PABST S., STRASSER W.: Continuum-based strain limiting. *Computer Graphics Forum 28*, 2 (2009), 569–576. 2, 3, 7

[Tsi06] TSIKINIS K. D.: Better cloth through unbiased strain limiting and physics-aware subdivision. 2

[WOR10] WANG H., O'BRIEN J., RAMAMOORTHI R.: Multiresolution isotropic strain limiting. *ACM Trans. Graph. 29*, 6 (Dec. 2010), 156:1–156:10. 2, 7

Appendix A: Explicit Formulas for Constraint Projection

Here we give the explicit formulas for the constraints to ease the implementation of our method. Remember that we assume $\mathbf{p}_0 = \mathbf{q}_0 = \mathbf{0}$.

Strain Based Constraints

Let $\mathbf{c}_i$ the columns of $\mathbf{Q}^{-1}$ and $\mathbf{f}_i$ be the columns of $\mathbf{F}$ i.e.

$$[\mathbf{c}_1, \mathbf{c}_2] = \mathbf{Q}^{-1} \tag{24}$$

$$[\mathbf{f}_1, \mathbf{f}_2] = \mathbf{F} \tag{25}$$

and

$$[\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3] = \mathbf{Q}^{-1} \tag{26}$$

$$[\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3] = \mathbf{F} \tag{27}$$

for triangles and tetrahedra, respectively. Then the entries of $\mathbf{S}$ can be computed as

$$\mathbf{S}_{ij} = \mathbf{f}_i \cdot \mathbf{f}_j = (\mathbf{P}\mathbf{c}_i) \cdot (\mathbf{P}\mathbf{c}_j), \tag{28}$$

where $i, j \in \{1,2\}$ for triangles and $i, j \in \{1,2,3\}$ for tetrahedra. The derivatives of the components of $\mathbf{S}$ w.r.t. the particle positions, needed in the PBD approach are

$$\nabla \mathbf{S}_{ij} = \left[\nabla_{\mathbf{p}_1}, \nabla_{\mathbf{p}_2}\right] \mathbf{S}_{ij} = \mathbf{f}_j \mathbf{c}_i^T + \mathbf{f}_i \mathbf{c}_j^T \tag{29}$$

$$\nabla \mathbf{S}_{ij} = \left[\nabla_{\mathbf{p}_1}, \nabla_{\mathbf{p}_2}, \nabla_{\mathbf{p}_3}\right] \mathbf{S}_{ij} = \mathbf{f}_j \mathbf{c}_i^T + \mathbf{f}_i \mathbf{c}_j^T \tag{30}$$

for triangles and tetrahedra respectively and

$$\nabla_{\mathbf{p}_0} \mathbf{S}_{ij} = -\sum_{k=1}^{d} \nabla_{\mathbf{p}_k} \mathbf{S}_{ij}, \tag{31}$$

where $d = 2$ for triangles and $d = 3$ for tetrahedra. Following [MHR06] we get the particle projection vectors w.r.t. $\mathbf{S}_{ij}$ as

$$\Delta \mathbf{p}_k = -\lambda w_k \nabla_{\mathbf{p}_k} \mathbf{S}_{ij}, \tag{32}$$

where w_k is the inverse mass of particle k and

$$\lambda = \frac{\mathbf{S}_{ii} - s_i^2}{\sum_k w_k \left|\nabla_{\mathbf{p}_k} \mathbf{S}_{ii}\right|^2}, \tag{33}$$

$$\lambda = \frac{\mathbf{S}_{ij}}{\sum_k w_k \left|\nabla_{\mathbf{p}_k} \mathbf{S}_{ij}\right|^2}, \tag{34}$$

$$\lambda = 2\frac{\sqrt{\mathbf{S}_{ii}} - s_i}{\sum_k w_k \left|\nabla_{\mathbf{p}_k} \mathbf{S}_{ii}\right|^2} \sqrt{\mathbf{S}_{ii}} \tag{35}$$

for Eqs. (9), (10) and (14), respectively. For the modified shear constraint function given in Eq. (15), the gradient is

$$\nabla \tilde{\mathbf{S}}_{ij} = \frac{1}{|\mathbf{f}_i||\mathbf{f}_j|} \nabla \mathbf{S}_{ij} - \frac{|\mathbf{f}_j|^2\, \mathbf{f}_i \mathbf{c}_i^T + |\mathbf{f}_i|^2\, \mathbf{f}_j \mathbf{c}_j^T}{|\mathbf{f}_i|^3 |\mathbf{f}_j|^3}\, \mathbf{S}_{ij}. \tag{36}$$

Material Coordinates for a Triangle

The texture coordinates $\mathbf{u}_i = (u_i, v_i)$ of the triangle vertices cannot be used directly as material coordinates for the rest state because they might contain stretching. To compute the material coordinates $\mathbf{q}_0$, $\mathbf{q}_1$, and $\mathbf{q}_2 \in \mathbb{R}^2$ we need an orthonormal local frame. Let the rest positions of the vertices of the triangle in world space be $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^3$. We can compute two world space tangential vectors $\mathbf{t}_u$ and $\mathbf{t}_v \in \mathbb{R}^3$ along the u and v axes as

$$(\mathbf{t}_u, \mathbf{t}_v) = (\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0)\,(\mathbf{u}_1 - \mathbf{u}_0, \mathbf{u}_2 - \mathbf{u}_0)^{-1} \tag{37}$$

These tangents give us the local frame to transform the global positions into material coordinates as

$$(\mathbf{c}_1, \mathbf{c}_2) = [\mathbf{n}_1, \mathbf{n}_2]^T\, (\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0), \tag{38}$$

where $\mathbf{n}_1 = \frac{\mathbf{t}_u}{|\mathbf{t}_u|}$ and $\mathbf{n}_2 = \frac{\mathbf{t}_v}{|\mathbf{t}_v|}$.

To make sure $\mathbf{t}_u$ and $\mathbf{t}_v$ are normal to each other, the latter can alternatively be computed as the cross product of $\mathbf{t}_u$ with the triangle normal.

Bending Constraint

Let $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ and $\mathbf{p}_4$ be the particles of a bending element consisting of the two triangles $(\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_4)$ and $(\mathbf{p}_2, \mathbf{p}_4, \mathbf{p}_3)$. The bending angle ϕ can be computed via the two triangle normals as

$$\phi = \arccos\left(\frac{(\mathbf{p}_3 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)}{|(\mathbf{p}_3 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)|} \cdot \frac{(\mathbf{p}_4 - \mathbf{p}_2) \times (\mathbf{p}_3 - \mathbf{p}_2)}{|(\mathbf{p}_4 - \mathbf{p}_2) \times (\mathbf{p}_3 - \mathbf{p}_2)|}\right) \tag{39}$$

The spatial derivatives correspond to the bending mode of [BMF03] and are

$$\nabla_{\mathbf{p}_1} \phi = |\mathbf{e}|\, \mathbf{n}_1 \tag{40}$$

$$\nabla_{\mathbf{p}_2} \phi = |\mathbf{e}|\, \mathbf{n}_2 \tag{41}$$

$$\nabla_{\mathbf{p}_3} \phi = \frac{(\mathbf{p}_1 - \mathbf{p}_4) \cdot \mathbf{e}}{|\mathbf{e}|} \mathbf{n}_1 + \frac{(\mathbf{p}_2 - \mathbf{p}_4) \cdot \mathbf{e}}{|\mathbf{e}|} \mathbf{n}_2 \tag{42}$$

$$\nabla_{\mathbf{p}_4} \phi = \frac{(\mathbf{p}_3 - \mathbf{p}_1) \cdot \mathbf{e}}{|\mathbf{e}|} \mathbf{n}_1 + \frac{(\mathbf{p}_3 - \mathbf{p}_2) \cdot \mathbf{e}}{|\mathbf{e}|} \mathbf{n}_2, \tag{43}$$

where

$$\mathbf{e} = \mathbf{p}_4 - \mathbf{p}_3 \tag{44}$$

$$\mathbf{n}_1 = \frac{(\mathbf{p}_3 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)}{|(\mathbf{p}_3 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)|^2} \tag{45}$$

$$\mathbf{n}_2 = \frac{(\mathbf{p}_4 - \mathbf{p}_2) \times (\mathbf{p}_3 - \mathbf{p}_2)}{|(\mathbf{p}_4 - \mathbf{p}_2) \times (\mathbf{p}_3 - \mathbf{p}_2)|^2}. \tag{46}$$

The signs of all the derivatives have to be flipped if $(\mathbf{n}_1 \times \mathbf{n}_2) \cdot \mathbf{e} > 0$.

Volume / Area Conservation Constraints

From Eq. (22) we have

$$C_{\text{volume}}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = \det(\mathbf{P}) - \det(\mathbf{Q}) \tag{47}$$

$$= \mathbf{p}_1^T (\mathbf{p}_2 \times \mathbf{p}_3) - \mathbf{q}_1^T (\mathbf{q}_2 \times \mathbf{q}_3) \tag{48}$$

and its derivatives

$$\nabla_{\mathbf{p}_1} C_{\text{volume}} = \mathbf{p}_2 \times \mathbf{p}_3 \tag{49}$$

$$\nabla_{\mathbf{p}_2} C_{\text{volume}} = \mathbf{p}_3 \times \mathbf{p}_1 \tag{50}$$

$$\nabla_{\mathbf{p}_3} C_{\text{volume}} = \mathbf{p}_1 \times \mathbf{p}_2 \tag{51}$$

$$\nabla_{\mathbf{p}_0} C_{\text{volume}} = -\mathbf{p}_2 \times \mathbf{p}_3 - \mathbf{p}_3 \times \mathbf{p}_1 - \mathbf{p}_1 \times \mathbf{p}_2 \tag{52}$$

$$\tag{53}$$

Similarly we can derive a constraint for preservation of triangle area as

$$C_{\text{area}}(\mathbf{p}_1, \mathbf{p}_2) = |\mathbf{p}_1 \times \mathbf{p}_2|^2 - |\mathbf{q}_1 \times \mathbf{q}_2|^2. \tag{54}$$

Its derivatives are

$$\nabla_{\mathbf{p}_1} C_{\text{area}} = 2\,\mathbf{p}_2 \times (\mathbf{p}_1 \times \mathbf{p}_2) \tag{55}$$

$$\nabla_{\mathbf{p}_2} C_{\text{area}} = 2\,\mathbf{p}_1 \times (\mathbf{p}_2 \times \mathbf{p}_1) \tag{56}$$

$$\nabla_{\mathbf{p}_0} C_{\text{area}} = -2\,\mathbf{p}_2 \times (\mathbf{p}_1 \times \mathbf{p}_2) - 2\,\mathbf{p}_1 \times (\mathbf{p}_2 \times \mathbf{p}_1) \tag{57}$$

$$\tag{58}$$

with the corresponding particle projection vectors

$$\Delta \mathbf{p}_k = -\lambda w_k \nabla_{\mathbf{p}_k} C_{\text{volume(area)}}, \tag{59}$$

where

$$\lambda = C_{\text{volume(area)}} / \sum_k w_k \left|\nabla_{\mathbf{p}_k} C_{\text{volume(area)}}\right|^2. \tag{60}$$