Robust Skin Weights Transfer via Weight Inpainting

Rinat Abdrashitov
Epic Games
Canada
rinat.abdrashitov@epicgames.com

Kim Raichstat
Epic Games
USA
kim.raichstat@epicgames.com

Jared Monsen
Epic Games
USA
jared.monsen@epicgames.com

David Hill
Epic Games
USA
david.hill@epicgames.com

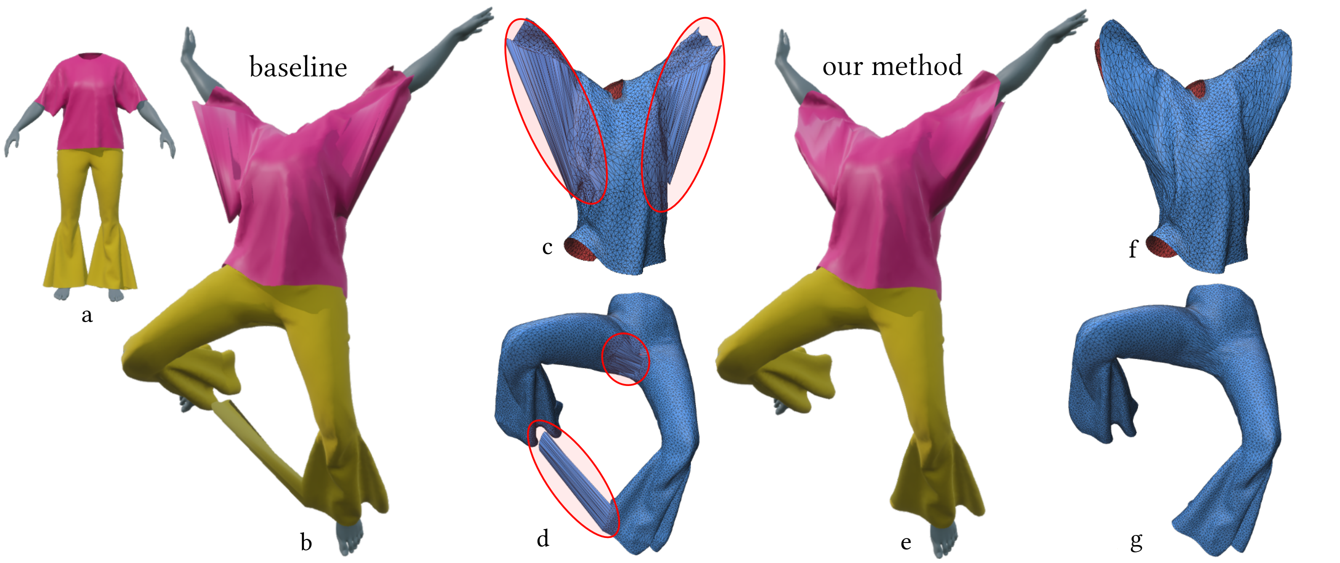


Figure 1: Skin weights are transferred from a rigged body to a t-shirt and pants (a). Current production methods fail to produce ready-to-use assets (b). Upon closer inspection, areas around the armpits (c), the crotch, and lower legs (d) are skinned incorrectly. In contrast, our method (e) does not suffer from the same issues (f, g). 3D models are the property of Epic Games, Inc., used with permission.

ABSTRACT

We present a new method for the robust transfer of skin weights from a source mesh to a target mesh with significantly different geometric shapes. Rigging garments is a typical application of skin weight transfer where weights are copied from a source body mesh to avoid tedious weight painting from scratch. However, existing techniques struggle with non-skin-tight garments and require additional manual weight painting. We introduce a fully automatic two-stage skin weight transfer process. First, an initial transfer is performed by copying weights from the source mesh only for those vertices on the target mesh where we have high confidence in obtaining the ground truth weights from the source. Then, we automatically compute weights for all other vertices by interpolating the weights computed in stage one. This approach is robust and easy to implement in practice, yet it far outperforms the methods used in existing commercial software and previous research works.

CCS CONCEPTS

• **Computing methodologies → Animation**.

KEYWORDS

Skin weights, Animation, Rigging

SA Technical Communications '23, December 12–15, 2023, Sydney, NSW, Australia

ACM Reference Format:
Rinat Abdrashitov, Kim Raichstat, Jared Monsen, and David Hill. 2023. Robust Skin Weights Transfer via Weight Inpainting. In *SIGGRAPH Asia 2023 Technical Communications (SA Technical Communications '23), December 12–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3610543.3626180

1 INTRODUCTION

High-quality virtual 3D garments are ubiquitous in film and games [Stuyck 2022] and are an important element in bringing digital characters to life. In real-time applications, the garments are often fully or partially deformed via linear blend skinning (LBS), which despite known artifacts, enjoys popularity due to its simplicity and speed [Jacobson et al. 2014]. These qualities make LBS a common choice for cloth deformation as an alternative to slower but more realistic physics-based methods. The virtual garment mesh is bound to the skeleton via skinning weights, where each vertex on the mesh is assigned a weight representing the amount of influence that a given bone has on the vertex. The rest pose vertex position is transformed by a linear combination of bone transformation matrices weighted by the appropriate skinning weights.

Due to the cloth being deformed on top of the body mesh, the cloth skinning pipeline often starts with weight transfer from the rigged body with subsequent manual edits. The amount of edits required depends on the type of clothing. If the cloth is skin tight (i.e., fitted T-shirt), then only a minimal post-edit is needed. However, for non-skin-tight clothing like loose T-shirts, dresses, flared pants, and many others, the result of the transfer produces artifacts. Fig. 1(b-d) and Fig. 3(b-f) show examples of the issues that arise from transferring weights using existing methods, like the cloth being incorrectly bound to nearby but unrelated character parts.

We introduce a new method for transferring skinning weights that significantly improves on all these issues (Fig. 1(e-g), Fig. 3(g-k)). Current state-of-the-art production tools [Autodesk Inc. 2023] are based on the simple algorithm that boils down to closest point query: for each vertex on the target mesh, find the closest point the source mesh and use barycentric coordinate interpolation to compute the weight that will be assigned to the current target vertex. Unlike previous methods, we do not copy the weights from the source mesh for every single target vertex. Instead, we only copy weights for those vertices where we have high confidence of getting ground truth weights. For all other vertices, we compute a solution based on the copied ground truth data. The solution is computed automatically by minimizing a combination of the Dirichlet and the Laplacian energies subject to fixed constraints posed as a quadratic problem which is easy to solve. This makes our method fast, and robust, requiring no manual post-processing and ready to be deployed as a drop-in replacement for existing methods in a production software application.

2 RELATED WORK

Despite being an important artist tool for skinning meshes, the topic of weight transfer has received little attention. [Yoon et al. 2016] proposed a method that constructs moving planes for use as the projection reference planes. The clothing vertices on a plane are then projected onto the skin on the same plane. The method, however, requires the manual selection of key joints and a pivot point to construct the planes. And while showing improved results in the armpit and shoulder areas of the clothing, it's not clear how well the method will generalize to other parts of the body. [Got 2019] build a transfer tool based on the Geodesic Voxel Binding algorithm [Dionne and de Lasa 2013] and rely on the voxelization of both the source and the target meshes. It performs closest point queries by computing geodesic distances inside the voxelized meshes, taking into consideration the models' topology. While this avoids artifacts where a target vertex is assigned weights from a point on the source mesh that is geodesically far (i.e., the bottom of the pants in Fig. 1d), other issues (armpits, crotch areas in Fig. 1c) were not addressed. Also, the quality depends on the voxel grid resolution.

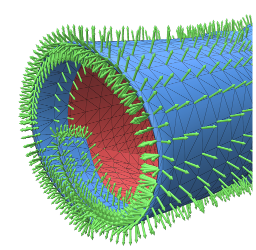
Automatic skin weight computation methods that bind a mesh to a skeleton do not directly apply to our problem, as there is a high chance that the resulting weights will cause interpenetration between the body and the clothing during the animation. Additionally, computing tetrahedral meshes [Jacobson et al. 2011] of non-watertight clothing meshes or getting a large data set for training [Xu et al. 2020] may not be feasible in practice. User-friendly interfaces for skinning [Bang and Lee 2018] require user interactions while we seek an automated solution. Hence, to the best of our knowledge, the only other automated methods for transferring weights are based on the closest point method as described in [Autodesk Inc. 2023], the implementation of which is closed-source. Therefore, we use our own implementation of the closest point method as our baseline and compare it to our new approach. We have confirmed that using [Autodesk Inc. 2023] directly produces very similar results to our implementation for our test cases.

3 METHOD

The input to our method are the skinned source mesh $\mathbf{V}_s \in \mathbf{R}^{|\mathbf{V}_s|\times 3}$, $\mathbf{F}_s \in \mathbf{R}^{|\mathbf{F}_s|\times 3}$ and the target mesh $\mathbf{V}_t \in \mathbf{R}^{|\mathbf{V}_t|\times 3}$, $\mathbf{F}_t \in \mathbf{R}^{|\mathbf{F}_t|\times 3}$ without skinning weights, where $\mathbf{V}$ and $\mathbf{F}$ are vertex position and triangle face index matrices. We assume that two meshes are aligned.

3.1 Closest Point Matching

First, for every vertex on the targer mesh, we want to check if its closest point on the source mesh is a good match. We check if the point is within a threshold distance D away and its normal does not deviate by more than Θ angle. The default values for D and Θ are chosen as $D = 0.05 * d_{box}$ and $\Theta = 35°$, where d_{box} is the target mesh bounding box diagonal length. The closest point usually lies on a triangle, so we use its barycentric coordinates to find its position and normal based on the triangle vertices. This approach is common among non-rigid surface registration techniques where the goal is to find matches between a template mesh and the target surface [Yoshiyasu et al. 2014]. Now we have two sets of vertices on the target mesh, set S_{match} contains all the vertices for which a good match was found on the source mesh and set $S_{nomatch}$ contains all the vertices without a match (Fig. 2).



Often, clothing is multilayered, where the vertices of the inner layer have their normals pointing towards the body (see the inset showing a sleeve example) and hence will fail the normal test. So after we create the initial $S_{nomatch}$ set, we do an optional pass on that set with flipped normals to check if matches exist and address potentially multilayered fabric.

For every vertex in S_{match}, we simply copy the weights from its corresponding match whose weight is computed using barycentric interpolation of the weights of the source vertices. However,

Figure 2: Vertex set S_{match}, highlighted in blue, represents vertices whose weights we copied directly from the source mesh. While for the vertex set $S_{nomatch}$, highlighted in white, we automatically compute the smooth weights. 3D models are the property of Epic Games, Inc., used with permission.

we now have vertices in $S_{nomatch}$ for which we do not copy the weights from the source mesh and instead want to compute them automatically based on the vertex weights in S_{match}. Inspired by the image inpainting and surface hole-filling algorithms, we propose an approach for *weight inpainting* that allows us to infer the smooth skinning weights for vertices in $S_{nomatch}$.

3.2 Skinning Weights Inpainting

We pose the problem of computing the missing weights as an optimization problem subject to constraints:

$$\underset{\mathbf{W}}{\operatorname{argmin}} \quad trace(\mathbf{W}^T(-\mathbf{L} + \mathbf{L}\mathbf{M}^{-1}\mathbf{L})\mathbf{W}) \tag{1}$$

$$\text{subject to} \quad \sum_{j=0}^{m} \mathbf{W}(i,j) = 1 \quad \text{for any vertex } i \tag{2}$$

$$\mathbf{W}(i,j) >= 0 \quad \text{for any vertex } i \text{, bone } j \tag{3}$$

$$\mathbf{W}(k,:) = \mathbf{s}_k \quad \text{for all } k \in S_{match} \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{|\mathbf{V}_t| \times m}$ is a sparse matrix where m is the total number of bones in the skeleton, and the $\mathbf{W}(i,j)$ entry is the influence of a vertex i by bone j. The matrix is sparse since, in most real-time applications, the number of bones influencing a vertex is approximately 4-12. $\mathbf{L}$ is the symmetric stiffness matrix (i.e., cotangent Laplacian), and $\mathbf{M}$ is the diagonal lumped mass matrix (with Voronoi area/volume $\mathbf{M}_i$ of vertex v_i on each diagonal entry i). Hence Eq. 1 minimizes a combination of Dirchlet ($-\mathbf{L}$) and Laplacian ($\mathbf{L}\mathbf{M}^{-1}\mathbf{L}$) energies as proposed by [Bang and Lee 2018]. The motivation for this combination is the fact that the Dirichlet term has good interpolation qualities but is not necessarily smooth at the boundaries (i.e., the transition between S_{match} and $S_{nomatch}$). The Laplacian term is smooth at the boundaries, but it doesn't guarantee that values won't be negative [Jacobson et al. 2012], even if the values at the boundaries are. In our experiments, this combination gives the closest approximation to the artist-painted weights.

Partition of unity constraint [Jacobson et al. 2011] in Eq. 2 ensures that if the same transformation $\mathbf{T}$ is applied to all joints, the mesh will be transformed by $\mathbf{T}$. We do not enforce this property explicitly since it's guaranteed if the weights at the boundaries sum to 1.

Non-negativity constraint in Eq. 3 ensures that vertices do not move in the opposite direction to the transformation. Because of the Laplacian energy term, we are not guaranteed to have non-negative weights even if all of the weights at the boundaries are non-negative. However, in all of our experiments, the resulting weights are only slightly negative (≥ -0.03) so in practice, we simply clamp the negative weights to 0 and re-normalize. This allows us to avoid enforcing an inequality constraint which requires an implementation of the active-set or the interior-point QP solver.

The constraints in Eq. 4 ensure that the skinning weights for any vertex $v_k \in S_{matched}$ for which we found a good match are set to the corresponding row vector of weights $\mathbf{s}_k$ copied over from the source mesh as described in Section 3.1.

3.3 Implementation

In the end, we are left with a quadratic energy with a fixed value constraints which can be reduced to an unconstrained problem (see Section 2.2.1 [Jacobson 2013]) for every column $\mathbf{w}$ in $\mathbf{W}$:

$$\underset{\mathbf{w}_U}{\operatorname{argmin}} \quad \mathbf{w}_U^T \mathbf{Q}_{UU} \mathbf{w}_U + \mathbf{w}_U^T \mathbf{Q}_{UI} \mathbf{w}_I \tag{5}$$

where $\mathbf{Q} = -\mathbf{L} + \mathbf{L}\mathbf{M}^{-1}\mathbf{L}$, the submatrix $\mathbf{Q}_{UU}$ is extracted by slicing the rows/columns by the indices of vertices in the set $S_{nomatch}$ and $\mathbf{Q}_{UI}$ by slicing the rows by indices in the set $S_{nomatch}$ and columns in S_{match}. Equivalently we slice the column vector $\mathbf{w}$ to get $\mathbf{w}_U, \mathbf{w}_I$.

Taking the derivative of Eq. 5 with respect to $\mathbf{w}_U$ and setting the result to zero gives us a linear system

$$\mathbf{Q}_{UU}\mathbf{w}_U = -\mathbf{Q}_{UI}\mathbf{w}_I \tag{6}$$

where $\mathbf{Q}_{UU}$ is positive semi-definite and can be efficiently solved using the Cholesky decomposition. We could further skip the solve if $\mathbf{w}_I$ is a zero-vector, i.e., none of the vertices in $S_{nomatch}$ are affected by the current bone. So in practice, the actual number of times we need to solve Eq. 6 is much less than the total number of bones m. As a post-processing step, the user can apply a few iterations for Laplacian smoothing for any vertex in set $S_{nomatch}$. This is especially beneficial in areas like the armpits and the crotch.

3.4 Artist-Control

Sets S_{match} and $S_{nomatch}$ in Fig. 2 are computed as binary masks in Section 3.1. However, if we allow the user to modify those masks before applying weight inpainting (Sec. 3.2) we can offer greater control over the results and fix some of the limitations. In Fig. 3a, we show an example of a user-painted mask that indicates that the collar vertices should be excluded from the S_{match} even if a good match is found on the body. Instead, they are added to $S_{nomatch}$, and their weights are automatically computed via weight inpainting.

4 RESULTS

Loose clothing/Dresses: When using the standard closest point method, a wrong or suboptimal match can be found. This happens because while the pairs of vertices are close to each other in world space, they are far apart in shape space. Fig. 1 shows T-shirt's sleeve vertices being matched to the torso (Fig. 1c), and the left leg of the flared pants has a single vertex matched to the right leg (Fig. 1d), leading to incorrect skinning. Fig. 4 in the Supplementary Materials shows how the bottom of the dress has a sharp break in the middle due to vertices being matched to the closest points on the legs (Fig. 4b,c). Fig. 5 shows a poncho with incorrect skinning around the arm and the back areas (Fig.5 c.d). Instead, our method results in smooth deformation (Fig. 4d,e and Fig. 5f,g).

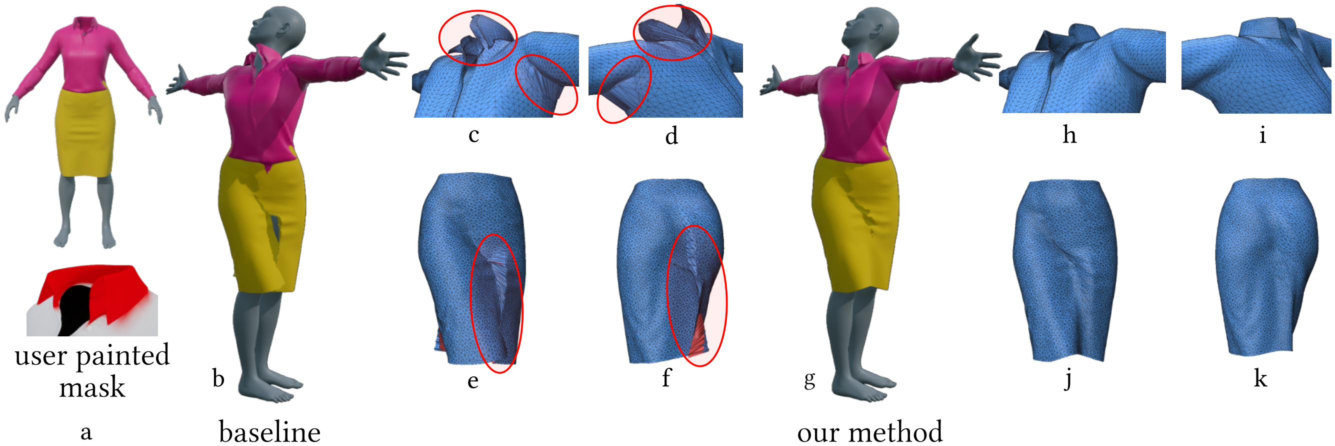


Figure 3: Skin weights are transferred from a rigged body to a dress shirt and a pencil skirt (a). An additional user-painted mask is provided to indicate vertices that are forced to have their weights computed automatically (Sec. 3.4). Current production methods fail to produce ready-to-use assets (b). Upon closer inspection, the collar and the armpits (c, d), and the front (e) and back (f) of the skirt are skinned incorrectly. In contrast, our method (g) does not suffer from the same issues (h, i, j, k). 3D models are the property of Epic Games, Inc., used with permission.

Creases: Less noticeable yet important issues arise in areas such as the armpits and crotch in more skin-tight clothing. When using the closest point method, the vertices in the armpit area are split between being influenced by the torso bones and the arm bones. So when the character raises the arm, edges connecting the two vertices that are influenced by a different set of bones end up being stretched and require manual cleanup (Fig. 3c,d). The same issue happens in the crotch area where some vertices are assigned to the left leg while others are to the right without smooth falloff (Fig. 1d).

Artist-control: Collars tend to inherit skinning weights from the upper neck and jaw due to their close proximity. When the head is then rotated, the collar follows the jaw (Fig. 3c,d). This is not desirable, as collars usually tend to be rigid. In practice, they are rigged with additional corrective bones. Our method in Sec. 3.1 also assigns weights from those regions since the vertices pass the distance and the normal checks. However, the user can paint a mask to force the weight in the collar region (Fig. 3a) to be automatically computed (Sec. 3.4), thus improving the results (Fig. 3h,i).

5 LIMITATIONS AND FUTURE WORK

Frequently, rigs include corrective bones designed to address problems with specific poses to achieve a high degree of realism. Their influences are custom painted, and they are associated with a small specific area of the model, allowing for precise refinements (e.g., upper arm deformation). If a large part of the area influenced by the corrective bone is part of the $S_{nomatch}$ set, then our optimization might fail to accurately reconstruct the corrective bone weights.

Similarly to the closest point method, our method does not guarantee that no interpenetration between the body and the skin-tight garments will occur for extreme poses. Additionally, small interpenetrations in extreme poses are possible in the areas with a large number of vertices in $S_{nomatch}$.

Our method assumes that all clothing parts are connected. However, it's often the case that clothing is modeled as a set of disconnected patterns (i.e., the T-shirt sleeve is disconnected from the torso) or has small parts like buttons. We leave dealing with more complex topologies to future work.

6 CONCLUSION

We introduce a new production-ready method for the automatic transfer of the skin weights from one mesh to another that is robust and simple to implement in practice. As we have shown in the figures and the supplementary video, it far outperforms current existing methods, especially in more challenging cases involving transferring weights from rigged bodies to non-skin-tight clothing at a small increase in computation cost.

REFERENCES

Autodesk Inc. 2023. Copy smooth skin weights, Maya 2024. https://help.autodesk.com/view/MAYAUL/2024/ENU/?guid=GUID-7D895BB0-1522-4388-96E0-4245127F90AB.

Seungbae Bang and Sung-Hee Lee. 2018. Spline interface for intuitive skinning weight editing. *ACM Transactions on Graphics (TOG)* 37, 5 (2018), 1–14.

Olivier Dionne and Martin de Lasa. 2013. Geodesic voxel binding for production character meshes. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 173–180.

Emeline Got. 2019. *Developement of a new automatic skinning system*. Master's thesis. Universitat Politècnica de Catalunya.

Alec Jacobson. 2013. *Algorithms and interfaces for real-time deformation of 2d and 3d shapes*. Ph. D. Dissertation. ETH Zurich.

Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine. 2011. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.* 30, 4 (2011), 78.

Alec Jacobson, Zhigang Deng, Ladislav Kavan, and John P Lewis. 2014. Skinning: Real-time shape deformation. In *ACM SIGGRAPH 2014 Courses*. 1–1.

Alec Jacobson, Tino Weinkauf, and Olga Sorkine. 2012. Smooth shape-aware functions with controlled extrema. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1577–1586.

Tuur Stuyck. 2022. *Cloth simulation for computer graphics*. Springer Nature.

Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. 2020. Rignet: Neural rigging for articulated characters. *arXiv preprint arXiv:2005.00559* (2020).

Seung-Hyun Yoon, Taejoon Kim, Ho-Won Kim, and Jieun Lee. 2016. Transferring skin weights to 3D scanned clothes. *ETRI Journal* 38, 6 (2016), 1095–1103.

Yusuke Yoshiyasu, Wan-Chun Ma, Eiichi Yoshida, and Fumio Kanehiro. 2014. As-conformal-as-possible surface registration. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 257–267.

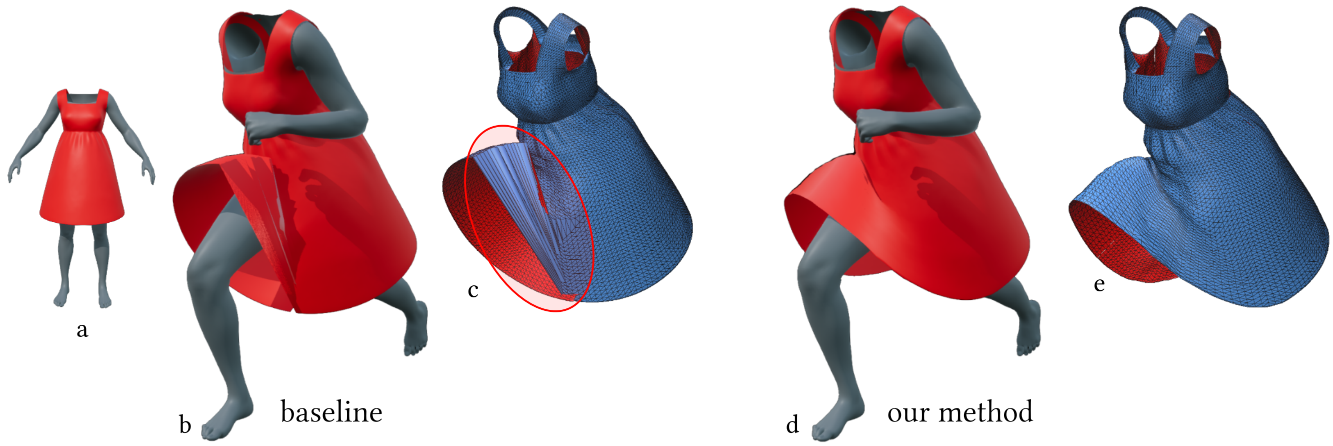


Figure 4: Skin weights are transferred from a rigged body to a babydoll type dress (a). Current production methods fail to produce ready-to-use assets (b). Upon closer inspection, the bottom of the dress (c) is skinned incorrectly. In contrast, our method (d) does not suffer from the same issues (e). 3D models are the property of Epic Games, Inc., used with permission.

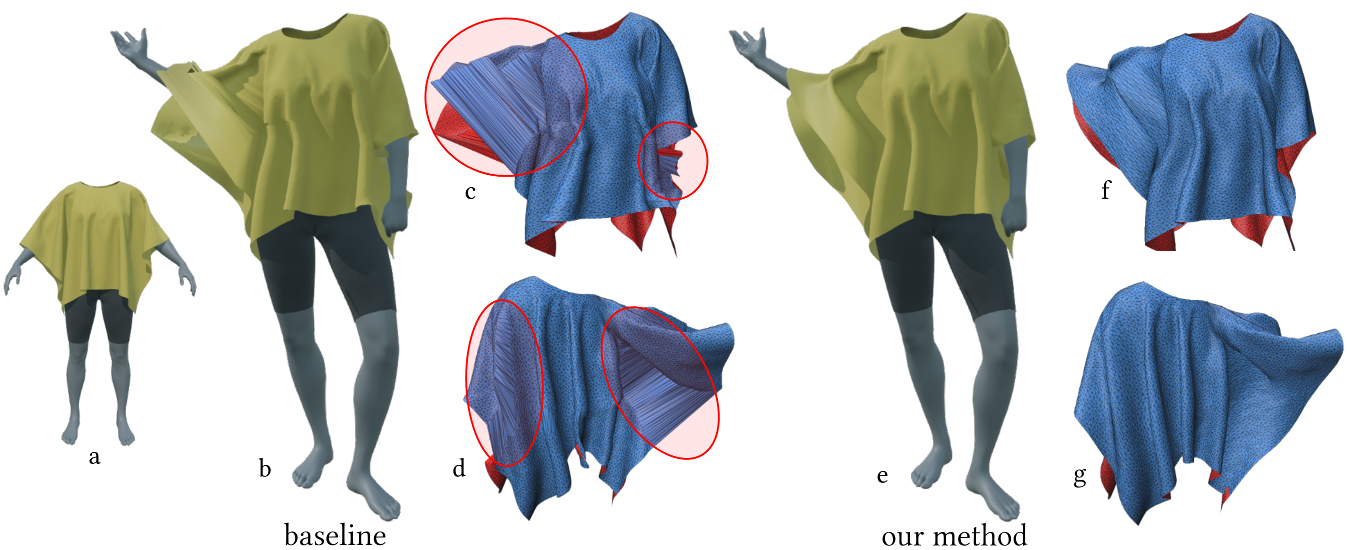


Figure 5: Skin weights are transferred from a rigged body to a poncho (a). Current production methods fail to produce ready-to-use assets (b). Upon closer inspection, the areas near the arms and the back (c,d) are skinned incorrectly. In contrast, our method (e) does not suffer from the same issues (f,g). 3D models are the property of Epic Games, Inc., used with permission.