Vis Comput (2013) 29:525–534
DOI 10.1007/s00371-013-0816-2

ORIGINAL ARTICLE

Image warping for retargeting garments among arbitrary poses

Hadi Fadaifard · George Wolberg

Published online: 30 April 2013

Abstract We address the problem of warping 2D images of garments onto target mannequins of arbitrary poses. The motivation for this work is to enable an online shopper to drag and drop selected articles of clothing onto a single mannequin to configure and visualize outfits. Such a capability requires each garment to be available in a pose that is consistent with the target mannequin. A 2D deformation system is proposed, which enables a designer to quickly deform images of clothing onto a target shape with both fine and coarse controls over the deformation. This system has retargeted thousands of images for retailers to establish virtual dressing rooms for their online customers.

Keywords Shape deformation · Image warping · Moving least squares · As-rigid-as-possible shape manipulation

1 Introduction

The online experience in shopping for garments has remained largely unchanged over the years. Consumers are typically presented with a myriad of clothing images presented on mannequins or flat surfaces. The ability to drag and drop clothing onto a single mannequin to configure and visualize outfits is considered to be highly useful. A few companies, such as *Looklet* and *Schway*, already help users design custom looks by allowing them to virtually dress model mannequins with clothing items available on their websites. To facilitate this capability, all clothing items must either be placed on the mannequins when obtaining their pictures, or be deformed from different poses to fit the target mannequins. The former approach is generally more expensive and involves more manual labor. Moreover, the latter approach allows showcasing any clothing item, which may have been previously photographed for other purposes. Therefore, having a system that allows designers to manipulate images of clothing items to fit a target shape is desirable. Retargeting of images of garments among arbitrary poses, however, is a challenging problem, since the images come from disparate sources, including mannequins and human models in many poses, sizes, and shapes.

This paper describes a deformation system for warping images of garments from any pose onto a target mannequin upon which a fashion ensemble can be created by the user. We treat this problem as an exercise in 2D shape deformation that is governed by user-specified control points. Various image deformation techniques exist in the graphics literature. However, to the best of our knowledge, none provide a general framework capable of handling the challenges that arise when designing a system for this problem. The main difficulties faced by the system are allowing the user to perform the deformations with as few operations as possible, while, at the same time, providing fine control over the shapes of the deformed images. Additionally, providing the user with the ability to assign depth values to different parts of the items that are being deformed is another crucial feature needed in such a system.

Figure 1 shows an overview of the deformation pipeline of our system. First, the input shape of the garment is triangulated (Fig. 1(a)). Next, the user performs coarse adjustments to the input shape (Fig. 1(b)). Finally, the user makes

H. Fadaifard
Brainstorm Technology LLC, New York, USA
e-mail: fadaifard@brainstormLLC.com

G. Wolberg (✉)
City College of New York, City University of New York, New York, USA
e-mail: wolberg@cs.ccny.cuny.edu

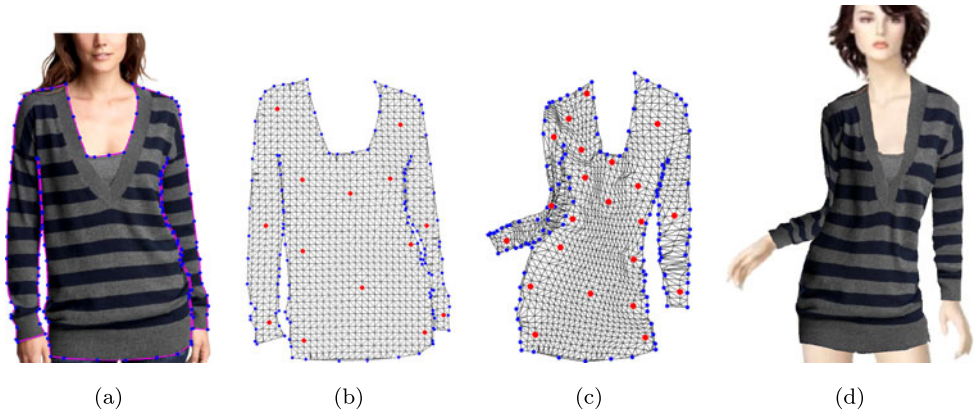


Fig. 1 Deformation pipeline: (**a**) the user traces the contour of the garment to define a closed region; (**b**) the triangulated mesh of the region with user-specified interior (*red*) and exterior (*blue*) control points; (**c**) the user deforms the mesh to align it with a target mannequin by inserting, deleting, and moving the control points; (**d**) the final result of the garment retargeted onto the desired mannequin

finer adjustments to the shape of the garment (Fig. 1(c)) to obtain the desired result (Fig. 1(d)).

In this work, we improve and employ known deformation techniques in a *complete* system for the garment retargeting problem. We extend the image-space MLS deformation approach of Schaefer et al. [10] to object-space. This simple extension is used as an initial guess to the as-rigid-as-possible (ARAP) deformation scheme of Sorkine and Alexa [11]. As shown in the next section, this improves the convergence rate of ARAP and reduces the possibility of convergence to incorrect minima. An additional feature of our 2D deformation system is the capability of assigning arbitrary depth values to the control points, which can then be used to determine the depth values at all mesh vertices. This feature is required when rendering deformed objects with overlapping parts. Finally, we show how interior and exterior control points enable the system to furnish intuitive local-global control over the shape of the deformed mesh: exterior points accommodate the fine local detail along the boundary, while interior points perform coarser global shape adjustments.

The proposed deformation system was designed to be easy to use and operate at interactive rates. The behavior of the system adheres to the following criteria: *Ease of use:* an iterative procedure is used to require minimal user input in quickly creating natural deformations; *Fine control:* exterior control points provide the user with fine control over the deformations; *Global control:* interior control points provide the user with global control over the deformations.

2 Related work

The first ARAP formulation introduced by Alexa et al. [1] was for morphing between compatibly triangulated 2D shapes. Igarashi et al. [3] used the ARAP criterion as the main constraint in designing a deformation system that enabled manipulating 2D objects in real time. However, they reported slow performances for meshes with more than 300 vertices on a 1-GHz system. In [10], Schaefer et al. proposed an MLS-based image-space deformation system capable of deforming images with thousands of pixels in real time. They derived closed-form solutions for affine, similarity and rigid control points and lines, which could be used to guide the deformation. In [13], the authors propose a 2D deformation system using a non-linear least squares optimization scheme, where the objective is to preserve two geometric properties of the shape that is being deformed: the Laplacian coordinates of the boundary curve of the shape and local areas inside the shape.

In [11], an ARAP system for deforming 3D triangular surfaces is proposed. Starting with an initial guess for the deformation, their approach iteratively improves the deformation by minimizing the non-rigidity of the deformation in vertex cells over the whole mesh. Their approach manipulates a 3D shape through its surface, rather than the *volume* enclosed by the surface. Therefore, the approach falls into the same category as Igarashi's [3], while being mathematically simpler. The individual deformation cells used by the two approaches, however, are different. In [3], each cell consists of a single triangle, while in [11], the cells are defined on a per-vertex basis and the extent of each cell corresponds to the 1-ring (star) neighborhood of the vertex. In our experiments, we noticed that as the amount of overlap between neighboring cells increases, the resulting deformation becomes smoother. Moreover, in [3], only one ARAP iteration is performed as opposed to multiple iterations in [11].

In [5], the authors propose a more general deformation scheme for deforming images using a similar local-global scheme to the one used in [11]. Their method is more general as it allows incorporating information about the image content (energy) in the set of allowable transformations the deformation may employ. They show applications of their system in content-aware image resizing and 2D image deformations.

In [2], the authors propose an extension of the MLS-based image deformation of [10] to 3D shapes. However, the approach uses the Euclidean distance as the default metric and also does not attempt to minimize the non-rigidity

of the deformation within the cells of the deformed surface (i.e., in the sense of [11]).

One of the main shortcomings of the MLS-based deformation approach of [10] is the lack of support for non-convex objects. We show how their MLS approach can be extended to handle non-convex objects. In more recent work [5, 12], the shortcomings of the MLS approach compared to other approaches have been solely attributed to its use of Euclidean distance as the driving metric in the deformation. However, since the approach does not explicitly attempt to minimize the local non-rigidity of the deformation around each point of interest, it is not truly ARAP as defined in [1]. Empirically, we show that besides the issues arising from the employed metric, the MLS approach enforces a weaker rigidity constraint and that its deformation results can be improved.

More recently, in [4], a deformation system is proposed where linear blending weights are used to produce smooth deformations for combinations of points, bones, and cages. Maximum Entropy Coordinates and Mean Value Coordinates are examples of other techniques that may be used for deforming the garments. However, these techniques do not guarantee that no local or global scaling will be introduced during the deformation process.

In [7], a web-based 3D system is presented where users can try on physically simulated garments on virtual representations of themselves. It is important to note that our approach operates entirely in 2D and makes no attempt at cloth simulation due to the constraints of our image-based input.

3 Deformation framework

The deformation of the input shape is guided by the movement of a set of control points. The objective is to update the positions of all the points in the shape in a manner that is consistent with the movement of these control points, and is also "natural." Since the shapes being manipulated generally correspond to physical objects, it has been argued that the most natural deformations are those that minimize the non-rigidity energy of the deformed object [1]. Even though garments cannot be regarded as rigid objects, in our experiments we noticed that the rigidity constraint leads to more desirable deformations compared to other types of deformations such as as-similar-as-possible (ASAP), Complex Barycentric Coordinates [12], and Conformal Maps. Therefore, in this work we follow the same as-rigid-as-possible (ARAP) approach as in the works of refs. [1, 3, 5, 10, 11].

This section is organized as follows: in Sect. 3.1, we describe the coarse deformation process in our system. In Sect. 3.2, we describe how complex overlapping scenarios can be handled by our vertex depth assignment procedure. Finally, in Sect. 3.3, we describe the fine deformation stage of our system.

3.1 Coarse deformation

In coarse deformation, the user makes rough adjustments to the shape of the input using a few interior control points. The initial deformation in this mode is first obtained using object-space MLS (Sect. 3.1.1), and then is iteratively improved using ARAP (Sect. 3.1.2). Our system also allows the user to perform rotations about arbitrary control points (Sect. 3.1.3).

The input mesh is represented by $M = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_n\}_{n=1}^{N}$ is the set of mesh vertices and $\mathcal{E} = \{e_{ij} \in \mathcal{V} \times \mathcal{V} | v_i \text{ is connected to } v_j\}$ is the set of edges that connect the vertices. It is assumed that the topology of the mesh remains the same throughout the deformation. The ordered sets $P = \{p_n\}_{n=1}^{N}$ and $Q = \{q_n\}_{n=1}^{N}$ contain the 2D positions of the vertices in the original and deformed meshes, respectively. Similarly, ordered sets $A = \{a_m\}_{m=1}^{M}$ and $B = \{b_m\}_{m=1}^{M}$ respectively contain the source and target positions of the interior control points.

The objective is to determine a warp function $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ that yields the positions of the vertices in the deformed mesh based on the movements of the control points from a_m to b_m. In the ideal case, ψ must satisfy the following properties [10]: smoothness, interpolation, identity, and deformation constraints. Simultaneously satisfying all of these properties may not be possible. For example, the deformation constraints generally lead to warp functions, which are not smooth.

In real-time deformation systems, as the user moves the control points, a sequence of warp functions are constructed. The deformation is guided with a sequence of point-sets $\mathcal{A} = \mathcal{A}^1, \ldots, \mathcal{A}^K$ describing the source and target positions of the control points. Given two such sequences $\mathcal{A}$ and $\mathcal{B}$, in addition to the above properties for the warp function, we require the following property to hold:

– *trajectory invariance:* If $\mathcal{A}^1 = \mathcal{B}^1$ and $\mathcal{A}^K = \mathcal{B}^K$, then the deformed meshes corresponding to the two sequences should be the same.

This guarantees the system does not produce undesirable results similar to the one shown in Fig. 2(b).

The warp function ψ is obtained by minimizing an energy functional over the mesh vertices $\mathcal{V}$. We now discuss the initial guess and the deformation constraints, which are the main considerations that impact the solution for ψ.

Initial guess One aspect of the ARAP framework, which is generally ignored in the previously mentioned works, is its need for an initial guess for the deformation. Since the non-rigidity energy functional that is minimized in the process may have multiple local or global extrema, the initial guess for the deformation plays an important role in both the convergence rate of the approach and in preventing undesirable

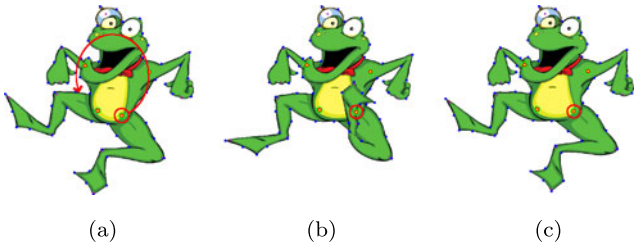


Fig. 2 Comparison of ARAP deformation using different initial guesses: (**a**) original shape, (**b**) previous frame as initial guess (PFAIG), (**c**) object-space MLS

deformations. Previous works generally advocate the use of the "previous frame as the initial guess" (PFAIG) [5, 11]. Alternate solutions for initial guesses have also been used. For example, an initial guess based on the Least Squares Conformal Map (LSCM) is discussed in [5]. However, the authors recommend using PFAIG to achieve convergence to a minimum with fewer iterations.

The main problem with using PFAIG is that the deformation error is accumulated across frames. In many instances, and especially under extreme deformations, this accumulated error can lead to two types of problems: slower convergence rates and/or convergence to a wrong minimum. For example, in Fig. 2, we compare the deformation results between PFAIG and our proposed approach discussed in this paper. In both cases all control points were returned to their original configuration after a few user manipulations. However, in Fig. 2(b), the model did not return to its initial shape. This is in violations of the identity and trajectory invariance criteria for deformations, which were mentioned previously. The initial guess in Fig. 2(c) was obtained using an extension of the image-space MLS-based deformation approach of [10]. The computational overhead of this step is negligible (less than 4 milliseconds on meshes with approximately 3 K vertices on a 2 GHz CPU). In Sect. 3.1.1, we describe the approach in detail. Note that in [3], instead of using PFAIG, the authors always use the original shape as the initial guess. Although doing this may not result in the undesirable effects shown in Fig. 2(b), the approach will always have slower convergence rates than both PFAIG and our approach.

Deformation constraints In general, an ARAP framework results in natural-looking deformations. However, under extreme deformations, the rigidity constraint leads to foldovers and other undesirable effects. For example, in Fig. 3 we compare the performance of an ARAP deformation system with a relaxed ARAP system. As can be seen in Fig. 3(a, b), foldovers develop in the neck of the giraffe when all mesh cells are required to remain as rigid as possible. On the other hand, by relaxing the rigidity constraints, more desirable results, which do not contain any foldovers, can be obtained.

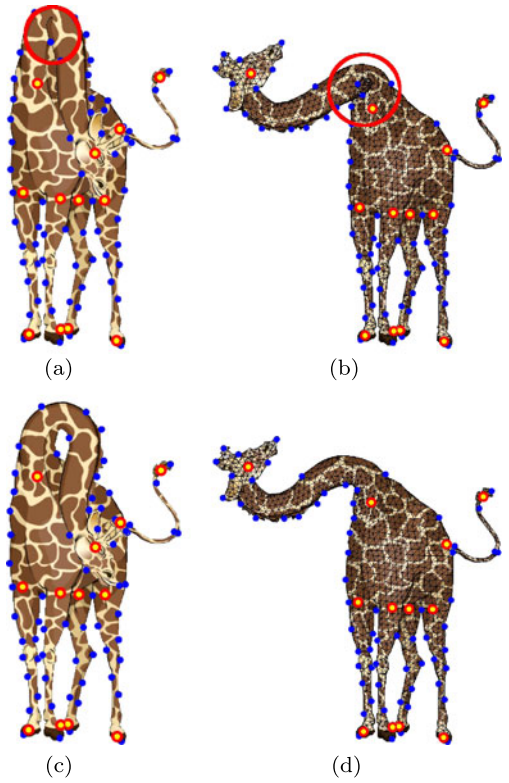


Fig. 3 Preventing foldovers in deformations: deformation results using (**a**, **b**) ARAP, and (**c**, **d**) relaxed ARAP. The original model is given in Fig. 5

This is shown in Fig. 3(c, d). Relaxed ARAP is briefly discussed in Sect. 3.1.2.

3.1.1 Object-space MLS for 2D shape deformation

We extend the image-space Moving Least Squares (MLS) deformation approach of [10] to object-space and use it to obtain the initial guess needed for the ARAP deformation. The approach has very limited overhead (in the order of a few milliseconds for meshes with a few thousand vertices) and is easy to implement.

Given the movements of the control points from their source positions P to target positions Q, the objective is to obtain the positions of the mesh vertices such that the shape undergoes the least amount of non-rigid transformations. Globally, satisfying the rigidity constraint for all arbitrary configurations of source and target control points is not feasible. Instead, the constraint is enforced on small local neighborhoods near each point in the domain of ψ. A warp function is sought at each mesh vertex that depends only on the movement of the control points by minimizing an error functional in the least-squares sense. In particular, for each vertex v_n a separate local warp function $\psi_n : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is

found that minimizes

$$E_n = \sum_{m=1}^{M} \theta\big(d(a_m, v_n)\big)\big\|b_m - \psi_n(a_m)\big\|^2, \tag{1}$$

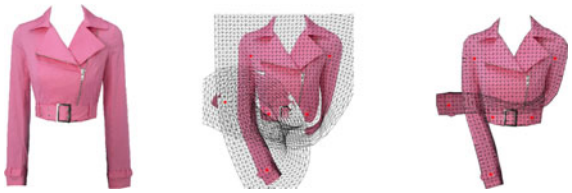
where $d : \mathbb{R}^2 \times \mathcal{V} \to \mathbb{R}$ is a function measuring the geodesic distance between a point in $\mathbb{R}^2$ and a mesh vertex, and $\theta : \mathbb{R} \to \mathbb{R}$ is a non-negative monotonically decreasing weight function. This approach for deforming 2D shapes using Moving Least Squares is based on the work in [10]. One important difference is that the work in [10] was limited to convex shapes, whereas our use of geodesic distance extends our technique to arbitrary shapes.

The geodesic distance between any two vertices on the mesh is defined as the length of the shortest path that runs along the edges between the two vertices. Let $g : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ be a function that returns the geodesic distance between two mesh vertices. Then, the geodesic distance between a mesh vertex v_n and an arbitrary point $p \in \mathbb{R}^2$ is computed as $d(p, v_n) = \|p' - p\| + g(v', v_n)$, where v' denotes the mesh vertex nearest to p, and p' refers to that vertex position. In our case, p refers to the *source* position of a control point $a_m \in A$. As a result, the geodesic distance only needs to be computed from each control point $a_m \in A$ to all mesh vertices. Note that this computation can be performed offline, since it is specified in terms of the anchored source positions, and not the varying target locations. We use Dijkstra's shortest path algorithm to efficiently compute these distances.

In order for ψ to be local and interpolating at the control points, the following respective conditions must hold: $\theta(r)$ must be rapidly decreasing as $r \to \infty$ and $\lim_{r\to 0}\theta(r) = \infty$ [6]. These two conditions reduce the set of possible choices for θ to asymptotic functions whose asymptotes coincide with the x and y axes. The hyperbolic cosecant function $\theta(r) = \operatorname{csch}(r/\alpha)$ and the rational function $\theta(r) = r^{-\alpha}$, for some $\alpha > 0$, are two examples of good candidates for the weight function. In our experiments, we noticed that weight functions with a faster rate of decay result in more desirable (i.e., rigid) deformations. For any $\alpha_1, \alpha_2 > 0$, $\operatorname{csch}(r/\alpha_1)$ decreases more rapidly than $r^{-\alpha_2}$ as $r \to \infty$. This, theoretically, makes $\operatorname{csch}(r/\alpha_1)$ a more suitable choice as the weight function for the deformation. However, the fast rate of decay of csch introduces numerical instabilities. Therefore, $\theta(r) = r^{-\alpha}$ is used as the weight function in all calculations that follow. The same choice of weights was also used in the original MLS formation of [8] and MLS-based deformation scheme of [10].

Since we are looking for a rigid deformation, ψ_n is restricted to belong to the class of rigid transformations. Therefore, the warp function at each vertex v_n is given as

$$\psi_n(x) = R_n x + T_n. \tag{2}$$



(a) original image (b) image-space MLS (c) object-space MLS

Fig. 4 Comparison of image-space vs. object-space MLS deformations. The object-space deformation results are preferred since the movement of control points on one sleeve does not induce extreme foldover

Consequently, the error functional of Eq. (1) becomes

$$E_n = \sum_{m=1}^{M} \theta_m^n \|b_m - R_n a_m - T_n\|^2, \tag{3}$$

where $\theta_m^n = d(a_m, v_n)^{-\alpha}$ for some $\alpha > 0$, R_n is a rotation matrix, and T_n is a vector representing translation. It is important to note that the MLS formulation in Eq. (3) attempts to derive a rigid transformation based exclusively on the source and target positions of the control points. This is considered to impose a weaker rigidity constraint because it does not explicitly incorporate the local neighborhood about each mesh vertex v_n.

Figure 4 compares the results for image-space and object-space MLS deformations. Notice that our object-space MLS deformation results are superior to the image-space results because the movement of control points in one part of the shape does not affect unrelated mesh vertices that may lie nearby. For instance, the movement of control points on the sleeve only affects the sleeve region and does not affect the position of mesh vertices on the nearby torso.

3.1.2 As-rigid-as-possible deformation

As noted earlier, the object-space MLS deformation procedure described in Sect. 3.1.1 is not really "as rigid as possible" in the sense of [1, 11]. This limitation is due to the fact that a different rigid transformation is obtained for each vertex in the input mesh and no attempt is made to ensure the transformations are locally the same. However, the approach results in a good approximation and can be used as an initial guess for an iterative ARAP procedure that refines the deformation. We adapt the ARAP 3D deformation method of [11] to iteratively improve the 2D results of Sect. 3.1.1.

The ARAP procedure iteratively minimizes the total non-rigidity energy of our initial MLS deformation. This energy is defined as the sum of the local non-rigidity energies in the vertex cells covering the whole mesh. Each vertex cell C_n is defined using vertex n and its 1-ring neighbors, given by set $\mathcal{N}(n) = \{k | e_{nk} \in \mathcal{E}\}$.

Each iteration of the minimization in the ARAP procedure consists of two steps. In the first step, for each vertex

Fig. 5 Comparison of MLS and ARAP MLS deformation results. Notice that MLS version suffers from excessive distortion along the neck near the head

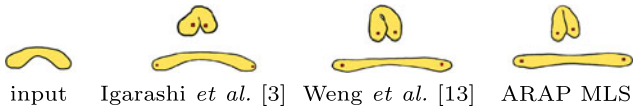


Fig. 6 Comparison of deformation results between our ARAP MLS approach and other techniques. Our approach clearly outperforms that of [3]. However, in this simple example, it is not easy to visually evaluate the performance of our approach against Weng's. Due to lack of proper initialization in Weng's technique, their results can be expected to degrade for more complicated examples, as in Fig. 2

cell C_n in the original mesh, we solve for a rotation matrix that best maps, in a least-squares sense, the positions of the vertices in C_n onto their corresponding positions in the deformed mesh. The sum of squared differences in these positions defines the local non-rigidity in the cell. In the second step, the recovered rotation matrices are used to construct and solve a sparse linear system to recover vertex positions that minimize the total non-rigidity energy of the deformation over the whole mesh. The two steps can be iterated to improve the deformation results. It has been shown that this iterative algorithm is guaranteed to converge to a local minimum [5].

Figure 5 compares MLS and as-rigid-as-possible MLS (ARAP MLS) deformation results. The giraffe example in Fig. 5 is chosen to show the superior performance of ARAP MLS over the object-space MLS under extreme deformations. In Fig. 6 we compare the coarse deformation results of our approach to those of [3] and [13]. Our ARAP procedure with MLS initialization converged to a solution after approximately 20 iterations. The total deformation time for meshes with 2 K vertices was approximately 45 milliseconds on a 2.4 GHz CPU. The offline precomputation time to update the MLS and ARAP parameters after user changes was approximately one second.

One can also relax the rigidity constraint in local regions of the deformed object to reduce the effects of foldovers (Fig. 3). This is done by first detecting triangles in the deformed object where foldovers occur. The ARAP procedure is then re-applied to the 2D shape, but this time allowing the foldover triangles and a small neighborhood near them to undergo similarity transformations with a pre-specified maximum amount of allowed scaling. This option serves to improve the deformation results, as shown in Fig. 3.

3.1.3 Rotation about control points

In some instances, the user may wish to deform the input mesh by rotating sections of the object around a control point. To allow this, we associate a rotation angle with each control point. Rotation about the control points is performed in two steps. First, an initial guess for the deformation is obtained. Second, the rigidity of the deformation is iteratively improved using the same approach as in Sect. 3.1.2.

The initial deformation is obtained by rotating each mesh vertex v_n by the rotation angle associated with its nearest interior control point c_m. Let p_n denote the position of v_n, and R_m be the rotation matrix associated with c_m. The new position q_n of v_n is then obtained as $q_n = R_m(p_n - c_m) + c_m$.

The initial deformation, however, results in undesirable artifacts in the deformed mesh. The result can be improved by applying the ARAP procedure as described in the Sect. 3.1.2. In this case, since the initial deformation is poor, more ARAP iterations are required to improve the results. This considerably increases the computation time. However, the initial deformation generally only modifies the positions of a few vertices on the mesh. Therefore, the ARAP deformation procedure only needs to be applied to the vertex cells where the residual of the initial deformation is large. The modified ARAP procedure for rotations about control points is given as follows:

1. Using the initial deformation results, for each vertex cell C_n, obtain rotation matrix R_n as described in Sect. 3.1.2.
2. For each cell C_n, compute the residual

$$r_n = \sum_{k \in \mathcal{N}(n)} w_{nk} \left\| (q_k - q_n) - R_n(p_k - p_n) \right\|^2. \tag{4}$$

3. Construct and solve the sparse linear system

$$\mathbf{Lq} = \mathbf{h}, \tag{5}$$

where $\mathbf{L}$ is the discrete Laplace operator, $\mathbf{q}$ is a vector containing the unknown positions q_n, and vector $\mathbf{h}$ contains the corresponding equalities for each q_n on the right-hand side of Eq. (4). In order to speed up the computation, add all vertices with residual $r_n < \epsilon$, for some $\epsilon \geq 0$, as constraints to the system.
4. Repeat the above steps K times.

3.2 Handling overlapping elements

An important feature of the deformation system is the ability to handle shapes with overlapping elements. We resolve these overlaps by considering each mesh vertex to have a depth value that is initialized via user-specified control points. This is an exercise in scattered data interpolation,

whereby a smooth depth function is fitted through the control point values.

Let $A = \{a_m\}_{m=1}^{M}$ denote the sparse set of control point positions and let $Z = \{z_m\}_{m=1}^{M}$ denote their associated depth values. Our objective is to find a smooth function $\xi : P \to \mathbb{R}$ that maps the mesh vertices in P to depth values, with the following properties:

- *Smoothness:* the function must be smooth everywhere, i.e., be at least C^1-continuous.
- *Interpolation:* the depth values at the control points must be the same as the user-specified values; i.e., $\xi(a_m) = z_m$, for $m = 1, \ldots, M$.
- *Identity:* if all control points have the same depth value, then the mesh should remain flat; i.e., if $z_m = K$, $\forall m \in |Z|$, then $\xi(p_n) = c, \forall p_n \in P$, where c is a constant value.

Note that the desired function has exactly the same properties as the warp function we sought in Sect. 3.1.1. Therefore, we apply the MLS technique to solve for the depth function. At each mesh vertex v_n we seek to find $\xi_n : \mathbb{R}^2 \to \mathbb{R}$, minimizing

$$E_n = \sum_{m=1}^{M} \theta\big(d(a_m, v_n)\big(z_m - \xi_n(a_m)\big)^2\big), \tag{6}$$

where $\theta(r) = r^{-\alpha}$ for some $\alpha > 0$ and $r \in \mathbb{R}$, and $d(a_m, v_n)$ returns the geodesic distance between mesh vertex v_n and control point a_m, as defined in Sect. 3.1.1. We know the function ξ_n cannot depend on the initial depth values of the control points, since they are all initially placed on the flat mesh and, as a result, have the same values everywhere. Therefore, the local function we are looking for is of the form $\xi_n(x) = c_n$; i.e., a constant function. The error functional becomes

$$E_n = \sum_{m=1}^{M} \theta_m^n (z_m - c_n)^2. \tag{7}$$

where $\theta_m^n = d(a_m, v_n)^{-\alpha}$. The minimizing c_n for the energy functional is found by taking the partial derivative of E_n with respect to c_n and setting the result to zero: $\frac{\partial E_n}{\partial c_n} = -2\sum_m \theta_m^n (z_m - c_n) = 0$; we obtain $c_n = \frac{\sum_m \theta_m^n z_m}{\sum_m \theta_m^n}$. Therefore, the depth value of mesh vertex v_n can be efficiently computed as $\xi_n(p) = c_n$ for all $p \in \mathbb{R}^2$. Recomputation is necessary only when the user modifies the height value of a control point. This method enables the user to represent shapes with complex overlapping elements, as shown in Fig. 7.

3.3 Boundary deformation

As shown earlier, the ARAP MLS approach can be used to deform 2D shapes in a natural manner with few control points. However, the user may still want to further refine the deformation, particularly along the boundaries. This is achieved by enclosing the boundary with a control polyline. The polyline vertices define the exterior control points. By allowing the user to move these points, fine control is given over the shape of the object's boundary. In particular, each line segment in this polyline is an MLS control line. A closed-form solution for deforming the object using these lines is given in [10]. Alternatively, these control lines may be densely sampled to yield MLS control points that will govern the deformation.

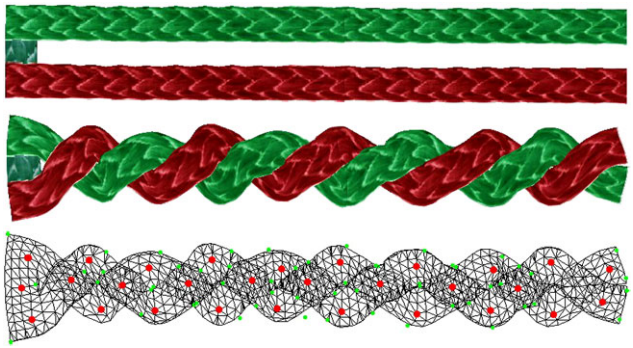


Fig. 7 Example of complex overlapping elements. *Top row*: original ropes; *middle* and *bottom rows*: interlaced ropes and their underlying mesh structure with control points shown in red

In order for the interior and exterior control points to interact with each other in a consistent and natural manner, the system has two deformation modes: coarse and fine. In the coarse (fine) deformation mode, the user can deform the interior (boundary) by moving the interior (exterior) control points. When one set of points are moved, the other set of points are subject to the deforming forces and follow the shape in tandem.

Since the exterior control points are not part of the triangulated mesh but move with the mesh as it deforms, their coordinates are given relative to the mesh vertices. Let $p \in \mathbb{R}^2$ denote the position of an exterior control point and let $p_m, p_n \in P$ denote the positions of the two closest vertices on the undeformed mesh to p, such that $\|p_m - p_n\| > \epsilon$ for some small $\epsilon > 0$. Then, the relative coordinates of p with respect to the mesh are given in terms of the following inner products:

$$p_u = \langle (p - p_n), \hat{u} \rangle, \tag{8}$$

$$p_v = \langle (p - p_n), \hat{v} \rangle, \tag{9}$$

where $\hat{u} = \frac{p_m - p_n}{\|p_m - p_n\|}$ and $\hat{v} = \hat{u}^{\perp}$ form an orthonormal basis at p_n. The global coordinates of p can be obtained from $\hat{u}$ and $\hat{v}$ by

$$p = p_u \hat{u} + p_v \hat{v} + p_n. \tag{10}$$

Fig. 8 Deforming a dress to match a target shape. Deformation using (**a**) interior control points only, and (**b**) interior and exterior control points. Notice that (b) requires exterior control points to perform finer adjustments to the shape boundary. The *red* and *blue dots* denote the interior and exterior control points, respectively; (**c**) and (**d**) show the original shape and the final deformation results

Let p'_m and p'_n denote the new positions of p_m and p_n on the deformed mesh, respectively. The new position of the control point on the deformed mesh is given as

$$p' = p_u \hat{u}' + p_v \hat{v}' + p'_n, \tag{11}$$

where $\hat{u}' = \frac{p'_m - p'_n}{\|p'_m - p'_n\|}$, $\hat{v}' = \hat{u}'^{\perp}$.

When the deformation mode is changed from coarse to fine, the original mesh is set to be the deformed mesh and the interior control points are subjected to the mesh deformation. The exterior control points define the endpoints of control lines that can be used to adjust the shape of the boundary. Note that as a result of the coarse deformation, the control lines which initially enclosed the original shape may cross into the shape. This, however, does not seem to degrade the deformation results in the fine adjustment phase.

Figure 8 shows an example where a dress is deformed to fit a target model. Note that the deformation results of ARAP MLS (Fig. 8(a)), though close to the desired shape, still require finer adjustments. This is especially evident in the regions near the wrists and shoulders of the model. The shortcomings of the ARAP MLS in those regions are *not* a result of the failure of the technique in enforcing the rigidity constraint; the shape of the underlying mesh near the armpits prevents the arms from correctly rotating around the shoulder joints without distorting the geometry and texture in the surrounding regions. Figures 8(b) and (d) show the improved deformation results with the additional help of exterior control points.

3.3.1 Automatic exterior control point extraction

Manual specification of the exterior control points is undesirable as it may be too tedious for complex shapes. Instead, we would like the system to be able to pick the most appropriate locations for these control points while keeping their numbers at a minimum. In our system, the control points are picked at the locations on the contour curve where the unsigned curvature is locally maximum.

In the continuous setting, the unsigned curvature at a point on an arc-length parameterized regular curve, $S(t)$, $S : [0, 1) \to \mathbb{R}^2$, is given as

$$\kappa(t) = \left\| \frac{\partial \hat{T}}{\partial t}(t) \right\|, \tag{12}$$

where $\hat{T}(t) = \frac{\partial S}{\partial t}(t)$ is the unit tangent vector at t. The goal is to reliably and quickly estimate the curvature values in the discrete setting, whereby a regularly sampled 2D curve is represented with the ordered point-set $R = \{r_i\}_{i=1}^{I}$, where $r_i \in \mathbb{R}^2$ and I is the number of points on the curve. Our approach is based on the curvature scale-space theory [9], where the derivatives are computed by convolving the sample data with Gaussian derivatives. We obtain the tangent vectors of the sample points by performing circular convolution of the data with the first-order Gaussian derivative

$$T = R * G^1_\sigma, \tag{13}$$

where G^1_σ denotes the discrete first-order Gaussian derivative kernel with standard deviation σ and $*$ is the discrete circular convolution operator. Note that the vectors $t_i \in T$ obtained this way do not necessarily have unit length and their orientations may be incorrect. Therefore, we obtain new normalized tangent vectors $\hat{t}_i$ with correct orientations by $\hat{t}_i = s \frac{t_i}{\|t_i\|}$, where $s = \langle t_i, r_{(i+1)\%I} - r_i \rangle$. The vector-set $\hat{T} = \{\hat{t}_i\}_{i=1}^{I}$ contains the normalized vectors with correct orientations. The normal directions at the data points are then computed by convolving $\hat{T}$ with a first-order Gaussian derivative in the same manner as before. Let $N = \{n_i\}$ be the vector-set obtained in this manner. The unsigned curvature at data point r_i is given as $\kappa_i = \|n_i\|$.

Sample points whose unsigned curvatures are greater than their immediate neighbors' are then picked as the exterior control points. Note that the robustness of the curvature computations and the number of extracted control points can be adjusted by changing the standard deviation of the Gaussian kernel, σ, used in the derivative computations. We use a default value of $\sigma = \frac{I}{\beta}$, for some $\beta > 0$, which produces acceptable results for all 2D shapes. However, β is a parameter that the user can adjust to alter the number of exterior control points that are automatically extracted.

4 Results

Figure 9 shows snapshots of our deformation system during various stages of the warping process for retargeting garments among various poses. The top row of Fig. 9 depicts images of two dresses that need to be warped to fit different poses. These dresses are shown overlaid upon target models in the second row of the figure. It is clear that major adjustments are needed. The third row of Fig. 9 shows the deformation results using only interior control points. Further refinements to the shape boundary are made by adding and moving exterior control points, as shown in the fourth row of the figure. Final results of the deformation process are shown in the bottom row of Fig. 9.

The total time required for a user to warp and obtain the final result for each dress in Fig. 9 was approximately three minutes. This performance exceeded the expectations of the commercial users who employed our system to make thousands of retargeted garment images for use in virtual dressing rooms of retailer websites. The underlying mesh in each dress in Fig. 9 consists of approximately 1700 triangular faces. Each ARAP MLS deformation iteration, on average, took 45 ms on a single 2.4 GHz Intel® CPU core. In each iteration, the computation time is dominated by the ARAP deformation step.

5 Conclusion

In this paper, we described a deformation system for warping images of garments onto target mannequins of arbitrary poses. The rationale for this work is that input imagery of garments may come from varied sources. However, in order to create a compelling online shopping experience, it is useful for a consumer to drag and drop images of clothing onto a target mannequin to visualize a customizable fashion ensemble. This can only be achieved if the images available to the user have already been warped to be readily aligned on the target mannequin. This warping/alignment problem was treated as an exercise in 2D shape deformation that is governed by user-specified control points. A balance was maintained to allow the user to perform the deformations with as few operations as possible, while, at the same time, providing fine control over the shapes of the deformed images.

We built upon recent advances in MLS and ARAP shape manipulation, and extended MLS from the image-space to the object-space domain to handle the concave shapes that are prevalent in this application. We also exploited ARAP deformations to yield more natural warps that improve the MLS results. In addition, our system enabled the user to correctly deform objects with overlapping elements. Finally, we showed how the interior and exterior control points enable the system to furnish intuitive local-global control over the shape of the deformed mesh.

Fig. 9 Snapshots of our garment retargeting system. Each column of the five rows respectively depicts an input dress, an overlay of the dress on a target model, partial deformation results using interior control points, additional refinement using exterior control points, and the final result

The work described in this paper has led to a deformation tool for retargeting images of garments onto mannequins of arbitrary poses. It supersedes commercially available tools such as the Puppet Warp module in Adobe® Photoshop® CS5, which enables deformations using only interior con-

trol points. The absence of exterior control points in Puppet Warp makes it difficult to perform local shape adjustments, as required for carefully retargeting images of garments. Our use of interior and exterior control points for coarse and fine shape editing modes facilitates a high-speed workflow to handle vast image collections.

In future work, we intend to augment the process of making finer adjustments to the boundary of deformed objects by using fewer control points. We will experiment with cage-based deformation techniques that use mean value coordinates, harmonic coordinates, Green coordinates, and complex barycentric coordinates [12].

References

1. Alexa, M., Cohen-Or, D., Levin, D.: As-rigid-as-possible shape interpolation. In: SIGGRAPH '00 (2000)
2. Cuno, A., Esperança, C., Oliveira, A., Cavalcanti, P.: 3D as-rigid-as-possible deformations using MLS. In: Proceedings of the 27th Computer Graphics International Conference, pp. 115–122 (2007)
3. Igarashi, T., Moscovich, T., Hughes, J.F.: As-rigid-as-possible shape manipulation. ACM Trans. Graph. **24**(3), 1134–1141 (2005)
4. Jacobson, A., Baran, I., Popović, J., Sorkine, O.: Bounded biharmonic weights for real-time deformation. ACM Trans. Graph. **30**(4), 78:1–78:8 (2011)
5. Karni, Z., Freedman, D., Gotsman, C.: Energy-based image deformation. In: SGP '09: Proceedings of the Symposium on Geometry Processing (2009)
6. Levin, D.: The approximation power of moving least-squares. Math. Comput. (1998). doi:10.1090/S0025-5718-98-00974-0
7. Magnenat-Thalmann, N., Kevelham, B., Volino, P., Kasap, M., Lyard, E.: 3d web-based virtual try on of physically simulated clothes. Comput.-Aided Des. Appl. **8**, 163–174 (2011)
8. McLain, D.H.: Drawing contours from arbitrary data points. Comput. J. **17**(4), 318–324 (1974)
9. Mokhtarian, F., Bober, M.: Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization. Kluwer Academic, Dordrecht (2003)
10. Schaefer, S., McPhail, T., Warren, J.: Image deformation using moving least squares. ACM Trans. Graph. **25**(3), 533–540 (2006)
11. Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. In: SGP '07: Proceedings of the Fifth Eurographics Symposium on Geometry Processing (2007)
12. Weber, O., Ben-Chen, M., Gotsman, C.: Complex barycentric coordinates with applications to planar shape deformation. Comput. Graph. Forum (2009). doi:10.1111/j.1467-8659.2009.01399.x
13. Weng, Y., Xu, W., Wu, Y., Zhou, K., Guo, B.: 2d shape deformation using nonlinear least squares optimization. Vis. Comput. **22**, 653–660 (2006)

Hadi Fadaifard received his Ph.D. degree in Computer Science from the City University of New York (CUNY) in 2011, and his B.Sc. degree in Computer Science from the City College of New York in 2003. From 2006 to 2011, he was a Research Computing Fellow at the CUNY Graduate Center. He is currently a Research Scientist at Brainstorm Technology LLC. His interests include shape deformation, shape matching, and face recognition.

George Wolberg is a Professor of Computer Science at the City College of New York. He received his B.Sc. and M.Sc. degrees in Electrical Engineering from Cooper Union in 1985, and his Ph.D. degree in Computer Science from Columbia University in 1990. He was an early pioneer of image morphing, and has conducted research on warping, interpolation, registration, 3D reconstruction, and structure from motion. Prof. Wolberg is the recipient of an NSF Presidential Young Investigator Award, CCNY Outstanding Teaching Award, and NYC Mayor's Award for Excellence in Science and Technology. He is the author of *Digital Image Warping*, the first comprehensive monograph on image warping and morphing.