Interactive Editing of Upholstered Furniture

Christopher Schwartz, Patrick Degener, Reinhard Klein
Institute for Computer Science II, University of Bonn

ABSTRACT

Fast visualization of industrial parts for rapid prototyping is nowadays eased by the fact that CAD construction data is readily available in most cases. Upholstery constitutes an important exception as its shape is not given a priori but the result of complex physical interactions between hard bodies, soft cushioning and elastic sheets. In this paper we propose an interactive visualization and editing method for upholstery that infers physically plausible surfaces from a sewing pattern. Our method supports fast design decisions by allowing easy and intuitive modifications of the inferred surface at any time.
We also propose a reconstruction method for point clouds that is specifically targeted at upholstery. We argue that the sewing pattern encodes important information about shape and material deformations of the final surface and consequently use it as a prior in our reconstruction algorithm. The practicability of our method is demonstrated on two real world data sets.

Keywords: Upholstery, Mesh Optimization, Mesh Editing, Surface Modeling, Reconstruction

1 INTRODUCTION

In many modern design and production processes an early and realistic visualization of the product is of high importance as it not only allows timely marketing but also reveals errors and supports decisions during the design phase. To this extent, visualization tools must be fast and flexible so that designers can quickly evaluate different design options or modifications of a prototype.

As most production pipelines are nowadays fully or in large parts automatized, the products shape can usually be directly derived from CAD data or construction plans which eases visualization. An important exception is, however, upholstery: For furniture, cushions or car seats only the sewing patterns, the shape of the cushioning material and the solid upholstery frame are known (see Figure 3).

Even though the frame might roughly resembles the upholstered furniture at first sights, the actual shape of the surface usually differs strongly as it is determined by the complex interaction of frame, cushioning and fabric. In this context, the concern of this paper is to infer a physically plausible surface model of the upholstered object for visualization. Moreover, to enable rapid prototyping we aim at a surface model that allows easy, intuitive and interactive manipulation. Furthermore, in some cases (possibly incomplete) 3*D* point measurements might be available for an upholstered furniture in addition to sewing patterns. In this paper we therefore also propose a novel method to reconstruct the upholstered object's shape from such measurements.

To visualize upholstery the fabric surface can be obviously inferred by physical simulation. In fact, the simulation of textiles or cloth has a long tradition in computer graphics. Early approaches introduced either continuous [21] or discrete [3] deformation models for elastic sheets which have been subsequently improved e.g. with respect to speed [22], stability [7] or to handle materials with special properties (e.g. [9]). Most of these approaches naturally handle interactions with rigid bodies and even interactions between cloth and soft bodies has been considered, e.g. in [8].

For an early visualization in the design process the physical simulation approach in general is, however, not well suited: First, the interaction between fabric, cushioning and frame is involved and requires measurement and specification of material and friction parameters for all involved materials. Second, small design

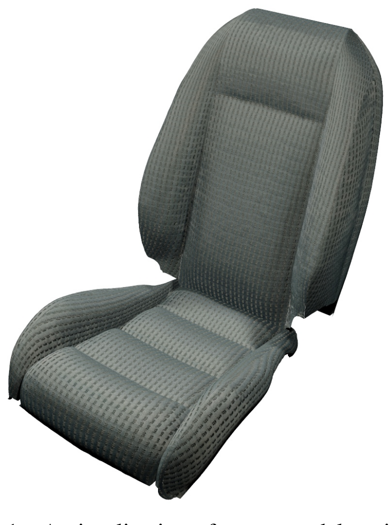
Figure 1: A visualization of a seat model retrieved with our method.

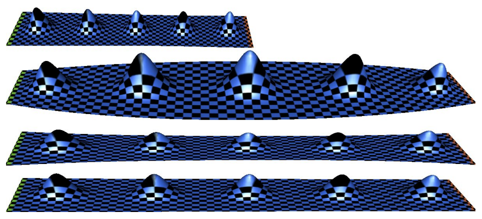
Figure 2: A patch undergoing an anisotropic deformation. The deformation model of [16] can be parametrized to mimic the behaviour of different materials.

modifications like adding a fancy seam must be actually modelled and implemented in a physically correct manner. This is far more complex than directly editing the surface. For these reasons we abandon strict physical accuracy in this paper. Nevertheless the results of the reconstruction should be at least physically plausible and mimic natural effects like folds or wrinkles.

To allow easy and intuitive modifications of surfaces, a number of editing metaphors have been proposed [10, 17, 11, 12] that allow the user to specify changes to the surface by sketching. In particular, the approach of Nealen et al. [15] uses a set of curves as surface representation, that can be arbitrarily modified or extended.

Targeted at general surface editing these approaches have no notion of an underlying parametrization. For upholstery, however, a parametrization is crucial to represent stretch and shear of cover materials. Furthermore, the parametrization corresponds directly to the sewing pattern used in the upholstery. Since in the above approaches constrained curved can only sketched directly on the surface, patterns and seams between patterns can not be defined. For interactive design of plush toys, Mori and Igarashi [14] therefore maintain the sewing pattern as internal shape representation during editing. To verify that the constructed pattern is valid, their method actually computes the surface shape using a simple reconstruction method. While their approach yields good results for relatively simple stuffed objects, the obtained surface deformations are in general not physically plausible (see Figure 10).

Non-rigid template surface fitting for reconstruction was proposed by Marschner et al. [13]. Their approach fits a mesh template to a target mesh by iteratively minimizing distances between closest point pairs similar to the ICP algorithm [4]. For well-posedness a smoothness term is added in the minimization that produces smooth surfaces in areas with missing data. Alternative regularizations have been proposed in [1] and [2] that improve the preservation of template details by enforcing locally affine transformations instead of general smoothness.

Similar to the above approaches our reconstruction algorithm can also be regarded as a generalized ICP algorithm. However, our regularizations is based on the mesh deformation model from [16] that penalizes anisotropic local transformations. This mimics the behaviour of textiles and gives physically more plausible results for upholstery (see Figure 2). While a physically plausible deformation behaviour is not always desirable for general template fitting (for example, local surface scaling is physically not plausible but often necessary to accommodate differences in surface area) it is crucial for upholstery reconstruction.

In [20] Stoll et al. present a non-rigid template fitting method for reconstruction of point sampled surfaces which also generalized the classical ICP algorithm. To prevent overfitting and to preserve template details their approach resorts to Laplacian surface editing [19], a linear deformation technique which is known to yield physically implausible deformations [6]. Moreover, their method adapts fitting weights to accommodate for template vertices without reliable corresponding point on the sampled surface which requires a rather costly matrix refactorization in each optimization step. We circumvent this problem by proposing a novel type of constraint that does not requires matrix refactorizations.

In summary, our paper makes the following contributions: We present an interactive reconstruction method for upholstery from sewing patterns and upholstery frame that is based on the mesh optimization method in [16]. Several extension to this mesh optimization are described, most notable $G1$-continuous seams and force field constraints. We propose a curve based representation that allows fast and interactive changes of the surface that using a sewing pattern metaphor similar to [14]. In contrast to the latter it shows physically more plausible behaviour and can simulate different materials. In addition, our representation supports direct drawing of additional constraint curves into the sewing pattern. We exemplify the power of this concept by a simply but effective tool for adding fancy seams.

Last but not least, we also show how to reconstruct the fabrics surface from possibly incomplete or partial $3D$ point measurements if these are available. To this extent, we describe a further extension to the mesh editing method from [16]. The practicability of our approach is exemplified on two real world data sets.

2 PROBLEM STATEMENT AND OVERVIEW

For the following we assume that for a piece of upholstered furniture a sewing pattern and an upholstery frame are given (see Figure 3). The pattern is given as a set of closed planar curves that define the outlines of patches. Individual patches are further annotated with sewing instructions or seams illustrated by dashed arrows in Figure 3a. Besides seams between patches, in the construction some parts of the sewing pattern are

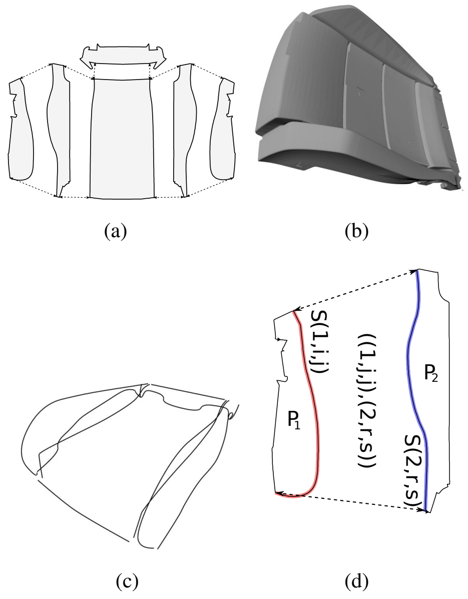

Figure 3: (a) sewing pattern with sewing instructions (b) upholstery frame (c) fixations curves on the upholstery frame. (d) seam between patches P_1 and P_2

also fixated at the upholstery frame in $3D$. Figure 3c shows an example of a set of fixation curves on a frame.

Now, given a sewing pattern, fixation curves and seams the first problem that we tackle in this paper is to compute a visibly plausible approximation of the fabrics surface. To this extent, Section 3 proposes a suitable surface representation and a basic reconstruction algorithm. Moreover, we describe a set of interactive editing operations on the reconstructed surface. In Section 4 an extension to the basic reconstruction algorithm is described that improves on it by simulating a cushioning between upholstery frame and surface.

Finally, we consider in Section 5 the case where a set of point samples $S \subset \mathbb{R}^3$ on the fabric surface is given as an additional input. The point set S might be obtained by laser range scanning and can be incomplete, noise and disconnected. The problem is then to reconstruct a closed fabric surface that approximates the set S while adhering to the given sewing pattern and seaming constraints. After exemplifying the proposed algorithms on real world models in Section 6 we close with a short conclusion in Section 7.

3 BASIC RECONSTRUCTION AND EDITING

3.1 Representation

At the heart of our representation is the sewing pattern $\mathbf{P} = \{P_1, \dots, P_n\}$, a set of simple closed curves P_i in $2D$ that define the outlines of patches. In our implementation these are given as either polygons or B-spline curves. Seams between patches are represented by tuples $((i,r,s),(j,r',s'))$ denoting a seam between the curve segment $P_i([r,s])$ and another $P_j([r',s'])$ (see Figure 3d). We collect all seams between patches in the set $\mathscr{PS}$.

As described in the previous section there are also fixation seams between fabric and frame. We therefore keep a list with fixation curves $\mathbf{C} = \{C_1, \dots, C_m\}$ on the surface of the upholstery frame which are also given as either B-splines or polygons in $3D$. In contrast to patch outlines P_i these curves might be open. Fixations seams are represented analogously to patch seams: we store for each seam between a curve segments $P_i([r,s])$ and a fixation curve C_j a tuple $((i,r,s),j)$ in a set $\mathscr{FS}$.

Our idea to derive the fabrics shape given these inputs is to simulate an elastic sheet cut from the sewing pattern that adheres to the sewing and fixations. If we assume a sufficiently stiff material like e.g. leather or paper that resists bending to some degree, the sewing pattern $\mathbf{P}$ together with fixation curves $\mathbf{C}$ and seams $\mathscr{PS}, \mathscr{FS}$ do in fact determine the shape of this sheet. By using the elastic sheet metaphor, these sets can therefore be indeed regarded as a representation for the upholstered shape. Later we will augment this representation by further user defined constraint to enable interactive editing. The next section describes the choice of an appropriate interactive sheet simulation.

3.2 Basic Reconstruction

Although in principle strictly physically based deformation models like e.g. for cloth simulation can be used to simulate elastic sheets, physical accuracy is, as argued in the introduction, not required in our case and the relatively high computational effort thus not justified. An alternative constitute interactive mesh editing methods that are less accurate but tuned to run interactively even for large models. Among a multitude of deformation models for elastic sheets we chose the mesh editing method of Paries et al. [16] as it is very flexible and provides physically plausible results at interactive frame rates. In contrast to other mesh editing method [6, 5], the deformation model of [16] can emulate the behavior of materials with different Poisson ratios, i.e. rubber, leather, or textiles via a single parameter (see Figure 2). This is of high importance for our application as it guarantees physically plausible shear or stretch in the texture map. In the following we summarize the most important concepts from [16].

The mesh editing algorithm minimizes iteratively a non-linear deformation energy on the surface by solving two linear systems in least squares sense and a parallel low-dimensional eigenvalue problem. In each step, the first linear systems $L1$ updates the positions of all vertices while the second system $L2$ optimizes a local frame in each vertex. Both steps aim at the preservation of local surface details. The method gains its speed from the fact, that the system matrices of the two linear systems do not change during the iterations and thus can be prefactored using sparse direct solvers.

In addition, the original formulation allows to constrain the position and the local frame at arbitrary vertices. This is realized by adding additional rows to both linear systems: positional constraints are added in the first system while constraints to the local frames - orientation constraints - are added to the second. Although the original formulation does not describe pure *position constraints* or pure *orientation constraints*, these can be added in straightforward manner by skipping the corresponding rows from the other linear system.

To reconstruct upholstery with this mesh optimization algorithm, we first triangulate the interior of each curve P_i of the sewing pattern $\mathbf{P}$ using a planar Delaunay triangulator [18]. For later reference we label boundary vertices positioned at $P_i(t)$ with their parameter value t. With the resulting meshes M_i^0, we initialize the mesh optimization that eventually yields optimized meshes M_i with the same topology that locally resemble M_i^0. As the meshes M_i^0 are flat and the above described iterations of the mesh optimization will thus minimize bending in the surface. In the absence of further constraints the patches M_i will therefore remain flat.

To impose sewing constraints, we compute for each seam $((i,r,s),j) \in \mathscr{FS}$ the vertices on the boundary of M_i^0 that correspond to $P_i([r,s])$, i.e. with label values $r \leq t \leq s$ and constrain their position to $C_j((t-r)/(s-r))$.

Finally, sewing constraints between patches have to be imposed. For each inter patch sewing constraint $((i,r,s),(j,r',s')) \in \mathscr{PS}$ boundary vertices with corresponding labels have to be identified. More precisely, a vertex on $P_i([r,s])$ labeled with t must be sewed to the vertex on $P_j([r,s])$ with the label $t' = \frac{s'-r'}{s-r}(t-r)+r'$. However, as usually a vertex with that label does initially not exist in M_j^0, we add it by subdividing the corresponding boundary edge.

To actually impose a sewing constraint for two corresponding vertices v and v' we add an additional row to the first linear system in the mesh optimization, to minimize the quadratic distance between v and v'. In the formulation of [16] the coordinates of all vertices are collected in a matrix. If k and k' denote the row indices of v and v' respectively, the additional row equals 1 at column index k and -1 at k'. All other entries are zero.

3.3 Interactive Editing

In our implementation the user interface is divided into two windows (see Figure 4). There is a *pattern window* showing patterns laid-out in $2D$ with seams indicated by lines. Pattern curves can be modified and rearranged. Moreover, seams between patches can be defined. In the main $3D$*-window* the reconstructed model and the fixation curves $\mathbf{C}$ are shown. In both windows, the user interacts with the reconstructed model only by adding or changing curves, either patch boundaries or fixations.

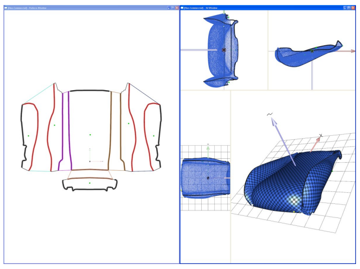
Figure 4: The user interface of our interactive editor.

When two points on a patch curve are selected it is possible to create a new fixation curve in the $3D$-window. This also creates a fixation seam between patch and the created curve. It is possible to display a polygon model of the upholstery frame for guidance. The user interface and its functions are exemplified in the accompanying video.

Modifying a fixation curve in the $3D$-window changes the position constraints of the associated vertices (see previous section). As the mesh optimization is tuned to handle changes in positional constraint very efficiently, the shape of the reconstructed surface updates interactively at about $10-15$ fps. Modifications on a pattern curve in the pattern window results in instant re-triangulation of the patches and re-initialization of the mesh optimizations. Although this is computationally more expensive than changes of fixation curves, patch boundaries can be edited at interactive frame rates.

3.4 C^1 and G^1 Continuous Seams

So far, inter patch seams constrain only the positions of adjacent patches, i.e. the resulting surface is C^0 continuous at such seams. In some cases, however, higher orders of smoothness desirable (see Figure 8 bottom). In our systems, the user can therefore choose C^1 or G^1 smoothness for individual inter patch seams.

To impose C^1 continuity at a seam, we first identify corresponding vertices as described in section 3.2. In the mesh optimization first order derivatives at a vertex v are explicitly represented by a local frame which is in turn given as quaternion q_v (see [16]). Analogous to positional constraints, we add four similar rows for each pair of corresponding vertices v,w along the seam to the second linear system of the mesh optimization to minimizes the squared difference $\|q_v - q_w\|^2$. This effectively enforces nearly identical local frames at both sides of the seam and thus C^1 continuity.

In upholstery seams are often placed to reduce stretch of the fabric. This is most effective if discontinuities in the texture or fabric pattern are allowed at the seam, i.e. discontinuous tangent vectors at the seam. Nevertheless, the surface shape should be smooth. In other words, the surface should be G^1 continuous at the seam. To allow this weaker form of continuity, we have to enforce identical normal vectors at corresponding vertices along the seam. More precisely, for corresponding vertices v and w with frames encoded by quaternions q_v, q_w and vertex normals n_v, n_w we require

$$\mathbf{R}[q_w]\mathbf{R}[q_v]^{-1}n_v = n_w \tag{1}$$

where $\mathbf{R}[q]$ denotes the rotation matrix associated with the quaternion q. The above expression simple requires that both vertex normals coincide in the local frame coordinates of vertex w.

Unfortunately, equation 1 is nonlinear in the quaternion components and thus cannot be represented in the second linear system of the optimization anymore. As the linearity of this step is crucial for the performance of the method, we use the following approximation: In each step of the iterative mesh optimization we compute the quaternions q_{vw} and q_{wv} that rotate n_v onto n_w and vice versa. We then add the following constraints to the linear system:

$$q_v = q_{vw}q_w \quad \text{and} \quad q_w = q_{wv}q_v$$

To avoid matrix updates that require an expensive refactorization of the linear system, we evaluate the right hand part of both equation and constrain q_v and q_w to the computed values using the build-in orientation constraint mechanism.

3.5 Handle Curves and Fancy Seams

Although results of the basic reconstruction algorithm are plausible, the information encoded by fixation curves are sparse and there might be the need to add surface details in some parts of the upholstered object. To provide additional flexibility, we therefore allow the user to draw addition handle curves H_k into the interior of the planar pattern patches after a surface has been reconstructed. In contrast to patch boundaries $\mathbf{P}$ these curves can also be open.

Each handle curve H_k is converted to a polygon and triangulated into the respective patch M_i^0. The corresponding edges in the reconstructed $3D$ surface M_i are then collected into a polygon H_k'. We then add an additional fixation seam mapping H_k onto H_k' and allow the user to manipulate both H_k and H_k' interactively. In this way, the surface can be easily modified to meet the user's needs. If H_k is given as a spline, it is also convenient to approximate H_k' by a $3D$ spline for editing.

Figure 5 shows an important application of handle curves. Fancy seams do not connect between patches but are of purely decorative nature. Nonetheless these seams have a great impact on the object's appearance. Using handle curves, a fancy seam can be easily realized by adding triples of parallel curves as shown in the figure and translating the $3D$ handle curve in the middle.

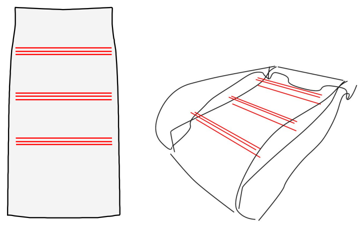
Figure 5: Adding fancy seams. Left: additional handle curves H_k in the pattern. Right: corresponding $3D$ handle curves H_k'.

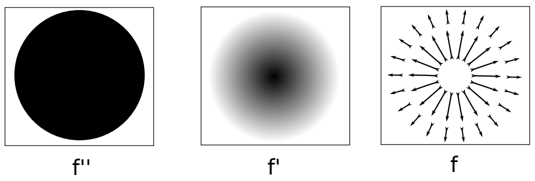

Figure 6: The three steps to create the force field f (see text).

4 SIMULATING CUSHIONING

The basic reconstruction algorithm described in the last section relies only on the sewing pattern and fixation curves. The actual shape of the upholstery frame has not been considered so far. However, as described in the introduction frame and cushioning have significant influence on the actual shape of upholstered furniture.

To simulate cushioning and to incorporate the frame's shape in the interactive reconstruction we propose a simple force field based approach: First, we compute a force field $f : \mathbb{R}^3 \to \mathbb{R}^3$ from the upholstery frame's geometry that assign each point in the inside a force pointing toward the cushions outside. We then extent the mesh optimization to push the reconstructed surface outwards along the force field.

4.1 Creating the Force Field

In our approach, the actual force field f is defined by trilinear interpolation of a discrete grid $f_{i,j,k}$ surrounding the upholstery frame. To construct this grid we start with a binary voxel grid $f'' : \mathbb{N}^3 \to \mathbf{R}^3$ defined as

$$f''_{i,j,k} = \begin{cases} 1, & (i,j,k) \text{ is inside the frame} \\ 0, & otherwise \end{cases}$$

From this voxel grid, we then calculate for each voxel the city-block-distance to the nearest voxel with $f''_{l,m,n} = 0$ and store it in f' (see Figure 6). Finally, to obtain an outside pointing force field f we take discrete partial derivatives of f' scale them with the distance value stored in f':

$$f_{i,j,k} = \begin{pmatrix} \frac{f'_{i+1,j,k} - f'_{i-1,j,k}}{2} \cdot f'_{i,j,k} \\ \frac{f'_{i,j+1,k} - f'_{i,j-1,k}}{2} \cdot f'_{i,j,k} \\ \frac{f'_{i,j,k+1} - f'_{i,j,k-1}}{2} \cdot f'_{i,j,k} \end{pmatrix}$$

To simulate extra cushioning a user specified amount of dilation steps can be applied the auxiliary binary voxel grid f'.

4.2 Force Field Constraints

For cushioning simulation we first construct an initial surface using the basic reconstruction algorithm described in Section 3.2. Then a force to each vertex is applied that is equivalent to the value of f at its location.

To integrate the force field f into the mesh optimization without breaking the linearity of the two sub steps $L1$ and $L2$, we propose a simple extension that fits into the least squares frame work: If a force f_v should be imposed to a vertex v located at p_v, we compute a ghost position

$$p'_v = p_v + \frac{1}{2} f_v$$

and add a soft positional constraint to the linear system $L1$, i.e. we add rows corresponding to the equation:

$$p_v = p'_v$$

As the linear system $L1$ is solved in least squares sense, this effectively minimizes the quadratic energy $E_f := \|p_v - p'_v\|^2$. The force imposed on v in the minimization is thus equivalent to the negative gradient of this energy which is given by

$$-\frac{\partial E_f}{\partial p_v} = -2(p_v - p'_v) = f_v$$

Although the ghost position p'_v has to be updated in each iteration, the system matrix of the linear system $L1$ remains constant and thus no expensive refactorization is required.

Using the above described force field constraints, the force field f can be trivially integrated with the mesh optimization by evaluating the field at all vertex positions in each iteration. However, we found in our experiments that the robustness of the cushioning simulation can be improved, by applying only the fraction of the force f that is orthogonal to the surface. Furthermore, we want the cushioning to pull the surface only outwards, i.e. in the direction of its normal. We thus use the following force assignment

$$f_v = \max\left(0, \langle n_v | f(p_v) \rangle\right) n_v$$

where n_v denotes the vertex normal at v.

5 SURFACE RECONSTRUCTION FROM 3D MEASUREMENTS

In this section we assume that besides frame and sewing pattern also a set of point measurements S on the surface is given. As described in Section 2 this measurements can be obtained e.g. from range scans and can be incomplete or have holes. In this case, our general approach can also be used to reconstruct a complete surface from these samples.

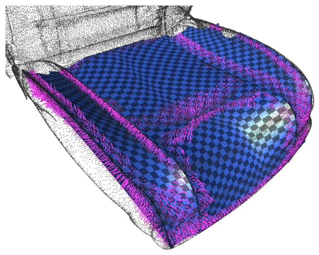

Figure 7: Point cloud fitting: initialized mesh with closest pairs

Even the most basic reconstruction method described in Section 3.2 usually yields a surface that already approximates the actual fabric surface (see Figure 7). To fit this initial surface to the point set S we use an ICP-like approach. The original iterative closest point (ICP) algorithm [4] searches a single rigid transformation that aligns two point sets so that their distance is minimized. It iterates between finding pairs of closest point in the two point sets and minimizing the distances between these point pairs until it eventually converges.

Similarly, we start by computing for every vertex v positioned at p_v its nearest neighbor $s_v \in S$ using a fast kd-tree indexing structure. The distance $||p_v - s_v||$ is evaluated and all vertices whose values is below a certain threshold are marked. This is illustrated in Figure 7 where pairs of marked vertices and closets point samples are connected by red lines.

To minimize the distance between vertices and selected samples s_v we again use force field constraints: To each marked vertex v we apply the force

$$f_v = s_v - p_v$$

pointing towards the point sample s_v. For all unmarked vertices we set $f_v = 0$. Finally, a mesh optimization step is computed. Closest point search and mesh optimization are iterated until the surface converges.

The reason why we have chosen force field constraints instead of simple positional constraints $p_v = s_v$ is that the set of marked vertices can change significantly in each iteration. Adding of removing positional constraints from $L1$, however, changes the structure of the system matrix and thus would require a refactorization in each iteration which is infeasible.

Contrary to the original ICP our method does not solve for a single rigid body transformation but performs an optimization of the whole surface. The resulting mesh approximates the point cloud while minimizing surface bending and material stretch or shear. Certainly, seams between pattern patches and fixation seams are nevertheless maintained.

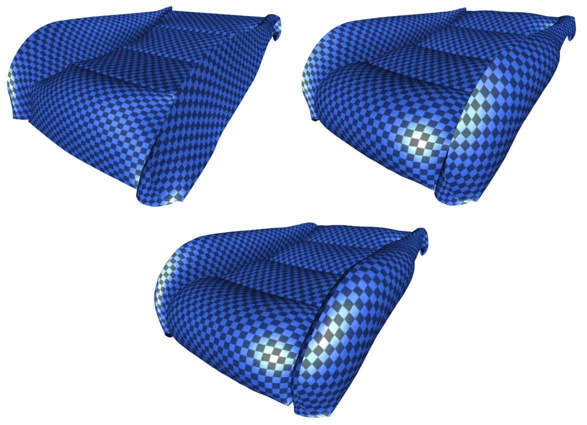
Figure 8: Top left: result after basic reconstruction. Top right: same object with cushioning simulation. Bottom: result after selecting $G1$ smoothness for the two seams along the two armrests.

6 RESULTS

We applied our method to two real world data sets to demonstrate its applicability. For both data sets a sewing pattern and upholstery frame were given. In addition, a laser range scan of the constructed seat was given for the second data set.

The results for the first data set are shown in Figure 8. The fixation curves and the frame for this data set are those shown in Figure 3. As visible in the figure, the result of the basic reconstruction already resembles a car seat. However, the effects induced by cushioning are clearly missing. The example shown in the middle of Figure 8 demonstrates that the cushioning simulation can add these effects.

For visualization of material deformations, we used a checker board pattern texture and mapped it to the surface patches M_i using the positions of the corresponding mesh elements in M_i^0 as uv-coordinates. Material deformations become thus visible in deviations of the checker boards from equally sized squares. As visible in the figure, deformations appear plausible and resemble the shearing and stretching of natural materials.

In the bottom of the figure an application for G^1-continuous seams is presented: The two seams running along the armrest of the cushion meet at an sharp angle after cushioning simulation. As in this case clearly a smooth transition is desired, we selected $G1$ continuity for these seams. The results shown in the bottom of Figure 8 is a smooth surface while the texture map is still discontinuous as visualized by the checker boards.

For the second data set we used the surface reconstruction from $3D$ measurements. As shown in the accompanying video the fitting runs interactively and converges after less than 5 seconds. The result of the reconstruction is shown in Figure 9 while the original point cloud and the initialization is shown in 7. The fancy seams in this example were not added by the handle

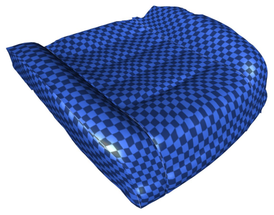
Figure 9: Result of point cloud fitting.

	# vertices	**# constraint**	**FPS**
basic reconstruction	11094	2073	15
cushioning	11094	11094	13
cushioning + G^1-seams	11331	11736	11
point fitting	8026	8026	10

Table 1: Timings of the mesh optimization

curves but reconstructed from detail in the point set. Please note that the sewing pattern of this data set is not symmetric as it contains a notch for the belt on the right hand side.

In Table 1 we collected timings for the mesh optimization using the extensions proposed in this paper. The rows correspond to the reconstruction algorithm, cushioning simulation using force field constraints for all vertices, cushioning simulation with two additional G^1 seams, and point cloud fitting with a point set size of $105,126$. All tests were conducted on a 2.4GHz machine with a GeForce 8800 used for acceleration of the mesh optimization as descibed in [16]. Even with the relatively expensive nearest neighbor search and a large number of constraints the optimization runs at interactive frame rates for both data sets.

Finally, we present in Figure 10 a comparison between our method and the plush toy design tool Plushie [14]. To this extent, we build a small cushion using a quadratic patch of fabric (see figure) and sewed it to a trapezoidal fixation curve. For better comparison we also added a cube shaped cushioning to our result. While in the result obtained by Plushie the fabric is locally scaled to compensate for the smaller lower edge of the trapezoid, this behaviour is seldom observed for real materials: most materials like leather, textiles etc. rather form out folds and wrinkles to reduce compression of the fabric. This expected behaviour can be observed with our method.

7 CONCLUSION

In this paper we proposed an visualization and interactive editing system for upholstery designed to support early and fast decisions in industrial design. The system is build around a basic reconstruction method that infers an initial shape from sewing pattern and fixation curves. Besides manual editing of this initial shape, we proposed extension to simulate cushioning and for fitting to given point measurements. Our approach is based on the mesh optimization from [16] but extents it in several important ways: we describe ways to realize

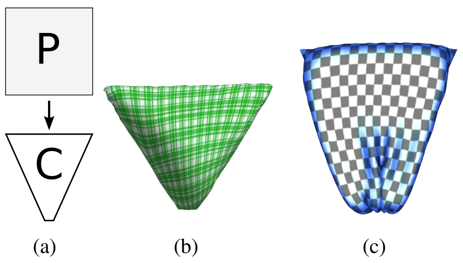

Figure 10: Comparison of our method with Plushie: (a) setup: a quadratic shaped pattern P is sewed to a trapezoid shaped fixation curve C. (b) result using Plushie (c) result using mesh editing (bolstered by a quadratic shaped cushion)

G^1 seams, incorporate a force field and to fit the mesh interactively to a point cloud.

Currently our method fails to produce satisfying results if the shape of the upholstery frame differs strongly from the actual cushioning. In these cases, deriving a force field as described in Section 4 might not provide a suitable approximation for the forces expelled by the cushioning. We therefore want to to explore simple and intuitive ways to edit the force field f or to roughly model the shape of cushioning material.

Acknowledgements The results shown in this paper are based on sewing patterns, upholstery frame data and 3*D* scans kindly provided by the Volkswagen AG.

REFERENCES

[1] B. Allen, B. Curless, and Z. Popovic. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graph*, 22(3):587–594, 2003.

[2] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid ICP algorithms for surface registration. In *CVPR*. IEEE Computer Society, 2007.

[3] D. Baraff and A. P. Witkin. Large steps in cloth simulation. In *SIGGRAPH*, pages 43–54, 1998.

[4] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell*, 14(2):239–256, 1992.

[5] M. Botsch, M. Pauly, M. H. Gross, and L. Kobbelt. Primo: coupled prisms for intuitive surface modeling. In *Symposium on Geometry Processing*, volume 256, pages 11–20. Eurographics Association, 2006.

[6] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE Trans. Vis. Comput. Graph*, 14(1):213–230, 2008.

[7] K.-J. Choi and H.-S. Ko. Stable but responsive cloth. In *SIGGRAPH*, pages 604–611. ACM, 2002.

[8] F. Cordier, P. Volino, and N. Magnenat-Thalmann. Integrating deformations between bodies and clothes. *Journal of Visualization and Computer Animation*, 12(1):45–53, 2001.

[9] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun. Efficient simulation of inextensible cloth. *ACM Trans. Graph*, 26(3):49, 2007.

[10] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A sketching interface for 3D freeform design. In *SIGGRAPH*, pages 409–416, 1999.

[11] L. B. Kara and K. Shimada. Sketch-based 3D-shape creation for industrial styling design. *IEEE Computer Graphics and Applications*, 27(1):60–71, 2007.

[12] O. A. Karpenko and J. F. Hughes. Smoothsketch: 3D free-form shapes from complex sketches. *ACM Trans. Graph*, 25(3):589–598, 2006.

[13] S. R. Marschner, B. K. Guenter, and S. Raghupathy. Modeling and rendering for realistic facial animation. In *Rendering Techniques*, pages 231–242. Springer, 2000.

[14] Y. Mori and T. Igarashi. Plushie: an interactive design system for plush toys. *ACM Trans. Graph*, 26(3):45, 2007.

[15] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. Fibermesh: designing freeform surfaces with 3D curves. *ACM Trans. Graph*, 26(3):41, 2007.

[16] N. Paries, P. Degener, and R. Klein. Simple and efficient mesh editing with consistent local frames. In *Pacific Conference on Computer Graphics and Applications*, pages 461–464. IEEE Computer Society, 2007.

[17] R. Schmidt, B. Wyvill, M. C. Sousa, and J. A. Jorge. Shapeshop: Sketch-based solid modeling with blobtrees. In *Sketch Based Interfaces and Modeling*. Eurographics Association, 2005.

[18] J. R. Shewchuk. Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. In *WACG*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer, 1996.

[19] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. *SGP*, pages 179–188, 2004.

[20] C. Stoll, K. Zachi, C. Rössel, H. Yamauchi, and H.-P. Seidel. Template deformation for point cloud fitting. *Symp. on Point-Based Graphics*, pages 27–35, 2006.

[21] D. Terzopoulos, J. C. Platt, A. H. Barr, and K. W. Fleischer. Elastically deformable models. In *SIGGRAPH*, pages 205–214. ACM, 1987.

[22] P. Volino and N. Magnenat-Thalmann. Implementing fast cloth simulation with collision response. In *Computer Graphics International*, page 257, 2000.