EUROGRAPHICS 2014 / B. Lévy and J. Kautz
(Guest Editors)

Volume 33 (2014), Number 2

Detection and reconstruction of freeform sweeps

M. Bartoň [1] H. Pottmann [1,2] J. Wallner [3]

[1] KAUST [2] TU Wien [3] TU Graz

Abstract

We study the difficult problem of deciding if parts of a freeform surface can be generated, or approximately generated, by the motion of a planar profile through space. While this task is basic for understanding the geometry of shapes as well as highly relevant for manufacturing and building construction, previous approaches were confined to special cases like kinematic surfaces or "moulding" surfaces. The general case remained unsolved so far. We approach this problem by a combination of local and global methods: curve analysis with regard to "movability", curve comparison by common substring search in curvature plots, an exhaustive search through all planar cuts enhanced by quick rejection procedures, the ordering of candidate profiles and finally, global optimization. The main applications of our method are digital reconstruction of CAD models exhibiting sweep patches, and aiding in manufacturing freeform surfaces by pointing out those parts which can be approximated by sweeps.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Computational Geometry and Object Modeling—

Introduction and motivation. The topic of this paper is to detect if freeform surfaces can be generated by simple processes, in particular sweeping. This can mean digitally reconstructing the sweep which generates a given shape, but it can also mean information relevant for manufacturing. These are complex tasks which become increasingly important with today's trends towards totally free forms in many fields of engineering and design: Large scale undertakings, such as realizing freeform architecture (Fig. 1), manufacturing free forms required in engineering (Fig. 2), or any kind of mass production, obviously benefit enormously from a cheap and effective way of manufacturing.

A guiding principle in this area is the decomposition of complex parts into simpler ones, which can be built cheaply and efficiently, without being obviously 'simple' to the untrained eye. Similar thoughts have e.g. led architects to design building skins of freeform appearance which are in fact not genuinely freeform in a two-dimensional way, but can still be described by one-dimensional entities (like the translational surface used for the Hippo house in the Berlin zoo). A very general class of surfaces which are simple from the manufacturing viewpoint contains the surfaces generated by the motion of a profile curve through space. Manufacturing processes based on them are the following:

- Hot-wire foam cutters are tools to cut materials like polystyrene. They are acting on the principle that a straight line L (i.e., the hot wire) is moving relative to the workpiece, thereby acting as a knife. The boundary of the volume cut away by this process is the surface swept by L during its motion. Such a surface is a ruled surface. It is also possible to employ curved hot blades instead of straight wires under tension, and in this way to generate more general surfaces.

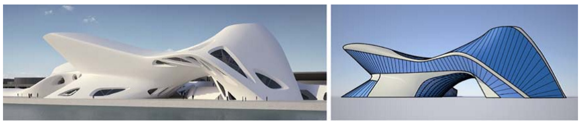


Figure 1: *Large parts of an architectural design may be representable as a sweep surface. In this particular case the simplest kind of sweep, namely ruled surfaces, are used.* Left: *design by Zaha Hadid architects.* Right: *ruled surfaces approximating this design, holes omitted (*www.evolute.at*).*

- In building construction, temporary or permanent underconstructions might be necessary, e.g. when pouring concrete onto wooden formworks. The complex task of formwork assembly is considerably simplified, if the formwork can be made from straight elements (as a ruled surface) or from elements which are not straight but at least which are all the same shape. The resulting surface is a surface generated by the motion of a single curve.

Figure 2: *Formworks for concrete tunnel of variable cross section (Silvretta reservoir, cf.* www.spezialschalungen.com*).*

Contributions. This paper revolves around the above-mentioned class of surfaces, which are generated by the motion of a profile curve through space, and which we call *sweep surfaces* (see Fig. 3). In particular, we do the following:

- ⋆ We evaluate the degree of movability of a given curve within a surface (i.e., to what extent can a curve move such that it remains close to the surface). This evaluation is fast, and is a key ingredient in other algorithms.
- ⋆ An exhaustive search finds planar profiles whose movement generates a given surface or a part of it. It combines profile evaluation, profile comparison, intelligent exploration of the search space, and global optimization.
- ⋆ We decompose any given surface into parts which are well approximated by sweep surfaces. Thus locally the two-dimensional geometry information contained in the surface is compressed into one-dimensional information, namely, the shape of a curve plus the information on how it moves.
- ⋆ The overall goal is to gain information on efficient manufacturing according to the techniques described above, avoiding the very expensive building of molds and similar genuinely two-dimensional methods.

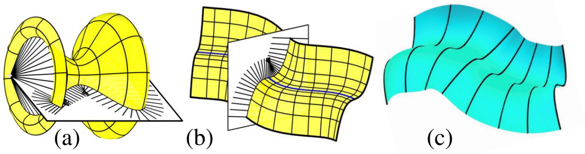


Figure 3: *Sweep surfaces of increasing geometric complexity generated by planar profiles. (a) Kinematic surface (here: surface of revolution). (b) Surface generated by a planar profile which is also a principal curvature line, the motion of the profile being defined by slip-free rolling (moulding surface aka profile surface, cf.* [PHOW04]*). (c) General sweep.*

Previous work. The mathematical questions basic to this paper are (i) detecting if a certain surface is a sweep surface, and (ii) approximation of a surface by a sweep surface. Unfortunately they have not been solved, and to the authors' knowledge no differential-geometric infinitesimal characterization of sweep surfaces has been found.

There is, however, previous work on "simpler" classes of sweep surfaces (for a general discussion, see [VM02]). Approximation of surfaces by ruled surfaces is studied e.g. by [CP99], [EW13] and [FP10]. Surfaces swept by circles have been briefly treated by [BPK*11] and more extensively by [BSK*13]. An interesting case is the recognition of *kinematic* surfaces which are movable within themselves (like surfaces of revolution, see Fig. 3a). They are the topic of [PHOW04, HOP*05]. [GG04] investigated segmentation of surfaces into kinematic parts, and [AS13] studied algorithms in this context from the viewpoint of robustness and independence of coordinate transforms. [PCL98] and [PHOW04] use the ability to recognize rotational surfaces to reconstruct *moulding surfaces* (also known as profile surfaces, see Fig. 3b). They represent the class of sweeps by planar profiles where the paths of individual points are orthogonal to the profile. They are also studied by [KV13] who reconstruct the profile curves from the property that they are principal curvature lines. A general treatment of sweep surfaces however has been missing so far.

As to architecture and building construction, freeform formworks comprise an active field of study. We do not survey the literature, but only highlight a pneumatic method [DK09] for achieving free forms and point to a survey on fabric formworks [VWB11], which is a large topic.

1. Analyzing curves and surfaces.

Profile curves generating sweep surfaces have certain properties which are easy to verify, so one can quickly determine if a certain curve can possibly serve as a profile. In this section we develop such tests which form the basis of the algorithms of Section 2.

1.1. Analyzing curves: assessing movability. We start with the analysis of *movability* of a curve within a surface S. This analysis is based on the velocities experienced by the points of a rigid body which moves. If t denotes time and $\mathbf{x}(t)$ is the trajectory of a point, the velocity vector of that point is denoted by the symbol $\dot{\mathbf{x}}(t)$: we write $\frac{d}{dt}\mathbf{x}(t) = \dot{\mathbf{x}}(t)$. At any time instance t the velocity state of the body is described by 6 parameters

$$\mathbf{C} = (c_1, c_2, c_3, \bar{c}_1, \bar{c}_2, \bar{c}_3)^{\mathsf{T}} = \begin{bmatrix}\mathbf{c}\\ \bar{\mathbf{c}}\end{bmatrix}, \qquad (1)$$

with $\mathbf{c}$ as vector of angular velocity and $\bar{\mathbf{c}}$ as translational component of motion. There is the well known formula

$$\dot{\mathbf{x}} = \bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x}, \qquad (2)$$

see e.g. [PW01]. Now consider a curve moving through space. Assume it to be densely sampled by vertices $\mathbf{x}_i$ ($i = 1, \ldots, n$), each experiencing a certain velocity, $\dot{\mathbf{x}}_i = \bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i$. If the curve moves within S, then these velocity vectors must be tangent to S. With the unit normal vector $\mathbf{n}_i$ in the point $\mathbf{x}_i$, this condition reads $\mathbf{n}_i^{\mathsf{T}}(\bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i) = 0$, i.e.,

$$\begin{bmatrix}\mathbf{x}_i \times \mathbf{n}_i\\ \mathbf{n}_i\end{bmatrix}^{\mathsf{T}} \begin{bmatrix}\mathbf{c}\\ \bar{\mathbf{c}}\end{bmatrix} = 0 \quad (i = 1, \ldots, n). \qquad (3)$$

The question if the given curve is movable within the given surface S can be answered by solving the above system of n linear equations in the 6 unknowns $\mathbf{C} = \left[\begin{smallmatrix}\mathbf{c}\\ \bar{\mathbf{c}}\end{smallmatrix}\right]$.

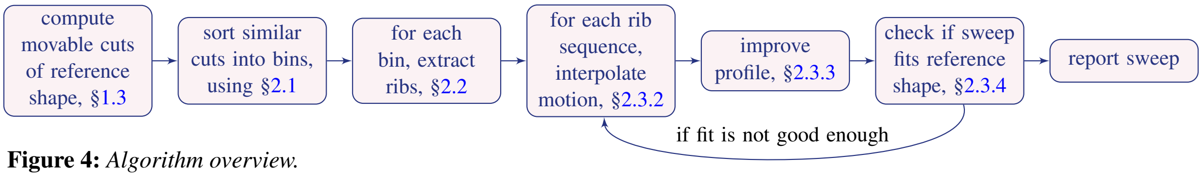

compute movable cuts of reference shape, §1.3
sort similar cuts into bins, using §2.1
for each bin, extract ribs, §2.2
for each rib sequence, interpolate motion, §2.3.2
improve profile, §2.3.3
check if sweep fits reference shape, §2.3.4
report sweep
if fit is not good enough

Figure 4: *Algorithm overview.*

By minimizing the weighted sum of squares

$$F(\mathbf{C}) = \sum_{i=1}^{n} w_i \big(\mathbf{n}_i^{\mathsf{T}}(\bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i)\big)^2 \qquad (4)$$

under the side condition $\|\mathbf{C}\| = 1$, we compute a nonzero least-squares solution of (3), thus finding the velocity state which moves the given curve in a manner as tangential as possible to the given surface (see Fig. 5). The weights w_i will be set later. Rewriting the target function,

$$F(\mathbf{C}) = \begin{bmatrix} \mathbf{c} \\ \bar{\mathbf{c}} \end{bmatrix}^{\mathsf{T}} \Big(\sum_i w_i \begin{bmatrix} \mathbf{x}_i \times \mathbf{n}_i \\ \mathbf{n}_i \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \times \mathbf{n}_i \\ \mathbf{n}_i \end{bmatrix}^{\mathsf{T}} \Big) \begin{bmatrix} \mathbf{c} \\ \bar{\mathbf{c}} \end{bmatrix} = \mathbf{C}^{\mathsf{T}} \mathbf{A} \mathbf{C},$$

shows that the minimizer of F is a unit eigenvector of the 6 by 6 matrix $\mathbf{A}$, corresponding to the smallest eigenvalue λ (then also $\min F(\mathbf{C}) = \lambda$). The case $\lambda = 0$ indicates exact movabililty of the given collection of sample points. We use

$$f_i = \Big(\mathbf{n}_i^{\mathsf{T}} \frac{\dot{\mathbf{x}}_i}{\|\dot{\mathbf{x}}_i\|} \Big)^2 = \cos^2 \sphericalangle(\dot{\mathbf{x}}_i, \mathbf{n}_i) \qquad (5)$$

as the *movability* of the vertex $\mathbf{x}_i$, which is expressed in terms of the angle between the surface and the velocity (recall that $\dot{\mathbf{x}}_i = \mathbf{c} \times \mathbf{x}_i + \bar{\mathbf{c}}$). The value f_i is small if the movement of the vertex $\mathbf{x}_i$ is nicely tangential to the surface.

1.2. Analyzing curves: extracting movable parts. We here determine if a curve has parts capable of moving tangentially to the given surface S. For that, we make use of the possibility to assign weights to vertices when assessing movability: We iteratively downweight vertices which do not move tangentially. Eventually the "movable" parts of the given curve consist of vertices whose weight is still large.

Algorithm 1 (Weight Iteration)**.** *Vertices $\mathbf{x}_i$ with normal vectors $\mathbf{n}_i$, $i = 1, \ldots, n$ are given. Initialize weights $w_i \longleftarrow 1$.*

1. *Find the velocity state $\mathbf{C} = \begin{bmatrix} \mathbf{c} \\ \bar{\mathbf{c}} \end{bmatrix}$ which best fits the given data, by minimizing $F(\mathbf{C})$ according to Equ.* (4)*.*
2. *Compute movability values f_i according to* (5)*, based on velocities $\dot{\mathbf{x}}_i = \mathbf{c} \times \mathbf{x}_i + \bar{\mathbf{c}}$, and recompute weights by*

   $$w_i \longleftarrow 1/(1 + \beta f_i^{\gamma}),$$

   so w_i is smaller if $\mathbf{x}_i$ does not move tangentially.
3. *Go to 1. unless maximum number of iterations is reached.*
4. *Find the connected components $\mathbf{x}_{i_0}, \ldots, \mathbf{x}_{i_1}$ of the set of vertices whose weights are above a certain threshold ε. For each component report its gliding energy*

   $$E_{gliding}(\mathbf{x}_{i_0}, \ldots, \mathbf{x}_{i_1}) = \frac{1}{i_1 - i_0 + 1} \sum_{i=i_0}^{i_1} (1 - w_i)^2 f_i.$$

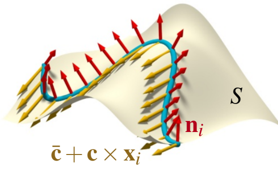


Figure 5: *Movability of a curve (blue) within a surface S. Individual vertices $\mathbf{x}_i$ experience optimal velocities (yellow) tangential to S.*

In our examples we choose $\gamma = 2$, $\beta = 400$, and (mostly) 3 iterations. As to the weight threshold, we let $\varepsilon = \frac{2}{3}$. Fig. 6 illustrates this weight iteration. The precise form of the formulae for weights and gliding energy is not important: their final form as printed in this paper is the result of experiments.

1.3. Analyzing surfaces: search for movable cuts. The main applications we consider in this paper rest on a comprehensive answer to the question which parts of a given surface S can be generated, at least approximately, by the movement of a *planar* profile curve.

For this end we need to exhaustively search all planar cuts of S and determine if they can possibly serve as profiles; those who cannot are discarded and not further considered by the algorithms of Section 2.

The exhaustive search starts with a regular sampling of the unit sphere (we take the vertices of one of Buckminster Fuller's geodesic spheres) to sample the set of normal vectors. Secondly, for each normal there is an interval of planes which intersect S, which is regularly sampled too. Thus we have sampled the space of planes which intersect S. For each plane α we assess movability of the intersection curve $S \cap \alpha$, using the method above. Since the sample was rather coarse we adaptively refine the sampling density in the neighbourhood of such planes which produce cuts of low gliding en-

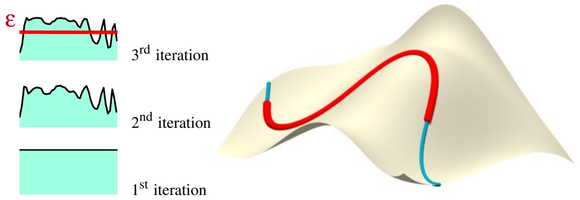


Figure 6: Left: *Plot of weights w_i associated to the vertices $\mathbf{x}_i$ of a curve during weight iteration.* Right: *The red polyline consists of vertices whose weights are above the threshold value ε in the 3rd iteration, implying that it is movable.*

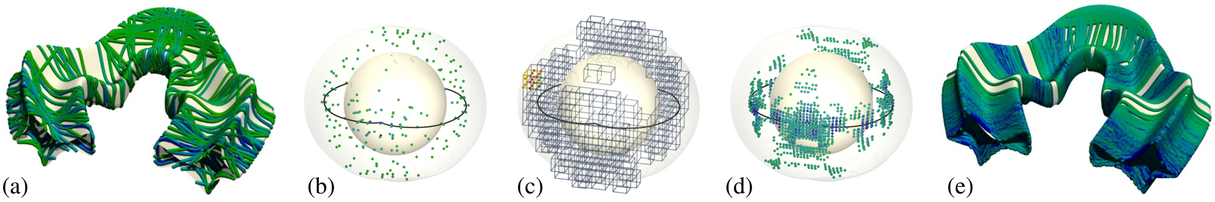


Figure 7: *Verifying our search procedure for movable cuts by means of a surface S which is generated by the movement of a planar profile. (a) A coarse sampling of plane space yields 2420 planar cuts of S. We compute $E_{gliding}$ for each and visualize those with small values of $E_{gliding}$ (the 5-th percentile, which has 206 members). The color of these "most movable" planar cuts corresponds to the value of $E_{gliding}$. (b) In plane space, each planar cut is visualized as a point: A plane with equation $\mathbf{n}^\mathsf{T}\mathbf{x} = d$ is visualized as the point "$d \cdot \mathbf{n}$". The moving planar profile corresponds to the black path in plane space. (c) For each of those "good" planes, we consider a neighbhourhood which appears as a box in plane space, and increase sampling density there. (d), (e) After sampling density has been increased, there are 667 good cuts (again, the 5-th percentile of the $E_{gliding}$ statistics).*

ergy. The result is a set of planar cuts which are infinitesimally movable within the given surface, see Fig. 7.

2. Fitting sweep surfaces.

This section shows how for any given surface S we find those parts which can be represented by the motion of a planar profile. Our algorithm is based on the data collected by the methods of §1 and consists of the following stages:

- ⋆ A set of planar cuts of S is distributed into bins, each containing cuts which partially match in the sense that a rigid body motion transforms one onto the other.
- ⋆ From each bin we select a sparse set of cuts which can be ordered in the manner of sequential positions of a profile curve which moves in space.
- ⋆ These initial data undergo optimization and produce a patch of sweep surface which fits the given surface S.

Figure 4 gives a detailed overview of this algorithm.

2.1. Comparing two curves by partial shape matching.

It is well known that planar curves are uniquely determined by their curvature as a function of arc length [dC76]. This implies the following criterion:

Two planar curves $\mathbf{b}, \mathbf{b}'$ in space are congruent (i.e., there is a rigid body motion mapping one to the other) $\iff$ they have the same length and their signed curvatures $\kappa(s)$, $\kappa'(s)$, considered as functions of arc length, agree up to reversal of sign and/or reversal of direction.

We can therefore recognize parts of polylines as congruent by finding matching parts in the plots of their respective curvatures as functions of arc length, see Fig. 8. This is much simpler than a direct attempt at registration of one curve onto the other (which would have to deal with the 6 degrees of freedom of the group of Euclidean motions).

Solution by Dynamic Programming. We treat this matching problem in a manner very similar to the common substring problem, and solve it by dynamic programming [CLRS05]. The curves to be matched are evenly sampled by vertices $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and $\mathbf{x}'_1, \ldots, \mathbf{x}'_{n'}$, resp., so that the distance of successive vertices is a constant. A partial matching $i' = \varphi(i)$ of vertices, such as illustrated by Fig. 8 is a diagonal walk connecting elements (i_1, i'_1) and (i_2, i'_2) within the $n \times n'$ matrix of possible vertex correspondences: We have either $\varphi(i) = i'_1 - i_1 + i$ or $\varphi(i) = i'_1 + i_1 - i$. Matchings which are too short are not useful for our applications, so we require

$$\text{length}(\varphi) = |i_2 - i_1| = |i'_2 - i'_1| > l_{min}.$$

We set up a similarity energy E which is based on comparing the curvatures κ_i, κ'_j at vertices $\mathbf{x}_i, \mathbf{x}'_j$, resp.:

$$E_{\kappa,\kappa'}(\varphi) = \frac{1}{\text{length}(\varphi)} \sum\nolimits_{i=i_1}^{i_2} |\kappa_i - \kappa'_{\varphi(i)}|. \qquad (6)$$

and find φ such that $E_{\kappa,\kappa'}(\varphi)$ is small but φ is still long. The unknowns in this optimization problem are the indices i_1, i'_1, i_2, i'_2 (under the side condition above). The longest match φ whose energy is below a certain threshold is then the sought-after partial shape match of curves. We find φ via dynamic programming, using an $n \times n'$ cost matrix C, with $C_{j,j'} = \min\{E_{\kappa,\kappa'}(\varphi) \mid \varphi$ partially matches polylines $\mathbf{x}_1, \ldots, \mathbf{x}_j$ and $\mathbf{x}'_1, \ldots, \mathbf{x}'_{j'}\}$. We omit the details and refer to [CLRS05] instead.

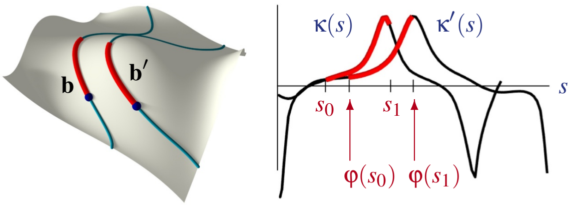


Figure 8: *Partial match (red) between two planar cuts of a surface. The match is discovered by comparing the respective curvature plots $\kappa(s)$, $\kappa'(s)$ and finding a length-preserving mapping φ such that $\kappa(s) \approx \kappa'(\varphi(s))$ for $s \in [s_0, s_1]$.*

Remark. User-defined parameters in the above procedure are l_{min} and the energy threshold. We set l_{min} such that curves shorter than 20% of the bounding box diameter are pruned away. For the energy threshold, see Fig. 15.

Remark. For evaluating the energy E in Equ. (6) we use the robust curvatures of [PWHY09], but E could be based on any similarity measure (e.g. based on deviation of tangent fields [CEY97], see also [BB82] for 2D shape descriptors).

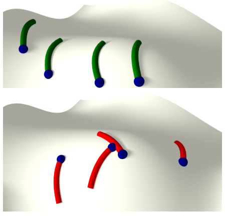


Figure 9: *Congruent curve segments arranged in a good (top) and a bad (bottom) manner, as far as future construction of sweep surfaces is concerned. Any sweep surface based on the bottom sequence exhibits a sharp twist.*

2.2. Finding sequences of proto-profiles. Every valid partial match between two curves gives rise to a family of similar curves, which consists of all other curves matching the initial ones. An exhaustive search of all candidates needs $O(N^3)$ time, with N as number of curves. Since the purpose of matching is to find a profile whose smooth motion generates the surface under consideration, we reject any pairing of curves where the angle between their planes exceeds a certain threshold (mostly 25°), see Fig. 9. The result of this search is an unordered set of curve parts trimmed to equal length, looking like Fig. 10, left, not yet like Fig. 9.

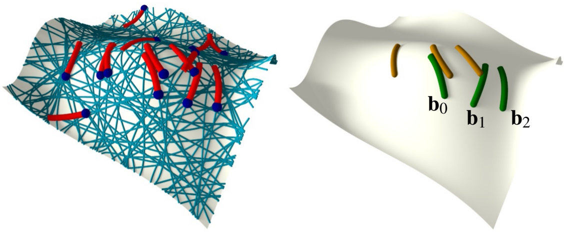


Figure 10: *Extracting rib sequences from a set of similar curves (left, in red). Choosing weights* $(\omega_1, \omega_2) = (1, 1)$ *and* $(0.6, 1)$*, resp., produces the yellow, resp. green, ribs.*

Sparsification and ordering of proto-profiles. We proceed with further treatment of the set of near-congruent curves we have found (see Fig. 10, left, for an example). Our aim is to extract a sequence of curves, called *ribs*, which are arranged in the manner of a moving profile. The relative position of successive ribs should be like in Fig. 9, top, and not like in Fig. 9, bottom. For that purpose we must assess the 'goodness' of the relative position of two curves $\mathbf{b}$, $\mathbf{b}'$ of equal length. For that, we define:

- d_{min} is the minimal distance of curves $\mathbf{b}$, $\mathbf{b}'$.
- d_{haus} is the Hausdorff distance of curves $\mathbf{b}, \mathbf{b}'$.
- δ is the diameter of the family of curves which is under investigation at the moment.
- $\mathbf{t}_i = \frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|}$ is unit tangent vector of $\mathbf{b}$ (similar for $\mathbf{b}'$).

From these quantities we constuct the combined energy

$$E_{order}(\mathbf{b}, \mathbf{b}') = -\frac{d_{min}}{\delta} + \omega_1 \frac{1}{n} \sum_i \mathbf{t}_i^\mathsf{T} \mathbf{t}'_i + \omega_2 \frac{d_{min}}{d_{haus}}.$$

Maximizing E_{order} then favours curves in proximity (1st summand), favours "parallel" curves (2nd summand), and penalizes tangential misalignment, i.e., the sliding of $\mathbf{b}'$ along $\mathbf{b}$ (3rd summand). E_{order} is used to extract and order a sparse rib sequence as follows: We start with an unordered set of curves such as produced by exhaustive search, pick an element $\mathbf{b}_0$ and find $\mathbf{b}_1$ such that $E_{order}(\mathbf{b}_0, \mathbf{b}_1) \to \max$. We then inductively find the curve $\mathbf{b}_{i+1}$ by maximizing $E_{order}(\mathbf{b}_i, \mathbf{b}_{i+1})$, under the side condition that $\mathbf{b}_{i-1}$ and $\mathbf{b}_{i+1}$ do not lie on the same side of the plane which carries $\mathbf{b}_i$. An analogous recursion finds curves $\mathbf{b}_{-1}, \mathbf{b}_{-2}, \ldots$ (see Fig. 10 for an example. For this paper we used $\omega_1 = 0.6$, $\omega_2 = 1$).

2.3. Computing sweeps by interpolation and optimization. A nicely ordered sequence of ribs, such as shown by Figure 10 is converted into a surface by interpolation. That surface patch subsequently undergoes optimization. The following paragraphs describe the individual parts of our optimization procedure; for a detailed overview of the entire algorithm see Figure 4.

2.3.1. *Hermite Interpolation of motions.* The design of curves by Hermite interpolation is a standard procedure of geometric modeling, which has been transferred to the design of motions by [HPR04]. The vertices and tangent vectors used for curve interpolation are thereby replaced by positions of rigid bodies and velocity fields, respectively. Recall that such a velocity field has the form given by Equation (2) and is encoded in a vector $\mathbf{C} = \left[\begin{smallmatrix}\mathbf{c}\\ \bar{\mathbf{c}}\end{smallmatrix}\right]$, cf. Equation (1). Figure 11a illustrates that kind of Hermite data. [HPR04] perform "ordinary" Hermite interpolation in the linear space of affine transformations, followed by closest point projection onto the subset of Euclidean motions (using a distance measure constructed from the vertices which move).

2.3.2. *Estimating velocities for Hermite interpolation.* Assume two curves $\mathbf{b}, \mathbf{b}'$ close to a reference surface S. They are sampled uniformly by vertices $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and $\mathbf{x}'_1, \ldots, \mathbf{x}'_n$. We also compute closest point projections $\mathbf{x}_i^*$ of vertices $\mathbf{x}_i$ onto the reference surface S, and the normal vectors $\mathbf{n}_i$ there.

We seek a velocity state $\mathbf{C} = \left[\begin{smallmatrix}\mathbf{c}\\ \bar{\mathbf{c}}\end{smallmatrix}\right]$ belonging to a sweep which after a timestep of magnitude $\Delta t = t' - t$ moves all $\mathbf{x}_i$'s along S, to eventually reach $\mathbf{x}'_i$. This wish is expressed by minimizing the following quadratic function $F(\mathbf{C})$:

$$F_{\mathbf{b},\mathbf{b}'}(\mathbf{C}) = F_{tang} + \mu_1 F_{reg} + \mu_2 F_{pivot}, \quad \text{where} \qquad (7)$$

$$F_{tang} = \sum_i \left(\mathbf{n}_i^\mathsf{T} (\mathbf{x}_i + \Delta t \cdot \dot{\mathbf{x}}_i - \mathbf{x}_i^*)\right)^2, \quad \dot{\mathbf{x}}_i = \bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i,$$

$$F_{reg} = \sum_i \|\mathbf{x}_i + \Delta t \cdot \dot{\mathbf{x}}_i - \mathbf{x}_i^*\|^2,$$

$$F_{pivot} = \sum_i \|\mathbf{x}'_i - \mathbf{x}_i - \Delta t \cdot \dot{\mathbf{x}}_i\|^2.$$

The parts of F have the following meaning: F_{tang} is small if

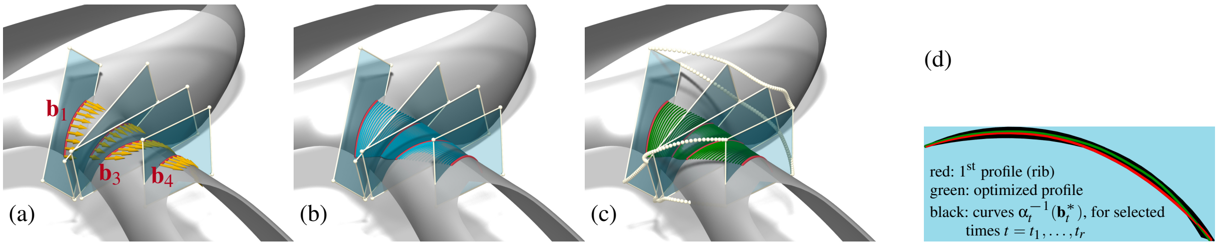


Figure 11: *Generating and optimizing sweeps from a sequence* $\mathbf{b}_1, \ldots, \mathbf{b}_4$ *of ribs. (a) The procedure of §2.3.2 yields velocities for each rib (shown in yellow). (b) These Hermite data define a motion (cf. §2.3.1), which is applied to the profile* $\mathbf{b}_1$ *and generates a sweep surface. (c) One round of profile optimization according to §2.3.3 yields this sweep. (d) Sweeping any planar profile* $\mathbf{b}$ *(here* $\mathbf{b} = \mathbf{b}_1$*) along a surface S does not recreate S exactly. This image visualizes the traces* $\alpha_t^{-1}(\mathbf{b}_t^*)$ *of the reference shape S in the moving plane which carries* $\mathbf{b}$ *before optimization (referring to the notation employed in §2.3.3).*

the vertex $\mathbf{x}_i$ moves tangential to the reference surface. F_{reg} is a regularizer, we chose $\mu_1 = 0.002$. F_{pivot} is small if the velocity state causes $\mathbf{x}_i$ to move close to its partner $\mathbf{x}_i'$. We choose $\mu_2 = 0.1$.

The velocity states $\mathbf{C}_i$ associated with ribs $\{\mathbf{b}_i\}_{i=i_{min}}^{i_{max}}$ are now computed as minimizers of $F_{\mathbf{b}_i,\mathbf{b}_{i-1}} + F_{\mathbf{b}_i,\mathbf{b}_{i+1}}$ (two summands occur for all indices except for $i = i_{min}$ and $i = i_{max}$). The time t_i associated with ribs $\mathbf{b}_i$ is chosen as $t_i = i$. We proceed with C^1 Hermite interpolation, using the method of [HPR04] as a black box (see Fig. 11a and Fig. 11b).

2.3.3. *Improving proto-profiles.* Having obtained (by Hermite interpolation) a continuous motion of an initial profile curve $\mathbf{b}$ and a sweep surface, we now compare that sweep with the reference shape S — see Figure 11. The position of the curve at time t is denoted by $\alpha_t(\mathbf{b})$, where α_t is a rigid body transformation. If S were generated by that profile in an exact manner, then all curves $\alpha_t(\mathbf{b})$ would lie in S. Computing the intersection (temporarily denoted by $\mathbf{b}_t^*$) of their plane with S would yield $\alpha_t(\mathbf{b})$ itself. Since this is in general not the case, we *improve* the profile $\mathbf{b}$ by replacing it by a common approximant of the trace curves $\alpha_t^{-1}(\mathbf{b}_t^*)$. Figure 11 displays such curves and their common approximant. To numerically perform this procedure we compute the intersection curves $\mathbf{b}_t^*$ for a finite number (say, 8 between each rib pair) of intermediate values t and use the tangent-distance minimization method introduced by [BI98] for approximation.

2.3.4. *Checking the sweep surface for a good fit.* Hermite interpolation (§2.3.2) and improving the profile (§2.3.3) are performed in an alternating way, until the resulting sweep surface is close enough to the reference shape S. We check this condition by means of the intermediate profiles $\alpha_t(\mathbf{b})$ which occurred in §2.3.3. The bounding box of S having diameter 1, we declare a sweep to be good enough if its distance to S does not exceed a certain threshold ε_{prox}, which typically is in the range of 0.001 to 0.01. Having now collected all ingredients of our algorithm, let us refer to Figure 4 again which depicts how they are called in succession.

3. Results and discussion

Applications. *Application: Surface analysis.* Fig. 12 shows our analysis of a freeform architectural design ('Skipper library"). One can see that the convex nature of this design allows us to cover it almost completely with sweep patches. Similarly, Figure 13 shows results for the "Yas Marina Hotel", Abu Dhabi, and the top of the proposed "Lilium Tower", Warszawa. Here a complete covering would require us to use rather small patches. Fig. 20 (Heydar Aliev Cultural center, Baku) exhibits similar properties.

Application: Digital Surface Reconstruction. Fig. 14 demonstrates how we digitally reconstruct a given sweep surface which is given as a mesh S (see Fig. 7 for a similar example). We here do not run our entire algorithm, but visualize only the search for movable planes. One can clearly see that up to sampling density in plane space, and apart from some spurious false positives, we reconstruct exactly the planes which correspond to the original motion generating S.

Application: Analysis of CAD models. Since the form of ma-

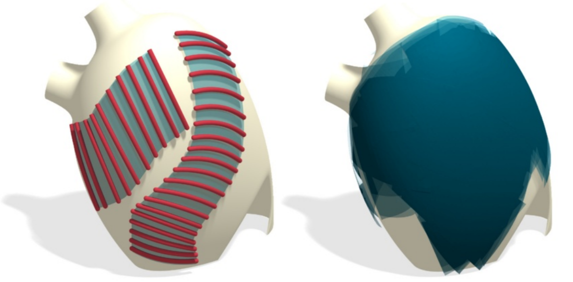
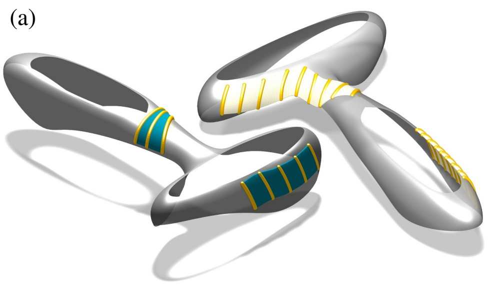


Figure 12: *Analyzing the 'Skipper library' design by Formtexx. The left hand figure shows the largest patches found, while the right hand figure shows all 119 patches detected by our exhaustive search which lie closer to the reference geometry than 0.001 times its bounding box diameter, and which are large enough not to fall below the algorithm's size threshold.*



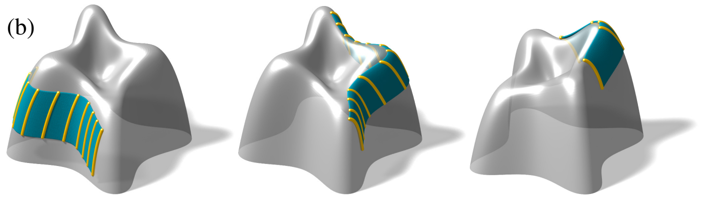


Figure 13: *Here we show several of the larger sweep patches we detected on the skin of architectural designs. (a) Yas Marina Hotel, Abu Dhabi, 2009, by Asymptote Architecture. (b) Top of Lilium Tower (to be constructed in Warszawa; by Zaha Hadid Architects).*

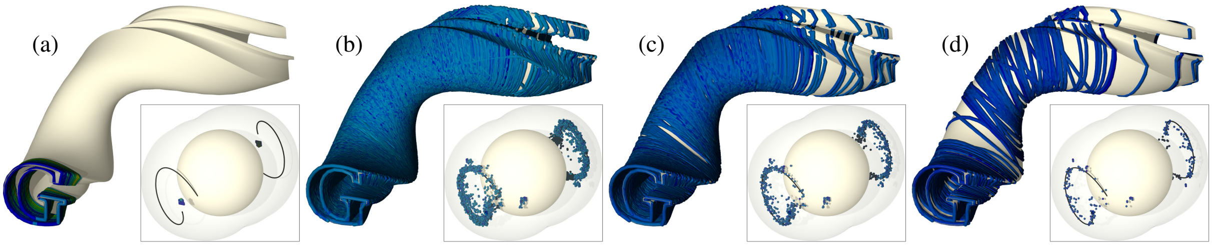


Figure 14: *Digital reconstruction of a sweep surface S. S is generated by sweeping a profile in the shape of the letter G. In plane space (see insets), the continuum of planes carrying the generating profile appears as a curve. Subfigures (b)–(d) illustrate the 'movable' curves and their planes found by the analysis procedure of §1.3. The individual subfigures correspond to different gliding energy thresholds. Apart from some spurious sections which accidentally are movable, we basically reconstruct the exactly movable ones.*

Reference shape		pre-	Initializing planes				Matching curves				Optimizing sweeps			Total time [sec]		
Fig. No.	#vertices	proc	N_{ini}	g [%]	# iter	N_{good}	α_{match}	N_{match}	F_{match}	ε_{match}	N_{opt}	ε_{prox}	N_{sweeps}	T_{ini}	T_{match}	T_{opt}
7a	5k	0.1	2420	5	1	206								8		
7e	”	”	”	5	2	667								21		
12	32k	0.6	1452	1	3	142	25	167	536	.01	152	.001	119	286	234	86
13a	5k	0.01	310	5	3	137	60	241	637	.1	310	.01	49	17	184	57
13b	82k	1.6	310	5	3	164	60	162	342	.1	185	.005	122	97	854	152
14b	10k	0.3	1936	5	3	4002								78		
14c	”	”	”	3	3	574								78		
14d	”	”	”	1	3	194								78		
16a	9k	0.15	665	5	3	151	25	166	312	.01	194	0.01	112	63	212	68
16e	.4k	0.01	312	5	3	196	25	229	471	.01	356	0.01	263	14	241	87
17	5k	0.1	176	1	3	380	45	403	814	.01	612	0.005	315	52	529	72
18a	200k	15	496	5	3	132	25	144	724	.01	516	.001	292	172	312	962
18b	”	”	114	5	3	84	25	95	300	.01	211	.001	138	144	148	423
18e	414k	10	186	5	3	76	25	87	291	.01	229	.01	216	342	671	582
18d, 20	”	”	”	”	”	”	”	”	”	”	”	.05	111	”	”	624
18c	”	”	”	”	”	”	”	”	”	”	”	.001	36	”	”	871
19a	10k	0.3	1210	1	3	1720								42		

Figure 15: *Statistics and timings for the examples shown in the paper. The first columns show the number of vertices in the mesh used to represent the reference shape, and the amount of time spent preprocessing. • The "initialization" columns show the number N_{ini} of planes we cut the reference shape with initially; and the number N_{good} of good planes taken away to the next step in the algorithm. Goodness is decided by the value of the gliding energy being in the lowest g-th percentile. Further we show the number of iterations used for adaptive denser sampling of plane space. • The "matching" columns show the angle threshold α_{match} used for pruning pairs of planes; the number N_{match} of "good" curve segments we perform matching on; the number F_{match} of families of near-congruent curves which were found; and the threshold used in matching: We consider a match φ good enough if $E_{\kappa,\kappa'}(\varphi) \leq \varepsilon_{match} \cdot E_{\kappa,0}(\cdot)$. • The "optimization" columns show the number N_{opt} of initial sweep patches; the distance threshold ε_{prox} for deviation of the sweep from the reference geometry; and the number N_{sweeps} of sweep patches found.*

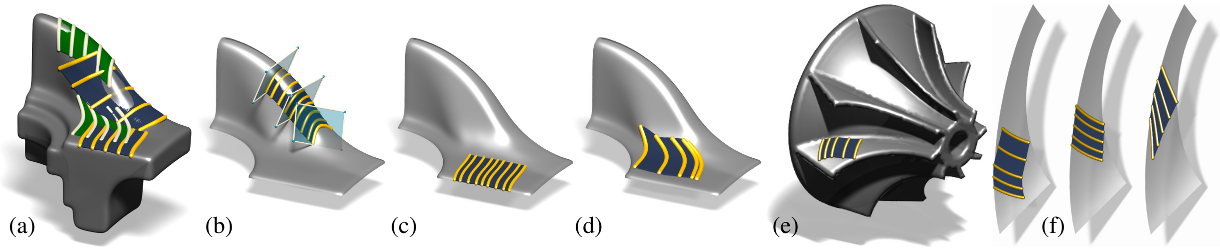


Figure 16: *CAD models like the ones shown in (a), (e) often exhibit simple surface parts which are trimmed away before analysis is applied to the freeform parts of the model (b,c,d,f). The sweep patches we found on the latter are not consistent enough to convince us that sweepability is a design feature here.*

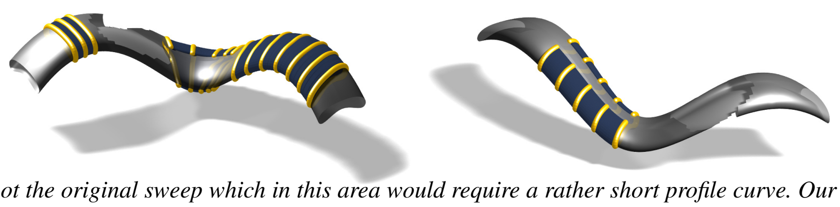


Figure 17: *Analyzing incomplete data. Here our algorithm is applied to a sweep surface S which has a big hole in it. In the undisturbed areas of S, the original sweep is recovered. Near the hole we find a different sweep which approximates S within tolerance, and not the original sweep which in this area would require a rather short profile curve. Our algorithm has not been extended to handle closed profiles, as evident from the right hand figure.*

chine parts often must match their function, CAD models usually exhibit simple kinds of surfaces, e.g. parts which are planar or which have rotational symmetry. Figures 16a and 16e show two examples, where we trim away the simple surface parts and apply sweepability analysis to the rest.

Stability. We have tried to verify the veracity and stability of our approach by various means:

- ⋆ Varying the proximity threshold between the values 0.01 and 0.001 of the bounding box diameter yields information on stability w.r.t. to tight proximity bounds when checking the fit of sweeping patches in §2.3.4. See Fig. 18c–e for such a test.
- ⋆ Varying the number of planes used in the exhaustive search for “movable” cuts in §1.3 yields information on stability w.r.t. coarse sampling of plane space. This information is valuable when one wants to quickly assess the sweepability of a reference shape in seconds rather than minutes. See Fig. 18a,b for such a test.
- ⋆ Running the algorithm on a reference shape which is already known to be generated by a planar profile yields information on the fundamental question of false negatives. In our experience, existing sweeps are detected in a satisfactory manner. Figures 7 and 14 show examples of such tests. Fig. 17 illustrates missing data on an otherwise exact sweep.
- ⋆ Running the algorithm on a reference shape which is generated by a planar profile up to local deformation yields information on how stable the “sweepability” property is. Such a deformation series is shown by Fig. 19.

Convergence. One can prove that any given surface S can be completely covered (approximately) by sweep patches. That covering is far from optimal but can be described explicitly, and its existence implies that our algorithm is able to completely cover S if we just increase the sampling density and decrease the minimum permitted length of profile curves. We construct such a covering, namely by strips of predefined length but possibly small width: we move circular arcs such that they are in second order contact with S along arbitrarily chosen paths. By Meusnier’s theorem [dC76] we only have to choose the circle radius sufficiently small.

Implementation details. At the moment only an academic implemention of the algorithm is available which requires the user to set parameters. Statistics concerning the number of variables, of samples, of results, of iterations are given in Fig. 15, together with various thresholds. It is not surprising that the difficult problem of recognizing ‘sweeping’ parts of a given shape is time-consuming. The most complex shape contained in the table required half an hour for computation. Timings refer to a desktop PC with a 3.33 GHz processor.

Since we do not know of other work which solves the same problem, we found it hard to do a meaningful comparison with other work.

Limitations. The main limitations of our algorithm lie in its complexity and the time-consuming nature of individual steps. Since the problem of detecing sweep surfaces is known to be difficult, this is hardly suprising. However, standard procedures of 3D reconstruction [VM02], like trimming away simple surface parts, and segmentation along sharp edges, will typically leave only smaller freeform parts for inspection by our algorithm.

When applying the algorithm to shapes which are appar-

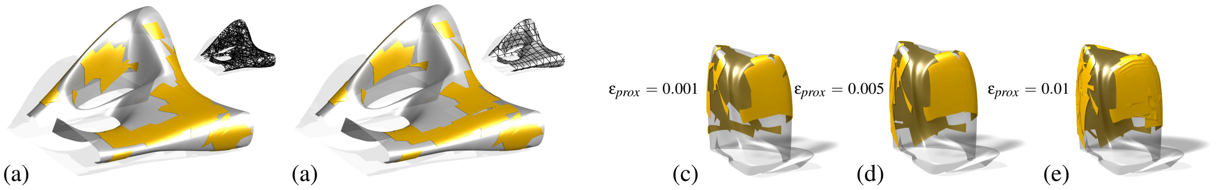

(a)
(a)
$\varepsilon_{prox} = 0.001$
(c)
$\varepsilon_{prox} = 0.005$
(d)
$\varepsilon_{prox} = 0.01$
(e)

Figure 18: *Stability of our algorithm. Subfigures (a), (b) illustrate the effect of thinning set of planar sections on the number of sweep patches found. This architectural design is part of the Heydar Aliev cultural center built in Baku by Zaha Hadid Architects. Subfigures (c)–(e) illustrate how we improve coverage by sweep patches via lowering the proximity threshold ε_{prox}. This architectural design was proposed for the National Holding Headquarters, Abu Dhabi. In both cases, the change in coverage is not as big as the change in the intial algorithm parameters might suggest, which implies good stability of the method.*

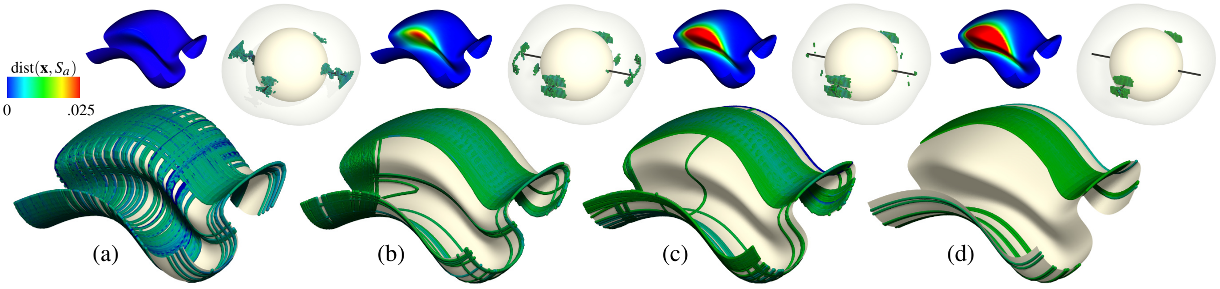


Figure 19: *Behaviour of the sweepability property under deformation. This series of images demonstrates reconstruction of a double-sweep surface S_a which is locally deformed into surfaces S_b, S_c, S_d. The amount of deformation is illustrated by inset figures which show the deviation from S_a (S_a has bounding box diameter 1). (a) The reference shape S_a is in parametric form described by $\mathbf{x}(u,v) = \mathbf{b}(u) + \mathbf{b}'(v)$. It therefore is doubly a sweep: The profile $\mathbf{b}$ moves by translations along the profile $\mathbf{b}'$, likewise $\mathbf{b}'$ moves along the profile $\mathbf{b}$. The planes carrying the various copies of $\mathbf{b}$ appear as a straight line in plane space, and so do the planes carrying the various copies of $\mathbf{b}'$. These planes are detected by our algorithm, up to sampling density of plane space. Subfigures (b)–(d) illustrate the effect of deformation, by analyzing the surfaces S_b, S_c, S_d in the same way. The best cuts (most movable cuts, actually the 1-st percentile according to $E_{gliding}$) are shown, proving that the profiles not affected by the deformation are still reconstructed.*

ently not smooth, the user has two choices: One is to consider the small features of the given shape as intentional and run the algorithm with an appropriately dense set of initial planes which is able to capture all details. The other one is to consider the small features as noise, and make sure that partial matching is based on a similarity measure which is robust w.r.t. noise.

We should remark here that the apparently much less than full coverage of shapes by sweep patches – as evident from various figures – does not indicate a limitation of the algorithm, but rather a limitation of the shapes under investigation: Our algorithm will cover the remaining parts of the reference shape if the size thresholds in our algorithm are set lower (cf. remarks on convergence).

Conclusion. We have presented an algorithmic approach to a difficult geometric problem for which so far no analytic solution has been presented, namely, detecting if a surface can be represented by sweeping of a planar profile, or at least approximately so. Our method successfully solves this problem. It relies on a coarse exhaustive search in the space of planes, combined with adaptive refinement and pruning, an intelligent method of partial curve matching, and optimization of sweep patches.

In summary we believe that we have presented a substantial contribution to the understanding of shapes: This topic is interesting from the purely geometric point of view, since we are now able to determine if a shape is genuinely two-dimensional, or it is in fact simpler and can be defined by one-dimensional entities alone. The topic is also interesting from the viewpoint of applications, in particular manufacturing, and underconstructions in freeform architecture.

Future research. One direction of future research is an extension of our algorithms to the case of non-planar profiles. Unfortunately at the moment this seems out of reach. Another direction is to establish connections with NC milling, aiming at tool motions which can generate large parts of a surface in a single sweep.

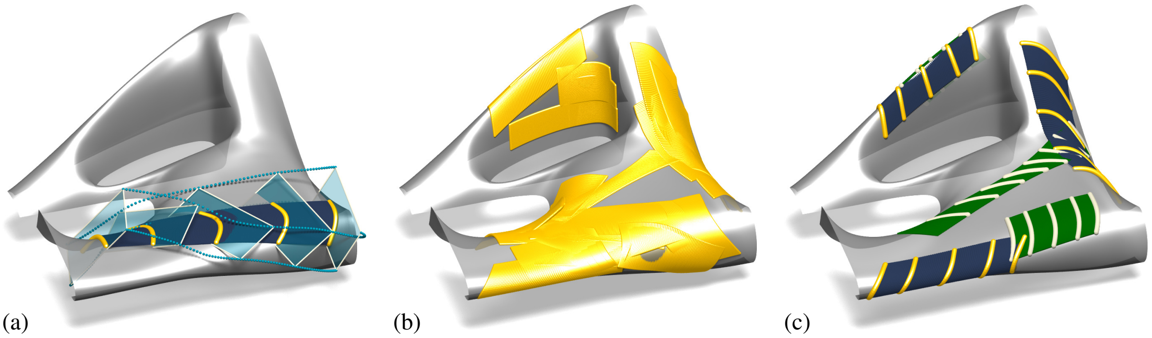


Figure 20: *Finding sweep patches in the skin of the Heydar Aliev cultural center, Baku. (a) For one large detected patch, the correspondong motion and planar profile is shown in detail. (b) Partial covering by sweep patches. (c) Selected larger patches together with the profiles they are generated from.*

Acknowledgments This research has been supported by the European Community's 7th Framework Programme under grant agreement 286426 (*GEMS*).

References

[AS13] Andrews J., Séquin C. H.: Generalized, basis-independent kinematic surface fitting. *Computer-Aided Design 45*, 3 (2013), 615–620. doi:10.1016/j.cad.2012.10.047.

[BB82] Ballard D. H., Brown C. M.: *Computer Vision*. Prentice Hall, New Jersey, USA, 1982. Downloadable from homepages.inf.ed.ac.uk.

[BI98] Blake A., Isard M.: *Active Contours*. Springer, 1998. doi:10.1007/978-1-4471-1555-7.

[BPK*11] Bo P., Pottmann H., Kilian M., Wang W., Wallner J.: Circular arc structures. *ACM Trans. Graphics 30* (2011), 101:1–101:11. Proc. SIGGRAPH. doi:10.1145/1964921.1964996.

[BSK*13] Bartoň M., Shi L., Kilian M., Wallner J., Pottmann H.: Circular arc snakes and kinematic surface generation. *Computer Graphics Forum 32*, 2 (2013), 1–10. Proc. Eurographics. doi:10.1111/cgf.12020.

[CEY97] Cohen S., Elber G., Yehuda R. B.: Matching of freeform curves. *Computer Aided Design 29*, 5 (1997), 369–378. doi:10.1016/S0010-4485(96)00075-9.

[CLRS05] Cormen T. H., Leiserson C. E., Rivest R. L., Stein C.: *Introduction to Algorithms*. The MIT Press, 2005.

[CP99] Chen H. Y., Pottmann H.: Approximation by ruled surfaces. *J. Comput. Appl. Math. 102* (1999), 143–156. doi:10.1016/S0377-0427(98)00212-X.

[dC76] do Carmo M. P.: *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[DK09] Dallinger S., Kollegger J.: Pneumatic formwork for concrete and ice shells. In *Int. Conf. Textile Composites & Inflatable Structures*, Kröplin B., Oñate E., (Eds.). CIMNE, Barcelona, 2009. Downloadable from fabricforming.org.

[EW13] Elber G., Wang C.: Multi-dimensional dynamic programming in ruled surface fitting, 2013. technical report.

[FP10] Flöry S., Pottmann H.: Ruled surfaces for rationalization and design in architecture. In *LIFE in:formation. On Responsive Information and Variations in Architecture*. 2010, pp. 103–109. Proc. ACADIA. downloadable from cumincad.architecturez.net.

[GG04] Gelfand N., Guibas L.: Shape segmentation using local slippage analysis. In *Symp. Geom. Processing*. 2004, pp. 219–228. doi:10.2312/SGP/SGP04/219-228.

[HOP*05] Hofer M., Odehnal B., Pottmann H., Steiner T., Wallner J.: 3D shape recognition and reconstruction based on line element geometry. In *10th IEEE Int. Conf. on Computer Vision* (2005), vol. 2, pp. 1532–1538. doi:10.1109/ICCV.2005.2.

[HPR04] Hofer M., Pottmann H., Ravani B.: From curve design algorithms to the design of rigid body motions. *The Visual Computer 20* (2004), 279–297. doi:10.1007/s00371-003-0221-3.

[KV13] Kovács I., Várady T.: Reconstructing swept surfaces from measured data. In *The Mathematics of Surfaces XIV*, Cripps R. et al., (Eds.). IMA, 2013, pp. 327–344.

[PCL98] Pottmann H., Chen H.-Y., Lee I. K.: Approximation by profile surfaces. In *The Mathematics of Surfaces VIII*, Cripps R., (Ed.). 1998, pp. 17–36.

[PHOW04] Pottmann H., Hofer M., Odehnal B., Wallner J.: Line geometry for 3D shape understanding and reconstruction. In *Computer Vision — ECCV 2004, Part I*. Springer, 2004, pp. 297–309. doi:10.1007/b97865.

[PW01] Pottmann H., Wallner J.: *Computational Line Geometry*. Springer, Heidelberg, 2001. doi:10.1007/978-3-642-04018-4.

[PWHY09] Pottmann H., Wallner J., Huang Q., Yang Y.-L.: Integral invariants for robust geometry processing. *Comput. Aided Geom. Design 26* (2009), 37–60. doi:10.1016/j.cagd.2008.01.002.

[VM02] Várady T., Martin R. R.: Reverse engineering. In *Handbook of Computer Aided Geometric Design*, Farin G., Hoschek J., Kim M. S., (Eds.). Elsevier, 2002, ch. 26. doi:10.1016/B978-044451104-1/50027-7.

[VWB11] Veenendaal D., West M., Block P.: History and overview of fabric formwork: using fabrics for concrete casting. *Structural Concrete 12*, 3 (2011), 164–177. doi:10.1002/suco.201100014.