Deformation by discrete elastica

Flavio Bertini
Department of Computer Science and Engineering
University of Bologna
Mura Anteo Zamboni 7
40126 Bologna, Italy
fbertini@cs.unibo.it

Serena Morigi
Department of Mathematics
University of Bologna
P.zza Porta San Donato 5
40126 Bologna, Italy
serena.morigi@unibo.it

ABSTRACT

Modeling a physically plausible deformation of a flexible object from its naturally planar (curved) state is a fundamental and challenging topic in digital surface processing with applications to computer animation and game design. We propose a new variational model for detail preserving surface-based deformation of a body based on total curvature energy. We demonstrate the efficacy of the model with several examples which enhance the realism of the deformation.

Keywords

Mesh deformation, total curvature, surface processing, elasticity.

1 INTRODUCTION

The great interest in computer animation and game design has led over time to the development of a large amount of deformation and editing methodologies for discrete models [12]. Surface based-deformation methods have been recently widely investigated and represent an emerging alternative both to *rigging* (i.e., adding a skeleton to control and animate a mesh) which is the classical way in the graphics industry to efficiently design poses and deformation, and *caging* which uses control structure like lattices to immerse the object and propagate the deformation.

In this paper we propose a new variational surface-based deformation formulation which improves traditional Laplacian surface editing by using the total curvature as a better aesthetic measure for deformation of elastic bodies.

The basic idea of the variational design approach is to measure the quality of a surface in terms of a certain curvature-based energy. In general, a curvature energy may be expressed in terms of principal curvatures of a surface: mean ($H = k_1 + k_2$), Gaussian ($K = k_1 k_2$), and total ($k_1^2 + k_2^2$) curvature. The relationship between curvature and bending energy first appeared in an explicit form in Euler's elastica, curves minimizing the integral of squared curvature.

Let us denote by $\mathscr{M}$ a two-manifold surface, parameterized by a function $X : \Omega \subset \mathbb{R}^2 \to \mathscr{M} \subset \mathbb{R}^3$, where Ω is an open reference domain.

Thin flexible structures are governed by a surface bending energy of the form

$$E(\mathscr{M}) := \int_{\mathscr{M}} \alpha + \beta(H - H_0)^2 - \gamma K d\mathscr{M}, \quad (1)$$

the so-called Canham-Helfrich model [14], where H_0 denotes the spontaneous curvature which plays an important role in thin-shell. For $\alpha = H_0 = 0$ and $\beta = 1, \gamma = 2$, the Canham-Helfrich model reduces to the **total curvature energy**

$$E_T(\mathscr{M}) := \int_{\mathscr{M}} (H^2 - 2K) d\mathscr{M} = \int_{\mathscr{M}} (k_1^2 + k_2^2) d\mathscr{M}, \quad (2)$$

which approximates the bending energy of a thin plate manifold. In (2) the principal curvatures k_1 and k_2 depend non-linearly on the surface $\mathscr{M}$. Let us call the surfaces minimizing (1) elastica surfaces because they generalize the famous Euler's elastica curves. This energy is invariant under rigid motions and uniform scaling of the surface $\mathscr{M}$, that is, it is invariant under Möbious transformations.

In (2) the first term ($E_B(\mathscr{M}) := \int_{\mathscr{M}} H^2$) represents the well-known Willmore energy [33], successfully used in surface fairing and restorations [9].

According to the Gauss-Bonnet Theorem from differential geometry [11], the integral over a disk region

 ISBN 978-80-86943-74-9

$\mathscr{M} \subset \widetilde{\mathscr{M}}$ can be expressed as an arclength integral over the boundary $\partial\mathscr{M}$ as follows

$$\int_{\mathscr{M}} K d\mathscr{M} = 2\pi - \oint_{\partial\mathscr{M}} k_g ds, \tag{3}$$

where k_g is the geodesic curvature of the boundary curve $\partial\mathscr{M}$. This implies that the Gaussian curvature of $\mathscr{M}$ depends only on a collar neighborhood of $\partial\mathscr{M}$: if we make any modification to $\mathscr{M}$ away from the boundary, the total curvature is unchanged (as long as $\mathscr{M}$ remains topologically a disk).

That is, the second term in (2) remains constant on surfaces with fixed boundaries and fixed normals on the boundary. This is the main reason why the surface-based deformation strategies proposed so far limit the deformation model to the Willmore energy.

However, when the deformation acts on some interior region of a closed compact manifold $\widetilde{\mathscr{M}}$, then $\mathscr{M}$ has fixed boundaries but changing normals, and for an open manifold the deformation region can even have free boundaries and normals. Moreover, a deformation can easily change even the topology of the surface, thus changing the Gaussian curvature of it. These observations are the major motivations for the work presented here.

Therefore, the second term in (2) cannot be neglected in the energy optimization process, and only including it in the deformation process will allow us to handle a wider class of free-form deformations.

Moreover, both the energy $E_T(\mathscr{M})$ and $E_B(\mathscr{M})$ are invariants under all Möbious transformations (in particular, rigid motions and uniform scaling of the surfaces). In extending the energies from smooth surfaces to the discrete case (polyhedral surfaces) we will require this property to remain true.

In this paper we are interested in modeling a physically plausible deformation of a flexible object from its naturally planar (curved) state to an equilibrium configuration due to external localized forces interactively applied by the user (by imposing geometrical constraints). As energy of deformation associated with the elastically deformable object we consider the total curvature functional for the justification given above. This energy is employed to define the internal elastic force of the object, expressed as δE, a first variational derivative of the potential energy of deformation. To deal effectively with boundaries we introduce appropriate boundary conditions which guarantee, when it is possible, to satisfy G^1 continuity conditions at the boundary of the domain.

The rest of the paper is organized as follows: in section 2, some related work on mesh deformation is presented and in section 3, we discuss some potential energies of deformations and a new variational deformation model which includes the effect of the total curvature. Linear and nonlinear deformation constraints are described in section 4. In section 5, the details on the discretization are given, and we present the deformation algorithm based on the iterative alternating strategy. The effect of the total curvature energy and other examples are discussed in section 6. Some limits and concluding remarks are made in section 7.

2 RELATED WORKS

Deformable curve, surface, and solid models gained popularity in computer vision and computer graphics after they were proposed in the mid 1980s by Terzopoulos et al. [30]. Terzopoulos introduced the theory of continuous (multidimensional) deformable models in a Lagrangian dynamics setting, based on deformation energies in the form of (controlled-continuity) generalized splines [31]. Since then, many results have been presented in that direction. Nowadays, most existing shape deformation approaches can be classified as *space deformation* and *surface deformation* according to the way in which they act on the object to be deformed [4]. Space deformation techniques modify objects by deforming their embedded space, while surface-based methods define the deformation directly on the surface of the object.

All space deformation methods use some form of control structure like a lattice to immerse the object then every structure deformation is propagated to the object itself. The robustness and the efficiency of these methods are strongly affected by the control structure complexity even if they are less affected by the complexity or the triangles' quality of the original surface [27]. In *Free-Form Space Deformation* points of the object are expressed as a linear combination of the structure control points and blended with some different bases functions: Bézier [23], B-spline [13] [17], T-splines [27] and many others. The unnaturalness of the deform through a control structure has led to the development of the *Direct Manipulation FFD* [15]: the user directly moves the object points and the system computes the control point displacements. The *Radial Basis Functions*-based approach proposed in [3] is equally innovative, it improves upon FFD and DMFFD due to its handle point nature and the deformation function physically used.
Nonlinear methods are presented in [28], where an energy functional optimizes the local deformation gradients, and in [6], where the object is voxelized and the deformation is driven by a nonlinear elastic energy.

There are a plenty of different surface deformation methods, well summarized in the detailed survey [7]. The *Transformation Propagation* linearly propagates the deformation within a region and the main challenge is how to define the propagation function (e.g. using

 ISBN 978-80-86943-74-9

geodesic distances [2] or Euclidean distances [22]). The *Shell-Based Deformation* techniques minimize two physically-inspired deformation energies: stretching and bending [32]. The *Multi-scale Deformation* [8] mainly decomposes the object into two frequency bands: high frequency for details and low frequency for global shape. Two detail-preserving techniques have been proposed as surface deformation methods based on differential coordinates: the *Gradient-Based Deformation* [35] and the *Laplacian-Based Deformation* [26],[18]. The former uses the original surface gradients as target in the least-squares sense for the deformed surface, the latter is similar but it uses the Laplacian operator on vertices.
The linear methodologies can be solved very efficiently but can lead to counterintuitive results for large-scale deformations, they have obvious limitations and in some circumstances they even fail. Nonlinear deformation techniques overcome these limitations, but they require more complex numerical schemes [4].
Typically, some constraints are added to improve quality results: *Pyramid coordinates* [24], handle-aware isoline technique [1], volumetric graph Laplacian [16], skeleton-based inverse kinematics [25] and shell-based minimization coupled by a nonlinear elastic energy [5].

Another important issue is the deformation metaphor which concerns the manner in which the user defines the deformation: by *handle point deformation* the user moves some "handle" points of the object, by *curve-based deformation* the user imposes deformation by sketching curves and by *control point deformation* the user manages the object in an indirect way [12].

3 ENERGIES OF DEFORMATION

The deformation of a nonrigid body is a change in shape or size of an object due to applied forces: pulling or pushing (compressive) forces, shear, bending or torsion (twisting). A deformation is termed *elastic* if the undeformed or reference shape restores itself completely, upon removal of all external forces, *inelastic* otherwise. A pioneer work in modeling inelastic deformations simulating behaviors such as viscoelasticity, plasticity, and fracture, has been proposed in [30] for use in computer graphics animation of nonrigid objects. More advanced physically-based PDE models are considered in [21].

In this work, we focus on elastic deformations of a non-rigid model during the deformation phase neglecting the phase responsible for recovering the reference shape.

Let $\mathscr{M}$ be a fixed reference surface, parameterized by a function $X : \Omega \subset \mathbb{R}^2 \to \mathscr{M} \subset \mathbb{R}^3$. A deformation is a function d that maps $\mathscr{M}$ to a certain deformed model $\mathscr{M}'$, by adding to each point $X(u,v) \in \mathscr{M}$ a displacement vector $d(u,v)$, such that $\mathscr{M}' = X'(\Omega), X' = X + d$.

A reasonable approximation for elastic thin-shell energy which measures stretching and bending is the following (see [4] for details)

$$\int_\Omega k_s \|I' - I\|^2 + k_b \|II' - II\|^2 du dv, \tag{4}$$

where I (I') and II (II') represent the first and second fundamental forms for $\mathscr{M}$ ($\mathscr{M}'$), k_s and k_b weight the matrix norms and determine resistance to stretching and bending, respectively. The matrix norms in this function make it highly nonlinear, leading to a difficult nonlinear optimization problem. It is therefore common to simplify (linearize) this objective function by replacing the change of the first and second fundamental forms by the first-order and second-order partial derivatives of the displacement function d [7].

The (simplified) **thin plate energy** is given by:

$$E_B(d) = \frac{1}{2} \int_\Omega k_b (\|d_{uu}\|^2 + 2\|d_{uv}\|^2 + \|d_{vv}\|^2) du dv. \tag{5}$$

The stretching or **membrane energy** is defined by the functional

$$E_M(d) = \frac{1}{2} \int_\Omega k_s (\|d_u\|^2 + \|d_v\|^2) du dv. \tag{6}$$

Deformations of non-rigid surfaces (so called *thin-shells*) require both energies, while for a deformable solid only the local stretching within the object is considered.

In order to keep the parametrization of the surface $\mathscr{M}$ as close to isometric as possible, Ω is typically chosen to be equal to the initial surface $\mathscr{M}$, such that $d : \mathscr{M} \to \mathbb{R}^3$ is defined on the manifold $\mathscr{M}$ itself. As a consequence, the Laplace operator $\triangle$ w.r.t. the parametrization X turns into the Laplace-Beltrami operator $\triangle_\mathscr{M}$ w.r.t. the manifold $\mathscr{M}$. Therefore when the parametrization is isometric,

$$E_B(d) \simeq \frac{1}{2} \int_\mathscr{M} k_b \|\triangle_\mathscr{M} d\|^2 d\mathscr{M}, \tag{7}$$

and

$$E_M(d) \simeq \frac{1}{2} \int_\mathscr{M} k_s \|\nabla_\mathscr{M} d\|^2 d\mathscr{M}. \tag{8}$$

The minimization of these functionals, performed efficiently by applying variational calculus, yields to their Euler-Lagrange equations which are

$$\triangle_\mathscr{M}^2 d = 0, \tag{9}$$

for (7), and

$$-\triangle_\mathscr{M} d = 0, \tag{10}$$

for (8), subject to suitable boundary constraints. We denote by E_C the variational deformation proposed in [7]

 ISBN 978-80-86943-74-9

where stretching and bending are combined together, given by

$$-k_s \triangle_{\mathscr{M}} d + k_b \triangle^2_{\mathscr{M}} d = 0. \quad (11)$$

We observe that in surface smoothing similar functionals are applied to X itself instead of their displacements [10].

Note that the linearization in (7) and (8) causes artifacts for large deformations, as we will verify in the result section 6.

We propose instead to minimize the total curvature energy (2), which leads to the Euler-Lagrange equation

$$\triangle_{\mathscr{M}} H(d) - 2H(d)(H^2(d) - K(d)) = 0, \quad (12)$$

subject to natural boundary conditions. Equation (12) is a fourth-order partial differential equation, (the term $\triangle_{\mathscr{M}} H(X)$ involves fourth-order surface derivatives) satisfied for an elastica surface. To be well posed it requires two independent boundary conditions. By natural boundary conditions we mean that no continuity conditions are specified at the boundary points, but the continuity is implied by the "outer" part incident to the boundary of $\mathscr{M}$. The elastica flow has been proposed for surface fairing and repairing in [34]. Note that $\sqrt{H^2 - K}$ is the half-difference of the principal curvatures and also in the discrete setting the property $H^2 - K \geq 0$ has to be guaranteed.

In a modeling application, one can be interested either to a dynamic time dependent simulation, or directly to solve the rest state of the deformation process. For the former, one typically deals with the associated geometric flow $\partial \mathscr{M} / \partial t = -\nabla(E(d))$ by the steepest descent method. The latter means solving the Euler-Lagrange formulation subject to user-defined boundary constraints, which will be discussed in section 4. This typically means to fix certain surface regions $\mathscr{F} \subset \mathscr{M}$, and to define displacements for the so-called handle (target) regions $\mathscr{H} \subset \mathscr{M}$. In an interactive application $\mathscr{M}$ has to be recomputed by solving the PDE each time the user manipulates the boundary constraints, for instance by moving the handle region $\mathscr{H}$.

Considering the deformation energy together with linear or nonlinear constraints represented by a generic $\Phi(\cdot)$, the constrained deformation can be formulated by

$$min_{X'} E(X' - X) \quad \text{subject to} \quad \Phi(X'). \quad (13)$$

4 LINEAR AND NONLINEAR DEFORMATION CONSTRAINTS

First we consider positional constraints that can be either incorporated as **hard** or **soft** constraints. The hard positional constraints are preferred in classical editing tools where the exact position should be achieved, while in a sketch-based system soft constraints are actually advantageous, since they allow the user to place imprecise locations to hint the desire shape, without specifying it exactly.

Representing the coordinate map X by real-valued functions (x, y, z), defined on $\mathscr{M}$, the prescribed target positions C are imposed by the following constraints:

$$\frac{1}{2} \int_{\mathscr{M}} (X' - C)^2 d\mathscr{M}. \quad (14)$$

Intuitive, detail-preserving constraints can be obtained by preserving local differential properties under deformation.

Let $\delta = \triangle_{\mathscr{M}}(X)$, the surface deformation is obtained by

$$\frac{1}{2} \int_{\mathscr{M}} (\triangle_{\mathscr{M}} X' - \delta)^2 d\mathscr{M}. \quad (15)$$

Eq. (15) forces the new position to resemble its undeformed Laplacian as closely as possible, that is, in view of the fact that $\triangle_{\mathscr{M}} X = -H\mathbf{n}$, with $\mathbf{n}$ outward surface normals, it preserves the local curvatures of the undeformed surface. Laplacian deformation methods are mainly based on the minimization of (15), see [26],[7].

This deformation constraint tries to preserve the orientation of the normals w.r.t. the global coordinate system, whereas in reality they should rotate with the deformed surface. This technique as well as the gradient-based deformation [35] fail to yield intuitive results for translational deformations as it clearly shown in section 6, since a translation does not cause a change in surface gradients, or normal vectors. This *translation insensitivity* is an inherent limitation of most approaches based on differential coordinates.

A correct deformation should retain the local surface features, that is their relative orientation and possibly their size. Therefore, several variants to the linear model (15) have been proposed for finding local rotations of the geometric details: some require additional input of the rotation of the handle, others use implicit optimizations, and multiresolution approaches estimate the local transformation from the deformation of the base surface.

Introducing the local transformations T, which can be restricted to rotation and isotropic scaling, the differential representations δ of the rest mesh X are transformed into the deformed pose: $\widehat{\delta} = T\delta$, The deformed positions X' are then obtained by replacing (15) with the following nonlinear term

$$\frac{1}{2} \int_{\mathscr{M}} (\triangle_{\mathscr{M}} X' - \hat{\delta}(X'))^2 d\mathscr{M}. \quad (16)$$

The term $\hat{\delta}(X')$ is a nonlinear function because it includes the effects of local rotations, thus (16) leads to a nonlinear least-squares problem, while (15) leads to a linear least-squares problem. We impose the nonlinear constraints on the set $\mathscr{M} \setminus (\mathscr{F} \cup \mathscr{H})$.

 ISBN 978-80-86943-74-9

5 MODEL DISCRETIZATION

We are interested in evaluating the constrained variational problem involving the curvature energy (2) for discrete surfaces, i.e., triangular meshes M which represent a piecewise-linear approximation of the smooth surfaces $\mathscr{M}$. Let M be defined by a set T of triangles $T_i, i = 1,\ldots,N_t$, that cover M, and a set X of vertices $X_i, i = 1,\ldots,N_v$, where $X_i \in \mathbb{R}^3$ is the ith vertex, $X_i = (x_i, y_i, z_i) \in \mathbb{R}^3$, with associated normal vector $\mathbf{n}_i$.

Let us briefly introduce the key ingredients for the discretization on M of the proposed variational deformation formulation. Since for a smooth surface $\Delta_{\mathscr{M}} = 2H\mathbf{n}$, see [11], a discrete approximation of the mean curvature vector $H_i\mathbf{n}_i$ associated to the vertex X_i can be derived using the following discrete form

$$H_i\mathbf{n}_i = L(X_i) = \frac{1}{2A_i}\sum_{j\in N(i)} w_{ij}(X_j - X_i), \qquad (17)$$

where L represents the discretization of the local Laplace-Beltrami operator $\Delta_{\mathscr{M}}$ on M, $N(i)$ is the set of 1-ring neighbor vertices of vertex X_i, A_i is the Voronoi area surrounding X_i, and the weights w_{ij} are positive numbers which satisfy the normalization condition $\sum_{j\in N(i)} w_{ij} = 1$. Different geometric discretizations of the Laplacian can be obtained for different choices of the weights in (17), the most common and used in our discretization, introduced by Meyer et al. in [19], is

$$w_{ij} = (\cot\alpha_{ij} + \cot\beta_{ij}), \qquad (18)$$

where α_{ij} and β_{ij} are the two angles opposite to the edge in the two triangles sharing the edge (X_j, X_i).

The notion of Gaussian curvature extends to such discrete surface M and is supported on the vertices $X_i \in X$. In fact, to keep the Gauss-Bonnet theorem true, we must take

$$\int_{\mathscr{M}} K d\mathscr{M} := \sum_i K_i, \quad K_i = \frac{1}{A_i}(2\pi - \sum_{j\in N(i)} \theta_j), \qquad (19)$$

where θ_i are the incident internal angles at X_i. The simple formula (19) is a standard for triangular meshes [29].

Considering that the displacement vector is $d = X' - X$, then the deformation energy models on the surface represented by the mesh M can be discretized as follows:

E_M (10)	$Ld = 0$	$LX' = LX$
E_B (9)	$L^2 d = 0$	$L^T LX' = L^T LX$
E_C (11)	$k_s Ld + k_b L^2 d = 0$	$(k_s L + k_b L^2)X' = (k_s L + k_b L^2)X$
E_T (12)	$L^2 d - 2LGd = 0$ $G = H^2 - K$	$(L^2 - 2LG)X' = (L^2 - 2LG)X$

Each of these models leads to a generic linear system $AX = b$ with a sparse $N_v \times N_v$ coefficient matrix. Note that, in general, the evaluation of $\Delta_{\mathscr{M}} H$ at X_i (for X_i either being an inner, that is $X_i \in X\backslash(\mathscr{F}\bigcup\mathscr{H})$, or an outer vertex, that is $X_i \in \partial\mathscr{M}$) involves 2-ring neighbor vertices of X_i. Some of them may be inner vertices, and the remaining are outer vertices. The inner vertices are treated as unknowns in the discretized equations and the outers are incorporated into the right-hand side.

A way to enforce *soft positional constraints*, is to incorporate them in a penalty formulation of the discrete energy functional

$$\min_{X'} E(X' - X) + \frac{\lambda}{2}\|X' - C\|^2 \qquad (20)$$

where $\lambda > 0 \in \mathbb{R}^n$ is the penalty coefficient, and C is the vector of prescribed vertex positions.

In order to approach the interpolation of the constraints C, the parameter λ has to be chosen sufficiently large. However, the condition number of the matrix grows with λ, then a higher weight can cause numerical problems.

Considering the discretization of (13) with total curvature energy (2), positional constraints (20) and detail-preserving constraints (16), and using the fact that $\hat{\delta}(X')$ are unknowns in the deformation process, we propose the following energy functional minimization to solve the mesh deformation problem:

$$\begin{gathered}\min_{X',\hat{\delta}} \mathscr{L}(X', \hat{\delta}),\\ \mathscr{L}(X', \hat{\delta}) := E(X' - X) + \tfrac{\lambda_1}{2}\|X' - C\|^2 + \tfrac{\lambda_2}{2}\|LX' - \hat{\delta}\|^2.\end{gathered} \qquad (21)$$

The minimum of (21) can be determined by the alternating minimization procedure, namely, for $k = 0, 1, \ldots$, we solve successively

$$\begin{aligned}\hat{\delta}^{(k+1)} &= \operatorname{argmin}_{\hat{\delta}} \mathscr{L}(X'^{(k)}, \hat{\delta})\\ X'^{(k+1)} &= \operatorname{argmin}_{X'} \mathscr{L}(X', \hat{\delta}^{(k+1)}).\end{aligned} \qquad (22)$$

Since $\mathscr{L}(X', \hat{\delta}^{(k+1)})$ is continuous differentiable in X', the solution $X'^{(k+1)}$ of the second minimization in (22) is obtained by imposing

$$\begin{aligned}0 = \quad & \nabla_{X'}\mathscr{L}(X', \hat{\delta}^{(k+1)}) =\\ & (L^2 - 2LG)(X' - X) + \lambda_1(X' - C) + \lambda_2(L^T(LX' - \hat{\delta}^{(k+1)})),\end{aligned} \qquad (23)$$

where $G = H^2 - K$.

In matrix-vector form, the solution of (23) for the new mesh vertices X', is given by solving the overdetermined system

$$\begin{bmatrix} (L^2 - 2LG) \\ \mathbf{0} \quad \sqrt{\lambda_1} I_n \\ \mathbf{0} \quad \sqrt{\lambda_2} L^T L \end{bmatrix} X' = \begin{bmatrix} (L^2 - 2LG)X \\ \sqrt{\lambda_1} C \\ \sqrt{\lambda_2} L^T \hat{\delta}^{(k+1)} \end{bmatrix} \qquad (24)$$

where the block $A = (L^2 - 2LG)$ represents the internal energy, $I_n \in \mathbb{R}^{n\times n}$ is the identity matrix which requires

 ISBN 978-80-86943-74-9

a resorting of the rows of L, and $C \in \mathbb{R}^n$ is a vector of elements c_i for each of the n positional constraint. The system has dimension $(N_v + 2n) \times N_v$ and it is full rank, thus it has a unique solution in the least-squares sense. The linear system of equation (24) is solved by the conjugate gradient method where we terminate the iterations as soon as the norm of the residual is less than or equal to 10^{-4}. The use of an iterative solver allows us to avoid storing the large dimension matrices, the only requirement is matrix-vector products.

The use of $L = D\overline{L}$ and $K = D\overline{K}$ with the area-scaling matrix $D_{ii} = 1/A_i$ in (24) allows to extend the invariance under rigid transformations and uniform scaling property of the energy E_T to the discrete setting.

To update $\widehat{\delta}^{(k+1)}$, the first minimization in (22) gives $\hat{\delta}^{(k+1)} = LX'^{(k)}$. In particular, following [16], at each step we use the two-phase procedure:

- (STEP 1) For each vertex X_i with $N(i)$ neighbors, solve for $\mu^i = (\mu^i_1, \mu^i_2, \ldots, \mu^i_{N(i)})$:

$$\sum_{j \in N(i)} \mu^i_j ((X_j - X_i) \otimes (X_{j-1} - X_i)) = \delta_i, \qquad (25)$$

  where δ_i are the Laplacian coordinates before deformation, and $(X_j - X_i) \otimes (X_{j-1} - X_i)$ is the normal vector to the triangle X_i, X_j, X_{j-1} on X_i. The overdetermined linear system (25) can be represented in matrix-vector form as

$$A_i \mu^i = \delta_i \qquad (26)$$

  where A has dimension $3 \times N(i)$ and solved by SVD method.

- (STEP 2) Plug the computed μ^i_j in

$$d_i(X') = \sum_{j \in N(i)} \mu^i_j ((X'_j - X'_i) \otimes (X'_{j-1} - X'_i)), \quad (27)$$

  with X' vertices of the deformed mesh M'. Since the μ^i are the same before and after deformation, $d_i(X) = T_i \delta_i$ for local rotations T_i. Finally the Laplacian coordinates are normalized as follows:

$$\widehat{\delta}(X_i) = \frac{d_i}{\|d_i\|} \|\delta_i\|. \qquad (28)$$

This means that we use the Laplacian coordinates of the previous iterative step as the target direction, while taking the magnitude of the original Laplacian coordinates as the target magnitude. Since the directions to be preserved are those at the rest position, then STEP1 can be done as preliminary step, while STEP2 updates the new Laplacian coordinates at each alternating step k.

In case we want to enforce *hard positional constraints*, that is constrained vertices which lie in the exact prescribed location, we set $\lambda_1 = 0$ in the functional (21), and solve (24) replacing the matrix block A with a reduced matrix $A_n \in \mathbb{R}^{(N_v - n) \times (N_v - n)}$ as follows. Forcing n constraints leads to the elimination in A of the n rows corresponding to the constrained vertices ($X_i \in \mathscr{F} \cup \mathscr{H}$) and moving the corresponding columns to the right-hand side.

We should remark that the original mesh without the constrained vertices $\mathscr{F}$ and $\mathscr{H}$ is a reduced mesh with, in general, more than one connected components. The associated connectivity matrix should be a reduced rank matrix. However, the A_n matrix is not the connectivity matrix of such a reduced mesh since the elements of A_n are the same of the original connectivity matrix, that is computed on the entire mesh.

The final deformation algorithm simply iterates two simple and efficient steps which are respectively responsible for improving the estimation of the local transformations, and vertex positions. At each iteration, fixing X', $\widehat{\delta}^{(k+1)}$ are updated by using STEP2, then the computed approximations for $\widehat{\delta}^{(k+1)}$ are used to solve the linear least-squares problem (24). The algorithm for nonlinear deformation is here summarized:

ALGORITHM
Discrete Elastica Nonlinear Deformation (DEND)
INPUT: undeformed vertex set X,
OUTPUT: deformed vertex set X'
Set $X^{(0)} = X$, $\widehat{\delta}(X^{(0)}) = L(X)$, k=0
STEP 1: Compute μ^i, $\forall X_i \in X$ by (25)
Repeat
  STEP 2: Compute $\widehat{\delta}(X'^{(k+1)})$ by (27) and (28)
  Solve the linear LS problem (24)
  k=k+1
until $\|X'^{(k)} - X'^{(k-1)}\| < 1 \cdot 10^{-3}$

6 DEFORMATION RESULTS

In this section we consider some examples to show how the proposed DEND algorithm performs to deform structured and unstructured polygonal meshes. All the examples have been produced on a standard consumer-level LINUX PC by using the `Meshviz` software, a GUI application for geometric surface processing developed at the University of Bologna, Italy, based on OpenGL graphics library and C language. The current version of `Meshviz` offers interactive tools for experimentation with the proposed deformation algorithm but it does not provide any tangential remeshing feature. This would prevent from degenerated meshes which may adversely affect the deformation performance. A preprocessing step of mesh optimization can alleviate this problem [20]. Moreover, the DEND algorithm has no collision detection which would allow for handling collision occurring between deformed parts of a deformable body. Collision detection is a complex constraint which increases considerably the complexity of

 ISBN 978-80-86943-74-9

the deformation model. The Meshviz editing tools are easy and intuitive and allow the user to interactively select by mouse regular/irregular regions $\mathscr{F}$ that he/she wants to keep fixed so as the areas $\mathscr{H}$ that he/she will drag (rotating, translating) to a target position; the positions of the remaining vertices $X \setminus (\mathscr{F} \cup \mathscr{H})$ will be determined by the DEND algorithm. The computational time of the entire process to set up a new pose for an object depends mainly on the dimension of the free vertex set $X \setminus (\mathscr{F} \cup \mathscr{H})$ which affects the solution of (24) in DEND algorithm. Therefore, for medium size objects like those used for the shown examples the deformations are achieved in real-time, while optimizations of the linear solver will be needed to obtain real-time deformations of larger meshes.

We demonstrate that our approach enables to apply small to large deformations on middle-large detailed meshes while keeping the shape of the details in their natural orientation.

Example 1. In this example we compare the surface-based mesh deformation techniques described in section 3. The DEND Algorithm presented in section 5 has been suitably modified to manage the different energy deformations by changing the block matrix A, and hard/soft constraints, by setting $\lambda_1 = 0$ or $\lambda_1 = 1$, respectively.

The deformations are performed on the original undeformed bar mesh with 856 vertices (Fig.1(a)) and cylinder mesh with 1088 vertices, illustrated in Fig.1 (b). Fig.1, first column, shows the results of 135° twist of the bar mesh, and second column reports the 120° bending of the cylinder mesh. In particular, we apply (21) with the deformation energies E_B (9) (Fig.1(e)-(f)), E_C (11) (Fig.1(g)-(h)), and E_T (12) (Fig.1(m)-(n)), with non-linear hard constraints ($\lambda_1 = 0, \lambda_2 = 1$). For comparison, we also show in Fig.1(i)-(l) the deformation obtained by the PRIMO system [5], courtesy from the author's web page, which present different, but still physically plausible shape deformations. However, the cylinder deformation presents a more concentrate bending in the middle, while using E_T (12) (Fig.1(n)) we get a uniform deformation along the shape. In each figure the red vertices represent the vertices $\mathscr{F}$ that are fixed; the blue vertices are the handle (target) constraints $\mathscr{H}$; the green area contains the remaining unconstrained vertices $X \setminus (\mathscr{F} \cup \mathscr{H})$ whose position is computed by the DEND algorithm. Fig.1(c)-(d) show the resulting deformations by using the thin plate energy E_B defined in (9) applied together with linear constraints (15) instead of nonlinear (16). The results clearly show the weaknesses of the linear deformation approach.

In linear theory the behaviour of the deformable model is physically correct only for small displacements (about 10% of the mesh size), it is less realistic for larger deformations. It is the main disadvantage of linear elasticity. In Fig.1 all the deformations are obtained in a single step. Nevertheless, our deformation procedure by using E_T with non-linear constraints (Fig.1(m)-(n)), provides well-shaped and aesthetically pleasing results. Interactive deformation prevents large deformations, since each step remain reasonably small. Fig.2 illustrates two large deformations obtained by interactively applying 5 steps for a total 300° twisting on the bar model (Fig.2(a)), and 3 steps for a 180° bending on the cylinder mesh (Fig.2(b)). The other considered energies were not been able to produce similar good results.

Example 2. The example illustrated in Fig.3 shows how to apply a simple deformation to transform a plane into a bumpy plane. From an original plane model, shown in Fig.3 (a), the bumpy plane (2115 vertices) in Fig.3 (b) is achieved by anchoring the red vertices and translating the blue vertices. The bumpy plane is then deformed by bending the two sides of the mesh using the DEND Algorithm with E_T.

Example 3. In this example we show the flexibility of our variational deformation model implemented with DEND Algorithm on a range of examples, including both open and closed surfaces representing elastically deformable models.

Fig.4 shows the deformations of a thin plate model, represented by the flag (289 vertices) mesh, whose rest state is flat. Fig.4 illustrates a few snapshots of a flag waving simulation, obtained by translating the right side of the flag mesh and anchoring the left side.

The interactive deformation of a complex hand (1515 vertices) model is illustrated in Fig.5, and a torus twist (576 vertices) is shown in Fig.6.

A sequence of deformations obtained by the DEND algorithm using E_T by free interactive shape-editing steps is shown in Fig.7 on a free bending of dino (10098 vertices) mesh. When dragging the handle vertices, the deformed surface should retain the look of the original surface in a natural way. The smooth regions of the surface should remain smooth, but if the surface contains some geometric details, as in the dinosaur tail and body, the shape and orientation of these details should be preserved. As shown from these preliminary examples, the DEND algorithm allows for a natural detail-preserving deformation.

7 LIMITS AND CONCLUSIONS

The main requirement for physically based surface deformation is an elastic energy that measures how much an object has been deformed from its initial configuration. In this paper a new constrained variational surface-based deformation model is proposed, by exploiting the total curvature as a better aesthetic measure for deformation of elastic bodies.

 ISBN 978-80-86943-74-9

There are a number of crucial requirements on a shape deformation operation which make it a challenging problem: (i) the operation should be efficient enough for interactive work, (ii) it should provide local influence and detail preservation, (iii) the editing operation should naturally change the shape and simultaneously respect the structural detail, (iv) elastic bodies should not self-intersect as they deform.

The proposed variational model satisfies the requirements (ii) and (iii). The current version of the DEND algorithm in `Meshviz` does not allow for great performance, which will require an optimized implementation. Therefore, deformation editing can be achieved in real-time only for medium size objects like those used for the shown examples. Moreover, self intersections can be avoided by surrounding the surface of the object with bounding boxes and using collision dynamics. This is not yet available in our simple deformation software.

Nevertheless, we demonstrated that our deformation model yields realistic effects, while still maintaining computational tractability. Many open directions will be investigated, starting from considering total curvature energies that are minimized by a non-flat rest surface, following the more general energy in (1).

8 REFERENCES

[1] Au O. K.-C., Fu H., Tai C.-L. and Cohen-Or D., *Handle-Aware Isolines for Scalable Shape Editing*, ACM Trans. on Graphics, 26/3, (2007).

[2] Bendels G. H. and Klein R., *Mesh Forging: Editing of 3D-Meshes Using Implicitly Defined Occluders*, In Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, 207–217 (2003).

[3] Botsch M. and Kobbelt L., *Real-time shape editing using radial basis functions*. Computer Graphics Forum (Proc. Eurographics), Vol.24/3, 611–621 (2005).

[4] Botsch M., Kobbelt L., Pauly M., Alliez P., and Levy B., *Polygon Mesh Processing*. AK Peters - 2010.

[5] Botsch M., Pauly M., Gross M. and Kobbelt L., *PriMo: Coupled Prisms for Intuitive Surface Modeling*. In Proc. of Eurographics symposium on Geometry Processing, 11–20 (2006).

[6] Botsch M., Pauly M., Wicke M. and Gross M., *Adaptive Space Deformations Based on Rigid Cells*, Computer Graphics Forum, Vol. 26/3, 339–347 (2007).

[7] Botsch M. and Olga Sorkine O., *On Linear Variational Surface Deformation Methods*, IEEE Trans. on Visualization and Computer Graphics, 14/1, 213–230 (2008).

[8] Botsch M., Sumner R., Pauly M. and Gross M., *Deformation Transfer for Detail-Preserving Surface Editing*. In Proc. Vision, Modeling and Visualization, 357–364 (2006).

[9] Clarenz U., U. Diewald, G. Dziuk, M. Rumpf, R. Rusu, *A finite element method for surface restoration with smooth boundary conditions*, Computer Aided Geometric Design **21(5)**, 427–445 (2004).

[10] Desbrun M., M. Meyer, P. Schroeder and A. Barr, *Implicit fairing of Irregular meshes using diffusion and curvature flow*, Computer Graphics (SIGGRAPH '99 Proceedings), 317–324 (1999).

[11] do Carmo M.P., *Riemannian Geometry*, Birkhauser, Boston-Basel-Berlin, 1993.

[12] Gain J. & Bechmann D., *A survey of spatial deformation from a user-centered perspective*. ACM Trans. on Graphics, 27/4, 107:1–107:21 (2008).

[13] Griessmair J. & Purgathofer W., *Deformation of Solids with Trivariate B-Splines*. In Proceedings of Eurographics - EUROGRAPHICS '89, W. Hansmann, F.R.A. Hopgood and W. Strasser (Editors), Elsevier Science Publishers B.V. (North Holland) , 137–148 (1989).

[14] Helfrich W., *Elastic properties of lipid bilayers-theory and possible experiments*, Z. Naturforsch. C 28, 693-703 (1973).

[15] Hsu W. M., Hughes J. F. & Kaufman H., *Direct manipulation of free-form deformations*. Proc. of ACM SIGGRAPH, New York, 177–184 (1992).

[16] Huang J., Shi X., Liu X., Zhou K., Wei L.-Y., Teng S., Bao H., Guo B. & Shum H.-Y., *Subspace Gradient Domain Mesh Deformation*. ACM Trans. on Graphics, Vol. 25/3, 1126–1134 (2006).

[17] Lamousin H. J. & N. N. Waggenspack N. N., *NURBS-based free-form deformations*. IEEE Computer Graphics and Applications, Vol. 14/6, 59–65 (1994).

[18] Lipman Y., Sorkine O., Cohen-Or D., Levin D., Rössl C., Seidel H.-P., *Differential coordinates for interactive mesh editing*, In Proceedings of Shape Modeling International, IEEE Computer Society Press, 181–190 (2004).

[19] Meyer M., M. Desbrun, P. Schroeder,P., and A. Barr, *Discrete Differential Geometry Operators for Triangulated 2-Manifolds*, in: Proc. VisMath '02, Berlin-Dahlem, Germany, 237–242, (2002).

[20] Morigi S., *Geometric Surface Evolution with Tangential Contribution*, Journal of Computational and Applied Mathematics; 233, 1277–1287 (2010).

[21] Osher S., R. Fedkiw, *Level set methods and dynamic implicit surfaces*. Eds. S.S.Antman, J.E. Marsden, L. Sirovich - Springer Verlag, New

 ISBN 978-80-86943-74-9

York, 2003.

[22] Pauly M., Keiser R., Kobbelt L. & Gross M., *Shape Modeling with Point-Sampled Geometry*. ACM Trans. on Graphics, Vol. 22/3, 641–650 (2003).

[23] Sederberg T. W. & Parry S. R., *Free-form deformation of solid geometric models*. Computer Graphics, 20/4, 151–160 (1986).

[24] Sheffer A., & Kraevoy V., *Pyramid coordinates for morphing and deformation*. In Second International Symposium on 3D Data Processing, Visualization and Transmission, 68–75 (2004).

[25] Shi X., Zhou K., Tong Y., Desbrun M., Bao H. & Guo B., *Mesh Puppetry: Cascading Optimization of Mesh Deformation with Inverse Kinematics*. ACM Trans. on Graphics, Vol. 26/3 (2007).

[26] Sorkine O., Cohen-Or D., Lipman Y., Alexa M., Rössl C. & Seidel H.-P., *Laplacian Surface Editing*. Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, ACM Press, 179–188 (2004).

[27] Sieger D., Menzel S., & Botsch M., *A Comprehensive Comparison of Shape Deformation Methods in Evolutionary Design Optimization*. Proceedings of International Conference on Engineering Optimization - 2012.

[28] Sumner R. w., Schmid J. & Pauly M., *Embedded Deformation for Shape Manipulation*. ACM Trans. on Graphics, Vol 26/3, 80:1–80:7 (2007).

[29] Surazhsky T. and E.Magid and O.Soldea and G.Elber and E. Rivlin, *A comparison of Gaussian and mean curvatures estimation methods on triangular meshes*, In Proceeding ICRA 2003, 1021–1026 (2003).

[30] Terzopoulos D., K. Fleischer, *Deformable models*, The Visual Computer, Vol 4, 306–331 (1988).

[31] Terzopoulos D., *On matching deformable models to images*, Optical Society of America, Topical Meeting on Machine Vision, 160–163 (1986).

[32] Terzopoulos D., Platt J., Barr A. & Fleischer K., *Elastically Deformable Models*. ACM Siggraph Computer Graphics, Vol.21/4, 205–214 (1987).

[33] Willmore T.J., *Total Curvature in Riemannian Geometry* (John Wiley and Sons, New York, 1982).

[34] Yoshizawa S., *Fair Triangle Mesh Generation with Discrete Elastica*, Geometric Modeling and Processing, 119–123 (2002).

[35] Yu Y., Zhou K., Xu D., Shi X., Bao H., Guo B. & Shum H.-Y., *Mesh Editing with Poisson-Based Gradient Field Manipulation*. ACM Trans. on Graphics, 23/3, 641–648 (2004).

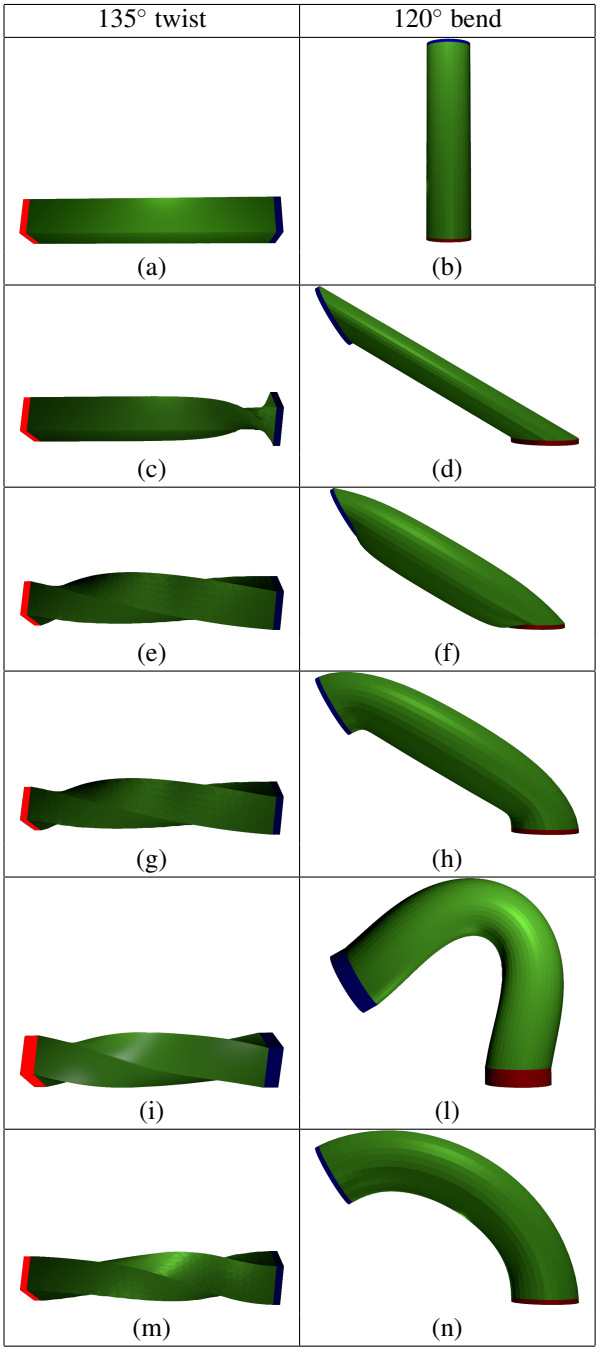
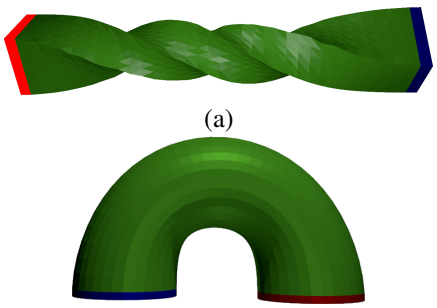


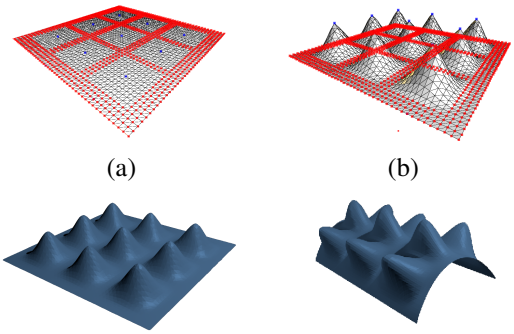
Figure 1: Deformation of the `bar` (left column) and `cylinder` (right column) meshes. The deformations are achieved by anchoring the red vertices and twisting/bending the blue vertices.

 ISBN 978-80-86943-74-9



(a) (b)

Figure 2: (a) 300° twist of the `bar` mesh; (b) 180° bend of the `cylinder` mesh.



(a) (b) (c) (d)

Figure 3: (a) Original `plane` model; (b) Bumpy plane achieved by deforming (a) anchoring the red vertices and translating the blue vertices; (c) undeformed shaded model; (d) deformation by bending the two sides of the mesh.

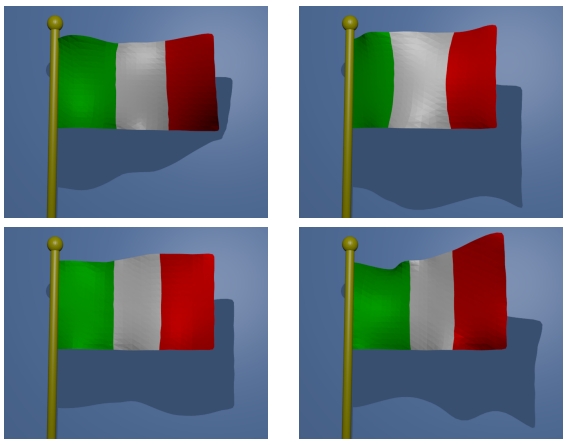
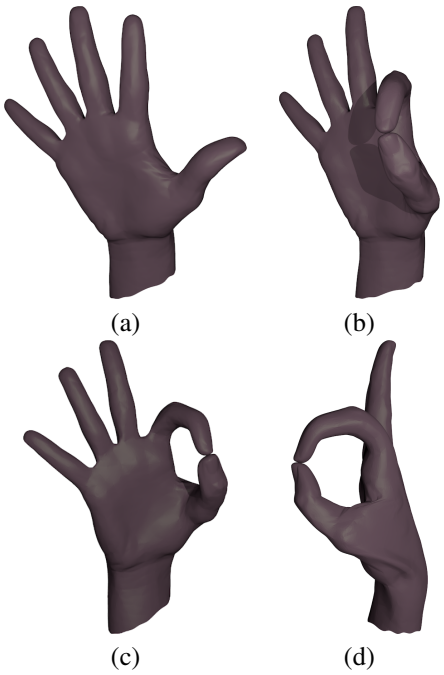


Figure 4: A sequence of `flag` deformations by translating the right side and anchoring the left side of the mesh.



(a) (b) (c) (d)

Figure 5: Interactive deformation of the `hand` mesh: (a) undeformed mesh; (b)-(c)-(d) deformed mesh from different points of view.

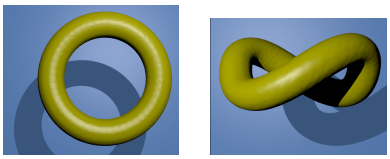


Figure 6: A twist of `torus` mesh.

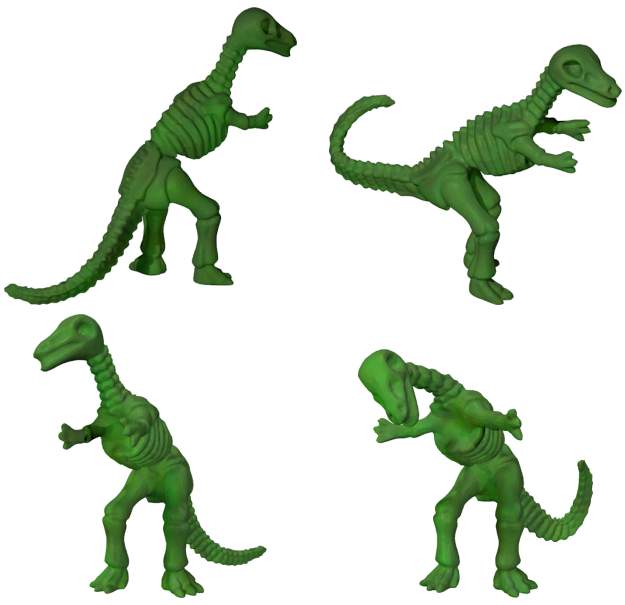


Figure 7: A sequence of `dino` deformations by free interactive shape-editing steps.

 ISBN 978-80-86943-74-9