Dynamic Harmonic Fields for Surface Processing

Kai Xu[a,b], Hao Zhang[*,b], Daniel Cohen-Or[c], Yueshan Xiong[a]

[a]*School of Computer Science, National University of Defense Technology, China*
[b]*School of Computing Science, Simon Fraser University, Canada*
[c]*School of Computer Science, Tel-Aviv University, Israel*

Abstract

Harmonic fields have been shown to provide effective guidance for a number of geometry processing problems. In this paper, we propose a method for fast updating of harmonic fields defined on polygonal meshes, enabling real-time insertion and deletion of constraints. Our approach utilizes the penalty method to enforce constraints in harmonic field computation. It maintains the symmetry of the Laplacian system and takes advantage of fast multi-rank updating and downdating of Cholesky factorization, achieving both speed and numerical stability. We demonstrate how the interactivity induced by fast harmonic field update can be utilized in several applications, including harmonic-guided quadrilateral remeshing, vector field design, interactive geometric detail modeling, and handle-driven shape editing and animation transfer with a dynamic handle set.

Key words: Harmonic fields, Dynamic update, Boundary constraints, Penalty method, Multi-rank updating

1. Introduction

Much work on geometry processing centers around the specification, computation, and utilization of functions and operators defined on surfaces. These surfaces are predominantly modelled in the discrete setting using meshes. A variety of functions and functional operators defined on meshes have been studied, laying the foundation and driving numerous applications in mesh processing. Some of these fundamental geometric applications include interpolation, sampling, filtering, remeshing, surface mapping, and many more.

We are interested in a particular class of functions on meshes — harmonic functions [2]. A harmonic function defines a scalar- or vector-valued field, thus it is also called a harmonic field. Harmonic fields can be defined as solutions to Laplace's equation with certain boundary conditions. In this paper, we consider boundary conditions of the Dirichlet type. Over the discrete manifold surface of a mesh, harmonic fields can be computed by solving a linear system set up by a discrete Laplacian-Beltrami operator, incorporating boundary conditions imposed at a set of *sites*. Site is a general term we use to refer to the locations of attributes which specify the boundary constraints of a harmonic field. Certain desirable properties of harmonicity, such as smoothness and concentration of local extrema only at the boundaries (the maximum principle), make harmonic fields suitable to use in a number of applications [1, 9, 11, 20, 29, 33, 37, 41].

Depending on the application, a site for a harmonic field can be a control handle in mesh deformation [1, 41], a feature point for shape matching [37] or morphing [20], a boundary or other constraint point for mesh parameterization [11, 14, 22], or a user-specified critical point in harmonic-guided quad-remeshing [9, 29, 33]. Common to all of these applications is the effective guidance provided by the choice of sites and their resulting harmonic fields. As site constraints directly influence results (see Figure 1), to be able to provide immediate visual feedback on the guidance fields, while inserting, deleting, or moving these constraints, can greatly improve quality control and design efficiency.

Most previous use of harmonic fields has been restricted to the static setting [1, 9, 11, 20, 29, 33, 37, 41]. When processing large mesh models with a dynamic set of sites, computing harmonic fields can be a major bottleneck, impairing interactive applications. Fisher et al. [12] propose interactive design of harmonic vector fields which allows dynamic change of constraint vectors. However, as we will discuss later, directly extending their method of constraint enforcement to scalar fields will lead to bi-harmonic rather than harmonic fields. Bi-harmonic fields, while preferred by several applications, are undesirable in others, such as handle-based shape deformation, since local extrema may be present at locations other than the boundaries.

Our main contribution is a method for fast updating of harmonic fields, both scalar and vector valued, under dynamic site conditions. Note that the same technique also applies to bi-harmonic fields. We utilize the penalty method [10, 36, 45] to enforce constraints which leads to approximate solutions to the Laplace equation. With suitably chosen penalty weights, the approximation error of the obtained solution is negligible. Compared to alternative formulations and execution of harmonic field computation, including direct elimination [13], substitution [9], and multi-grid solver [26], our method maintains the symmetry of the Laplacian system and takes advantage of

[*]Corresponding author. Tel.: +1 778-782-6843.
Email addresses: kaixu@nudt.edu.cn (Kai Xu), haoz@cs.sfu.ca (Hao Zhang), dcor@tau.ac.il (Daniel Cohen-Or), ysxiong@nudt.edu.cn (Yueshan Xiong)

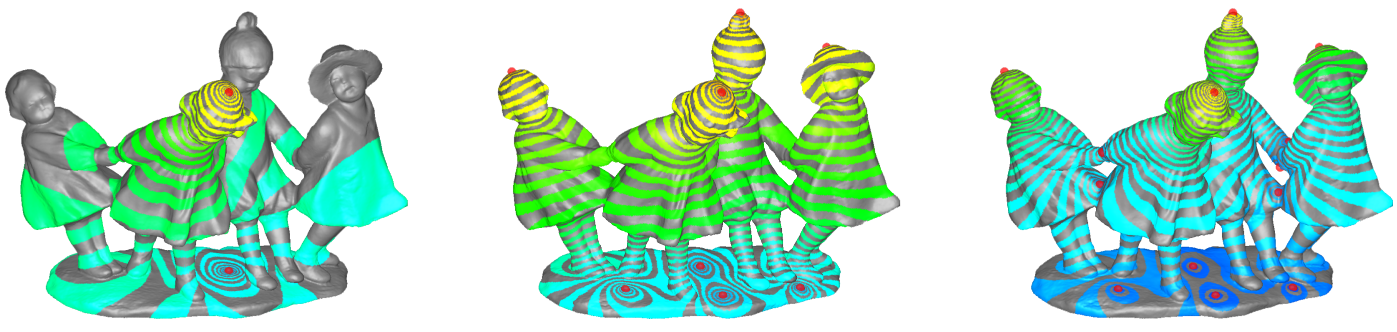
(a) Two sites. (b) Eight sites. (c) Thirteen sites.

Figure 1: Iso-contour plots of harmonic fields over the high-genus children model (304K vertices), where red dots mark the chosen sites. Iso-contour strips are colored via linear interpolation between blue (value 0) and yellow (value 1). Updating time is less than one second, allowing for exploration of varying design choices in real time.

the state-of-the-art algorithm of updating[1] and downdating of Cholesky factorization [7], allowing for real-time updating of solutions to constrained Laplacian systems.

We demonstrate the utility of interactive harmonic field update in a number of applications. These include harmonic-guided quad-remeshing, vector field design, interactive geometric detail modeling, and handle-driven shape editing and animation transfer with a dynamic handle set. In all these examples, a user is able to explore different design options guided by immediate visual feedback in real time, while working with hundreds of dynamic sites at a time, on very large mesh models.

2. Related works

There have been many uses of harmonic fields in geometry. In the volumetric setting, Joshi et al. [19] use harmonic coordinates for space deformation and Martin et al. [23] use harmonic functions for volumetric parameterization and trivariate B-spline fitting. More frequent use of harmonic fields still lies in surface processing. In handle-driven surface deformation, both the construction of reduced deformation models [1] and transformation propagation [1, 41] have utilized scalar harmonic fields. Interactive shape editing is achievable for large models since the reduced model and the harmonic fields can be pre-computed and reused during an editing session. However, when the handle set changes, the required recomputation of harmonic fields hinders performance.

Dong et al. [9] trace the integral lines of the gradient and co-gradient of a harmonic field for quad-remeshing, where user-specified sites serve as allowed singularity points. Tong et al. [33] compute two piecewise smooth harmonic functions whose iso-lines provide a quad-tiling. The key feature of their approach is an extension to the discrete Laplacian operator to allow line-type singularities and singularities with fractional indices, leading to more design flexibility. For surface-based shape correspondence, corresponding harmonic fields computed from matching sites on the two surfaces induce the mapping. Such examples include harmonic maps [11], feature-based non-rigid 3D registration [37], shape morphing [20], and animation transfer [41]. None of these applications allow updating of harmonic fields under dynamic site conditions.

To compute harmonic fields, one solves a linear system defined by the Laplacian-Beltrami operator, while incorporating *hard* constraints at the sites. Direct elimination [13] leads to a symmetric system to which Cholesky factorization applies. However, it is difficult to update the factorization when the set of sites change. An alternative is substitution [9], which results in a non-symmetric system. A multi-grid solver [26] can be applied and it is efficient, however there is no known fast scheme for handling dynamic sites. James and Pai [18] take the capacitance matrix approach to update LU decomposition. However, its efficiency is far from what can be accomplished by updating Cholesky factorization [5, 6, 7].

Sorkine et al. [31] enforce *soft* constraints at the sites in their progressive construction of geometry-aware bases. Their weighted least-squares (wLSQ) fitting solution, while smooth and suitable for shape approximation, is generally not a harmonic function. Instead, it is bi-harmonic which involves the bi-Laplacian, corresponding to a variational solution to the minimization of thin-plate energy [21]. On the other hand, the resulting normal equation is a symmetric linear system which admits efficient Cholesky and supports updates which involve adding a new row to the system. This is essentially equivalent to the rank-1 updating in [5].

Works on texture synthesis on surfaces [28, 34, 39], quad-remeshing [9, 33], and non-photorealistic rendering [15] have relied on designated vector fields on a given mesh. Their focus has been on how to obtain results based on a given vector field, which typically results from user input. A recent and thorough investigation into generic vector field design is due to Zhang et al. [43]. A variety of field editing operations are supported. Vector flow initialization, rotation, and reflection are per-vertex operations and do not require solving a system. Flow smoothing inside a given region, which is later utilized for singularity movement and pair cancellation, is carried out by solving a fresh constrained Laplacian equation using a bi-conjugate solver each time. None of these works addressed the problem of interactive field update or exploration.

[1]Note that "updating" of Cholesky factorizations is a specific term in its relevant literature and should not be confused with the general use of the word "update"; the distinction should be clear from the context.

Most relevant to our work, Fisher et al. [12] compute harmonic vector fields by interpolating discrete differential 1-forms given a sparse set of wLSQ constraints on mesh edges. The same method can work on 0-forms with wLSQ constraints on vertices to obtain a scalar field. However, this leads to the same bi-Laplacian system as in [31]. To facilitate interactive vector fields design, Fisher et al. [12] utilize incremental updating of Cholesky factorizations [5, 6]. Our method exploits a more efficient updating/downdating method based on the state-of-the-art dynamic supernodal algorithm of Davis and Hager [7], allowing for insertion/deletion of larger sets (up to hundreds) of sites on a dense mesh in real time.

3. Harmonic fields and constraints handling

Let u be a harmonic function, a solution to the Laplace equation $\Delta u = 0$ subject to Dirichlet boundary conditions. We regard u as a scalar function for now; these functions are harmonic 0-forms following Gu and Yau [14]. For vector-valued functions, each vector component can be treated separately. In the manifold setting, Δ is the Laplace-Beltrami operator. On a triangle mesh surface, Δ can be discretized in several ways [24, 38], sometimes resulting in a non-symmetric Laplacian matrix. We adopt the symmetric operator first derived by Pinkall and Polthier [27], since it leads to symmetric and positive-definite linear systems when computing harmonic fields, allowing for fast Cholesky factorization. For a closed manifold triangle mesh, the matrix is given by $\mathbf{L} = \mathbf{D} - \mathbf{W}$. $\mathbf{W}$ is defined by the well-known cotangent weights:

$$\mathbf{W}_{ij} = \frac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij}),$$

where α_{ij} and β_{ij} are opposite angles to edge (i, j) in the mesh and if (i, j) is not an edge, $\mathbf{W}_{ij} = 0$. The matrix $\mathbf{D}$ is a diagonal matrix of the row sums of $\mathbf{W}$.

Denote by $S \subseteq \{1, 2, \ldots, n\}$ the index set for the sites. The site constraints dictate that $u(i) = \mathbf{u}_i = s_i$ for all $i \in S$, where s_i is the prescribed value of the harmonic field at site i. Different ways to incorporate constraints into the Laplace equation lead to different linear systems. Direct methods are often used to solve the system since factorization of the system matrix can be reused for different constraint *values* at the sites. However, if the site *locations* change, the factorization is no longer applicable. Re-factoring the system is generally time-consuming. A better option is to update, for which the choice of methods to enforce site constraints is critical.

We compare three commonly used methods for constraint handling in solving partial differential equations (PDEs), e.g., via the finite element method (FEM), and show that the penalty method is the best choice in terms of efficiency, stability and support for factorization updates.

Direct elimination. This method [13] eliminates from the original system matrix the variables corresponding to the site vertices with known constraints. The unknown harmonic field values are computed by solving the re-arranged system,

$$\begin{bmatrix} \mathbf{L}_{\bar{S}\bar{S}} & \mathbf{L}_{\bar{S}S} \\ \mathbf{L}_{S\bar{S}} & \mathbf{L}_{SS} \end{bmatrix} \begin{pmatrix} \mathbf{u}_{\bar{S}} \\ \mathbf{u}_S \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{s} \end{pmatrix},$$

where $\mathbf{u}_{\bar{S}}$ is the vector of unknowns, $\mathbf{u}_S = (u_i)^{\mathrm{T}}_{i \in S}$ the vector of site vertices, and $\mathbf{s} = (s_i)^{\mathrm{T}}_{i \in S}$ the vector of corresponding pre-assigned site constraints. $\mathbf{L}_{\bar{S}\bar{S}}, \mathbf{L}_{\bar{S}S}, \mathbf{L}_{S\bar{S}}, \mathbf{L}_{SS}$ are corresponding block matrices in the Laplacian $\mathbf{L}$; $\mathbf{L}_{\bar{S}S} = \mathbf{L}_{S\bar{S}}$. The solution is $\mathbf{u}_{\bar{S}} = -\mathbf{L}^{-1}_{\bar{S}\bar{S}} \mathbf{L}_{\bar{S}S} \mathbf{u}_S$.

Since the modified system matrix $\mathbf{L}_{\bar{S}\bar{S}}$ is symmetric, efficient Cholesky factorization can be applied. However, when the set of sites change, so does the set of unknowns. It is difficult to update an existing Cholesky factorization since the structure of the system matrix changes with the unknowns.

Substitution. This method [9] does not eliminate any unknowns. It substitutes the diagonal element of each row corresponding to a site with 1 and other elements in the row with 0. Meanwhile, the corresponding entry of the right-hand side vector is substituted with the constraint value. This results in the linear system $\mathbf{A}\mathbf{u} = \mathbf{b}$, where

$$\mathbf{A}_{ij} = \begin{cases} \mathbf{L}_{ij} & \text{if } i \notin S, \\ 1 & \text{if } i \in S \text{ and } j = i, \\ 0 & \text{if } i \in S \text{ and } j \neq i. \end{cases} \quad \mathbf{b}_i = \begin{cases} 0 & \text{if } i \notin S, \\ s_i & \text{if } i \in S. \end{cases} \tag{1}$$

This sparse system is no longer symmetric and should be solved via LU decomposition, which is less efficient than Cholesky. More importantly, although an algorithm for updating LU factorization with a change of site locations is known [18], the state-of-the-art updating algorithm coupled with the supernodal method is available only for Cholesky factorization; this is significantly more efficient than updating of LU factorization.

Penalty method. This method finds extensive use and leads to good results in solving constrained variational problems [36, 45]. In computer graphics, it has appeared in shape approximation [30], vector field design [12], and rigid body simulation for fast and stable contact handling [10]. Other applications are found in solving PDEs for fluid simulation and physically-based deformable models. Here we show its effectiveness in constraint handling and computation for harmonic fields.

Roughly speaking, the penalty method converts a constrained optimization problem into an unconstrained one. Instead of minimizing the original objective function, the penalty method minimizes a weighted sum of this objective and a quadratic penalty term involving the constraints. A well-known example is the least-squares mesh of Sorkine and Cohen-Or [30],

$$\begin{aligned} \mathbf{u} &= \operatorname{argmin}_{\mathbf{x}}\{\|\mathbf{L}\mathbf{x}\|^2 + \alpha \sum_{i \in S} |\mathbf{x}_i - s_i|^2\} \\ &= \operatorname{argmin}_{\mathbf{x}}\{\|\mathbf{L}\mathbf{x}\|^2 + \|\mathbf{P}^{1/2}(\mathbf{x} - \mathbf{b})\|^2\}, \end{aligned} \tag{2}$$

where $\mathbf{L}$ is the unconstrained Laplacian matrix defined at the start of this section, $\mathbf{b}$ is defined as in (1), and $\mathbf{P}$ is the diagonal penalty (weight) matrix

$$\mathbf{P}_{ij} = \begin{cases} \alpha & \text{if } i \in S \text{ and } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Least-squares meshes minimize a weighted sum of the 2-norm of the Laplacian and the penalty term. The penalty factor, α, is used to tweak the importance of constraint satisfaction. Differentiating (2), we obtain the linear system

$$(\mathbf{L}^{\mathrm{T}}\mathbf{L} + \mathbf{P})\mathbf{u} = (\mathbf{L}^2 + \mathbf{P})\mathbf{u} = \mathbf{P}\mathbf{b}. \tag{3}$$

Note that we adopt a symmetric $\mathbf{L}$, thus $\mathbf{L}^{\mathrm{T}}\mathbf{L} = \mathbf{L}^2$. We see that the above yields a bi-harmonic solution, corresponding to the minimization of the thin-plate energy [21]. It is also the case when using the method of Fisher et al. [12], interpolating 0-form with least-squares constraints, to compute scalar fields. To obtain constrained harmonic functions, we instead minimize the membrane energy [21],

$$\mathbf{u} = \operatorname{argmin}_{\mathbf{x}}\{\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{L}\mathbf{x} + \frac{1}{2}||\mathbf{P}^{1/2}(\mathbf{x} - \mathbf{b})||^2\},$$

leading to the linear system

$$(\mathbf{L} + \mathbf{P})\mathbf{u} = \mathbf{P}\mathbf{b}. \tag{4}$$

It can be shown that the solution to (4) converges to that of the original constrained membrane energy minimization as α approaches infinity, as long as matrix $\mathbf{P}$ is singular. This is the case for us as the number of sites is smaller than the total number of mesh vertices (also the dimensionality of $\mathbf{L}$). While for the penalty weight, we choose $\alpha = 1.0\times10^8$ for all the examples demonstrated in this paper.

The penalty method has several advantages over the other two alternatives. First, in contrast to substitution, the systems in (3) and (4) are symmetric which admit fast Cholesky factorization. Second, the method does not change the set of unknowns, in contrast to direct elimination. In fact, it maintains the zero-nonzero structure of the system matrix for different sets of sites. Third, the addition of the diagonal penalty matrix in (4) can only improve the conditioning of the system matrix and hence its numerical stability. Last but not least, constraint handling via the penalty method facilitates fast Cholesky updating/downdating [7].

Although the use of penalty can only provide approximate constraint satisfaction, our experiments show that the resulting approximate solution is sufficiently close to those obtained by an exact method. In particular, maximum errors measured between solutions obtained by the penalty method and solutions obtained by the substitution method (an exact method) are in the order of 10^{-6}, with our choice of the penalty weight α.

For all the applications considered in this paper, the highly accurate solutions we obtain via the penalty method have proved to be sufficient. Therefore, we believe that the penalty method is the most advantageous in terms of speed, stability, and more importantly, factorization updating. Note also that while fast updating of Cholesky is applicable to both harmonic and bi-harmonic solutions, we focus only on harmonic fields, as harmonicity is desired in the applications considered.

4. Fast updating of harmonic fields

Lacking a fast harmonic field update scheme under dynamic site conditions, previous shape processing works typically assume that the boundary sets are generally not changed frequently. In practice however, design flexibility and immediate visual feedback are vital in an interactive setting, which then require dynamic site selection capabilities. This provides the user with an important way to interact with harmonic fields and hence to control shape processing behavior.

Model	#V	$\lvert S_{ins}\rvert$	$\lvert S_{del}\rvert$	Comp.	Updt.
Armadillo	173K	40	30	2.65	0.25
Armadillo	173K	200	0	2.65	0.63
Armadillo	173K	0	300	2.65	1.10
Raptor	42K	150	160	0.36	0.10
Raptor	84K	150	160	0.79	0.21
Children	304K	60	50	5.85	0.87
Children	304K	100	80	5.85	1.09
Children	304K	1	1	7.68	0.002

Table 1: Timing for re-computing (Comp.) vs. updating (Updt.) harmonic fields, recorded in seconds. The machine is a 2.5GHz Intel Core 2 Duo PC with 2GB of RAM.

We adopt the supernodal algorithm [7] for efficient updating and downdating of a sparse Cholesky factorization. Given an $n \times n$ sparse, symmetric positive definite matrix $\mathbf{A}$ with Cholesky factorization $\mathbf{A} = \mathbf{G}\mathbf{G}^{\mathrm{T}}$, supernodal methods gather the columns of $\mathbf{G}$ that have an identical or similar nonzero pattern into a set of dense submatrices, called supernodes of $\mathbf{G}$. Exploiting supernodal structures improves the locality for large sparse matrices, leading to higher performance in computation and memory access. However, most existing Cholesky updating schemes are non-supernodal since both updating and downdating can change and invalidate the supernodes. The algorithm we use [7] can detect supernodes dynamically as updating proceeds.

If a multi-rank modification to $\mathbf{A}$ can be written in the form of a matrix addition, i.e., $\bar{\mathbf{A}} = \mathbf{A} + \mathbf{R}\mathbf{R}^{\mathrm{T}}$ for updating and $\bar{\mathbf{A}} = \mathbf{A} - \mathbf{B}\mathbf{B}^{\mathrm{T}}$ for downdating, the supernodal algorithm of [7] can be utilized to update/downdate the Cholesky factorization of $\mathbf{A}$. Using the penalty method, the site constraints are imposed as an addition of the penalty matrix to the Laplacian matrix (4). As a result, inserting/deleting site constraints can be written as matrix additions:

$$\mathbf{L} + \bar{\mathbf{P}} = \mathbf{L} + \mathbf{P} + \mathbf{R}\mathbf{R}^{\mathrm{T}} - \mathbf{B}\mathbf{B}^{\mathrm{T}}, \tag{5}$$

where the modification matrices $\mathbf{R}$ and $\mathbf{B}$ have entries:

$$\mathbf{R}_{ij} = \begin{cases} \sqrt{\alpha} & i = j \in S_{ins}, \\ 0 & \text{otherwise}, \end{cases} \quad \mathbf{B}_{ij} = \begin{cases} \sqrt{\alpha} & i = j \in S_{del}, \\ 0 & \text{otherwise}, \end{cases}$$

with S_{ins} the set of indices for newly inserted site constraints and S_{del} the indices of site constraints to be deleted. It is obvious that the multi-rank updating procedure described above also applies to the bi-Laplacian (3). As a result, our method also supports fast updating of bi-harmonic fields.

The cost for update/downdate depends on the nonzero patterns of $\mathbf{L}$ and the pattern of the update. For a rank-one update/downdate operation, the cost is proportional to the number of changed entries in the Cholesky factor $\mathbf{G}$, which is typically much smaller compared to the size of the mesh, leading

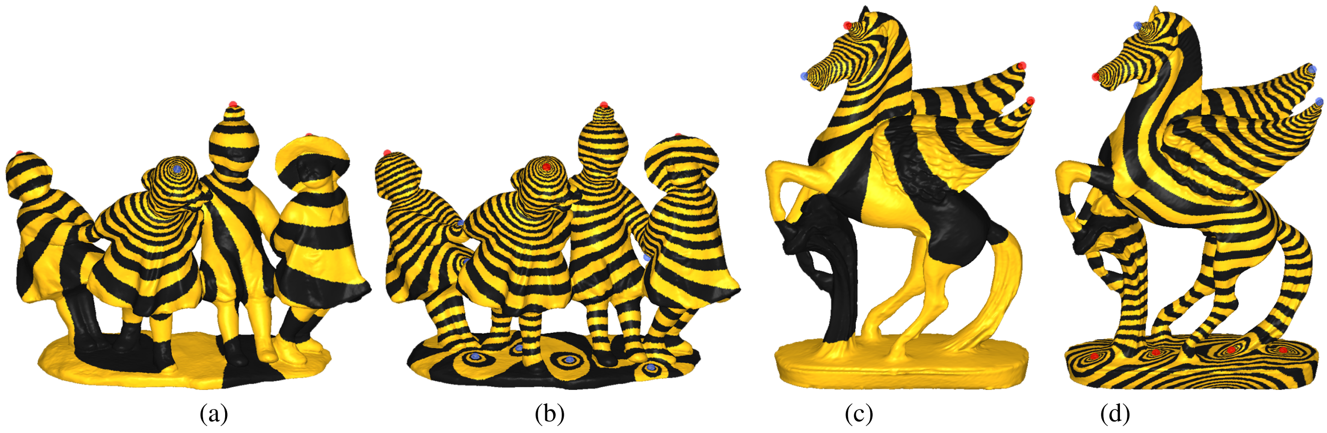

Figure 2: Iso-lines of harmonic scalar fields can be displayed to help steer quad-mesh design [9]. On high-genus models, interactive operations on site (critical point) selections allow the user to obtain harmonic fields that are more conforming to the shape — (b) and (d). (a) and (c): fields resulting from assigning sites to automatically extracted shape extremities.

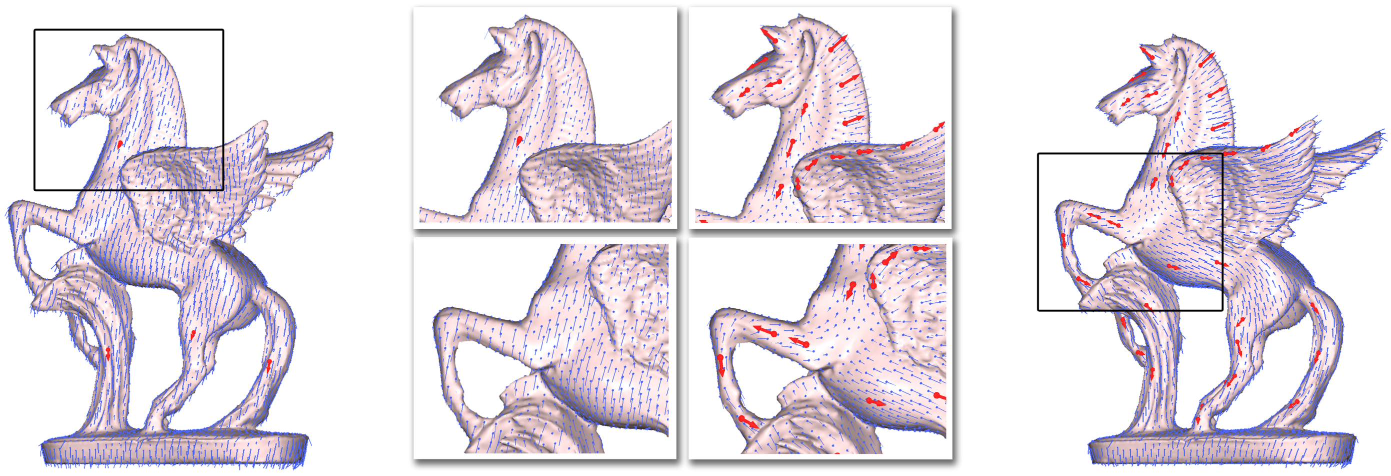
Figure 3: Interactive design of harmonic vector fields on the Pegaso model (120K vertices). The user specifies a few constraint vectors (red arrows) over the mesh surface. Our method updates the harmonic vector field accordingly in real time. With more constraints inserted, the resulting vector field is further refined everywhere (see right figure). Note that we only plot a sparse set of vectors in the field to facilitate visualization; our update scheme is applied to the entire dense set of mesh vertices.

to efficient results. Note also that diagonal modification does not change the nonzero patterns of the system matrix and its Cholesky factor. As a result, the supernodes do not change and our problem can be solved even more efficiently with the supernodal updating algorithm, since no dynamic detection is needed. More details about the algorithm can be found in [7]. The updated harmonic field can be easily derived from the updated Cholesky factorization via back substitution. An implementation of the dynamic supernodes algorithm for Cholesky is available in the CHOLMOD package [4].

In applications such as critical point selection for quad-remeshing [9], feature selection for shape matching [37], and boundary selection for parameterization [11, 22, 14], the number of inserted or deleted constraints is typically small, making the modification to the system matrix (5) low-rank. In the case of handle-driven shape editing, although the handle regions may contain a large number of vertices, harmonicity of the field ensures that we only need to sample along the region boundaries, as explained in Section 5.2. With our current implementation, we are able to process models of sizes up to a few hundred thousands vertices and update their harmonic fields with up to 300 inserted and deleted sites in real time. Table 1 lists some timing statistics comparing the updating of harmonic fields with our method and computing them from scratch after inserting and/or deleting sites. Evidently, updating can be done much more efficiently than re-computation, especially when the number of changed sites is relatively small.

5. Applications and results

We present several applications and visual results to demonstrate effective use of dynamic harmonic fields.

5.1. Interactive harmonic field design

Several surface processing applications, e.g., quad-remeshing and texture synthesis, can benefit from the use of harmonic fields in an interactive setting. One can build and

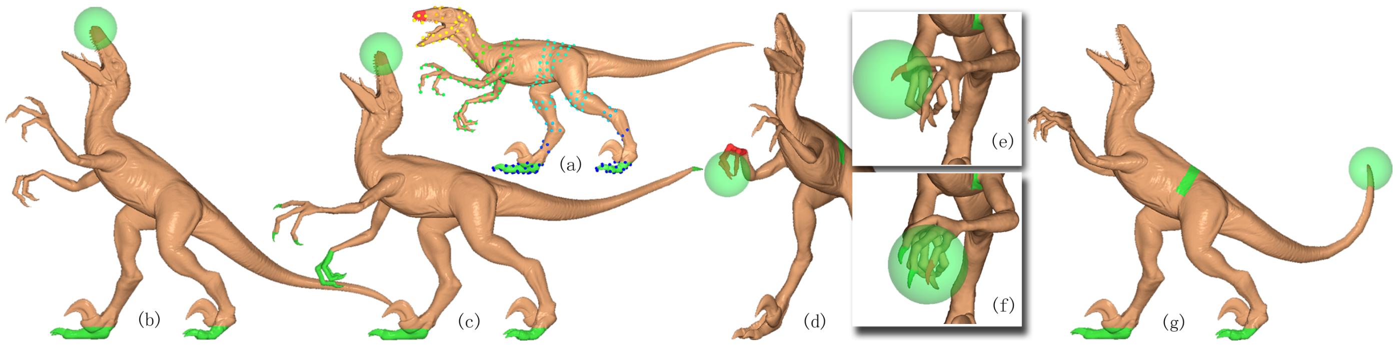

Figure 4: Hierarchical editing of the raptor (84K vertices). (a) shows our deformation model with colored dots representing deformation clusters. The raptor's head is first raised (b), using three coarse-level handles. Then, its left hand is manipulated to grasp the right one (d)-(f), with a new set of finer-level handles inserted. Finally, its tail is curled up (g). If all the coarse and fine-level handles were selected at the start (c), the effect of raising the head would be limited (c), unlike the one shown in (b).

control a smooth field over a mesh surface by placing and moving sites. In certain applications, e.g., quad-remeshing [9], the sites are preferably placed at feature points such as shape extremities. These extremities, e.g., the top of the children's heads and the wing tips of the Pegaso in Figure 2, can be extracted automatically via Poisson [9] or average geodesic distance fields [42] defined over a mesh. However, not all useful features are easy to detect this way, for instance, when extremities are merged into other parts of a shape like the feet of the Pegaso and children models. These sites are typically assigned manually to steer the mesh design. However, when a model is high-genus or the number of sites is large, the effect of site positioning on the resulting iso-lines becomes non-intuitive, making the design difficult.

As a result, immediate visual feedback is especially useful in interactive design of harmonic fields. Our fast updating method allows us to provide three interactive operations: site insertion, deletion, and movement, where the latter is implemented as a sequence of site deletions and insertions. These operations are all supported in real time even on a very dense mesh; see the last row of Table 1. Figure 2 compares the harmonic fields on two high-genus models obtained by interactive design versus those computed with extremities extracted from Poisson fields. Note that some recent works on quad-remeshing utilize the Laplacian eigenfunctions [8, 16] instead of harmonic fields. It would be interesting to see whether interactive control of singularity placements is possible within such a framework.

The use of harmonic fields is not restricted to scalar-valued attributes. Smooth tangential vector fields on surfaces have been applied to control rendering of surfaces, e.g., in texture synthesis [34] and non-photorealistic rendering [15]. By solving Laplace's equation for each direction separately, our method can compute a harmonic vector field given a set of constraint tangential vectors. Since the harmonic vectors are not necessarily tangential to the mesh surface away from the constraint sites, we project them onto estimated local tangent planes to obtain a tangential vector field that is approximately harmonic. Although less accurate, our method can be faster than [12] in terms of both computing and updating due to the simple formulation and more advanced updating method we employ. As a result, our method computes approximate harmonic vector fields interactively and in practice, these fields exhibit similar properties as true harmonic fields and work very well in applications such as texture synthesis. The constraint vectors selected through such an interactive design session can then be used to drive a more advanced method such as [12] to more accurately compute the final vector field. We have implemented four interactive operations to facilitate interactive design of tangent vector fields: insertion, deletion, movement, and rotation of the constraint vectors; see Figure 3.

5.2. *Shape deformation with dynamic handles*

In handle-driven shape deformation, harmonic fields have been used to guide the propagation of transformations from the manipulated handles to the rest of the shape [1, 41]. So far the deformation model is based on harmonic fields pre-computed for a set of prescribed handles. With our real-time harmonic field update, dynamic handles are now possible. Dynamic handles add more design flexibility and better control to interactive editing, as the user is not constrained by a particular selection of the handle set. Similarly, Laplacian [32] or Poisson [40] shape editing can also benefit from our method which can update the Cholesky factorization efficiently whenever the user changes the *region of interests* (ROI).

In what follows, we first use hierarchical shape editing as an example to illustrate the above point. Then we discuss site selection for harmonic field computation. Finally, we describe an improved deformation model which can be more quickly updated from a renewed harmonic field, further speeding up the use of dynamic handles.

Hierarchical shape editing. In many real-world design scenarios, an editing task has to be carried out in a hierarchical manner. Refer to the editing example illustrated in Figure 4. While raising the raptor's head is a higher-level pose change, best accomplished with a coarse handle set, moving its hands or fingers belongs to lower-level editing, which is more suitably controlled by a finer set of handles.

In general, one cannot select all the handles across the editing hierarchy at the beginning since anticipating the entire set of handles that will be needed is difficult. Even if this were possible, doing so would be limiting as the presence of finer-level handles may limit the degree of freedom required to achieve higher-level pose editing; see Figure 4(c). A coarse-to-fine approach requires finer-level handles to be inserted as editing proceeds, while these handles should be removed when we move up the editing hierarchy. With dynamic handles, the user can freely insert and remove handles at any time without sacrificing interactivity.

Site selection. A handle typically encompasses a small region of the manipulated shape, e.g., a hand of the raptor. For a dense mesh, a single handle may contain a large number of vertices or sites. If the number of sites to insert or remove is too large, real-time update may not be achievable. Fortunately, harmonicity of the guidance fields implies that treating all vertices in the handle regions as sites is not necessary. It suffices to only sample along the boundaries of the handle regions. Indeed, if the boundary points of a connected region take on a common harmonic field value and the region contains no other sites, then the whole region must take on that value due to the maximum principle. In practice, we only sample dozens of boundary vertices from each handle as sites, ensuring real-time performance.

Improved deformation model. With a renewed harmonic field due to handle insertion or deletion, generating plausible deformation over the whole mesh is still computationally demanding, especially for a dense model and a non-linear approach to handle rotations [1]. Thus the model representation needs to be reduced so that expensive operations are only performed on the reduced model, with results then propagated to all vertices.

An elegant reduced model utilizing harmonic fields is based on extracting the fields' iso-lines [1]. Our reduced model is built by clustering sampled points with the same harmonic value into a point set which we call a *deformation cluster*; see Figure 4(a). Each set of sites corresponding to a handle is also a cluster. Since sampling point sets is easier than extracting iso-lines, our model is more efficient to build. To each deformation cluster, we associate a rigid transformation. The mesh is then deformed by linearly interpolating the transformations at the deformation clusters based on a harmonic field. The clusters are sampled in overlapping fashion so that transformation propagates via points shared between neighboring clusters.

When the user drags a handle, we perform shape matching [25] on the corresponding handle cluster to find the optimal rigid motion for that handle. Due to cluster overlapping, the other clusters obtain reasonable rigid motions, also via shape matching. Keeping the local transformations rigid leads to detail-preserving deformation. We replace polar decomposition adopted by Müller et al. [25] by singular value decomposition (SVD), so as to ensure a pure rotation factorization [17]. Precomputation is only needed for shape matching. For each cluster, it only involves inverting a 3×3 matrix. This cost is negligible compared to that of Cholesky factorization and matrix multiplications as in Au et al.[1]. Overall, combining a more efficient reduced model and shape matching via SVD for pure rotation factorization, our deformation model can be quickly updated from a renewed harmonic field.

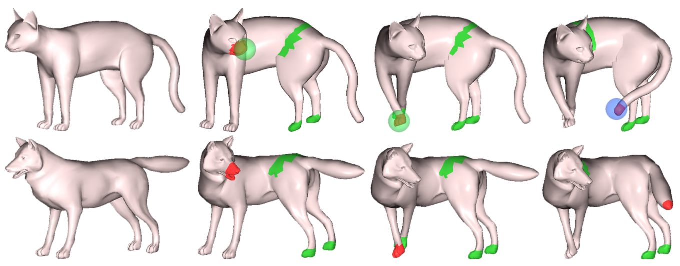
Figure 5: Deformation transfer with dynamic corresponding handles. The user manipulates the cat, changing handles in real time. The deformations are transferred to the wolf.

5.3. *Deformation transfer with dynamic handles*

Harmonic scalar fields, computed on two meshes with meaningful correspondence between their sites, provide a dense correspondence (iso-line correspondence) for deformation transfer between the two meshes [41, 1]. Dynamic handles can facilitate this process. For example, if the target deformation is not satisfactory, the user may wish to tune the source one through, say, changing handles. In harmonic-guided deformation transfer, changing of handles is more expensive since the harmonic fields of both meshes must be recomputed. Our fast update scheme allows us to apply our reduced deformation model to interactive editing and deformation transfer with a dynamic handle set.

Similar to other works [1, 41], our method requires the source and target shapes to have similar semantic correspondence and initial poses to ensure semantically meaningful and visually pleasing results. However, their sizes and shapes can be different. The corresponding handles on the two meshes are assigned interactively by the user. To avoid undesirable scaling on the target due to possible shape differences, we only transfer the rotational component of the rigid transformations and use shape matching (Section 5.2) to compute the local translation on each deformation cluster. Figure 5 shows an interactive deformation transfer session with dynamic handles.

5.4. *Interactive geometric detail modelling*

In contrast to detail-preserving shape editing, geometric detail modelling changes the appearance of a surface by modifying its high-frequency content. The user selects a region of interest (ROI) and cuts it away, forming a hole on the surface. The hole is filled by a smooth membrane or thin-plate base mesh computed with respect to the boundary of the hole. All the mesh vertices along the boundary curve serve as sites to define the harmonic or bi-harmonic base mesh. The base mesh can then be enriched with new geometric details, either synthesized by sample [3] or transferred from another shape [32].

As shown in Figure 6, the user selects a region of interest (ROI) by drawing a curve on the surface. The ROI is cut and then filled with a harmonic membrane, having the same connectivity as the original ROI. In the current experiment, the geometric details are modified simply by interpolating between the original surface over the ROI and the membrane. By changing

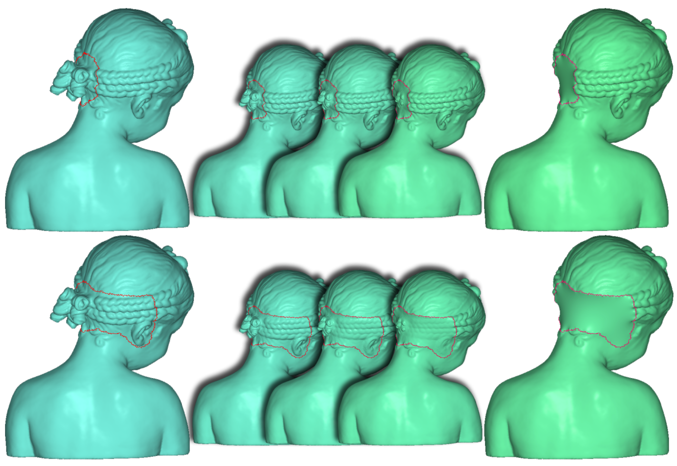

Figure 6: Interactive geometric detail modeling. A part of the Bimba model (115K vertices) is cut by drawing a curve on the surface (top row). A harmonic membrane is computed with respect to the boundary curve to fill the cut region and serve as the base for detail transfer. Geometric details are modified via a morphing between the original surface and the membrane (middle figures). When the user enlarges the boundary curve (bottom row), introducing 284 new sites, the new membrane can be updated in only 0.23 seconds.

the interpolation weight continuously, we obtain a morphing sequence as shown in the figure.

During interactive modelling, the ROI often needs to be changed frequently. However, for a dense mesh, computing the base mesh can be time-consuming. The key to note is that our method can update the base mesh in real time, according to the new boundary of the changed ROI, facilitating interactive geometric detail modeling. Note that our fast update scheme can also be applied to thin-plate base meshes since solving a bi-Laplacian also involves Cholesky factorization, as we have explained in Section 3.

6. Conclusion and future work

We present a method for quick updating of harmonic fields with respect to a dynamic set of sites. The penalty method is employed to enforce site constraints and this allows the use of fast updating and downdating to Cholesky factorization. We demonstrate that our method endows several surface processing applications with effective, interactive harmonic field guidance. The immediate visual feedback enabled by real-time dynamic harmonic fields adds design flexibility and interactive exploration to the user experience.

Our current software implementation allows us to interactively update several hundred sites on moderate large meshes, with up to several hundred thousand vertices. To alleviate the constraint on low-rank updating to Cholesky factorization so as to handle much larger sets of sites on even denser meshes, a GPU implementation of the Cholesky updating algorithms [7] is called for. In future work, we would first like to consider such a GPU implementation using data-oriented GPU programming models such as Nvidia's CUDA.

Using the penalty method to enforce site constraints, the obtained harmonic fields are approximate. Although these approximate solutions are highly accurate as judged by maximum approximation errors, this does not necessarily imply that all the theoretical properties of a harmonic filed, e.g., those related to the number of critical points, are preserved. These issues deserve further investigation.

Finally, as harmonic functions are almost ubiquitous in surface processing, we shall look into more applications, interactive or otherwise, which can benefit from dynamic harmonic fields. One such example is harmonic-guided shape matching. It is possible to incorporate harmonic fields into a deformation-driven shape correspondence framework [44] to obtain a significant performance boost, as the deformation corresponding to different sets of matching feature pairs can be obtained much more efficiently through field updates.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments. We are also grateful to Steven Bergner for proofreading the manuscript. This work was supported in part by an NSERC Grant (No. 611370), the Israel Ministry of Science and Education Tashtiot Grant "Advanced Methods For Reconstruction Of Three-Dimensional Synthetic Object", the Israel Science Foundation, as well as grants from the 863 Program of China (No. 2007AA01Z313 and No. 2008AA09Z124049) and National Natural Science Foundation of China (No. 60773022 and No. 60707030).

References

[1] Au, O. K.-C., Fu, H., Tai, C.-L., Cohen-Or, D., 2007. Handle-aware isolines for scalable shape editing. ACM Trans. on Graphics 26 (3), 83:1–83:10.

[2] Axler, S., Bourdon, P., Wade, R., 2001. Harmonic Function Theory, Second Edition. Springer.

[3] Bhat, P., Ingram, S., Turk, G., 2004. Geometric texture synthesis by example. In: Proc. of Symp. on Geometry Processing. pp. 41–44.

[4] Davis, T. A., 2006. User guide for cholmod. Tech. rep., University of Florida.

[5] Davis, T. A., Hager, W. W., 1999. Modifying a sparse Cholesky factorization. SIAM Journal on Matrix Analysis and Applications 20 (3), 606–627.

[6] Davis, T. A., Hager, W. W., 2001. Multiple-rank modifications of a sparse cholesky factorization. SIAM Journal on Matrix Analysis and Applications 22 (4), 997–1013.

[7] Davis, T. A., Hager, W. W., 2009. Dynamic supernodes in sparse cholesky update/downdate and triangular solves. ACM Trans. on Math. Software 35 (4).

[8] Dong, S., Bremer, P.-T., Garland, M., Pascucci, V., Hart, J. C., 2006. Spectral surface quadrangulation. ACM Trans. on Graphics 25 (3), 1057–1066.

[9] Dong, S., Kircher, S., Garland, M., 2005. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. Computer-Aided Geometric Design 22 (5), 392–423.

[10] Drumwright, E., 2008. A fast and stable penalty method for rigid body simulation. IEEE Trans. Vis. & Comp. Graphics 14 (1), 231–240.

[11] Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., Stuetzle, W., 1995. Multiresolution analysis of arbitrary meshes. In: SIGGRAPH. pp. 173–182.

[12] Fisher, M., Schröder, P., Desbrun, M., Hoppe, H., 2007. Design of tangent vector fields. ACM Trans. on Graphics 26 (3), 56:1–56:9.

[13] Floater, M., Hormann, K., 2002. Parameterization of Triangulations and Unorganized Points. Tutorials on Multiresolution in Geometric Modelling. Springer-Verlag, Heidelberg, pp. 287–316.

[14] Gu, X., Yau, S.-T., 2003. Global conformal surface parameterization. In: Proc. of Symp. on Geometry Processing. pp. 127–137.

[15] Hertzmann, A., Zorin, D., 2000. Illustrating smooth surfaces. In: SIGGRAPH. pp. 517–526.

[16] Huang, J., Zhang, M., Ma, J., Liu, X., Kobbelt, L., Bao, H., 2008. Spectral quadrangulation with orientation and alignment control. ACM Trans. on Graphics 27 (5), 147:1–147:9.

[17] Irving, G., Teran, J., Fedkiw, R., 2004. Invertible finite elements for robust simulation of large deformation. In: Proc. of Symp. on Computer Animation. pp. 131–140.

[18] James, D. L., Pai, D. K., 1999. Accurate real time deformable objects. In: SIGGRAPH. pp. 65–72.

[19] Joshi, P., Meyer, M., DeRose, T., Green, B., Sanocki, T., 2007. Harmonic coordinates for character articulation. ACM Trans. on Graphics 26 (3), 71:1–71:9.

[20] Kanai, T., Suzuki, H., Kimura, F., 1997. 3d geometric metamorphosis based on harmonic map. In: Proc. of Pacific Graphics. pp. 97–104.

[21] Kobbelt, L., Campagna, S., Vorsatz, J., Seidel, H.-P., 1998. Interactive multi-resolution modeling on arbitrary meshes. In: SIGGRAPH. pp. 105–114.

[22] Kraevoy, V., Sheffer, A., Gotsman, C., 2003. Matchmaker: Constructing constrained texture maps. In: SIGGRAPH. pp. 326–333.

[23] Martin, T., Cohen, E., Kirby, M., 2008. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. In: Proc. of ACM Symp. on Solid and Physical Modeling. pp. 269–280.

[24] Meyer, M., Desbrun, M., Schröder, P., Barr, A. H., 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In: Visualization and Mathematics III. pp. 35–57.

[25] Müller, M., Heidelberger, B., Teschner, M., Gross, M., 2005. Meshless deformations based on shape matching. In: SIGGRAPH. pp. 471–478.

[26] Ni, X., Garland, M., Hart, J. C., 2004. Fair morse functions for extracting the topological structure of a surface mesh. ACM Trans. on Graphics 23 (3), 613–622.

[27] Pinkall, U., Polthier, K., 1993. Computing discrete minimal surfaces and their conjugates. Experimental Mathematics 2 (1), 15–36.

[28] Praun, E., Finkelstein, A., Hoppe, H., 2000. Lapped textures. In: SIGGRAPH. pp. 465–470.

[29] Schall, O., Zayer, R., Seidel, H.-P., 2008. Controlled field generation for quad-remeshing. In: Proc. of ACM Symp. on Solid and Physical Modeling. pp. 295–300.

[30] Sorkine, O., Cohen-Or, D., 2004. Least-squares meshes. In: Proc. IEEE Conf. on Shape Modeling and Applications. pp. 191–199.

[31] Sorkine, O., Cohen-Or, D., Irony, D., Toledo, S., 2006. Geometry-aware bases for shape approximation. IEEE Trans. Vis. & Comp. Graphics 11 (2), 171–180.

[32] Sorkine, O., Lipman, Y., Cohen-Or, D., Alexa, M., Rössl, C., Seidel, H.-P., 2004. Laplacian surface editing. In: Proc. of Symp. on Geometry Processing. pp. 179–188.

[33] Tong, Y., Alliez, P., Cohen-Steiner, D., Desbrun, M., 2006. Designing quadrangulations with discrete harmonic forms. In: Proc. of Symp. on Geometry Processing. pp. 201–210.

[34] Turk, G., 2001. Texture synthesis on surfaces. In: SIGGRAPH. pp. 347–354.

[35] Vallet, B., Lévy, B., 2008. Spectral geometry processing with manifold harmonics. Computer Graphics Forum (Special Issue of Eurographics) 27 (2), 251–260.

[36] von Golitschek, M., Schumaker, L. L., 1990. Data fitting by penalized least squares. In: Algorithms for Approximation II. pp. 210–227.

[37] Wang, Y., Gupta, M., Zhang, S., Wang, S., Gu, X., Samaras, D., Huang, P., 2005. High resolution tracking of non-rigid 3D motion of densely sampled data using harmonic maps. In: Proc. Int. Conf. on Comp. Vis. pp. 388–395.

[38] Wardetzky, M., Mathur, S., Kälberer, F., Grinspun, E., 2007. Discrete laplace operators: No free lunch. In: Proc. of Symp. on Geometry Processing. pp. 33–37.

[39] Wei, L.-Y., Levoy, M., 2001. Texture synthesis over arbitrary manifold surfaces. In: SIGGRAPH. pp. 355–360.

[40] Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., Shum, H.-Y., 2004. Mesh editing with poisson-based gradient field manipulation. ACM Trans. on Graphics 23 (3), 644–651.

[41] Zayer, R., Rössl, C., Karni, Z., Seidel, H.-P., 2005. Harmonic guidance for surface deformation. Computer Graphics Forum (Special Issue of Eurographics), 601–609.

[42] Zhang, E., Mischaikow, K., Turk, G., 2006. Feature-based surface parameterization and texture mapping. ACM Trans. on Graphics 24 (1), 1–27.

[43] Zhang, E., Mischaikow, K., Turk, G., 2006. Vector field design on surfaces. ACM Trans. on Graphics 25 (4), 1294–1326.

[44] Zhang, H., Sheffer, A., Cohen-Or, D., Zhou, Q., van Kaick, O., Tagliasacchi, A., 2008. Deformation-drive shape correspondence. Computer Graphics Forum (Special Issue of Symp. on Geometry Processing) 27 (5), 1431–1439.

[45] Zienkiewicz, O. C., 1974. Constrained variational principles and penalty function methods in finite element analysis. Lecture Notes in Mathematics 363, 207–214.