Global Parametrization of Range Image Sets

Nico Pietroni[1]   Marco Tarini[1,2]   Olga Sorkine[3,4]   Denis Zorin[4]
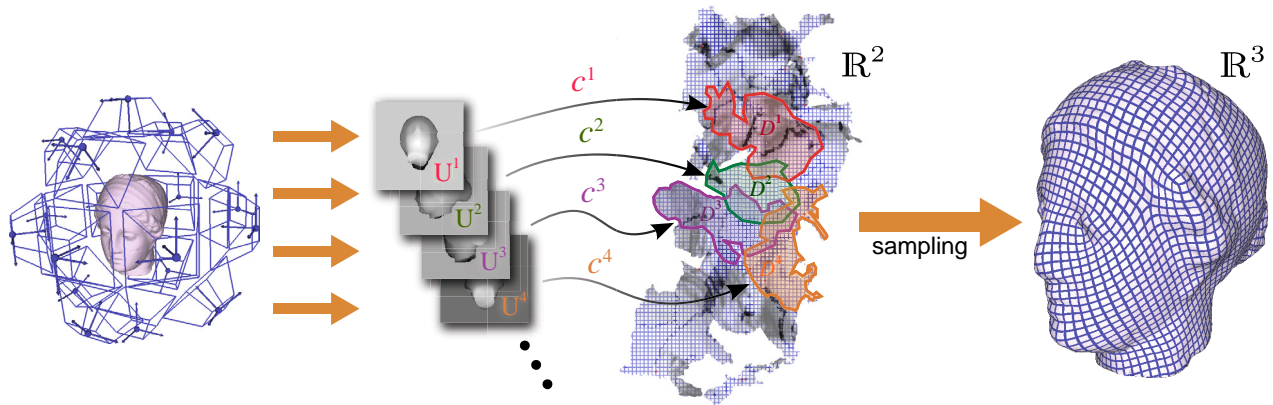
[1]ISTI, CNR, Italy   [2]University of Insubria, Italy   [3]ETH Zurich   [4]New York University

Figure 1: *A range image set and its parametrization. A set of range images $\mathcal{U}^i$ is captured by direct measurement of real world objects or by means of digital rendering of virtual objects. Each range scan $\mathcal{U}^i$ is mapped over a domain $\mathcal{D}^i$ by a parametrization function c^i, in a globally consistent way. Among other uses, a semi-regular quad remeshing of the original object can be obtained by regularly sampling $\cup \mathcal{D}^i$.*

Abstract

We present a method to globally parameterize a surface represented by height maps over a set of planes (range images). In contrast to other parametrization techniques, we do not start with a manifold mesh. The parametrization we compute defines a manifold structure, it is seamless and globally smooth, can be aligned to geometric features and shows good quality in terms of angle and area preservation, comparable to current parametrization techniques for meshes. Computing such global seamless parametrization makes it possible to perform quad remeshing, texture mapping and texture synthesis and many other types of geometry processing operations. Our approach is based on a formulation of the Poisson equation on a manifold structure defined for the surface by the range images. Construction of such global parametrization requires only a way to project surface data onto a set of planes, and can be applied directly to implicit surfaces, nonmanifold surfaces, very large meshes, and collections of range scans. We demonstrate application of our technique to all these geometry types.

CR Categories: I.3.5 [Computational Geometry and Object Modeling]: Geometric algorithms, languages, and systems

Links: ◆DL 📄PDF 🌐WEB

Keywords: geometry processing, parametrization, range scans

1 Introduction

A high-quality global parametrization greatly simplifies many operations on surfaces. Recent techniques made substantial progress in improving the quality and robustness of global parametrization. At the same time, the work on parametrization, with few notable exceptions, focuses on manifold meshes, rather than on other forms of geometric data. In this paper, we describe how global parametrization techniques based on solving the Poisson equation (or another PDE) on the surface can be extended to a surface represented by a set of projections to planes. In some cases (e.g., range scanning) raw surface data is directly given in this format. In many other cases, it can be easily computed from a given arbitrary geometry representation: for example, if a geometry description can be rendered with depth values, it can serve as the input to our algorithm.

Range image sets occupy an intermediate place between point clouds or triangle soups, and manifold meshes. On one hand, they exhibit a regular connectivity and implicitly define a global manifold structure for the object, with transition maps determined by reprojection. On the other hand, each point on the surface may be represented by multiple positions inside different range images, and the connectivities of different range images, while highly regular, are inconsistent with each other.

Our method directly recovers a global parametrization from a range image set, entirely avoiding the need to construct a consistent manifold mesh first; this parametrization itself can be used to create high quality regular meshes. This considerably reduces the complexity of the meshing pipeline, replacing a more difficult step of manifold mesh reconstruction by simple and robust projections, followed directly by parametrization and quadrangulation.

Our method is based on a novel discretization of the *seamless* global parametrization equations and constraints on a collection of overlapping triangles, in contrast to conventional discretization on a single mesh. Our parametrization is globally consistent (images of a point in each range image are assigned the same parametric coordinates), seamless and globally smooth. It has comparable area and angle preservation quality to similar approaches for meshes, and can be aligned to geometric features.

The main steps of our method are shown in Fig. 1. The set of partially overlapping range images covering the surface defines a manifold structure, with transitions determined by compositions of projections from the range images to the surface and back. The equations for the parametrization are discretized on each range image separately, with constraints ensuring that the projections of the same surface point to different range images have the same parametric position, up to an admissible transformation.

We apply our method in a number of scenarios: implicit surfaces, nonmanifold polygonal meshes, range images obtained from a scanner and pointsets (see an example in Fig. 2). In Sec. 8 we demonstrate examples for each setting.

2 Related work

The literature on parametrization and quadrangulation is extensive. We survey the most closely related work, and refer the reader to [Sheffer et al. 2006; Hormann et al. 2007] for broader reviews.

Our work is similar in spirit to the digital geometry processing approach in [Pauly and Gross 2001], where geometric data is processed on individual range images, exploiting their regular sampling. In their method, the results for individual images are combined in a separate phase using blending. In our construction, we eliminate the need for the blending stage which would make it difficult to achieve high parametrization quality.

A work addressing parametrization on surface representations different than polygonal meshes is [Zhang et al. 2010a], which handles the case of general point clouds. The connectivity information is extracted locally using geometric heuristics. A parametrization is found as a 2D embedding which maximizes isometry for connected point pairs. The point cloud is first cut into a disk-like region using geometric proximity-based clustering, and no consistency condition is ensured across cuts. When applied to range images, this approach ignores the implicit connectivity inside each range image. In contrast, in our approach this connectivity is exploited both to define the surface more robustly (relying less on geometric heuristics) and to increase the efficiency. Our parametrization is seamless and globally smooth, and can be aligned to geometric features if the tangent cross-field used to compute it is aligned.

All existing techniques for global parametrization we are aware of work with manifold meshes. A number of methods [Eck et al. 1995; Lee et al. 1998; Khodakovsky et al. 2003; Marinov and Kobbelt 2005; Daniels et al. 2009a; Daniels et al. 2009b; Pietroni et al. 2010; Tarini et al. 2010] use simplification techniques for constructing a conforming domain mesh. These techniques make it possible to obtain very coarse domain meshes, with good user control over the domain mesh size. While some degree of feature alignment is possible (see [Lee et al. 1998; Marinov and Kobbelt 2005]), it is difficult to preserve features during simplification. Other methods use global harmonic or conformal parametrizations with singularities [Gu and Yau 2003; Dong et al. 2006; Tong et al. 2006; Ben-Chen et al. 2008; Springborn et al. 2008; Kovacs et al. 2009]. While some of these methods offer a degree of control over the size and structure of the domain mesh (e.g., [Dong et al. 2006]), feature alignment is limited to determining positions of parametrization singularities. Huang et al. [2008] describe an algorithm for adding alignment and orientation control to the parametrization, but the domain mesh is still constructed independently of geometry. A recent technique of [Zhang et al. 2010b] uses a wave-based approach, which combines some of the features of spectral quadrangulation for feature alignment, and provides anisotropy control.

Following *feature-aligned* algorithms [Ray et al. 2006; Kälberer et al. 2007; Bommes et al. 2009], we find a parametrization of a
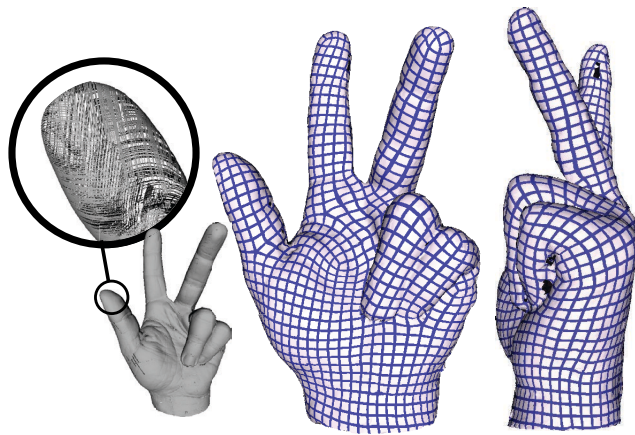
Figure 2: *Parametrization of a point cloud. The initial data is composed of 2.6M points.*

given shape by fitting its gradient to a given smooth *cross-field* capturing surface features (for example, smoothed principal curvature directions). In our case, better results are achieved if the field is defined consistently over the surface represented by range scans. As we discuss in Sec. 7, standard techniques to define a tangent cross-field on a surface can be easily adapted to this setting.

Our formulation is based on defining a manifold structure using transition functions. This type of approach was used to construct smooth surfaces (see e.g. [Grimm and Hughes 1995; Ying and Zorin 2004], and parameterize meshes over canonical domains [Grimm and Hughes 2004]. A more detailed survey of the relevant work can be found in [Grimm and Zorin 2006].

3 Algorithm overview

Global parametrization. For surfaces of topology other than a disk, global parametrization cannot be defined as a continuous embedding into the plane, as no such embedding exists. One common approach to global parametrization is to cut the surface to a disk, logically duplicating the points along each cut, and map the resulting disk to a plane. In the simplest case, no conditions are imposed on the distinct coordinates of the cut vertices, such that the parametrization on different sides of the cut is completely independent (this manifests in e.g. texture mismatch artifacts). In many parametrization applications, such as remeshing, this is not acceptable, and conditions are imposed on the seam to make the parametrization *seamless*. A common condition to require is that, as we cross the boundary, the parametric coordinates change by a $k\pi/2$ rotation and an integer translation (we call such rigid transforms *admissible*). In particular, these constraints make it possible to cover the surface with seamlessly matching quads, for the purposes of semi-regular remeshing or seamless texturing by rectangular texture maps. Typically, it is desirable for the parametrization to be as close as possible to isometric (i.e., preserve angles and areas), smooth, and with parametric lines aligned to features.

Following [Kälberer et al. 2007; Bommes et al. 2009], we compute the parametrization (u, v) as the constrained least-squares fit of parametric gradients to the target directions, obtained on each triangle by computing a cross-field on the surface:

$$E(q) = \sum_T A_T \|\nabla q^T - \mathbf{w}^T\|^2 \to \min,\qquad(1)$$

where the summation is over all triangles, $q = u + \sqrt{-1}v$ is the complex form of parametric coordinates, and $\mathbf{w}^T = \mathbf{u}^T + \sqrt{-1}\mathbf{v}^T$
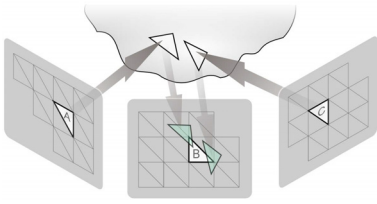
Figure 3: *Range image sets and overlapping triangles.*

D^i	a chart domain in a 2D range image (2D triangles in the discretization)
U^i	a chart domain on the surface (3D triangles in the discretization)
p	a point in D^i
$\bar{p}$	the corresponding 3D point in U^i
c^i	the map from U^i to D^i (projection P^i to the range image plane)
τ^{ij}	the transition map $c^j \circ (c^i)^{-1}$ between D^i and D^j,
q^i	the parametrization function for D^i mapping points to the parametric plane
q^i_m	the parametric value for corner m ($m = 1, 2, 3$) of a triangle
X^{ij}	the admissible transition function for parametrizations q^i and q^j of D^i: $q^j = X^{ij} q^i$, rotation r^{ij} by a multiple of $\pi/2$, and the integer translation t^{ij}
$\mathbf{w}^T$	the complex vector given by $\mathbf{u}^T + \sqrt{-1}\mathbf{v}^T$, where $\mathbf{u}^T$ and $\mathbf{v}^T$ are the two target parametrization gradients.

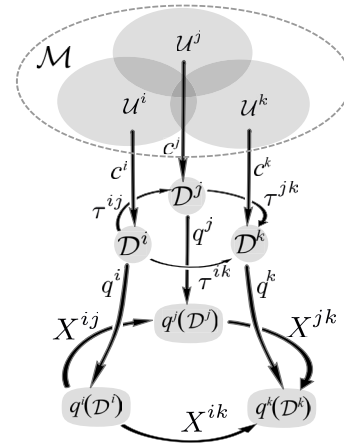**Table 1:** *The main symbols used in the paper.*

Figure 4: *Manifold charts, transition maps and parametrizations.*

is a complex vector representing the two target directions $\mathbf{u}^T$ and $\mathbf{v}^T$ for the parametrization gradients. Minimizing (1) amounts to solving a Poisson equation with special boundary conditions.

For points in different range images that correspond to the same point on the surface, we need to impose constraints on parametric positions to ensure a consistent parametrization: we require that these points are related by admissible transforms.

If arbitrary admissible transforms are allowed, the solution is not unique. Furthermore, the number of additional unknowns corresponding to these transformations can be very large. We describe an algorithm to efficiently eliminate most of these unknowns and ensure that the transformations can be determined consistently.

Once the Poisson equation discretization and a reduced system of constraints is constructed, the resulting linear system yields a parametrization of each range image which is consistent, up to an error dependent on the triangle size, with the parametrization of other range images.

Input data. The input to our algorithm consists of:

- A set of range images $\mathcal{S}^i$ of a surface (Fig. 3), i.e., rectangular arrays of scalar distances d^i_m from a uniform grid of positions $p^i_m = (x^i_m, y^i_m)$ in the plane of $\mathcal{S}^i$. Some values may be missing and are not used in the algorithm. Valid samples are connected into a uniform triangulation.

- A *cross-field*, i.e. four orthogonal unit-length vectors $\mathbf{u}_T, \mathbf{v}_T, -\mathbf{u}_T, -\mathbf{v}_T$ in the plane of each 3D triangle T of each range image. We explain how the cross-field can be constructed in Sec. 7. Our algorithm does not make any assumptions about the choice of the field, but the quality of the result depends on this choice.

4 Parameterizing a manifold structure

Before explaining how the discrete problem is formulated and solved, we start with a more abstract setting, to express the main ideas in a concise form.

A global seamless parametrization can be defined for a manifold $\mathcal{M}$ covered by overlapping sets $\mathcal{U}^i \subset \mathcal{M}$, with chart maps $c^i : \mathcal{U}^i \to \mathbb{R}^2$ mapping $\mathcal{U}^i$ to domains $\mathcal{D}^i = c^i(\mathcal{U}^i)$ in the plane (Fig. 4). In our method, $\mathcal{U}^i$ corresponds to triangles in $\mathbb{R}^3$, $\mathcal{D}^i$ to the triangles in range images and c^i to the projections (the notation used from this point on is summarized in Table 1).

The transition map between two charts $\mathcal{D}^i$ and $\mathcal{D}^j$ is denoted $\tau^{ij} = c^j \circ (c^i)^{-1}$, defined on $c^i(\mathcal{U}^i \cap \mathcal{U}^j)$. Transition maps τ^{ij} satisfy the standard *cycle condition* by construction:

$$\tau^{ik} = \tau^{jk} \circ \tau^{ij}, \quad \text{for points in } c^i(\mathcal{U}^i \cap \mathcal{U}^j \cap \mathcal{U}^k). \quad (2)$$

The *overlap graph* $G_{\text{ov}}(\mathcal{M})$ for an atlas has the domains $\mathcal{U}^i$ as nodes and an edge (i, j) connecting every two overlapping domains $\mathcal{U}^i$ and $\mathcal{U}^j$.

Conventional parametrization is defined as a map from $\mathcal{M}$ to the plane. In contrast, we define the parametrization on *charts*: for each $\mathcal{D}^i$, we define a parametrization map $q^i : \mathcal{D}^i \to \mathbb{R}^2$.

If $\mathcal{U}^i$ and $\mathcal{U}^j$ overlap, a point $p = c^i(\bar{p}) \in \mathcal{D}^i$ and another point $c^j(\bar{p}) = \tau^{ij}(p)$ in $\mathcal{D}^j$ correspond to the same point $\bar{p}$ on $\mathcal{M}$. We require that the parametric coordinates $\bar{p}$ gets in $\mathcal{D}^i$ and $\mathcal{D}^j$ differ by an admissible transformation X^{ij} (rotation by $k\pi/2$ + translation). This condition can be expressed as

$$q^j(\tau^{ij}(p)) = X^{ij}(q^i(p)) \quad (3)$$

for all pairs of overlapping domains $(\mathcal{U}^i, \mathcal{U}^j)$, and all points with $\bar{p} \in \mathcal{U}^i \cap \mathcal{U}^j$.

It is particularly convenient for our purposes to express parametrization using complex numbers. The admissible transforms X in this case all have the simple form $rq + t$, where r is the complex number corresponding to a $k\pi/2$ rotation (± 1 or $\pm\sqrt{-1}$) and t is a complex integer. *To simplify the presentation of the main idea, we assume that X^{ij} are fixed and known; we will later demonstrate how X^{ij} can be computed.*

Observe that compositions $q^i \circ c^i$ with domains $q^i(\mathcal{D}^i)$ form a new atlas for $\mathcal{M}$; the admissible transforms X^{ij} are the transition maps for this atlas. Therefore, they have to satisfy their own cycle condition, which we call *parametrization cycle condition*:
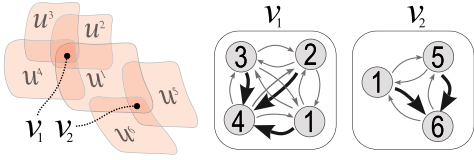
Figure 5: *Overlap graph G_{ov} for an atlas; a set of non-redundant constraints is shown with thick lines.*

$$X^{ik} = X^{ij} \circ X^{jk}, \text{ or equivalently}$$
$$r^{ik} = r^{ij}r^{jk}, \ t^{ik} = r^{jk}t^{ij} + t^{jk}. \tag{4}$$

These relations for admissible transform compatibility are central to our system construction.

With this notation, our general problem can be formulated as follows:

For a given set of admissible transforms X^{ij} satisfying the parametrization cycle constraints, find parametrizations q^i of domains $\mathcal{D}^i$, by minimizing an energy $\sum^i E(q^i)$ subject to the constraint (3) satisfied for all pairs of charts $(\mathcal{D}^i, \mathcal{D}^j)$, for points $\bar{p} \in \mathcal{U}^i \cap \mathcal{U}^j$.

Symmetrized constraints. The number of constraints between all pairs of overlapping charts is very large (recall that our charts are individual triangles); however most of them are made redundant by the cycle conditions. This can be understood as follows. We partition $\mathcal{M}$ into subdomains $\mathcal{V}_\ell$, such that all points in $\mathcal{V}_\ell$ share the same set I_ℓ of overlapping domains (Fig. 5). Let M_ℓ be the size of I_ℓ. For each point in $\mathcal{V}_\ell$, we have $M_\ell(M_\ell - 1)$ pairwise constraints (3). If cycle conditions are satisfied, we only need $M_\ell - 1$ constraints, obtained, e.g., by fixing a domain $\mathcal{U}^0 \in I_\ell$, and only considering constraints for pairs $(\mathcal{U}^0, \mathcal{U}^i)$, $i \neq 0$ (Fig. 5).

Instead of using an independent subset of constraints (3), we use the following M_ℓ *symmetrized constraints*: for each $\mathcal{V}_\ell$ and each $j \in I_\ell$, we require

$$q^j(p) = \frac{1}{M_\ell - 1} \sum_{m \neq j} X^{mj} q^m(\tau^{jm}(p)) \tag{5}$$

Proposition 1 *The constraint system* (3) *and the symmetrized system* (5) *are equivalent, if parametrization cycle conditions* (4) *are satisfied.*

This proposition is proved in the Appendix. The symmetrized system is still redundant, but much less so (only one unnecessary constraint per $\mathcal{V}_\ell$). This is the form we use in our computations, as explained in greater detail in the next section.

Singularities. For a manifold $\mathcal{M}$ of arbitrary genus, a seamless parametrizations may not exist: one easy way to see this is to observe that the admissible transition maps are isometric, so the parametrization defines a flat metric over the whole surface which may not exist, as the total Gaussian curvature needs to satisfy the Gauss-Bonnet theorem. A finite number of points (*singularities*) need to be excluded; we obtain these as the singularities of the field $\mathbf{w}$.

5 Discrete constraints and energy

We derive a discrete form of the problem introduced in the previous section, using the collection of triangles from all range images as the domains $\mathcal{U}^i$.

Notation. A triangle T lies on the plane $\pi(T)$, and a range image $\mathcal{S}$ is defined on the plane $\pi(\mathcal{S})$. For each range image, P^i is the orthogonal projection $P^i : \mathbb{R}^3 \to \pi(\mathcal{S}^i)$. For a range image $\mathcal{S}$, we distinguish between 3D triangles $\bar{T}$, with vertices $\bar{p}_m^T$, and triangles T in the plane of the range image with vertices p_m^T ($m \in \{1, 2, 3\}$).

We say that two triangles from the same range image are *edge-overlapping*, if they share an edge, and *vertex-overlapping* if they share a vertex, but not an edge. Two triangles T^1 and T^2 from two distinct range images $\mathcal{S}^1$ and $\mathcal{S}^2$ are termed *overlapping* if: (i) the projection $P^1(\bar{T}^2)$ overlaps T^1 in $\mathcal{S}^1$ (Fig. 5); (ii) the distance between triangle planes in 3D is less than a threshold ϵ_d, and (iii) the difference of normal directions in 3D is less than a threshold ϵ_n. These thresholds are the only parameters of the algorithm, in addition to the the target cross-field.

An initial set of overlaps determined using thresholds may not define a proper manifold (specifically, it may not be a *Hausdorff space*, cf. [Grimm 1996]). A simple example of this situation is if two triangles T_1 and T_1' are detected as overlapping for a triangle T_0, and the overlap areas intersect in T_0, but the pair (T_1, T_1') does not meet the overlap criterion. While this situation does not preclude computing a global parametrization, it may lead to nonmanifold quadrangulations, so we eliminate the problem by computing a closure of the overlap relation: if triangles T and T' overlap T_0, and the overlap areas intersect, we consider T and T' to be overlapping, even if the threshold conditions are not satisfied.

Conceptually, each triangle is parameterized separately: for each corner p_s^T of a triangle T we compute a parametric position $q_s^T = (u_s^T, v_s^T)$ (in practice most of these variables can be safely eliminated as described below).

We specialize the condition (3) to the case when $\mathcal{U}^i$ are triangles[1].

Our parametrizations on each triangle are linear, so we can formulate all constraints just in terms of values q_s^T of the parametrization at triangle corners.

Transition maps. The maps c^i are approximated by the projections P^i, so the transition map between triangles T^i and T^j from range maps i and j is approximated by $\tau^{ij} = P^j \circ (P^i)^{-1}$.

Next, we define a discrete version of the constraints (5). Importantly, due to piecewise linear discretization, only constrains between triangle pairs of the same range image can be imposed pointwise. Therefore we treat two cases separately: constraints between triangles from different range images are imposed in an averaged sense using a finite-element discretization, and constraints between edge-overlapping triangles are imposed exactly.

Constraints between triangles from different range images. As our parametrizations is linear inside each triangle, we can express each parametrization q^i in terms of linear basis functions $\phi_m(p)$, where $m = 1, 2, 3$ is the corner index in the triangle. Note that maps τ^{ij} are also linear. Since the uniform triangulation of the range image $\mathcal{S}^i$ and the triangulation obtained by remapping the triangulation of $\mathcal{S}^j$ to $\mathcal{S}^i$ do not generally coincide, we adopt a finite-element approach: we do not enforce the constraints pointwise; rather, we require that L^2 products with the hat basis functions agree, which yields the following expression:

$$q_m^i = \sum_k \frac{1}{M_i - 1} \sum_{j \in O_m} X^{ji} \sum_{n=1}^3 w_{mn}^{ij} q_n^j, \tag{6}$$

---

[1]Strictly speaking, $\mathcal{U}^i$ are required to be open sets; for rigorous treatment, we can extend each 3D triangle to an open set on the manifold $\mathcal{M}$ with singularities using an arbitrarily small extension beyond its boundary. In this way, any triangles sharing a vertex have a small overlap at the vertex unless the vertex is a singularity.

where the index sets O_m consist of indices of triangles T^j for which the remapped triangle $\tau^{ji}(T^j)$ in $\mathcal{S}^i$ overlaps T^i, and M_i is the number of triangles from other range images overlapping T_i; the weights w_{mn}^{ij} are obtained by integration (see Appendix).

This defines the complete set of constraints for triangles from different range images. As explained above, for a given set of mutually overlapping triangles, this set of constraints is redundant (there is one unneeded constraint). In our implementation, we retain the redundancy, and enforce the constraints with a penalty method. We found that the method is not sensitive to the choice of the penalty parameter (see Sec. 8); the corresponding penalty term is

$$E^{\text{pen}} = \sum_{T_i} \sum_{m=1}^{3} \left(q_m^i - \frac{1}{M_i - 1} \sum_{T_j \in O(T_i)} X^{ji} \sum_{n=1}^{3} w_{mn}^{ij} q_n^j \right)^2 \tag{7}$$

Constraints between triangles in the same range image. Constraints for edge-overlapping or vertex-overlapping triangles directly follow from (3) (similarly to [Kälberer et al. 2007; Bommes et al. 2009]).

For edge-overlapping triangles, the overlap area is a single line segment. If the shared vertices have indices 1 and 2 in both triangles, then we have

$$q_s^j = X^{ij} q_s^i = r^{ij} q_s^i + t^{ij}, \quad \text{for } s = 1, 2. \tag{8}$$

One can show that the constraints on vertex-overlapping triangles T_i and T_j follow from the the constraint on the chain of edge-overlapping triangles leading from i to j around the common vertex, hence we will not consider it further.

Discrete energy. For a single range image, the energy is just a standard finite-element discretization with the well-known cotangent weights. On a triangle T with vertex parametrization variables $q_m, m = 1, 2, 3$ and edge vectors e_m of the 3D triangle $\bar{T}$, the per-triangle energy (1) can be written as

$$E_T = A_{\bar{T}} \left(\sum_{m=1}^{3} \frac{e_m^{\perp}}{2A_{\bar{T}}} q_m - \mathbf{w}^T \right)^2,$$

yielding the expressions $(e_m \cdot e_n)/4A_{\bar{T}}$ for matrix coefficient (m, n) and $(e_m^{\perp} \cdot \mathbf{w}^T)/2$ for the right-hand side entry m.

In our case however, each point on the surface may be contained in multiple triangles. If the energy terms above are simply summed up to obtain the total energy, the energy density at points will depend on the number of overlapping triangles. To obtain a proper uniform weighting, we compute the per-triangle energy density $d_T = E_T/A_{\bar{T}}$ as above, and then we compute the weight for this density, which is in general different from the area of the triangle.
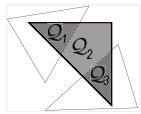
For a triangle T^0, consider all triangles T^i from other range images overlapping it. Intersections of T^0 and $\tau^{i0}(T^i)$ partition T^0 into a collection of polygonal domains $\mathcal{Q}_m$ (see inset), such that on each domain there is a fixed set of overlapping triangles. If the domain $\mathcal{Q}_m$ has N_m overlapping triangles, then we can average the densities from all triangles on this domain, and scale it by its area to get the energy: the 3D image of $\mathcal{Q}_m$ is $A(\mathcal{Q}_m)\frac{1}{N_m} \sum_{\ell} d_{\bar{T}^{\ell}}$, where the sum is over all triangles overlapping $\mathcal{Q}_m$, and $A(\mathcal{Q}_m)$ is the area.

Writing the total energy of the parametrization as a double sum over all domains $\mathcal{Q}_m^T$ and all triangles T, and rearranging, we get

$$E^{\text{total}} = \sum_{T} \left(\sum_{m} A(\mathcal{Q}_m^T)/N_m^T \right) d_T. \tag{9}$$

So to calculate the proper scaling factors for each triangle energy term, we need to compute the areas $A(\mathcal{Q}_m)$ and the numbers of triangles N_m overlapping each domain $\mathcal{Q}_m$. These areas can be efficiently approximated using hardware rendering.

The complete discrete problem. For fixed transforms $X^{ij} = (r^{ij}, t^{ij})$ for any pair of overlapping triangles (from the same or different range images) the discrete optimization problem is:

Minimize $E^{\text{total}} + \omega E^{\text{pen}}$ w.r.t. parametric positions q_m^i, subject to constraints (8) *for each edge.*

Variable translations. Up to this point, we have assumed the transforms X^{ij} to be known. In reality they are adaptively determined by the optimization to maximize the quality of the parametrization. As in [Kälberer et al. 2007; Bommes et al. 2009], the *rotational* parts r^{ij} of X^{ij} are determined by the input field (see below). The translational parts t^{ij} can be kept free, yielding to additional variables in the optimization process. As these must be integer, this becomes a *mixed-integer* problem, which we solve using the COMiSo solver from [Bommes et al. 2009]:

Minimize $E^{\text{total}} + \omega E^{\text{pen}}$ w.r.t. parametric positions q_m^i and integer translations t^{ij} subject to constraints (8) *for each edge.*

The minimum in general is not unique with respect to changes of some of the translations; in the next section, we describe how the number of variables is reduced, both to make the solution unique and to make the method practical.

6 Constraint setup and elimination

To define the constraints, we need to specify how these are obtained from a cross-field. If the constraints are implemented in the most direct form, with a constraint for each pair of overlapping triangles (T_i, T_j), the number of variables is very large (3 variables per triangle, and one translation variable for each triangle overlap); our goal is to reduce the number of variables to a number closer to the number of vertices in the range scan collection.

6.1 Rotational transforms from a cross-field

The cross-field is completely determined by $\mathbf{u}^T$ for each triangle (the other 3 cross-field vectors for T are obtained by rotation). The cross-field does not change if one of the other vectors is labeled $\mathbf{u}^T$. We do not assume any conditions on the initial labeling, but instead choose the labels so that the number of nonzero rotations r^{ij} is minimized.

Since the energy we use to compute the parametrization minimizes the difference between parametrization gradients and target vectors $\mathbf{w}^T$, and we expect the parametrization to be smooth, $\mathbf{u}^{T_j}$ in a triangle T_j should be as close as possible to $r^{ij} \mathbf{u}^{T_i}$ if T_i and T_j overlap. To set most r^{ij}s to zero, we construct a maximal spanning tree of the overlap graph G_{ov}, so that its restriction to each connected component of a range image is also a maximal spanning tree, and relabel all cross-fields (i.e., rotate $\mathbf{u}^T$ for all T) so that the rotation to the closest direction is zero across each tree edge. On the remaining edges, the rotation r^{ij} is determined by rotating $\mathbf{u}^{T_i}$ by $r^{ij} = k\pi/2$ to the nearest direction to $\mathbf{u}^{T_j}$. *Singularities* are exactly the points for which the sum of the resulting rotation angles of triangles sharing the vertex is not 0.

The rotations defined in this way do not necessarily satisfy the cycle condition for parametrization (4): there are simple examples when the closest match across overlaps results in a cycle for which the product of rotations is not identity. *In particular, any triangle overlapping a singularity in a different scan results in violation of the cycle condition.* Whenever we find an incorrect cycle condition,

I) 14276 integer variables   II) 4316 integer variables   III) 14 integer variables
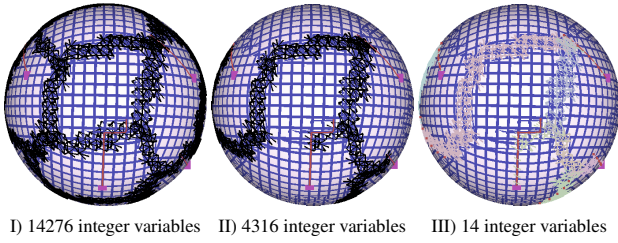
Figure 6: *The three steps of the integer variable reduction process.*

one of the triangles in the cycle is removed. In practice, very few triangles need to be removed. Triangles from the same scan cannot create cycle violations, so in areas covered by a single scan no triangles are removed.

6.2 Constraint and variable elimination

To create our constraint system, in the overlap graph G_{ov}, we label each edge with a translation variable index, unless the edge is labeled as *zero-translation edge* by the maximal spanning tree algorithm for the field, and then find all cycles of length 3 (3-cycles) by considering for each triangle all triangles overlapping it and each other. A constraint is created if there are nonzero translations in the 3-cycle. The constraints on translations due to 3-cycles are given by (4): $t^{ik} = r^{jk}t^{ij} + t^{jk}$.

We reduce the number of translation variables and of constraints in three steps, to drastically reduce the size of the constraint matrix on which Gaussian elimination is performed (see Fig. 6).

Step I: Uniqueness. For translations, we follow the same approach as for rotations to ensure uniqueness: translations across edges of the same maximal spanning tree of the overlap graph are set to zero.

Step II: Cycles with one nonzero translations. If, in a 3-cycle, two translations are zero, then the third one is zero. We iterate over all 3-cycles, and for every cycle with two zero translations, make the remaining edge a zero-translation edge.

Step III: Cycles with two nonzero translations. For these cycles, the constraint is always of the form: $t^{ij} = rt^{lm}$. This type of constraints provides a simple mechanism for variable elimination: Construct a graph of translation variables, with edges corresponding to the constraints. Then for each connected component of this graph, we can choose a single representative variable t^*. The remaining variables are expressed in terms of t^*, using 2-variable cycle constraints. This can be achieved by constructing a tree on the connected component with the root at t^*, doing a breadth-first propagation of coefficients, relating variables to t^*.

The number of constraints and translation variables remaining after this process is relatively small, and depends on the number of overlaps between scans and the number of singularities.

The remaining conditions form a linear system where equations correspond to cycles with three unknown translations. The mixed-integer solver of [Bommes et al. 2009] performs Gaussian elimination on it, identifying an independent set of translation variables.

6.3 Putting it all together

We summarize the overall parametrization process of a collection of range images endowed with cross-fields.

1. set rotation variables based on nearest cross-field direction in overlapping and edge-overlapping triangles;

2. eliminate few triangles to break 3-cycles that do not satisfy the parametric cycle conditions for rotations;

3. reduce the number of nonzero rotation and translation variables;

4. construct the linear system from the energy minimization equations and constraints;

5. solve the system using a mixed integer solver.

7 Preprocessing and quadrangulation

Computing the cross-field on the range scans. Our algorithm uses a cross-field defined over each triangle of the input range scans. We use a technique similar to [Bommes et al. 2009]. While [Bommes et al. 2009] originally targets the case of 2-manifold polygonal meshes, it can be easily extended to the more general case: instead of facet adjacency, we use the overlap graph G_{ov} to define the terms in the energy optimizing the cross-field smoothness (see [Bommes et al. 2009] for details of the energy definition).

Salient features can be identified by estimating principal curvature directions on the point cloud obtained as a union of all range scans.

Producing artificial range images. As discussed in Sec. 1, artificial range scans can be obtained by rendering of practically any form of digital geometric models (such as implicit surfaces, inconsistent meshes, point clouds). In each case, the resulting depth buffers serve as range scans. In this scenario, the global registration is known by construction. Any set of points of view can be chosen to "capture" the data. In most experiments, we used a fixed set of 26 orthographic views with directions uniformly distributed over the sphere, thus guaranteeing that the angle between a surface normal and the closest projection direction never exceeds 28 degrees.

To bypass any self-occlusions problem even in presence of complex shapes, a multi-pass depth-peeling is employed. Specifically, an initial depth image is obtained normally, then a sequence of renderings is performed from the same view direction. At each pass the GPU is instructed to discard fragments closer to the viewer than the previous pass (using a threshold to compensate for noisy or inconsistent data, if necessary). The sequence is stopped when an empty depth buffers is returned. All non-empty depth buffers are fed to the systems as separate range scans.

Processing captured range-scans. When range images are measured from real-world objects, we need to reciprocally align them by means of rigid or deformable registration procedure. The literature offers a wide array of effective alternatives, either user-assisted or fully automatic. In all our experiment we employed the global registration feature of the publicly available and open-source Mesh-Lab [Cignoni et al. 2008].

Quadrangulation. To construct a quadrangulation of a parameterized range image set, we trace isoparametric lines in each scan, and generate a quad for each loop of 4 vertices contained in the interior of a range image. Using the transition functions, overlapping quads are identified in parametric space and merged.

8 Results

In this section, we present an experimental evaluation of our method for a number of scenarios.

Nonmanifold meshes. We successfully tested our method on a number of meshes which often present a problem for parametrization due to topological flaws, such as nonmanifold edges gener-
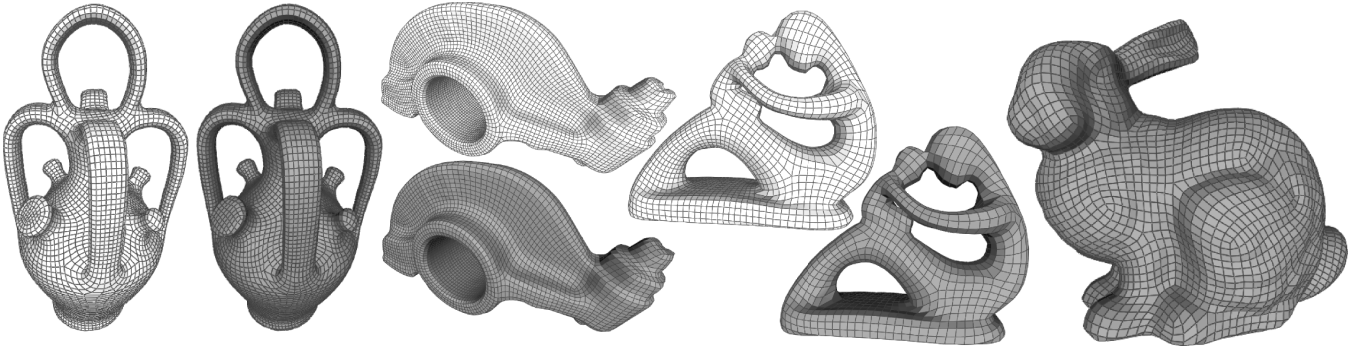
Figure 7: *Comparison of our method (darker meshes) vs. manifold meshes (lighter ones) quadrangulated with mixed integer quadrangulation. For each model, range images where produced (with depth peeling) from 18 view directions, at resolution 200×200.*

$\epsilon_n = 0.9$ $\epsilon_n = 0.6$ $\epsilon_n = 0.3$
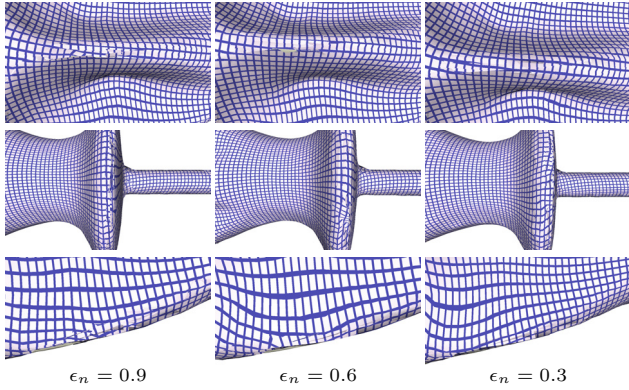
Figure 8: *Dependence of the parametrization on the algorithm parameter ϵ_n. The parameter ϵ_p is fixed at 0.3.*

Figure 9: *Meshes with topological problems and their parametrization (right). Top: a triangle soup obtained by separating triangles and randomly scaling/rotating. Bottom: an input mesh with several nonmanifold edges.*
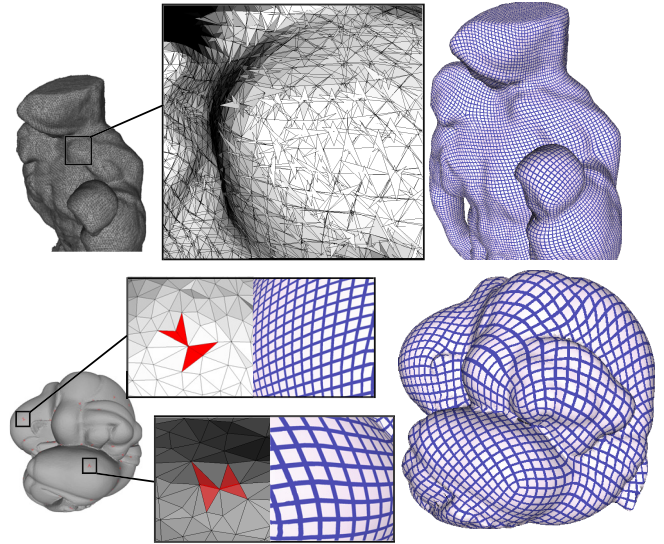
ated in the reconstruction process (Fig. 9, top), and even on triangle soups (Fig. 9, bottom) or point clouds (Fig. 2).

Implicit CSG surfaces. Our method makes it possible to construct manifold quad meshes with few singularities for implicit and CSG surfaces directly from those representations (see Fig. 10). The results can be used for boundary element method simulations and other tasks requiring a boundary representation.

Raw range scan data. Our method is a natural match for range scanning, which naturally produces the needed representation. We have tested our algorithm on several relatively simple examples (Fig. 11). We regard these tests as preliminary, as for complex range scans, a variety of well-known robustness problems need to be addressed, from outliers and noise to systematic scan warping requiring nonrigid registration. Our algorithm can be applied to range scans in two ways: we can use registered camera positions and range scans as is; alternatively, we can treat original scans as a set of overlapping meshes defining the geometry and generate new range images by rendering it.

Large meshes. Thanks to decoupling of target geometry and range image discretization, our method can be used to obtain coarse quadrangulations of large meshes; topology preservation is guaranteed up to range scan resolution, indirectly providing ability to simplify topology. A quad mesh is obtained directly, avoiding intermediate high-resolution surface extraction (Fig. 12).

Comparison with mixed-integer quadrangulation. As our approach most closely resembles mixed-integer quadrangulation, we compare the results of these two methods for several models, see

Fig. 7. It is not our goal to improve on the quality; rather we extend the method to the range image setting. We use the fields provided by the authors of [Bommes et al. 2009] (although, due to the resampling on the range scans, the positions of singularities are slightly shifted).

Dependence on parameters. The main parameters of our algorithm are the overlap detection thresholds ϵ_p, ϵ_n. We found the method to be relatively insensitive to ϵ_p (positional difference): even if set to 0, due to the high probability of intersections between range images, there are enough overlapping triangles. For large ϵ_p

Model	Scan size	Inc.	Red.	Poisson	Total
Rockerarm	150×150	26	4	21	51
Bunny	180×180	45	5	35	85
Igea	180×180	28	5	18	41
Galata	200×200	80	8	30	118

Table 2: *Timings in seconds of the different stages of our algorithm.* Inc. *stands for inconsistencies,* Red. *is variable reduction,* Poisson *is the Poisson equation solve time.*
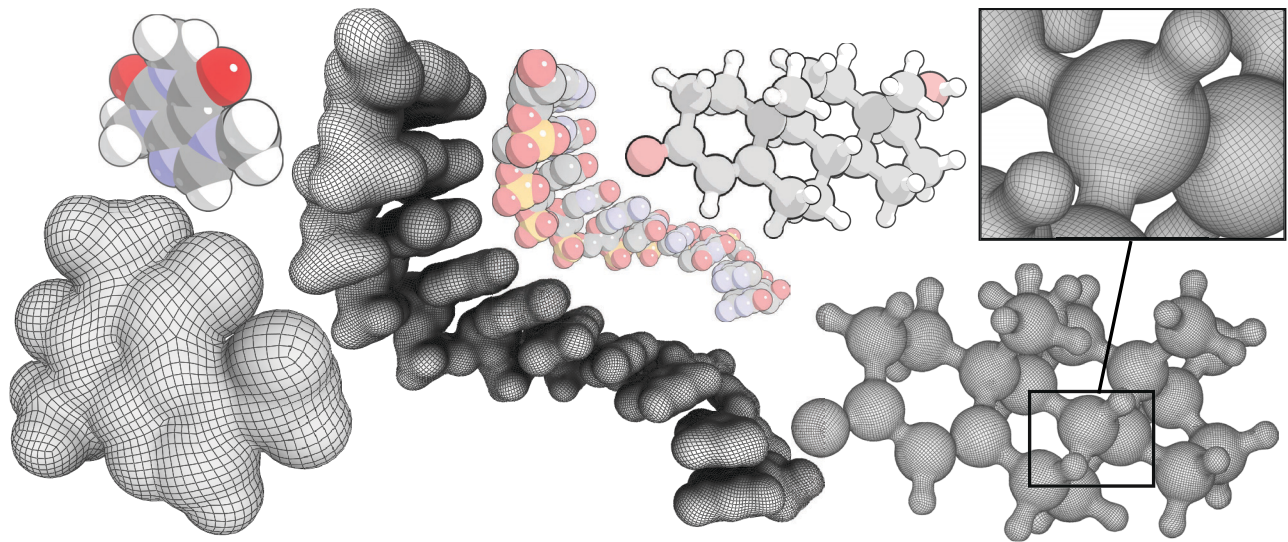
Figure 10: *Quadrangulation of molecular models originally available as implicit surfaces (renderings of the original implicit surface models performed with [Tarini et al. 2006]).*

triangles on opposite sides of thin features may be regarded as overlapping. Fig. 8 shows the dependence on ϵ_n. For noisy data, and low threshold ϵ_n, range images may become disconnected.

Performance. Our current implementation is not optimized for maximizing speed or minimizing memory usage. Table 2 reports timings for the various stages of the algorithm.

Limitations. Unlike reconstruction techniques based on isosurface extraction, in our method the result mesh can present holes which reflect missing data in input scans. If required, however, a hole-free reconstruction can be obtained by feeding to the system range-images obtained by ray-tracing isosurfaces, combining the two techniques. Our method is not sensitive to small amount of noise, but for larger noise the parametrization can become disconnected between range scans (Fig. 13).

9   Conclusion

We have proposed an algorithm for global mesh parametrization, extending previously proposed approaches to the range image setting. The most important advantage of using range images is that they can be obtained from any geometry type that can be rendered, so the complexity and robustness of the parametrization and quadrangulation process is decoupled from the details of a particular surface representation. We believe that this type of decoupling is increasingly essential for applying sophisticated algorithms to large data sets. This paper describes a basic implementation of the method, with a simple treatment of overlap detection (the main aspect of the method determining robustness), and little or no processing of input data. Dealing with less reliable data presents a set of additional challenges to explore in future work. In our implementation we do not take full advantage of the fact that sampling on each range image is regular. One can obtain large improvements in efficiency and scalability, if the data structures and the solver take advantage of regular mesh structure.

Acknowledgments

References

BEN-CHEN, M., GOTSMAN, C., AND BUNIN, G. 2008. Conformal flattening by curvature prescription and metric scaling. *Comp. Graph. Forum 27*, 2, 449–458.

BOMMES, D., ZIMMER, H., AND KOBBELT, L. 2009. Mixed-integer quadrangulation. *ACM Trans. Graph. 28*, 3, 77.

CIGNONI, P., CALLIERI, M., CORSINI, M., DELLEPIANE, M., GANOVELLI, F., AND RANZUGLIA, G. 2008. MeshLab: an open-source mesh processing tool. In *Sixth Eurographics Italian Chapter Conference*, Eurographics, 129–136.

DANIELS, J., SILVA, C., AND COHEN, E. 2009. Semi-regular quadrilateral-only remeshing from simplified base domains. *Comp. Graph. Forum 28*, 5, 1427–1435.

DANIELS, J., SILVA, C. T., AND COHEN, E. 2009. Localized quadrilateral coarsening. *Comp. Graph. Forum 28*, 5, 1437–1444.

DONG, S., BREMER, P., GARLAND, M., PASCUCCI, V., AND HART, J. 2006. Spectral surface quadrangulation. *ACM Trans. Graph. 25*, 3, 1057–1066.

ECK, M., DEROSE, T., DUCHAMP, T., HOPPE, H., LOUNSBERY, M., AND STUETZLE, W. 1995. Multiresolution analysis of arbitrary meshes. In *Proc. ACM SGGRAPH*, 173–182.

GRIMM, C., AND HUGHES, J. 1995. Modeling surfaces of arbitrary topology using manifolds. In *Proc. ACM SIGGRAPH*, 359–368.

GRIMM, C., AND HUGHES, J. 2004. Parameterization using manifolds. *International Journal of Shape Modeling 10*, 1, 51–82.

GRIMM, C., AND ZORIN, D. 2006. Surface modeling and parameterization with manifolds. In *ACM SIGGRAPH 2006 Courses*, 1–81.
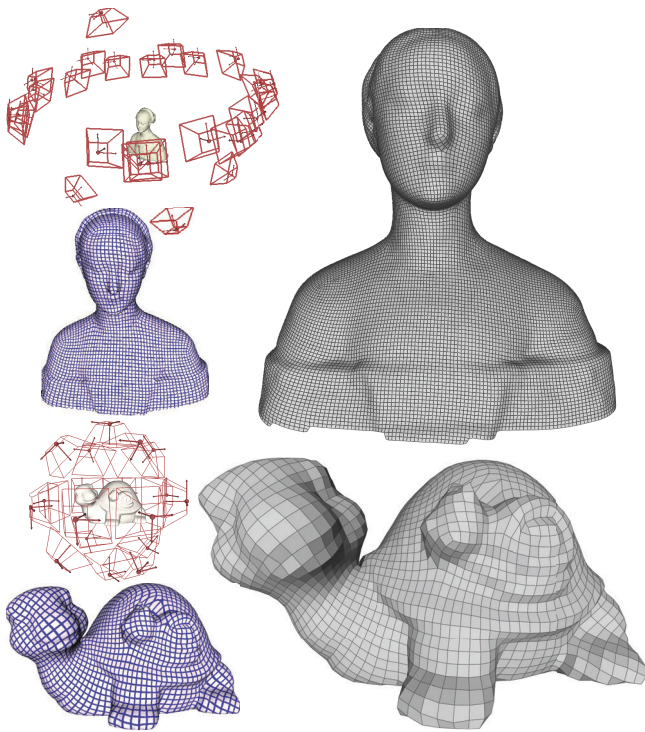
Figure 11: *Parametrization and quadrangulation of range scan sets. Top: a statue acquired with a laser scanner and the original range map positions. Bottom: an experiment using the aligned data coming from the laser scanner. This data has been resampled in screen space and merged into a single mesh.*

GRIMM, C. 1996. *Modeling Surfaces of Arbitrary Topology using Manifolds*. PhD thesis, Brown University.

GU, X., AND YAU, S. 2003. Global conformal surface parameterization. In *Symposium on Geometry Processing*, 127–137.

HORMANN, K., LÉVY, B., AND SHEFFER, A. 2007. Mesh parameterization: Theory and practice. *SIGGRAPH Course Notes*.

HUANG, J., ZHANG, M., MA, J., LIU, X., KOBBELT, L., AND BAO, H. 2008. Spectral quadrangulation with orientation and alignment control. In *Proc. ACM SIGGRAPH*.

KÄLBERER, F., NIESER, M., AND POLTHIER, K. 2007. Quad-Cover: Surface Parameterization using Branched Coverings. *Comp. Graph. Forum 26*, 3, 375–384.

KHODAKOVSKY, A., LITKE, N., AND SCHRÖDER, P. 2003. Globally smooth parameterizations with low distortion. *ACM Trans. Graph. 22*, 3, 350–357.

KOVACS, D., MYLES, A., AND ZORIN, D. 2009. Anisotropic harmonic quadrangulation. In *Symposium on Geometry Processing 2009 Poster*.

LEE, A., SWELDENS, W., SCHRÖDER, P., COWSAR, L., AND DOBKIN, D. 1998. MAPS: multiresolution adaptive parameterization of surfaces. In *Proc. ACM SIGGRAPH*, 95–104.

MARINOV, M., AND KOBBELT, L. 2005. Automatic generation of structure preserving multiresolution models. *Comp. Graph. Forum 24*, 3, 479–486.

PAULY, M., AND GROSS, M. 2001. Spectral processing of point-sampled geometry. In *Proc. ACM SIGGRAPH*, 379–386.
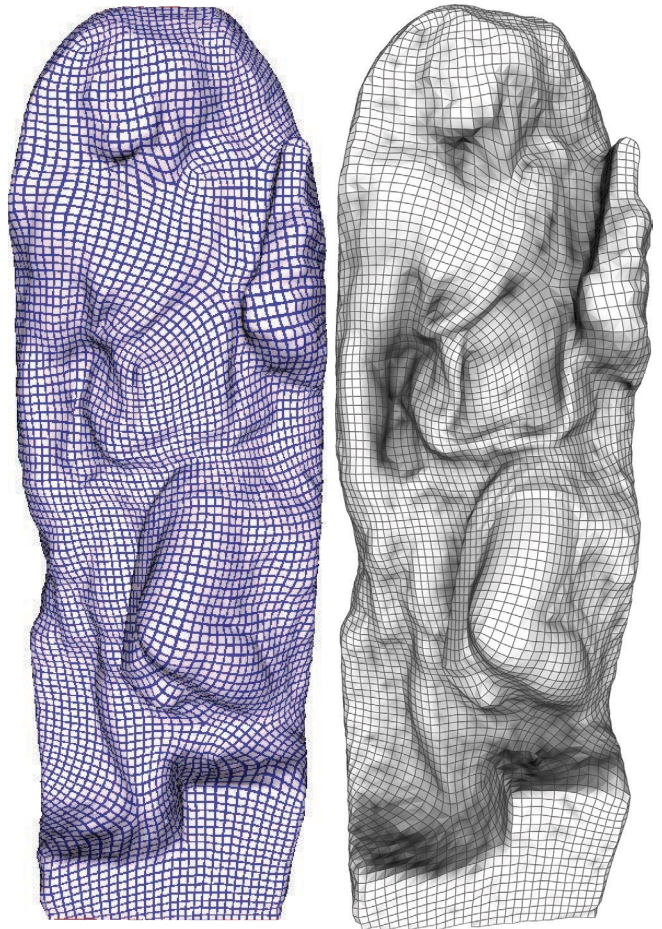
Figure 12: *Large meshes and their quadrangulations obtained using our algorithm. Top: the* Dying Galata*, the initial dataset is composed of 2M triangles and is remeshed to a 6.7K quad mesh. Bottom: the* Saint Matthew *(originally 8M triangles), its parametrization and the final remeshing at 12K quads.*
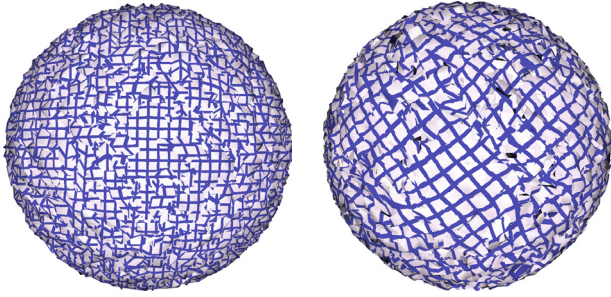
Figure 13: *Results with noisy data (left) and incomplete coverage (right). Noise was added to the depth values of the range scans.*

PIETRONI, N., TARINI, M., AND CIGNONI, P. 2010. Almost isometric mesh parameterization through abstract domains. *IEEE Trans. Vis. and Comp. Graph. 16*, 4, 621–635.

RAY, N., LI, W., LÉVY, B., SHEFFER, A., AND ALLIEZ, P. 2006. Periodic global parameterization. *ACM Trans. Graph. 25*, 4, 1460–1485.

SHEFFER, A., PRAUN, E., AND ROSE, K. 2006. Mesh parameterization methods and their applications. *Foundations and Trends® in Computer Graphics and Vision 2*, 2, 171.

SPRINGBORN, B., SCHRÖDER, P., AND PINKALL, U. 2008. Conformal equivalence of triangle meshes. *ACM Trans. Graph. 27*, 3, 77:1–77:11.

TARINI, M., CIGNONI, P., AND MONTANI, C. 2006. Ambient occlusion and edge cueing to enhance real time molecular visualization. *IEEE Trans. Vis. and Comp. Graph. 12*, 6.

TARINI, M., PIETRONI, N., CIGNONI, P., PANOZZO, D., AND PUPPO, E. 2010. Practical quad mesh simplification. *Comp. Graph. Forum 29*, 2.

TONG, Y., ALLIEZ, P., COHEN-STEINER, D., AND DESBRUN, M. 2006. Designing quadrangulations with discrete harmonic forms. In *Symposium on Geometry Processing*, 201–210.

YING, L., AND ZORIN, D. 2004. A simple manifold-based construction of surfaces of arbitrary smoothness. *ACM Trans. Graph. 23*, 3, 271–275.

ZHANG, L., LIU, L., GOTSMAN, C., AND HUANG, H. 2010. Mesh reconstruction by meshless denoising and parameterization. *Computers & Graphics 34*, 3, 198–208.

ZHANG, M., HUANG, J., LIU, X., AND BAO, H. 2010. A wave-based anisotropic quadrangulation method. *ACM Trans. Graph. 29*, 4.

A Proof of Proposition 1

We use the notation of Section 4 and Figure 5. Let $\mathcal{V}_\ell = \cap_m \mathcal{U}_m$, $m = 1 \ldots M_\ell$, for a collection of domains $\mathcal{U}_m$, such that for any domain $\mathcal{U}$ not in the collection, $\mathcal{V}_\ell \cap \mathcal{U} = \emptyset$. Consider a point $\bar{p} \in \mathcal{V}_\ell$, and its $p_m = c^m(\bar{p})$ in domains $\mathcal{U}_m$. Let $q_m = q^m(p_m) = q^m(c^m(r))$. Then the constraints (3) have the form $q_i = X^{ij} q_j$ and constraints (5) $q_j = \frac{1}{M_\ell - 1} \sum_{m \neq j} X^{mj} q_m$. Clearly, (3) implies (5), as the symmetrized constraints can be obtained by summing sets of constraints of the form (3). We prove the converse statement. As $X^{jj} = Id$ for any j, the symmetrized constraint for q_i can be rewritten in an equivalent form

$(M_\ell - 1)q_i + q_i = \sum_m X^{mi} q_m$, i.e., $q_i = \frac{1}{M_\ell} \sum_m X^{mi} q_m$. Using the cycle condition $X^{ij} \circ X^{mi} = X^{mj}$, we obtain $X^{ij} q_i = \frac{1}{M_\ell} \sum_m X^{mj} q_m = q_j$, i.e. the constraint of the form (3) for a pair (i, j) is obtained from combining the symmetrized constraints for i and j.

B Derivation of the discrete constraints between overlapping triangles

For a pair of functions $f(p)$ and $g(p)$, defined on a triangle, instead of pointwise equality $f(p) = g(p)$, we require that their L^2 products with linear basis functions coincide, i.e., $\langle f, \phi_k \rangle = \langle g, \phi_k \rangle$ for the hat basis functions ϕ_k corresponding to vertices of T, where $\langle a, b \rangle$ denotes the integral $\int ab \, dA$ and the integral is over the support of ϕ_k. The per-point compatibility condition $q^i(p) = X^{ji} q^j(\tau^{ij}(p))$ translates into $\sum_m q_m^i \langle \phi_m, \phi_k \rangle = \sum_n X^{ij} q_n^j \langle \phi_n \circ \tau^{ij}, \phi_k \rangle$. Following a common FEM practice, we approximate the integrals using quadratures.

For the product $\langle \phi_m, \phi_k \rangle$, using the standard lumped mass matrix approximation with quadrature points at triangle corners, we obtain non-zero values only for $k = m$. The left-hand side in this case reduces to $A_i/3 q_m^i$, where A_i is the area of T_i. On the right-hand side, a similar approximation can be used, but because $\phi_n \circ \tau^{ij}$ is defined on a different triangulation compared to ϕ_k, we find that a somewhat more accurate result is obtained by using a single point quadrature, with the function evaluated at the barycenter c_{ij} of the overlap: the integral is approximated as $\sum_n q_n^j \phi_n(\tau^{ij}(c_{ij})) \phi_m(c_{ij}) A_{ij}$, where A_{ij} is area of the overlap of T_i and $\tau^{ij}(T_j)$. We define the weights $b_{mn}^{ij} = A_{ij} \phi_n(\tau^{ij}(c_{ij})) \phi_m(c_{ij})$ The per-triangle integrals for left- and right-hand side become $A_i q_m^i = X^{ji} \sum_{n=1}^{3} b_{mn}^{ij} q_n^j$. The weights are defined as

$$w_{mn}^{ij} = \frac{b_{mn}^{ij}}{A_i} = \frac{A_{ij}}{A_i} \phi_n(\tau^{ij}(c_{ij})) \phi_m(c_{ij})$$

Adding the constraints for all triangles overlapping T_i, gives us a complete integral. Finally, we apply the symmetrization procedure discussed above to this set of constraints, and obtain

$$q_m^i = \frac{1}{M_i - 1} \sum_{j \in O_m} X^{ji} \sum_{n=1}^{3} w_{mn}^{ij} q_n^j, \tag{10}$$

where M_i is the number of triangles overlapping the triangle i. where the index sets O_m consist of indices of triangles T^j for which the remapped triangle $\tau^{ji}(T^j)$ in $\mathcal{S}^i$ overlaps T^i.