Eurographics Symposium on Geometry Processing (2007)
Alexander Belyaev, Michael Garland (Editors)

Generalized Surface Flows for Mesh Processing

I. Eckstein[1] J.-P. Pons[2] Y. Tong[3] C.-C. J. Kuo[1] M. Desbrun[3]

[1]University of Southern California
[2]WILLOW, INRIA / ENS / Ecole des Ponts, Paris, France
[3]Caltech

Abstract

Geometric flows are ubiquitous in mesh processing. Curve and surface evolutions based on functional minimization have been used in the context of surface diffusion, denoising, shape optimization, minimal surfaces, and geodesic paths to mention a few. Such gradient *flows are nearly always, yet often implicitly, based on the canonical L^2 inner product of vector fields. In this paper, we point out that changing this inner product provides a simple, powerful, and untapped approach to extend current flows. We demonstrate the value of such a norm alteration for regularization and volume-preservation purposes and in the context of shape matching, where deformation priors (ranging from rigid motion to articulated motion) can be incorporated into a gradient flow to drastically improve results. Implementation details, including a differentiable approximation of the Hausdorff distance between irregular meshes, are presented.*

1 Introduction

Geometric flows have been extensively used in mesh processing. In particular, surface flows based on *functional minimization* (i.e., evolving a surface so as to progressively decrease an energy functional) is a common methodology in geometry processing with applications spanning surface diffusion [Tau95], denoising of scanned meshes [DMSB99], shape optimization and surface design [Kob00, BS05, XPB06], minimal surfaces [PP93], (geodesic) shortest paths [BWK05], and animation [GHDS03]. Such *gradient flows* even extend to image processing where they appear, e.g., in the foundations of active contours [CKS97] and shape spaces [You99, KSMJ04]. While a large part of the image processing community implements gradient flows using an Eulerian methodology (typically, with level sets [Set99, OF03]—see also [MBWB02, DR04] in graphics), Lagrangian representations of surfaces based on *triangle meshes* are most common in graphics. In this Lagrangian setting, discretization of continuous flows is usually achieved through the use of discrete differential operators and/or using finite element techniques [PP93, DKS02, MDSB02]. In this paper, we point out that most (if not all) surface gradient flows used in mesh processing are based on the canonical L^2 inner product of vector fields—although this choice is rarely mentioned or acknowledged. In fact, changing this inner product is shown to be a powerful, versatile, and untapped approach to design novel *generalized* Lagrangian gradient flows of discrete surfaces.

Outline We first give a brief exposition of the mathematical background involved in the design of conventional gradient flows in Section 2, stressing the (often implicit) choice of the L^2 inner product over the space of vector fields on meshes. In Section 3, we then propose to alter this L^2 inner product through the introduction of an additional operator to offer a simple, versatile, and powerful way to design *generalized* gradient flows of discrete surfaces—an idea recently introduced in [CKPF05] and [SYM07] in the Eulerian setting. Finally, we show two particular applications of these generalized flows. In Section 4, we demonstrate that the use of Sobolev inner products is an effective way to regularize explicit gradient flows, similar to implicit fairing for the special case of the mean curvature flow, that we enhance to offer local volume control (Section 5). Finally, in Section 6, we show how deformation priors render generalized flows particularly relevant to shape matching: we introduce a series

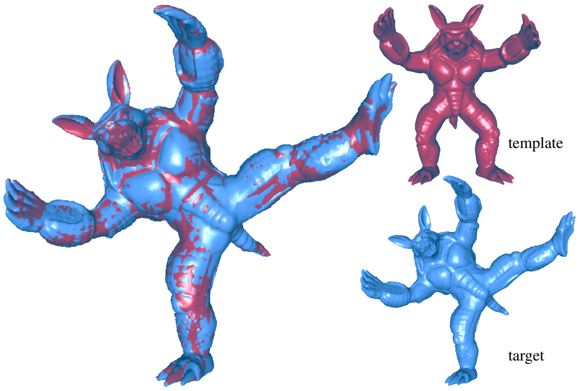

Figure 1: ***Articulated matching*** *of an Armadillo template mesh to a very different pose, using the Hausdorff gradient flow with a* ***quasi-articulated prior****. Our approach can embed such shape priors into conventional surface flows to provide,* e.g.*, robust and efficient shape matching with no markers. Notice that although a template arm initially coincides with a target mesh leg (resulting in a strong local minimum), the final pose is successfully recovered.*

of prior-based inner products to develop a multi-resolution shape matching procedure between two meshes through a generalized pseudo-Hausdorff distance gradient flow.

2 Mathematical Background on Gradient Flows

We start by giving a brief exposition of the mathematical background involved in the design of Lagrangian gradient flows. Note that we will refer to the Euclidean ($\mathbb{R}^3$) metric as $|.|$, and to its associated inner product as the conventional dot product "·" notation. Norms and inner products of vector fields are denoted as $\|.\|$ and $\langle .,. \rangle$ instead, to avoid confusion.

2.1 Continuous Definition of Gradient Flows

Let $\mathcal{S}$ be a surface in $\mathbb{R}^3$ (assumed with no boundary for simplicity), and suppose that we are given an energy functional $\mathcal{E}$ as a measure of surface "quality", i.e., the lower the energy, the "fairer" the surface is. A *gradient flow* is a motion of the surface that follows the steepest descent of the functional, i.e., the motion follows the "gradient" of the functional. However, in the continuous setting, the definition of the gradient of the energy functional is not straightforward: one cannot use the traditional concept of partial derivatives along each dimension since we are dealing with an infinite-dimensional space of deformation (thereafter denoted by F). Using the calculus of variations, we can however define a theory of curve and surface evolution [GH86] using the weaker notion of Gâteaux derivative. The Gâteaux derivative of $\mathcal{E}$ at $\mathcal{S}$ (denoted by $D\mathcal{E}(\mathcal{S})[\mathbf{v}]$ for its value in the direction $\mathbf{v} \in F$) is defined to be the rate of variation of the energy when the surface undergoes a deformation along a vector field $\mathbf{v}$:

$$D\mathcal{E}(\mathcal{S})[\mathbf{v}] \overset{\text{def}}{=} \left.\frac{d\mathcal{E}(\mathcal{S}+\varepsilon\mathbf{v})}{d\varepsilon}\right|_{\varepsilon=0} = \lim_{\varepsilon\to 0} \frac{\mathcal{E}(\mathcal{S}+\varepsilon\mathbf{v}) - \mathcal{E}(\mathcal{S})}{\varepsilon}\ . \quad (1)$$

Inner Product of Vector Fields Recall now that we wish to define a surface evolution that decreases the energy the fastest. However, the very *notion* of steepness requires a metric, i.e, a way to measure distances and angles. To that effect, we must equip our space F of all possible deformation fields with an inner product $\langle,\rangle_F$. With this additional structure on the space of deformations of a surface, we can now formally define the gradient of $\mathcal{E}$ *relative to the inner product* $\langle,\rangle_F$, denoted $\nabla_F\mathcal{E}(\mathcal{S})$, as the vector field such as:

$$\forall \mathbf{v} \in F,\ D\mathcal{E}(\mathcal{S})[\mathbf{v}] = \langle \nabla_F\mathcal{E}(\mathcal{S}), \mathbf{v}\rangle_F\ , \quad (2)$$

Here, we assume existence and uniqueness of this gradient for simplicity—these topics are outside the scope of this article; but be aware that these properties intimately depend on the choice of energy functional.

Gradient Flow Now that the gradient of the energy functional is properly defined, shape motion via functional minimization can be performed by evolving an initial surface $\mathcal{S}_0$ in the direction of the negative gradient to best reduce the energy, yielding:

$$\frac{d\mathcal{S}}{dt} = -\nabla_F\mathcal{E}(\mathcal{S}), \quad \mathcal{S}(0) = \mathcal{S}_0 \quad (3)$$

Once again, we assume that this flow is well defined without further analysis or justification. We note that if this evolution exists, it does decrease the energy $\mathcal{E}$ as expected:

$$\begin{aligned}\frac{d\mathcal{E}(\mathcal{S})}{dt} &= \lim_{\varepsilon\to 0} \frac{\mathcal{E}(\mathcal{S}(t) - \varepsilon\nabla_F\mathcal{E}(\mathcal{S})) - \mathcal{E}(\mathcal{S}(t))}{\varepsilon} \\ &= \langle \nabla_F\mathcal{E}(\mathcal{S}), -\nabla_F\mathcal{E}(\mathcal{S})\rangle_F = -\|\nabla_F\mathcal{E}(\mathcal{S})\|_F^2 \le 0\ .\end{aligned}$$

Choice of Inner Product In the construction of a gradient flow as we just reviewed, we remained agnostic vis-a-vis the inner product on F. In fact, every inner product yields its own gradient, and thus, a different gradient flow. However it is widely assumed, overtly or covertly, that the deformation space F is the Hilbert space of square integrable velocity fields $L^2(\mathcal{S},\mathbb{R}^3)$ equipped with the canonical L^2 inner product on $\mathcal{S}$:

$$\langle \mathbf{u},\mathbf{v}\rangle_{L^2} = \int_{\mathcal{S}} \mathbf{u}(\mathbf{x})\cdot\mathbf{v}(\mathbf{x})\ d\mathbf{x}\ , \quad (4)$$

where $d\mathbf{x}$ is the area element of the surface. Even with this choice of space for F (the only deformation space that we will consider in this paper), picking a different inner product is a degree of freedom for gradient flows that is rarely exploited in geometry processing—to the point where most classical gradient flows reported in the literature (mean curvature flow, geodesic active contours, etc) improperly refer to *the* energy gradient, implicitly assuming the L^2 inner product.

2.2 Discrete Gradient Flows

We now assume that the surface $\mathcal{S}$ is represented by a triangle (piecewise-linear) mesh $\mathbf{X}$, and the position of its vertices is denoted by $\mathbf{x_i}$. A piecewise linear vector field on $\mathcal{S}$ is a collection $\mathbf{V} = \{\mathbf{v}_i\}$ of vectors at the vertices, interpolated linearly over each triangle: $\mathbf{v} = \sum_i \mathbf{v}_i\,\phi_i$, where the linear, interpolating basis functions ϕ_i satisfy: $\forall \mathbf{x} \in \mathcal{S}$, $\sum_i \phi_i(\mathbf{x}) = 1$. Note that for this particular setup, a discrete surface flow exactly coincides to a motion of each vertex $\mathbf{x_i}$ along $\mathbf{v}_i$.

L^2 Inner Product and L^2-Gradient With this choice of discrete setup, the canonical L^2 inner product of two discrete vector fields $\mathbf{U}$ and $\mathbf{V}$ becomes particularly simple to express:

$$\langle \mathbf{U},\mathbf{V}\rangle_{L^2} = \mathbf{U}^T\mathbf{M}\,\mathbf{V}\ , \quad (5)$$

where $\mathbf{M} = \{M_{ij}\}$ is a symmetric definite positive matrix (called the *mass matrix* in the Finite Elements literature) defined through 3×3 blocks:

$$M_{ij} = \mathrm{Id}_3 \int_{\mathcal{S}} \phi_i(\mathbf{x})\phi_j(\mathbf{x})\,d\mathbf{x}\ .$$

Note that M is sparse, but not diagonal. A classical approximation, simplifying computations considerably with limited accuracy loss, is to use *mass lumping* which turns $\mathbf{M}$ into a diagonal matrix where M_{ii} is the area of the (Voronoi or barycentric) dual cell of $\mathbf{x}_i$ times the identity matrix Id_3 [DDCB00, MDSB02].

Discrete Gradient Flow If we call $\mathbf{X}$ the vector of all the positions $\mathbf{x}_i$ of the vertices of the mesh, the discrete counterpart of the L^2-gradient of a surface energy functional $\mathcal{E}$ is:

$$\nabla_{L^2}\mathcal{E}(\mathbf{X}) = \mathbf{M}^{-1}\frac{\partial\mathcal{E}}{\partial\mathbf{X}}(\mathbf{X})\ ,$$

where $\partial\mathcal{E}/\partial\mathbf{X}$ is the derivative of the energy with respect to the position of the vertices. The discrete L^2 gradient descent

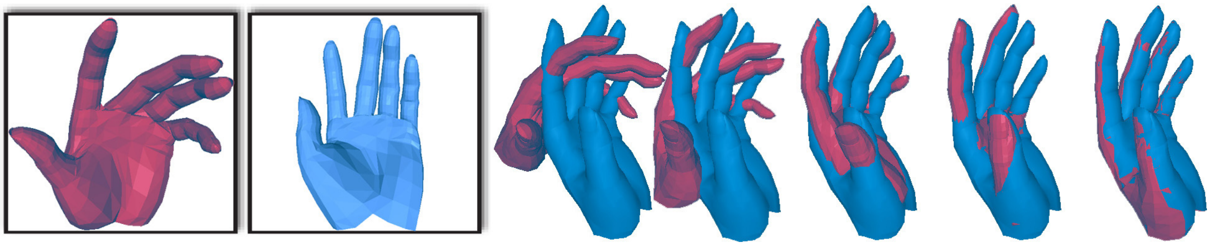

Figure 2: ***Quasi-Articulated flow:*** *matching two hands by minimizing their Hausdorff distance with a gradient flow incorporating an articulated prior (using 16 regions: 3 per finger + palm). The flow automatically morphs the red hand into the blue one, first mostly through a global rotation, then by adjusting each finger. Color mix on the final frame is due to the fact that the hands do* not *have the same sampling and connectivity.*

then follows straightforwardly:

$$\frac{d\mathbf{X}}{dt} = -\mathbf{M}^{-1}\frac{\partial \mathcal{E}}{\partial \mathbf{X}}(\mathbf{X}) \; . \tag{6}$$

Example The most canonical example of surface flow is certainly the *mean curvature flow* [GH86]. Its discrete treatment on meshes was first studied in [PP93], and more recently in [DMSB99], where the flow was formulated as

$$\frac{d\mathbf{x}_i}{dt} = -\frac{1}{2\mathcal{A}_i}\frac{d\mathcal{A}_i}{d\mathbf{x}_i} \; ,$$

with $\mathbf{x}_i$ denoting the i-th vertex of the mesh, and $\mathcal{A}_i$ denoting the 1-ring surface area of this vertex. This last term being proportional to the area of the barycentric dual of $\mathbf{x}_i$ (and thus, proportional to the lumped mass matrix diagonal term M_{ii}), one can see that this discrete flow corresponds to the L^2*-minimization of the area of the surface*—the exact analog of its continuous counterpart, called Laplace-Beltrami flow.

3 Generalized Gradient Flows

We now present how to design other inner product than L^2 on the space F of vector fields before showing how it results in a family of generalized gradient flows.

3.1 Designing New Inner Products

Recently, Charpiat *et al.* [CKPF05] have shown for the Eulerian setting that shape optimization through gradient descent can greatly benefit from the design of an application-dependent inner product on the deformation space. In particular, they noted that *for any self-adjoint positive-definite linear operator* $\mathcal{L}: F \to F$, a *new* inner product can be defined by

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{L}} = \langle \mathbf{u}, \mathcal{L}\,\mathbf{v} \rangle_{L^2} \; = \langle \mathcal{L}\,\mathbf{u}, \mathbf{v} \rangle_{L^2} \; . \tag{7}$$

This is evidently a very special type of inner product, as it is defined with respect to the L^2 norm. The advantage is however that *if* $\nabla_{L^2}\mathcal{E}(\mathcal{S})$ *exists, then* $\nabla_{\mathcal{L}}\mathcal{E}(\mathcal{S})$ *also exists and can be expressed as*:

$$\nabla_{\mathcal{L}}\mathcal{E}(\mathcal{S}) = \mathcal{L}^{-1}\nabla_{L^2}\mathcal{E}(\mathcal{S}) \; . \tag{8}$$

Indeed, one can verify that:

$$\forall \mathbf{v} \in F,\; D\mathcal{E}(\mathcal{S})\,[\mathbf{v}] = \langle \nabla_{L^2}\mathcal{E}(\mathcal{S}), \mathbf{v} \rangle_{L^2} = \left\langle \mathcal{L}\,\mathcal{L}^{-1}\nabla_{L^2}\mathcal{E}(\mathcal{S}), \mathbf{v} \right\rangle_{L^2}$$
$$= \left\langle \mathcal{L}^{-1}\nabla_{L^2}\mathcal{E}(\mathcal{S}), \mathbf{v} \right\rangle_{\mathcal{L}} \; .$$

In other words, this procedure is of significant practical interest because it allows to *modify any existing* L^2 *gradient flow*. We now show how to adapt this idea in the mesh setting.

3.2 Generalized Gradients

The discrete counterpart of the linear operator $\mathcal{L}$ is a matrix that we will denote as $\mathbf{L} = \{L_{ij}\}$. Then, using Eq.(7), the $\mathcal{L}$ inner product of two discrete vector fields $\mathbf{U}$ and $\mathbf{V}$ is expressed through

$$\langle \mathbf{U}, \mathbf{V} \rangle_{\mathcal{L}} = \langle \mathbf{L}\mathbf{U}, \mathbf{V} \rangle_{L^2} = \mathbf{U}^T\mathbf{M}\mathbf{L}\mathbf{V}. \tag{9}$$

Thus the $\mathcal{L}$-gradient of the energy can be easily expressed as

$$\nabla_{\mathcal{L}}\,\mathcal{E}(\mathbf{X}) = (\mathbf{M}\mathbf{L})^{-1}\frac{\partial \mathcal{E}}{\partial \mathbf{X}}(\mathbf{X}) = \mathbf{L}^{-1}\nabla_{L^2}\mathcal{E}(\mathbf{X}) \; .$$

Note that we made use of the self-adjointness of the operator $\mathcal{L}$, which in our discrete setting means that the $\mathbf{ML}$ matrix is symmetric (so that $\langle \mathbf{U}, \mathcal{L}\mathbf{V} \rangle_{L^2} = \mathbf{U}^T(\mathbf{ML})\mathbf{V} = \mathbf{U}^T(\mathbf{L}^T\mathbf{M})\mathbf{V} = \langle \mathcal{L}\mathbf{U}, \mathbf{V} \rangle_{L^2}$). In fact, the discrete $\mathbf{L}$ matrix is not necessarily symmetric; however, left-multiplying by the mass matrix $\mathbf{M}$ will make the final product symmetric. The reader can verify that all the inner products proposed in this paper trivially satisfy this property although their corresponding matrices $\mathbf{L}$ may not be symmetric by themselves.

3.3 Time Integration

Once our generalized gradient is defined, the associated generalized flow is obtained via time integration. A direct implementation of an *explicit* Euler time stepping scheme for a generalized gradient flow yields:

$$\mathbf{X}_{t+dt} = \mathbf{X}_t - dt\,(\mathbf{M}\mathbf{L})^{-1}\frac{\partial \mathcal{E}}{\partial \mathbf{X}}(\mathbf{X}_t) \; . \tag{10}$$

We will see that for most of our proposed generalized flows, the inverse of $\mathbf{ML}$ is given *analytically* (see, e.g., Eq.(16)), rendering the computational cost of time integration almost negligible. The only exception is the prior in Section 6.3.3, where $(\mathbf{ML})^{-1}$ has no closed form expression. Even in this case, integration can still be done efficiently: the matrix $\mathbf{ML}$ is such that multiplying it by a vector is not a quadratic, but a *linear* operation. Thus, a preconditioned conjugate gradient method (PCG) [BBC*94] will be, in practice, very efficient, as is the case in Section 6.3.3. We can also use *implicit* integration for our generalized gradient flows through a linearization of $\partial\mathcal{E}/\partial\mathbf{X}$ with respect to the position of the vertices. This modification, tantamount to keeping the metric of the surface constant over the integration step, has been advocated for the mean curvature flow in [DMSB99] as an easy way to remove the usual time step constraints inherent to explicit integration techniques and add robustness to mesh flows. We will suppose that such a linearization exists (as will be the case in all of our flows) such that: $\partial\mathcal{E}(\mathbf{X}_t)/\partial\mathbf{X} = \mathbf{A}_t\,\mathbf{X}_t + \mathbf{B}_t$, where

A is diagonal, positive definite matrix (the same argument can be made for sparse SPD matrices). Then, Eq.(10) can be rewritten by using the position of the vertices at the next time step in the evaluation of the gradient:

$$\mathbf{X}_{t+dt} = \mathbf{X}_t - dt\,(\mathbf{ML})^{-1}(\mathbf{A}\,\mathbf{X}_{t+dt} + \mathbf{B})\,.$$

That is, one can find the next mesh $\mathbf{X}_{t+dt}$ by solving the following linear system:

$$(\mathbf{ML} + dt\,\mathbf{A})\,\mathbf{X}_{t+dt} = \mathbf{ML}\,\mathbf{X}_t - dt\,\mathbf{B}\,. \qquad (11)$$

Interestingly, we now show that in some cases, this implicit integration of flows can be seen as a special case of the explicit integration of a *regularized* gradient flow.

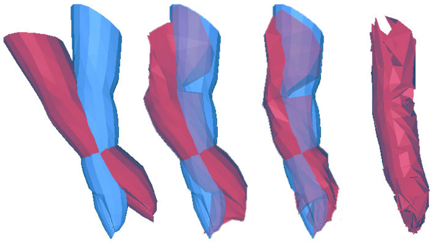

Figure 3: ***Incorrect matching*** *between two fingers obtained through a* ***straightforward*** L^2 ***gradient flow*** *of their Hausdorff distance. The absence of rigidity priors leads to significant spurious sliding and mismatch. The intermediate positions also exhibit large deformation.*

4 Application: Regularized Gradient Flows

As most existing gradient flows used in geometry processing are based on L^2, they all share the consequences induced by this particular choice of inner product. First, since the L^2 norm of a vector field completely disregards its spatial coherence, a conventional gradient flow may (and will, see Figure 3) produce highly non-rigid deformation, thus degenerate meshes. Second, one can also show that such gradient flows are potentially very susceptible to noise, and often get stuck in local minima of the energy. To remediate these flaws, Sundaramoorthi *et al.* [SYM07] proposed recently to use a regularizing inner product, namely, a Sobolev norm. They demonstrated the multiple advantages of this choice in the context of Eulerian (Level Sets) active contours.

Sobolev Inner product Our Lagrangian set-up can accommodate the same change of norm. For meshes, the Sobolev norm H^1 derives from the following inner product:

$$\langle \mathbf{u}, \mathbf{v} \rangle_{H^1} = \int_{\mathcal{S}} \mathbf{u}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) d\mathbf{x} + \lambda \int_{\mathcal{S}} \nabla \mathbf{u}(\mathbf{x}) \cdot \nabla \mathbf{v}(\mathbf{x}) d\mathbf{x}\,.$$

It is a simple matter of integration by part to show that this inner product corresponds to the linear operator $\mathcal{L}(\mathbf{u}) = \mathbf{u} - \lambda\Delta\mathbf{u}$, where Δ is the Laplace-Beltrami operator (discretized for instance in [PP93]), and λ is an arbitrary positive constant.

Sobolev Gradient Flow With this inner product, we can define the H^1-gradient of an energy $\mathcal{E}$, and perform explicit integration of the corresponding gradient flow:

$$\mathbf{X}_{t+dt} = \mathbf{X}_t - dt\,(\mathrm{Id} - \lambda\Delta)^{-1} \nabla_{L^2}\mathcal{E}(\mathbf{X}_t).$$

Consequently, a step of Sobolev gradient flow can be performed by solving the following linear system:

$$(\mathrm{Id} - \lambda\Delta)\mathbf{X}_{t+dt} = (\mathrm{Id} - \lambda\Delta - dt\,\nabla_{L^2}\mathcal{E})\mathbf{X}_t\,. \qquad (12)$$

The solution of this linear system now couples the motion of each vertex to the motion of the other vertices. This exemplifies the regularization effect: for an explicit integration of a L^2-gradient, vertices move basically independently of each other, while for H^1, vertices move in concord. Finally, we can extend the regularization scheme to higher Sobolev-like norms. For instance, a higher-order inner product can be defined through $\mathcal{L}(\mathbf{u}) = \mathbf{u} - \lambda\Delta\mathbf{u} + \mu\Delta^2\mathbf{u}$, leading to an additional term on each side of the linear system to solve.

This regularization is very reminiscent of a well-known smoothing method: it turns out that the explicit integration of the H^1-gradient flow of the surface area of a mesh is known as *implicit fairing* [DMSB99]. Indeed, we mentioned in Section 2.2 that the L^2-gradient of the area corresponds to the Laplace-Beltrami operator. Substituting this term in Eq. (12) and choosing $\lambda = dt$, we get the usual expression:

$$(\mathrm{Id} - dt\Delta)\,\mathbf{X}_{t+dt} = \mathbf{X}_t\,.$$

Seeing implicit fairing as simply being a regularized explicit mean curvature flow is another way of understanding its notable stability. The reader will have noticed that, the generalized mean curvature flow defined by the higher-order inner product corresponds once again to an implicit fairing, this time with higher-order operators—a known way to improve the filtering quality of this smoothing flow [DMSB99]. A similar regularizing effect can be used for any flow using Eq. (12) with the additional Δ^2 terms.

5 Application: Volume-Controlled Implicit Fairing

Now we take the above regularized scheme one step further by modifying it to allow a direct control over the local volume change during the flow. Once again, a simple modification of the inner product is all it takes.

Quasi-Volume Preserving Flows Let VP denote the subspace of all volume-preserving deformations, with NVP being its L^2-orthogonal complement, and let Π_{VP} and Π_{NVP} be the projection operators on the corresponding subspaces. We define a volume-controlled norm of a field $\mathbf{v}$ simply as:

$$\|\mathbf{v}\|^2_{\mathcal{L}_{\mathrm{VP}}} = \|\Pi\|^2_{\mathrm{VP}}\,\mathbf{v} + \frac{1}{\lambda}\,\|\Pi_{\mathrm{NVP}}\mathbf{v}\|^2\,,$$

where λ is a penalty factor on volume-preserving motion: a smaller λ will make such motion "cheaper", thus favored. Following the procedure from Eq.(7), we get:

$$\mathcal{L}_{\mathrm{VP}} = \Pi_{\mathrm{VP}} + \frac{1}{\lambda}\Pi_{\mathrm{NVP}}, \text{ thus: } \mathcal{L}_{\mathrm{VP}}^{-1} = \lambda\mathrm{Id} + (1-\lambda)\Pi_{\mathrm{VP}},$$

which can be used to obtain a new gradient flow, as in Eq.(8). For instance, a volume-controlled implicit fairing is achieved through solving:

$$(\mathcal{L}_{\mathrm{VP}} - \lambda\Delta)\mathbf{X}_{t+dt} = \mathcal{L}_{\mathrm{VP}}\mathbf{X}_t\,. \qquad (13)$$

Finally, the projection operator Π_{VP} on the subspace of locally volume-preserving motions can be implemented as follows. A deformation field $\mathbf{v}$ of a discrete mesh $\mathbf{X} = \{\mathbf{x}_i\}$ with a normal field $\mathbf{N} = \{\mathbf{n}_i\}$ and a mass matrix $\mathbf{M}$, causes a local volume change $\delta V_i = \mathbf{v}_i \cdot M_{ii}\,\mathbf{n}_i$ around each vertex $\mathbf{x}_i$, where $\mathbf{M}$ is assumed to be diagonal, i.e., the total area is $\mathcal{A} = \sum_i M_{ii}$. ($M_{ii}\,\mathbf{n}_i$ is akin to the usual area-weighted normal at a vertex.)

Then averaging δV_i over the one-ring area of $\mathbf{x}_i$ yields:

$$\overline{\delta V}_i = \frac{\delta V_i}{\sum_{j\in N^+(i)} M_{jj}},$$

where $N^+(i)$ denotes the vertex i and its immediate neighbors. Thus, the local volume-changing velocity component of each $\mathbf{x}_i$ becomes:

$$\bar{\mathbf{v}}_i = \mathbf{n}_i \sum_{j\in N^+(i)} \overline{\delta V}_j,$$

yielding: $\Pi_{\mathrm{VP}}(\mathbf{v}_i) = \mathbf{v}_i - \bar{\mathbf{v}}_i$. Note that controlling the volume locally ensures *global* volume control. With this procedure, λ is a "knob" that makes the flow more or less volume-preserving—it will become fully volume-preserving as λ goes to zero. Fig. 4 illustrates the obvious non-shrinking effect of this generalized flow compared to the original implicit fairing for a same amount of smoothing.

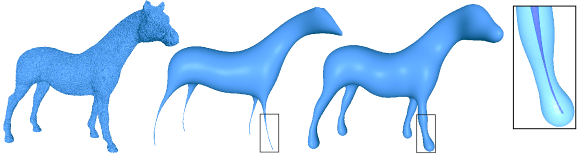

Figure 4: ***Implicit fairing with local volume control.*** *While a noisy horse model (left) becomes degenerate after aggresive implicit fairing (middle), a comparable amount of smoothing with local volume control avoid shrinking (right); closeup shows overlaid comparison.*

6 Application: Robust Shape Matching

6.1 Background

A particularly interesting application of surface flows is *shape matching*. Suppose that we have two 3D models (a *template* and an *instance*) of a same subject (be it an object, a whole human body, or just an organ) in different poses with no known correspondence (this is typically the case for most models acquired from 3D scanners and medical imaging). The shape matching problem amounts to recovering a relevant mapping (i.e., induced by a plausible deformation) between these two different poses—see Fig. 2 for an example of properly established correspondence. For shapes given as meshes or point clouds, matching can be done using only global registration if the two shapes are quite similar [CM91, BM92, MGPG04, GMGP05] —see Planitz *et al.* [PMW05] for a detailed survey—or using more complex approaches allowing for the matching of less similar shapes [You99, DGG03, CR03]. For widely different shapes, an alternative is to rely on Markov networks, approximate inferences, and geodesic nearness constraints to automatically compute Correlated Correspondences [ASP*04], and use these established correspondences as markers for more robust registration [ASK*05]. When no training set is available, a common procedure is to evolve the template shape, often initially positioned by the user, in the *direction of steepest descent of an energy measuring the distance to the instance shape*. However, due to the highly non-convex and non-linear nature of most energy functionals, an unsupervised gradient descent flow gets trapped in irrelevant states corresponding to local minima, seriously limiting its applicability and efficiency (see Figure 3). The use of markers as in [ACP02, ACP03, SKR*06] can help tremendously, but requires (potentially heavy) user supervision. Alternatively, for the class of near-isometric deformations (such as articulated motion), generalized multidimensional scaling (GMDS) has been proposed [BBK07a, BBK07b] based on the MDS method [EK03]. A mapping is formulated as a parametrization of one surface onto the other such that the distortion of pairwise geodesic distances is minimized. Yet without any prior knowledge about the shape structure, an optimal mapping can be extremely hard to recover, as shown in Fig. 6.

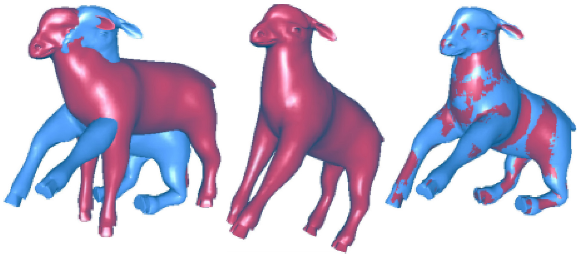

Figure 5: ***Non-rigid Shape Matching:*** *Here, an articulated template model of a lamb (red) is automatically deformed to fit a different pose (blue), by directly minimizing their Hausdorff distance with a gradient flow based on a quasi-articulated prior. To handle this example (~ 22000 vertices for each mesh) in less than two minutes, a multi-resolution scheme was used. Notice that our prior automatically makes the lamb move mostly rigidly first to best fit the target pose, then adjust the limbs non-rigidly. The color mix on the final frame shows that, even if both lambs do* not *have the same sampling and connectivity, their geometry matches well.*

Our Prior-based Approach We now explore an alternative where our generalized gradient flows are beneficially applied to embed *priors* (such as quasi-rigidity, quasi-isometry, or quasi-articulation) in the modeling of the gradient flow. The use of priors, often used in object tracking or shape fairing/completion [DTB06], consists in introducing a measure of the plausibility of the different configurations, and biasing the optimization process towards the most probable ones —thus improving robustness of the estimation.

6.2 Hausdorff Gradient Flow for Matching

A typical shape matching algorithm tries to deform the template to match the instance by reducing a geometric distance. Because a L^2-type distance is known to be too forgiving in comparing two shapes (as well as prone to numerous local minima), we resort to an approximation of the Hausdorff distance instead, based on a method recently introduced in [CFK05] in the context of level sets. The exact symmetric Hausdorff distance $d(\mathcal{S},\mathcal{T})$ between surface $\mathcal{S}$ and $\mathcal{T}$ is

$$d(\mathcal{S},\mathcal{T}) = \max\left[\max_{p\in\mathcal{S}}\min_{q\in\mathcal{T}} \|p-q\|\,, \max_{q\in\mathcal{T}}\min_{p\in\mathcal{S}} \|p-q\|\right]$$

Since this expression is not differentiable, we use instead a pseudo-Hausdorff distance $d_H(\mathbf{X},\mathbf{Y})$ between two distinct meshes $\mathbf{X} = \{\mathbf{x}_i\}_{i=1..P}$ and $\mathbf{Y} = \{\mathbf{y}_j\}_{j=1..Q}$. We refer the reader to Appendix A for the actual derivation of this distance, as it significantly differs from the original Eulerian description in [CFK05] to allow for a robust treatment of irregular sampling. The L^2 gradient flow based on our ap-

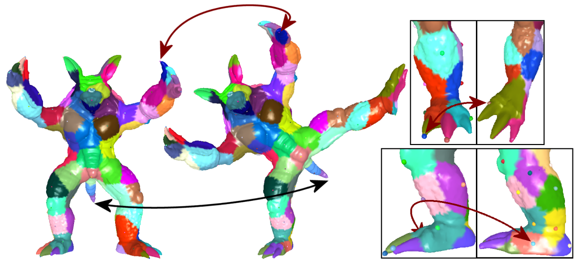

Figure 6: ***GMDS correspondences*** *(courtesy of [BBK07a]) between two different Armadillo poses, for a subset of 100 vertices, with the corresponding Voronoi regions shown in matching colors. While most regions are mapped adequately (e.g., the tail, shown with a black arrow), hands and legs still have many inaccurate correspondences (e.g., those shown with red arrows), which will result in unnatural deformations if further used for registration.*

proximate Hausdorff distance is then written as

$$\frac{d\mathbf{X}}{dt} = -\mathbf{M}^{-1}\frac{\partial\, d_H}{\partial \mathbf{X}}(\mathbf{X},\mathbf{X}^{\infty})\,, \tag{14}$$

where $\mathbf{X}^{\infty}$ is the instance mesh, and the mesh $\mathbf{X}$ is initialized as the template mesh. As extensively discussed previously, such a naive minimization is unable to yield relevant correspondences between two dissimilar shapes, as the energy landscape is too complex and non-linear to avoid getting stuck in one of the numerous local minima (see Fig. 3). While other similarity measures could be added to the Hausdorff distance for better matching (see the use of the normal field in [ZG04, RL01, CSAD04]), we will refrain from discussing them here for simplicity.

6.3 Constructing Prior-based Inner Products

To increase efficiency and robustness of shape matching as well as further exemplify the versatility of generalized flows, we present in this section three different possible alterations of a gradient flow.

6.3.1 Quasi-Rigid Deformation Prior

The case of quasi-rigid deformation prior, where a L^2-gradient flow is altered by penalizing non-rigid motion, was introduced in Charpiat *et al.* [CKPF05], and we adapt it here to our Lagrangian setting due to its relevance to our goals. To that effect, we denote by $\mathcal{A}$ the surface area of a given shape $\mathcal{S}$, and by $\mathbf{c} = \frac{1}{\mathcal{A}}\int_{\mathcal{S}} \mathbf{x}\, d\mathbf{x}$ its center of mass. With these two definitions, we decompose the space of rigid motions into the subspaces of translations, of instantaneous rotations around the center of mass, and of global scaling (denoted by respectively T, R and S). These subspaces are mutually orthogonal for the L^2 inner product. We finally denote by N the orthogonal complement of these subspaces, i.e. the space of strictly non-rigid motions. Thus, the space F of all deformations satisfies: $F = T \perp R \perp S \perp N$. A new inner product, altering the respective "strength" of the different components of a deformation, can now be expressed by following the general procedure of Eq.(7) and introducing:

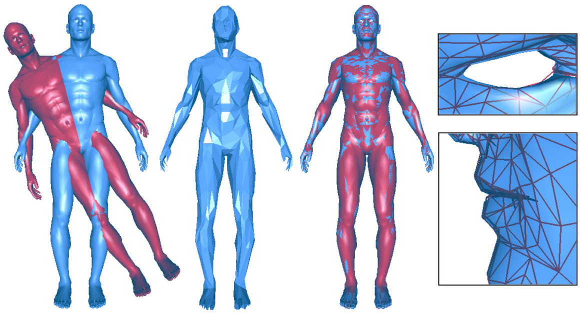

Figure 7: ***Quasi-Rigid Prior using Multiresolution:*** *Matching two full body meshes with different sampling (~ 28K vertices each) by minimizing their Hausdorff distance with a quasi-rigid gradient flow. To speed up computations, a multi-resolution scheme was employed (total time:* 1 *min). The coarsest resolution is shown in the middle. Closeups show the final alignment (template mesh in wireframe).*

$$\mathcal{L} = \lambda_T\, \Pi_T + \lambda_R\, \Pi_R\; + \lambda_S\, \Pi_S + \Pi_N\;, \tag{15}$$

where Π_T, Π_R, Π_S, Π_N denote orthogonal projections on different subspaces, and λ_T, λ_R, $\lambda_S > 0$ are penalty factors of translations, rotations, scalings, respectively. *The lower a penalty factor is, the stronger its corresponding type of motion will be in the new gradient flow*. Note that this inner product also has a straightforward interpretation in mechanics, since the $\mathcal{L}$-norm of a vector field can be seen as a kinetic energy summing the translational, rotational, and purely non-rigid residual kinetic energies of the motion.

Expression of New Gradient The new gradient $\nabla_{\mathcal{L}}\mathcal{E}(\mathcal{S})$ can be obtained from the L^2-gradient using Eq.(8) with a simple expression, since the inverse operator $\mathcal{L}^{-1}$ can be expressed in closed form as:

$$\mathcal{L}^{-1} = \mathrm{Id} + \frac{1-\lambda_T}{\lambda_T}\Pi_T + \frac{1-\lambda_R}{\lambda_R}\Pi_R + \frac{1-\lambda_S}{\lambda_S}\Pi_S\,. \tag{16}$$

Thus, the new $\mathcal{L}$-gradient is simply a linear combination of the L^2-orthogonal projections of the L^2-gradient on T, R and S. Each of these projections are found through the following equations, where the integrals are turned into discrete sums since $\mathbf{v}$ is piecewise linear:

$$\forall \mathbf{v} \in F, \quad (\Pi_S \mathbf{v})(\mathbf{x}) = \frac{\int_{\mathcal{S}} \mathbf{v}(\mathbf{y}) \cdot (\mathbf{y}-\mathbf{c})\, d\mathbf{y}}{\int_{\mathcal{S}} \|\mathbf{y}-\mathbf{c}\|^2\, d\mathbf{y}}\, (\mathbf{x}-\mathbf{c})$$

$$(\Pi_T \mathbf{v})(\mathbf{x}) = \frac{1}{\mathcal{A}}\int_{\mathcal{S}} \mathbf{v}(\mathbf{y})\, d\mathbf{y}\,,\;\; (\Pi_R \mathbf{v})(\mathbf{x}) = \mathbf{I}^{-1}\,\Omega \times (\mathbf{x}-\mathbf{c}),$$

where $\mathbf{I}$ is the discrete moment of inertia of the mesh:

$$\mathbf{I} = \int_{\mathcal{S}} \left[\|\mathbf{x}-\mathbf{c}\|^2\, \mathrm{Id}_3 - (\mathbf{x}-\mathbf{c})(\mathbf{x}-\mathbf{c})^T \right] d\mathbf{x},$$

and Ω is the discrete angular momentum

$$\Omega = \int_{\mathcal{S}} (\mathbf{x}-\mathbf{c}) \times \mathbf{v}(\mathbf{x})\, d\mathbf{x}.$$

The resulting generalized flow boost translations, rotations, and scaling from the initial gradient flow. One can verify that $\mathcal{L}$ fulfills our initial requirements for inner products, guaranteeing a proper generalized gradient flow. Figure 7 shows how this flow successfully aligns two full body meshes.

6.3.2 Quasi-Articulated Deformation Prior

We now extend the idea introduced above to motion of *articulated* bodies with quasi-rigid links. This prior will be particularly useful when used on mapping the hand in Figure 2 onto another deformed hand. The bone structure within each finger severely restricts the possible spectrum of deformation that a hand can usually undergo, and building this knowledge into the gradient flow will significantly improve the robustness of the matching procedure.

We will assume that a segmentation of $\mathcal{S} = \bigcup_{k=1..K} \mathcal{S}_k$ into K parts is known, and that the articulation connectivity graph is a *tree*: $G = \{(\mathcal{S}_i, \mathcal{S}_j) | \mathcal{S}_i$ is a predecessor of $\mathcal{S}_j\}$. Assume also that for each $\mathcal{S}_k$ a corresponding center of rotation (joint) c_k is given. We will denote by $\mathcal{R}$, $\mathcal{A}$ and $\mathcal{N}$ respectively the orthogonal subspaces of $\mathcal{R}$igid, $\mathcal{A}$rticulated and $\mathcal{N}$on-rigid motions of the full object. Similarly, $\mathcal{R}_k$, $\mathcal{A}_k$ and $\mathcal{N}_k$ denote the subspaces of rigid, articulated and non-rigid motions of the k^{th} articulated part. Notice that since we are dealing with connected links, the translation and scaling parts of $\mathcal{R}_k$ (described in the previous section) are irrelevant, and only the rotation part R_k is needed. Note also that each $\mathcal{A}_k$ depends solely on rotations—not only about its own center c_k, but also about its predecessors' joints. Thus, the projection $\Pi_{\mathcal{A}_k}$ on the articulated motion subspace $\mathcal{A}_k$ is best computed via a recursive formula through the tree:

$$\begin{cases} \Pi_{\mathcal{A}_1} = \Pi_{\mathcal{R}_1} \\ \Pi_{\mathcal{A}_k} = \Pi_{\mathcal{A}_{\mathrm{pred}(k)}} + \Pi_{\mathcal{R}_k}(\mathrm{Id} - \Pi_{\mathcal{A}_{\mathrm{pred}(k)}}) \end{cases}, \tag{17}$$

where $\mathrm{pred}(k)$ denotes the predecessor of link k in the tree structure and the root index is defined as 1. We can now obtain a weighted inner product by first isolating the rigid component of a deformation, then applying to the non-rigid residual an {articulated $\oplus$ non-rigid} decomposition of each part $\mathcal{S}_k$ of the object taken separately. Thus, the square $\mathcal{L}$-norm of a deformation field $\mathbf{v}$ is defined as:

$$\|\mathbf{v}\|^2_{\mathcal{L}} = \lambda \|\Pi_{\mathcal{R}}\mathbf{v}\|^2_{L^2} + \sum_k [\mu_k \left\|\Pi_{\mathcal{A}_k}\Pi_{\mathcal{N}}\mathbf{v}\right\|^2_{L^2} + \left\|\Pi_{\mathcal{N}_k}\Pi_{\mathcal{N}}\mathbf{v}\right\|^2_{L^2}],$$

where λ (resp. μ_k) is the penalty factor of the rigid motion of the full object (resp. of part $\mathcal{S}_k$). These penalty factors typically fulfill $\forall k$, $\lambda < \mu_k < 1$, so that rigid motion is favored over articulated motion, while the latter is favored over non-rigid motion. Finally, the linear operator $\mathcal{L}$ corresponding to the norm in Equation (6.3.2) becomes:

$$\mathcal{L} = \lambda \Pi_{\mathcal{R}} + \sum_k [\mu_k \Pi_{\mathcal{A}_k} \Pi_{\mathcal{N}} + \Pi_{\mathcal{N}_k}] \tag{18}$$

and the corresponding closed form inverse is:

$$\mathcal{L}^{-1} = \frac{1}{\lambda}\Pi_{\mathcal{R}} + \sum_k [\frac{1}{\mu_k}\Pi_{\mathcal{A}_k}\Pi_{\mathcal{N}} + \Pi_{\mathcal{N}_k}]\,. \tag{19}$$

Extension to Hierarchical Linked Structures We remark that the above ideas can be easily generalized to a *hierarchical structure*, i.e., a multi-level decomposition of the object, with different penalty factors both for the different parts and for the different levels. This approach allows to prioritize global motions over local ones, to obtain an "as-rigid-as-possible" articulated deformation. For instance, a hand can be a subtree in a body hierarchy, represented by single component on level k, but split into a palm and fingers on level $k+1$. This can be achieved with another recursive construction:

$$\|\mathbf{v}\|^2_{\mathcal{L},\mathcal{S}} = \lambda(\mathcal{S}) \left\|\Pi_{\mathcal{A}(\mathcal{S})}\mathbf{v}\right\|^2_{L^2} + \sum_{\mathcal{S}' \subset \mathcal{S}} \left\|\Pi_{\mathcal{N}(\mathcal{S})}\mathbf{v}\right\|^2_{\mathcal{L},\mathcal{S}'}\,,$$

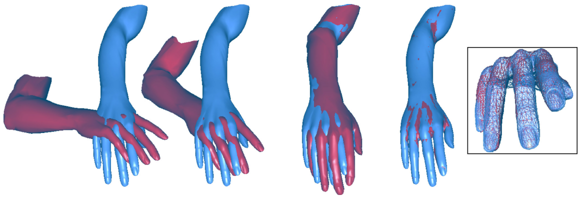

Figure 8: ***Quasi-Articulated Prior:*** *Matching two arms with the Hausdorff gradient flow incorporating a quasi-articulated prior (using 18 subparts: 16 for the hand, plus the upper and lower arm). Again, the mesh sampling of both arms is very different. Despite the presence of fingers in different positions (adding more local minima to the distance-based energy), our algorithm finds the right matching through a combination of global and local rotations, with a small amount of non-rigid deformation (for instance at the elbow). Final alignment of the fingers is shown in closeup on the right.*

where $\mathcal{A}(\mathcal{S})$ is the space of articulated motion on a part $\mathcal{S}$ containing subparts $\mathcal{S}'_k$, and $\lambda(\mathcal{S})$ is the corresponding penalty factor. To end the recursion correctly, we define:

$$\|\mathbf{v}\|^2_{\mathcal{L},\mathcal{S}} = \lambda(\mathcal{S}) \left\|\Pi_{\mathcal{A}(\mathcal{S})}\mathbf{v}\right\|^2_{L^2} + \left\|\Pi_{\mathcal{N}(\mathcal{S})}\mathbf{v}\right\|^2_{L^2}$$

for the leaf parts of the structure (i.e., the quasi-rigid deformation prior of Section 6.3.1).

Handling Joints A shortcoming of the above articulated priors is that they do not directly penalize disagreement between the motion of different parts at the joints (i.e., the nodes of the tree). As a result, a gradient flow derived from these priors may be discontinuous across parts (i.e., at the joints). To overcome this problem, we use a simple partition of unity of the shape $\mathcal{S}$ rather than a strict partitioning: by modeling the points close to the joints as simultaneously belonging to several parts with different probabilities ("blending"), we recover the smoothness of the deformation. Note that while in our case the probabilities are obtained simply by a local blending of the parts around the joints, this approach generalizes to any probabilistic segmentation if available.

6.3.3 Quasi-Isometric Prior

If no a-priori knowledge is available on the range of deformation that a particular shape can undergo, a reasonable assumption valid for a a wide class of models is *quasi-isometry*, i.e., that a surface deforms such that Euclidean distances between pairs of surface points, sampled from the whole surface or from a neighborhood of a certain size, are nearly preserved. This assumption implies a near-preservation of both angles and areas at the scale of the chosen neighborhood. A continuous inner product could be derived from this condition. For simplicity, we present a discrete version directly. This prior bears similarity in its goals to the "semi-local rigidification" from Charpiat *et al.* in [CMP*06] in the context of level sets, but it differs from this previous work in its foundations and realization. A closely related approach was recently and independently suggested in [KMP07].

Gradient for Quasi-Isometric Prior First, we note that the length preservation of an edge $e_{ij} = (\mathbf{x}_i, \mathbf{x}_j)$ is expressed as

$$(\mathbf{v}_i - \mathbf{v}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j) = 0\,.$$

From this condition, we can derive the following inner product of two discrete vector fields $\mathbf{U}$ and $\mathbf{V}$ directly:

$$\langle \mathbf{U},\mathbf{V}\rangle_{\mathcal{L}} = \langle \mathbf{U},\mathbf{V}\rangle_{L^2} + \lambda_{iso}\sum_{e_{ij}}\left[(\mathbf{u}_i-\mathbf{u}_j)\cdot(\mathbf{x}_i-\mathbf{x}_j)\right]\left[(\mathbf{v}_i-\mathbf{v}_j)\cdot(\mathbf{x}_i-\mathbf{x}_j)\right],$$

where λ_{iso} is a penalty factor. The associated operator is:

$$(\mathbf{ML\,V})_i = \mathbf{v}_i + \lambda_{iso}\sum_{j\in N_1(i)}\left[(\mathbf{v}_i-\mathbf{v}_j)\cdot(\mathbf{x}_i-\mathbf{x}_j)\right](\mathbf{x}_i-\mathbf{x}_j),$$

where $N_1(i)$ denotes the one-ring neighborhood of vertex i. This prior can be optimized by adapting the size of the neighborhood and the penalty factor λ_{iso} to the desired degree and scale of local rigidity, such that the $\mathbf{ML}$ matrix is still sparse and thus can be inverted rapidly using PCG. We then obtain a computationally efficient and versatile prior, which encompasses articulated motion without the need for a segmentation of the shape. Note that constraining local Euclidean distance is generally not sufficient to enforce a quasi-rigidity of the shape. Thus, one can add *fake* edges (joining a few opposite vertices in each limb) with our same isometric prior when added rigidity is desired.

6.4 Results of Prior-based Shape Matching

We show in a series of numerical experiments that upgrading the gradient flow of the approximate Hausdorff distance with our deformation priors of Section 6.3 yields an efficient method for shape matching. Note that in all the following examples, the sampling of the template and of the query meshes are (purposely) completely different. Unless specified otherwise, the meshes shown in this paper were obtained using E-Frontier's Poser. To accelerate convergence by several orders of magnitude, we implemented a multiresolution approach for the articulated examples: we precompute coarsified versions [GH97] of both the template and the instance mesh and run the Hausdorff gradient flow on those simpler meshes, before refining them back to the original resolution (using *pyramid coordinates* [SK04]) for final alignment, as illustrated by figure 7. This multiresolution implementation results in reasonable timings: most flows shown in this paper are done in a minute or less, the fastest being the hand in Figure 2 (10s.), while aligning the whole human body in Figure 7 takes 1 minute for $28K$ vertices. Note also that we used implicit integration for stability. Figs.5-8 demonstrate the ability of our quasi-articulated prior of Section 6.3.2 to generate a relevant matching between body parts in different poses (using $\lambda_R = \lambda_T$ varying from $5e$-3 to $5e$-4, and the μ's decreasing by a factor of 10 per level in the hierarchy). This prior can be used as soon as a segmentation (even a coarse one) of the template shape is available. In Figure 9 ($\lambda_{iso} = 50$), we show that our quasi-isometric can nonetheless accommodate articulated bodies with different poses without known segmentation, by enforcing the rigidity of the deformation only at a local scale. Finally, in Figure 10, we show the applicability of our method to noisy and incomplete meshes, with the example of a full arm and shoulder. Note that, although we obtain satisfying results in this experiment, the symmetric Hausdorff distance is not naturally suited to partial matches. Yet, our approach was able to resolve this matching procedure.

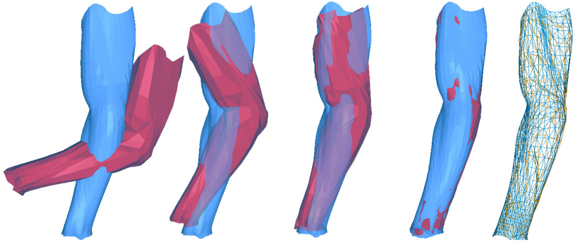

Figure 9: ***Quasi-Isometric Prior:*** *Matching two arms by minimizing their Hausdorff distance with a quasi-isometric gradient flow. Here, without any prior knowledge on the leg structure, the quasi-isometric gradient flow finds the appropriate matching. The last frame shows both meshes, to demonstrate the quality of the match despite the sampling disparity between the two models.*

Comparison to Related Work The above results are intended to show a "stress test" for the generalized flows methodology rather than the panacea of shape matching. Nevertheless, a closer look at shape matching literature points out the significant benefits that generalized flows bring. Several techniques [ACP03, BR04, PMG*05] essentially employ various non-rigid variants of ICP; one known shortcoming of ICP is its reliance on closest point-pairs of estimated correspondences, which are often erroneous unless the two meshes are pre-aligned. A common way to improve robustness is to add a relaxation term, affecting the quality of the final match. Our flow approach bypasses these issues due to the global nature of the Hausdorff distance functional, without sacrificing the quality of the final match. Allen *et al.* focused on human body matching [ACP03] with fairly similar poses, using a combination of locally-affine transformations (similarly to [FA96]) while enforcing a smoothly varying deformation field. To our knowledge, this setup does not have a simple way to build prior knowledge into the deformation, and rigid motion cannot be clearly separated (the same holds for [PMG*05]), necessitating the use of markers for each new object to avoid local minima. By comparison, our quasi-articulated prior only requires a one-time effort of segmenting the template mesh. Another popular way to constrain the deformation is via thin-plate splines [CR03, BR04]. Interestingly, adding a thin-plate energy term is similar to using the high-order Sobolev-like gradient (Section 4) of the original energy, making it a special case of our proposed setup.

7 Conclusion and Future Work

We have extended the notion of gradient flows of triangle meshes by pointing out that one can tailor the inner product on vector fields instead of using the canonical L^2. We gave details on implementation of such generalized geometric flows using explicit and implicit integration techniques. We applied this formalism to improve conventional approaches to shape matching, by first defining an approximate Hausdorff gradient flow, then by applying deformation priors (such as articulated bodies or quasi-isometry) to render the flow significantly more robust (i.e., less prone to local minima and erro-

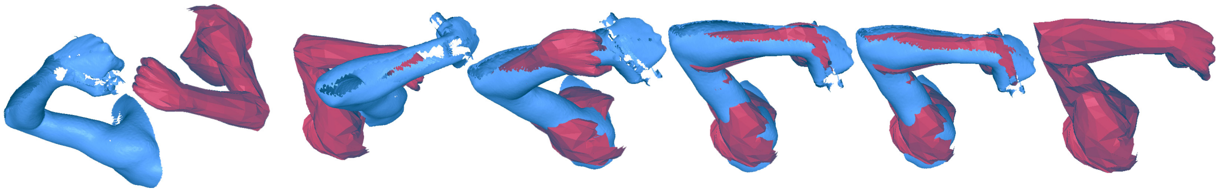

Figure 10: ***Partial Matching*** *of an incomplete scanned 3D mesh (in blue, courtesy of U. of Washington) with a template arm in a different pose. Despite the sparsity of the scan, a quasi-articulated prior (19 subparts, including fingers) on our Hausdorff-based flow gets a good match.*

neous matching). Finally, we use a multi-resolution approach to speed up convergence and add scalability. Our future work includes extending the toolbox of energies and priors for the particular case of shape matching. For instance, landmarks are often used in practice to alleviate the registration task (e.g. in [GMGP05]). Incorporating landmarks into our deformation energy is simple and could combine the strengths of both approaches. We also plan on further investigating the power of altering the norms in mesh flows in the more general context of shape representation and analysis. Promising results have been shown in this area [MM05, YM05, KMP07], where properly defining continuous morphs (homotopies) and geodesic distances in the space of curves requires altering the L^2 metric.

Acknowledgements We are grateful to Xiaohan Shi for sharing the deformed Armadillo model and to Michael and Alex Bronstein for testing their GMDS algorithm on it. This work was partially funded by NSF (CAREER CCR-0133983 and ITR DMS-0453145), DOE (DE-FG02-04ER25657), and Caltech's Center for the Mathematics of Information.

References

[ACP02] Allen B., Curless B., Popović Z.: Articulated body deformation from range scan data. *ACM Trans. on Graphics (SIGGRAPH) 21*, 3 (2002), 612–619.

[ACP03] Allen B., Curless B., Popović Z.: The space of human body shapes: reconstruction and parameterization from range scans. *ACM SIGGRAPH* (2003), 587–594.

[ASK*05] Anguelov D., Srinivasan P., Koller D., Thrun S., Rodgers J., Davis J.: Scape: shape completion and animation of people. *ACM Trans. on Graphics (SIGGRAPH) 24*, 3 (Aug. 2005), 408–416.

[ASP*04] Anguelov D., Srinivasan P., Pang H.-C., Koller D., Thrun S., Davis J.: The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Advances in Neural Information Processing Systems* (2004).

[BBC*94] Barret R., Berry M., Chan T., Demmel J., Donato J., Dongarra J., Eijkhout V., Pozo R., Romine C., van der Vonst H.: *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, 1994.

[BBK07a] Bronstein A., Bronstein M., Kimmel R.: Calculus of non-rigid surfaces for geometry and texture manipulation. *IEEE Trans. on Vis. and Comp. Graphics,* to appear (2007).

[BBK07b] Bronstein A., Bronstein M., Kimmel R.: *Joint intrinsic and extrinsic similarity for recognition of non-rigid shapes*. Tech. Rep. CIS-2007-01, Dept. of CS, Technion, 2007.

[BM92] Besl P., McKay N.: A method for registration of 3D shapes. *IEEE Transactions on PAMI 14*, 2 (1992), 239–256.

[BR04] Brown B., Rusinkiewicz S.: Non-rigid range-scan alignment using thin-plate splines. *Int. Symp. on 3D Data Processing, Visualization and Transmission* (2004), 759–765.

[BS05] Bobenko A. I., Schröder P.: Discrete willmore flow. In *Symp. on Geometry Processing* (July 2005), pp. 101–110.

[BWK05] Bischoff S., Weyand T., Kobbelt L.: Snakes on triangle meshes. *Bild. für die Medizin* (2005), 208–212.

[CFK05] Charpiat G., Faugeras O., Keriven R.: Approximations of shape metrics and application to shape warping and empirical shape statistics. *FoCM 5*, 1 (2005), 1–58.

[CKPF05] Charpiat G., Keriven R., Pons J.-P., Faugeras O.: Designing spatially coherent minimizing flows for variational problems based on active contours. In *International Conference on Computer Vision* (2005), vol. 2, pp. 1403–1408.

[CKS97] Caselles V., Kimmel R., Sapiro G.: Geodesic active contours. *IJCV 22*, 1 (1997), 61–79.

[CM91] Chen Y., Medioni G.: Object modeling by registration of multiple range images. In *IEEE Conference on Robotics and Automation* (1991), vol. 3, pp. 2724–2729.

[CMP*06] Charpiat G., Maurel P., Pons J.-P., Keriven R., Faugeras O.: *Generalized Gradients: Priors on Minimization Flows*. Tech. Rep. 06-23, ENPC-CERTIS, march 2006.

[CR03] Chui H., Rangarajan A.: A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding 89*, 2-3 (2003), 114–141.

[CSAD04] Cohen-Steiner D., Alliez P., Desbrun M.: Variational shape approximation. *ACM Trans. on Graphics 23*, 3 (2004), 905–914.

[DDCB00] Debunne G., Desbrun M., Cani M.-P., Barr A. H.: Adaptive simulation of soft bodies in real-time. In *Computer Animation* (May 2000), pp. 133–144.

[DGG03] Dey T., Giesen J., Goswami S.: Shape segmentation and matching with flow discretization. In *Workshop on Algorithms and Data Structures* (2003), pp. 25–36.

[DKS02] Dziuk G., Kuwert E., Schatzle R.: Evolution of elastic curves in $\mathbb{R}^n$: Existence and computation. *SIAM J. Math. Anal. 33*, 5 (2002), 1228–1245.

[DMSB99] Desbrun M., Meyer M., Schröder P., Barr A.: Implicit fairing of irregular meshes using diffusion and curvature flow. *ACM SIGGRAPH* (1999), 317–324.

[DR04] Droske M., Rumpf M.: A Levelset Formulation for Willmore Flow. *Interfaces & Free Boundaries 6* (2004), 361–378.

[DTB06] Diebel J. R., Thrun S., Brünig M.: A Bayesian method for probable surface reconstruction & decimation. *ACM Trans. Graphics 25*, 1 (2006), 39–59.

[EK03] Elad A., Kimmel R.: On bending invariant signatures for surfaces. *IEEE Trans. on PAMI 25*, 10 (2003), 1285–1295.

[FA96] Feldmar J., Ayache N.: Rigid, affine and locally affine registration of free-form surfaces. *International Journal of Computer Vision 18*, 2 (1996), 99–119.

[GH86] GAGE M., HAMILTON R.: The heat equation shrinking convex plane curves. *Journal of Diff. Geo. 23* (1986), 69–96.

[GH97] GARLAND M., HECKBERT P.: Surface simplification using quadric error metrics. *ACM SIGGRAPH* (1997), 209–216.

[GHDS03] GRINSPUN E., HIRANI A. N., DESBRUN M., SCHRÖDER P.: Discrete shells. In *ACM SIGGRAPH Symp. on Computer Animation* (2003), pp. 62–67.

[GMGP05] GELFAND N., MITRA N., GUIBAS L., POTTMANN H.: Robust global registration. In *Symp. on Geo. Processing* (2005), pp. 197–206.

[KMP07] KILIAN M., MITRA N. J., POTTMANN H.: Geometric modeling in shape space. In *ACM Transactions on Graphics (SIGGRAPH)* (2007).

[Kob00] KOBBELT L.: Discrete fairing and variational subdivision for freeform surface design. *The Visual Computer 16*, 3-4 (2000), 142–158.

[KSMJ04] KLASSEN E., SRIVASTAVA A., MIO W., JOSHI S. H.: Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Trans. on PAMI 26*, 3 (2004), 372–383.

[MBWB02] MUSETH K., BREEN D. E., WHITAKER R. T., BARR A. H.: Level set surface editing operators. 330–338.

[MDSB02] MEYER M., DESBRUN M., SCHRÖDER P., BARR A.: Discrete differential-geometry operators for 2-manifolds. In *Proc. of Int. Workshop on Vis. and Math.* (2002), pp. 52–58.

[MGPG04] MITRA N., GELFAND N., POTTMANN H., GUIBAS L.: Registration of point cloud data from a geometric optimization perspective. *Symp. on Geometry Processing* (2004), 23–32.

[MM05] MICHOR P., MUMFORD D.: Riemannian geometries of space of plane curves, 2005. Preprint.

[OF03] OSHER S., FEDKIW R.: *Level Set Methods and Dynamic Implicit Surfaces*, vol. 153 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2003.

[PKZ04] PENG J., KRISTJANSSON D., ZORIN D.: Interactive modeling of topologically complex geometric detail. *ACM Trans. Graph. 23*, 3 (2004), 635–643.

[PMG*05] PAULY M., MITRA N., GIESEN J., GROSS M., GUIBAS L.: Example-based 3D scan completion. *Symposium on Geometry Processing* (2005), 23–32.

[PMW05] PLANITZ B., MAEDER A., WILLIAMS J.: The correspondence framework for 3D surface matching algorithms. *Computer Vision and Image Understanding 97*, 3 (2005), 347–383.

[PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2*, 1 (1993), 15–36.

[RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the ICP algorithm. *Proc. 3DIM* (2001), 145–152.

[Set99] SETHIAN J. A.: *Level Set Methods and Fast Marching Methods*, 2nd ed., vol. 3 of *Monographs on Appl. Comput. Math.* Cambridge University Press, Cambridge, 1999.

[SK04] SHEFFER A., KRAEVOY V.: Pyramid coordinates for morphing and deformation. In *Int. Symp. on 3D Data Processing, Visualization and Transmission* (2004), pp. 68–75.

[SKR*06] STOLL C., KARNI Z., RÖSSL C., YAMAUCHI H., SEIDEL H.-P.: Template deformation for point cloud fitting. In *Symposium on Point-Based Graphics* (2006), pp. 27–35.

[SYM07] SUNDARAMOORTHI G., YEZZI A., MENNUCCI A. C. G.: Sobolev active contours. *IJCV 73*, 3 (2007), 345–366.

[Tau95] TAUBIN G.: A signal processing approach to fair surface design. In *Proc. of SIGGRAPH* (Aug. 1995), pp. 351–358.

[XPB06] XU G., PAN Q., BAJAJ C. L.: Discrete surface modeling using partial differential equations. *Computer Aided Geometric Design 23*, 2 (2006), 125–145.

[YM05] YEZZI A., MENNUCCI A.: Conformal metrics and true "gradient flows" for curves. In *International Conference on Computer Vision* (2005), pp. 913–919.

[You99] YOUNES L.: Optimal matching between shapes via elastic deformations. *Image & Vision Comp. 17*, 5 (1999), 381–389.

[ZG04] ZELINKA S., GARLAND M.: Similarity-based surface modelling using geodesic fans. *Symposium on Geometry Processing* (2004), 204–213.

A Pseudo-Hausdorff Distance and its Gradient

We define our approximation of the Hausdorff distance between two distinct meshes $\mathbf{X} = \{\mathbf{x}_i\}_{i=1..P}$ and $\mathbf{Y} = \{\mathbf{y}_j\}_{j=1..Q}$ as follows:

$$d_H(\mathbf{X},\mathbf{Y}) = \left[\frac{1}{P}\sum_i M^{\mathrm{x}}_{ii} f_i^{-1} + \frac{1}{Q}\sum_j M^{\mathrm{y}}_{jj} g_j^{-1}\right]^{\frac{1}{2\alpha}} - \varepsilon \qquad (20)$$

with $f_i = \frac{1}{Q}\sum_j M^{\mathrm{y}}_{jj} d_{ij}^{-\alpha}$, $g_j = \frac{1}{P}\sum_i M^{\mathrm{x}}_{ii} d_{ij}^{-\alpha}$ and $d_{ij} = |\mathbf{x}_i - \mathbf{y}_j|^2 + \varepsilon^2$,

where M^x_{ii} and M^y_{jj} are elements of the diagonal mass matrices of $\mathbf{X}$ and $\mathbf{Y}$, respectively (required to compensate for irregular sampling), $\varepsilon > 0$ and $\alpha \geq 0$. Proving that the above expression converges to the Hausdorff distance between the two meshes when $\alpha \to +\infty$ and the sampling of the two meshes increases follows the continuous proof of [CFK05]. An additional argument necessary to the proof is the elementary fact that for any set of positive values $\{u_i\}_{i=1..N}$,

$$\lim_{\alpha\to+\infty}\left(\frac{1}{N}\sum_{i=1}^{N} u_i^{\alpha}\right)^{1/\alpha} = \max_{1\leq i\leq N} u_i \,.$$

Note however that our Hausdorff approximation does not satisfy the triangular inequality. Nevertheless, it is symmetric with respect to the two meshes (while only sided distances are often used in practice for other methods, see for instance [PKZ04]), and only takes positive values. Finally, it has a decisive advantage over the real Hausdorff distance expression: it is *differentiable* with respect to the position of the vertices of the two meshes. Specifically, we have:

$$\frac{\partial d_H(\mathbf{X},\mathbf{Y})}{\partial \mathbf{x}_i} = \frac{(d_H(\mathbf{X},\mathbf{Y})+\varepsilon)^{1-2\alpha}}{P\cdot Q} M^{\mathrm{x}}_{ii} \sum_j \frac{\mathbf{x}_i - \mathbf{y}_j}{d_{ij}^{\alpha+1}} M^{\mathrm{y}}_{jj} (f_i^{-2} + g_j^{-2}) \,.$$

The complexity of computing d_H or its gradient is $O(P\cdot Q)$, which can be prohibitive when using large datasets. In practice, we restrict the sums in f_i and g_i (Eq.(20)) to only the ε-nearest neighbor pairs (found in constant time using a uniform partitioning of the domain), reducing the complexity to a more tractable $O(P) + O(Q)$ for coarsely aligned shapes, without a noticeable loss of accuracy. Further elimination of the computational bottleneck is made possible through the use of multi-resolution.