



Highly-Expressive Spaces of Well-Behaved Transformations: Keeping It Simple

Freifeld, Oren ; Hauberg, Søren; Batmanghelich, Kayhan ; Fisher, John W.

Publication date:
2015

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Freifeld, O., Hauberg, S., Batmanghelich, K., & Fisher, J. W. (2015). Highly-Expressive Spaces of Well-Behaved Transformations: Keeping It Simple. Paper presented at 15th International Conference on Computer Vision, Santiago, Chile.

DTU Library

Technical Information Center of Denmark

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Highly-Expressive Spaces of Well-Behaved Transformations: Keeping It Simple

Oren Freifeld
MIT CSAIL

freifeld@csail.mit.edu

Søren Hauberg
DTU Compute

sohau@dtu.dk

Kayhan Batmanghelich
MIT CSAIL

kayhan@csail.mit.edu

John W. Fisher III
MIT CSAIL

fisher@csail.mit.edu

Abstract

We propose novel finite-dimensional spaces of $\mathbb{R}^n \rightarrow \mathbb{R}^n$ transformations, $n \in \{1, 2, 3\}$, derived from (continuously-defined) parametric stationary velocity fields. Particularly, we obtain these transformations, which are diffeomorphisms, by fast and highly-accurate integration of continuous piecewise-affine velocity fields; we also provide an exact solution for $n = 1$. The simple-yet-highly-expressive proposed representation handles optional constraints (e.g., volume preservation) easily and supports convenient modeling choices and rapid likelihood evaluations (facilitating tractable inference over latent transformations). Its applications include, but are not limited to: unconstrained optimization over monotonic functions; modeling cumulative distribution functions or histograms; time warping; image registration; landmark-based warping; real-time diffeomorphic image editing. Our code is available at <https://github.com/freifeld/cpabDiffeo>

1. Introduction

Spaces of well-behaved transformations, particularly diffeomorphisms, play a key role in computer vision. Unfortunately, current state-of-the-art representations of highly-expressive diffeomorphisms are overly complicated. Thus, despite their potential power and mathematical beauty, their applicability is somewhat limited, especially when large datasets are involved or when computing resources and time are limited. Moreover, owing to their complexity, using powerful inference tools in such spaces remains challenging. Motivated by not only the practicalities of probabilistic modeling and statistical inference but also a desire to make diffeomorphisms broadly accessible, we propose a *representation* that combines simplicity, expressiveness, and efficiency. Particularly, we propose novel spaces of transfor-

Acknowledgement: O.F. and J.W.F. are partially supported by U.S. Office of Naval Research MURI program, award N000141110688, and VITALITE, which receives support from U.S. Army Research Office MURI, award W911NF-11-1-0391. S.H. was funded by the Danish Council for Independent Research, Natural Sciences.

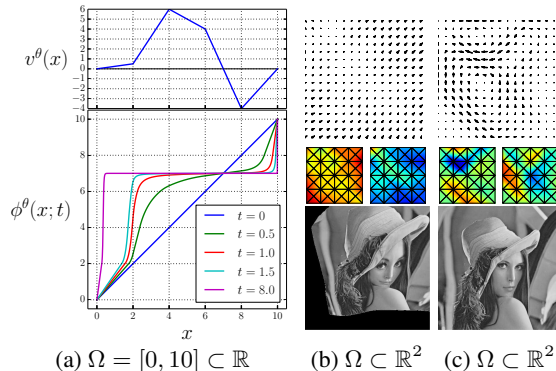


Figure 1: An $\Omega \rightarrow \Omega$ map, $\phi^\theta(\cdot, t) : \mathbf{x} \mapsto \phi^\theta(\mathbf{x}, t)$, derived from a CPA velocity field, $\mathbf{v}^\theta : \Omega \rightarrow \mathbb{R}^n$. (a) A 1D example. Note \mathbf{v}^θ vanishes at fixed points of $\phi^\theta(\cdot, t)$. (b) Top: A 2D \mathbf{v}^θ shown in select locations. Middle: Rather than a Middlebury-style visualization, we use one emphasizing the CPA property. The horizontal component, \mathbf{v}_h^θ (left), and the vertical component, \mathbf{v}_v^θ (right), are shown as heat maps whose colors range from blue = $-\lambda$, via green = 0, to red = λ where $\lambda = \max_{\mathbf{x} \in \Omega} \max(|\mathbf{v}_h^\theta(\mathbf{x})|, |\mathbf{v}_v^\theta(\mathbf{x})|)$. Bottom: $I_{\text{src}} \circ \phi^\theta(\cdot, 1)$ (where I_{src} appears in Fig. 2). (c) A 2D example with boundary and volume-preserving constraints.

mations that are based on (fast, highly-accurate) integration of Continuous Piecewise-Affine (CPA) velocity fields.

Existing spaces offer only subsets of the *benefits* of the proposed spaces: 1) high expressiveness; 2) finite dimensionality; 3) ease of implementation; 4) modest mathematical preliminaries; 5) convenient modeling choices; 6) ease of handling optional constraints; 7) fast and highly-accurate computations that lead to fast likelihood evaluations. These in turn facilitate the use of inference tools – typified by the case of analysis-by-synthesis methods – that are usually too expensive for rich transformation spaces. Possible *applications* are numerous, as we demonstrate with: image editing; optimization over monotonic functions; modeling Cumulative Distribution Functions (CDFs) and histograms with order-preserving geometry; time warps; landmark-based image warping and animation. Another related potential application is image registration.

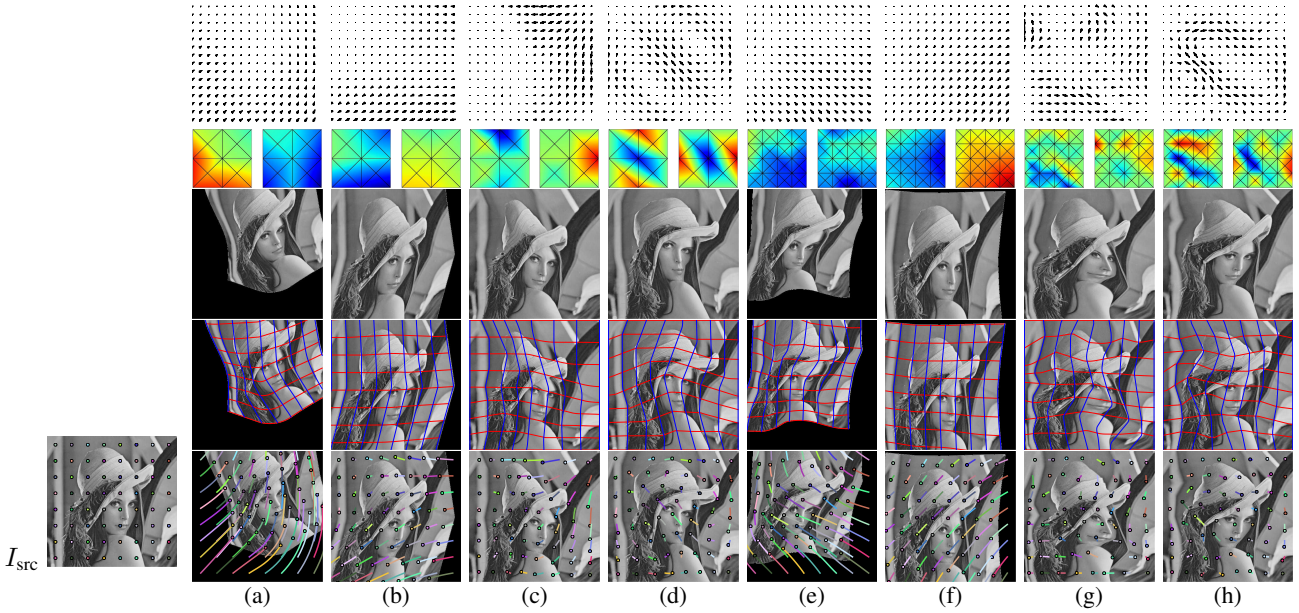


Figure 2: Samples from the prior (§ 3). (a-h) The top 3 rows echo those in Fig. 1b and Fig. 1c. The 4th row shows a deformed grid overlaid on the image. The 5th row shows select *trajectories*. Note that the trajectories and transformations are differentiable (hence continuous) but *not* piecewise affine. The tessellation in (e-h) is a *refinement* of the one in (a-d). (a) & (e): $T^\theta \in M$. (b) & (f): $T^\theta \in M^{\text{vp}}$. (c) & (g): $T^\theta \in M^\partial$. (d) & (h): $T^\theta \in M^{\partial, \text{vp}}$. See § 3 for details.

Henceforth, $n \in \{1, 2, 3\}$ and Ω is either the whole of \mathbb{R}^n or an interval/rectangle/box (in which case, $\Omega \subsetneq \mathbb{R}^n$).

CPA velocity fields. We base our method on spaces of CPA velocity fields (Fig. 1). The term ‘piecewise’ is w.r.t. \mathcal{P} , a *tessellation* (§ 3) of Ω . Let $\mathcal{V}_{\Omega, \mathcal{P}}$ be such a space. While $\mathcal{V}_{\Omega, \mathcal{P}}$ depends on Ω and \mathcal{P} , we will usually notationally suppress these dependencies, and will just write \mathcal{V} . One appeal of these spaces is that they are *finite-dimensional* and *linear* (but note that their elements, *i.e.* the velocity fields, are usually nonlinear). Let $d = \dim(\mathcal{V})$. The spaces \mathbb{R}^d and \mathcal{V} are identified with each other, where every $\theta \in \mathbb{R}^d$ is identified with exactly one element of \mathcal{V} , denoted by \mathbf{v}^θ , and vice versa. A finer \mathcal{P} implies a higher d and richer velocity fields (Figs. 2 and 3). Appealingly, easily-imposed optional constraints yield useful *linear subspaces* of \mathcal{V} .

From CPA velocity fields to trajectories. Modulo a detail (addressed in § 3.2) related to the case $\Omega \subsetneq \mathbb{R}^n$, any continuous velocity field, whether Piecewise-Affine (PA) or not, defines differentiable $\mathbb{R} \rightarrow \Omega$ *trajectories*. Particularly, if $\mathbf{x} \in \Omega$ then $\mathbf{v}^\theta \in \mathcal{V}$ defines a trajectory, $t \mapsto \phi^\theta(\mathbf{x}, t)$, such that $\phi^\theta(\mathbf{x}, 0) = \mathbf{x}$; see Fig. 2. This trajectory is the solution to the *integral equation*

$$\phi^\theta(\mathbf{x}, t) = \mathbf{x} + \int_0^t \mathbf{v}^\theta(\phi^\theta(\mathbf{x}, \tau)) d\tau, \quad \mathbf{v}^\theta \in \mathcal{V}. \quad (1)$$

Remark 1. Equation (1), whose solution, $\phi^\theta(\mathbf{x}, \cdot)$, appears both inside and outside the integral, should not be confused with the piecewise-quadratic $\Omega \rightarrow \mathbb{R}^n$ map, $\mathbf{y} \mapsto$

$\int_{\mathbf{0}_{n \times 1}}^{\mathbf{y}} \mathbf{v}^\theta(\mathbf{x}) d\mathbf{x}$. The latter, a popular tool in computer-vision [40] and numerical analysis, is unrelated to our work.

CPA-Based (CPAB) transformations. Modulo that detail, any continuous velocity field, whether PA or not, defines a transformation; *i.e.*, a map whose input and output are viewed as points in Ω . Particularly, letting \mathbf{x} vary and fixing t , $\mathbf{x} \mapsto \phi^\theta(\mathbf{x}, t)$ is an $\Omega \rightarrow \Omega$ transformation. Without loss of generality we set $t = 1$ and define $T^\theta(\cdot) \triangleq \phi^\theta(\cdot, 1)$. Since we integrate CPA velocity fields, we coin our transformations CPA-Based (CPAB). We write $T^\theta = \exp(\mathbf{v}^\theta)$ to indicate the relation between \mathbf{v}^θ and T^θ ; the rationale for this symbol is explained in our supplemental material (henceforth referred to as **Sup. Mat.**). We let

$$M \triangleq \exp(\mathcal{V}) \triangleq \{\exp(\mathbf{v}^\theta) : \mathbf{v}^\theta \in \mathcal{V}\} \quad (2)$$

denote the space of CPAB transformations, again notationally suppressing the dependencies on Ω and \mathcal{P} ; *i.e.*, formally we should write $M_{\Omega, \mathcal{P}}$. Importantly, M is nonlinear.

Remark 2. CPAB transformations are *not* CPA (except in degenerate cases): while T^θ is continuous, it is not PA.

CPAB transformations are “nice”. *E.g.*, they are *diffeomorphisms*, $(T^\theta)^{-1} \in M$, and the inversion is easy.

Remark 3. $\mathbf{x} \mapsto \phi^\theta(\mathbf{x}, t = 1)$ should not be confused with (the non-diffeomorphism, parametric optical-flow-like representation) $\mathbf{x} \mapsto \mathbf{x} + \mathbf{v}^\theta(\mathbf{x})$, the latter being only a *Taylor approximation* of the former. Thus, the way we utilize flexible parametrized vector fields is different from, *e.g.*, [31].

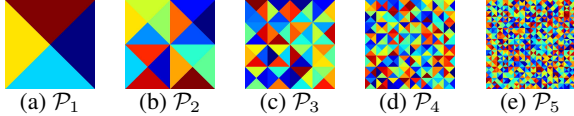


Figure 3: Tesselating a 2D region in several resolutions.

Integration of CPA velocity fields. Integral equations usually lack analytic solutions. Since CPA velocity fields are Lipschitz-continuous and almost-everywhere smooth, generic integration solvers are quite effective for them. However, we can do even better. *One of our contributions is showing that integration of such fields is given in either closed form (when $n = 1$) or almost closed form ($n > 1$).* Besides the obvious pluses (accuracy, computing time), this solution makes it easier to interpret the resulting trajectories/transformation and is key to the theorems in § 3.

A specialized numerical solver. In practice, that solution has one shortcoming. It requires tedious bookkeeping (if $n > 1$) and invoking certain routines (easy if $n = 1$ but cumbersome if $n > 1$). This is especially a hurdle in GPU implementations. We thus propose a practical alternative, *a specialized solver for integrating CPA velocity fields*, which is faster *and* more accurate than non-specialized solvers.

Convenient modeling, tractable inference. Smoothness priors on M are easy to build and use. As is common with nonlinear spaces of nonlinear transformations, the nonlinearity of $(\theta, \mathbf{x}) \mapsto T^\theta(\mathbf{x})$ prohibits closed-form posterior or maximum likelihood calculations; however, since $(\theta, \mathbf{x}) \mapsto T^\theta(\mathbf{x})$ is evaluated fast, so is $\theta \mapsto p(\text{data}|T^\theta)$. This facilitates the use of inference methods that rely on multiple likelihood evaluations such as most Markov Chain Monte Carlo (MCMC) methods. Lastly, CPAB transformations support coarse-to-fine analysis.

2. Related Work

CPA maps. In addition to fields such as numerical analysis and control theory, CPA maps are also used in computer vision [7, 40]. Our use of them differs from such works in that we *integrate* them as *velocity fields* to obtain *transformations*. The ODE equivalent to Eqn. (1) is $\frac{d\phi^\theta(\mathbf{x}, t)}{dt} = \mathbf{v}^\theta(\phi^\theta(\mathbf{x}, t))$. Unlike us, who solve it and exploit its link to transformations, Lin *et al.* [26] use it only as a regression model for fitting patterns in velocity data.

Pattern theory. Representing objects via transformations acting on them is a cornerstone in Grenander’s *pattern theory*. Our work is influenced by the impressive works initiated by Grenander and continued in the geometry-oriented subcommunities of computer vision and medical imaging. Due to space limits, we mention only a few: [15, 16, 8, 24, 42, 4, 1, 18, 17, 43, 30, 45, 9]. However, rather than focusing on complicated, possibly ∞ -dimensional spaces (whose both representations and asso-

\mathcal{P}	N_c	N_v	$D = 6N_c$	$d = 2N_v$
\mathcal{P}_1	4	5	24	10
\mathcal{P}_2	16	13	96	26
\mathcal{P}_3	64	41	384	82
\mathcal{P}_4	256	145	1536	290
\mathcal{P}_5	1024	1025	6144	1090

Table 1: Values of N_c , N_v , $D = \dim(\mathcal{V}'_{\Omega, \mathcal{P}})$, and $d = \dim(\mathcal{V}_{\Omega, \mathcal{P}})$ for the \mathcal{P} ’s shown in Fig. 3. See § 3.

ciated integration are in practice often discretized and/or approximated), or the geometry of transformation spaces, here we take a different approach, emphasizing simplicity and practicality, in order to provide *the larger computer-vision community* with a simple and powerful tool. Thus, our text avoids the direct use of differential geometry, modulo an *optional* discussion in the **Sup. Mat.** The ability to rapidly evaluate transformations accurately and to capture a wide range of deformations while keeping complexity low sets this work apart from other works on diffeomorphisms. Also, in applications involving N landmark pairs, most methods can deal with only small values of N , as their complexity is $\mathcal{O}(N^p)$, $p > 1$ (e.g., some methods invert a dense $nN \times nN$ matrix). Thus, e.g., they cannot utilize tools for dense-correspondence extraction (e.g., [25]), highlighting a disconnect with the larger computer-vision community. Ours is $\mathcal{O}(N)$ and most computations are parallelized.

Finitely-many affine building blocks. The *PA transformations* from [7] should be confused with neither PA/CPA velocity fields nor CPAB transformations; see Remark 2. In fact, PA transformations are neither differentiable (or even continuous) nor invertible. Closer to ours is the work of Arsigny *et al.* [3] who, like us, use finitely-many affine building blocks to build flexible stationary velocity fields. While they spatially average the blocks, we use a CPA constraint on them. A field in our space is usually not in their space (so their integration is inapplicable to it) and vice versa, preventing a direct comparison of *integrations*. Unlike in our integration method, they use an approximation throughout; *i.e., they must approximate the entire trajectory*. Their approximation differs from others in that it uses a weighted sum of matrix exponentials. As the approximation holds only near zero, they must divide the field by a large number (which is tied to #steps). To keep #steps small, they smartly generalize the scaling-and-squaring method. Their scheme is exact only for a single block; however, expressiveness requires more blocks, so the accuracy drops. It is thus unsurprising they focus on a small number of blocks, and that to save time they use their method only in the last stage of inference. For similar dimensions of spaces of velocity fields, we can use larger (hence fewer) steps than them since *most of our steps are exact*. Unlike them, for $n = \dim(\Omega) = 1$ (regardless of the number of blocks)

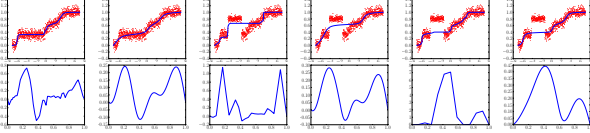


Figure 4: Monotonic regression on synthetic data (top) by inferring latent CPA fields (bottom), defined over 100 equal-length cells. In the 4 leftmost columns we used a Gaussian likelihood; in the 2 rightmost columns we used a robust (Geman-McClure) one. Columns 1,3,5 show ML solutions. The rest show MAP solutions with a smoothness prior. Regardless of outliers/likelihood/prior, the function is increasing (this is obtained effortlessly from the representation).

we solve exactly, while for $n > 1$ we solve exactly most of the trajectory, and use a numerical solver in only parts of it. Beyond the comparison of integrations, which favors ours, our *representation* has additional advantages such that it is simpler, suits better for a GPU implementation, handles constraints and several modeling choices more easily, and scales better with an increase in the number of blocks.

Discrete representations and approximations. Allasonnire et al. [1] efficiently *approximate* diffeomorphisms via non-differentiable maps. While we focus on a general-purpose representation, Diffeomorphic demons [43] is a fast registration tool based on discretely-defined fields, which as its authors note, may be inconsistent with a diffeomorphic framework and may not preserve orientation. They also cannot impose volume preservation. Lastly, a computer representation of a sequence of discrete fields requires plenty of memory. Such issues can be obviated by adapting demons to our compact and continuously-defined fields. More generally, approximations based on discretely-defined velocity fields and/or discrete diffeomorphisms are widely used, *e.g.*, in medical imaging [43]. Unlike these works, both our fields and transformations are continuously-defined.

Statistics on manifolds and tangent spaces. Like others, *e.g.* [3, 42, 11, 29, 13, 20], we deal with the nonlinearity of the space of interest by utilizing the linear tangent space at the identity. Though not explored here, other types of statistics on manifolds [32, 12, 34, 38, 36, 21, 19, 41, 28, 44, 10, 23, 14] may also be applicable.

CDF/histogram modeling. Working with CDFs is preferred to working with densities since the latter might not exist and, moreover, the L_p (or a sphere-based) distance between two densities might be arbitrarily large even if their *probability measures* are essentially the same. One principled approach to representing CDFs uses p -Wasserstein spaces, usually $p \in \{1, 2\}$, the $p = 1$ case is related to Earth Mover’s Distance [33]. A limitation of this approach is that it requires bounded p^{th} -moments and non-trivial computations. 1-Wasserstein methods also lack a full geometric

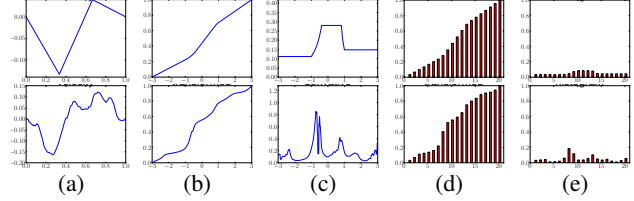


Figure 5: CDF/histogram representation. 1st row: $N_c = 3$. 2nd row: $N_c = 100$. F_0 , not shown, is a CDF of a uniform distribution on $J = [-3, 3]$. (a) $v^\theta \in \mathcal{V}^d([0, 1], \mathcal{P})$. (b) $F = T^\theta \circ F_0$. (c) $\frac{d}{dx}F$. Note it is *not* piecewise constant. H_0 , not shown, is a *cumsum* of a 20-bin uniform histogram. (e) $H = T^\theta \circ H_0$, a *cumsum* of a new histogram h (f) h .

space whose elements are always valid histograms/CDFs, complicating synthesis of new points. Methods that map a histogram/density to a sphere (*e.g.*, [35]) have such geometry but suffer from two issues. First, they do not respect the ordering of the bins or the real line. While in *some* applications of histograms bin ordering is immaterial, it is strange to ignore the ordering of \mathbb{R} . Second, large moves on the sphere lead to CDFs/histograms with negative values. The problems above do not exist in our method.

Image warping. Also related to our work is the problem of image warping [5, 2, 27, 6]. Unlike most methods, ours is diffeomorphic and handles additional constraints easily.

Stationary Velocity Fields. Here we focus on Stationary Velocity Fields (SVF). While not all diffeomorphisms can be defined via SVF, this popular approach (*e.g.*, [3]) to reduce computations and simplifying inference still captures a wide range of real-world deformations. Our CPA fields reduce computations and simplify inference even further. While it is possible to use non-stationary CPA fields (*i.e.*, $\theta = \text{func}(t)$), we are unaware of a specialized integration for these, so one may have to resort to standard solvers. Either way, there may still be a practical gain since the dimension of the inference problem will remain finite (*i.e.*, d times the number of time steps) and our arguments related to priors, constraints, and coarse-to-fine analysis still hold.

3. The Proposed Mathematical Representation

The lemmas/theorems below are proved in the **Sup. Mat.**

3.1. CPA Velocity Fields

A *tessellation* of Ω , denoted by \mathcal{P} , is a set of N_c closed subsets of Ω , $\{U_c\}_{c=1}^{N_c}$, also called *cells*, such that their union is Ω and the intersection of any two adjacent cells is just their shared border. An N_c' -cell tessellation \mathcal{P}' , $N_c' > N_c$, is a *refinement* of \mathcal{P} , a relation denoted $\mathcal{P} < \mathcal{P}'$, if each cell in \mathcal{P}' is a subset of a cell in \mathcal{P} . Unless stated otherwise, it is assumed that the cells are either intervals

```

Input:  $\mathcal{P} = \{C_c\}_{c=1}^{N_c}; (A_{1,\theta}, \dots, A_{N_c,\theta}); U =$  a sequence
of  $N_{\text{pts}}$  points in  $\Omega; t > 0; N_{\text{steps}}, n_{\text{steps}} \in \mathbb{Z}^+$ 
Output:  $\phi^\theta(U, t) \triangleq \{\phi^\theta(\mathbf{x}, t)\}_{\mathbf{x} \in U} \subset \Omega$ 
1  $\Delta_t \leftarrow t/N_{\text{steps}}; \delta_t \leftarrow \Delta_t/n_{\text{steps}}$ 
2 for  $c \in \{1, \dots, N_c\}$  do in parallel
3    $T_{A_c, \theta, \Delta_t} \leftarrow \exp(\Delta_t A_{c, \theta})$ 
4 for  $\mathbf{x} \in U$  do in parallel
5    $\mathbf{x}_0 \leftarrow \mathbf{x}$ 
6   for  $i \in \{1, \dots, N_{\text{steps}}\}$  do
7      $\tilde{\mathbf{x}}_{\text{temp}} = [\mathbf{x}_{\text{temp}}^T \ 1]^T \leftarrow T_{A_{\gamma(\mathbf{x}_{i-1}), \theta, \Delta_t}} \tilde{\mathbf{x}}_{i-1}$ 
8     if  $\gamma(\mathbf{x}_{i-1}) \neq \gamma(\mathbf{x}_{\text{temp}})$  then
9        $\mathbf{x}_i \leftarrow \mathbf{x}_{\text{temp}}$  // analytic update
10    else
// inter-cell bdry crossed
11     $\mathbf{x}_i \leftarrow \text{genericODEsolver}(\text{ini. con.}=\mathbf{x}_{i-1}, \text{step}$ 
size= $\delta_t, n_{\text{steps}}, \mathbf{v}(\cdot) : \mathbf{x} \mapsto A_{\gamma(\mathbf{x}), \theta} \tilde{\mathbf{x}}$ )
12   $\phi^\theta(\mathbf{x}, t) \leftarrow \mathbf{x}_i$ 

```

Algorithm 1: Integrating $\mathbf{v}^\theta \in \mathcal{V}$. See **Sup. Mat.** for details on the generic solver and a discussion on the hypothetical concern related to powers of $T_{A_c, \theta, \Delta_t}$.

(if $n = 1$), triangles ($n = 2$), or tetrahedra ($n = 3$). Let $\mathcal{P}_{\text{vert}} = \{\xi : \xi \text{ is a vertex of some } U_c \in \mathcal{P}\}$ and let N_v denote its cardinality; e.g., see Fig. 3 and Table 1. Rather than regarding $\mathcal{P}_{\text{vert}}$ as a set, we regard it as an ordered N_v -tuple (the ordering is arbitrary, but assumed fixed).

Fix \mathcal{P} and let $\mathbf{x} \in \Omega$. We define the membership function

$$\gamma : \Omega \rightarrow \{1, \dots, N_c\}, \gamma : \mathbf{x} \mapsto \min \{c : \mathbf{x} \in U_c\}; \quad (3)$$

i.e., if \mathbf{x} is not on an inter-cell border then $\gamma(\mathbf{x}) = c \iff \mathbf{x} \in U_c$, while a border point is assigned to the cell of the lowest index. A map, $f : \Omega \rightarrow \mathbb{R}^n$, is called PA (w.r.t. \mathcal{P}) if $\{f|_{U_c}\}_{c=1}^{N_c}$ are affine; i.e., $f(\mathbf{x}) = A_{\gamma(\mathbf{x})} \tilde{\mathbf{x}}$ where

$$A_c \in \mathbb{R}^{n \times (n+1)} \quad \forall c \in \{1, \dots, N_c\} \quad \tilde{\mathbf{x}} \triangleq \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}. \quad (4)$$

A map, $f : \Omega \rightarrow \mathbb{R}^n$, is called CPA if it is continuous and PA. If f is CPA w.r.t. \mathcal{P} it is CPA w.r.t. $\mathcal{P}' \succ \mathcal{P}$. A vector field \mathbf{v} (on Ω) is an $\Omega \rightarrow \mathbb{R}^n$ map viewed as the mapping of points to vectors; i.e., if $\mathbf{x} \in \Omega$ then we view $\mathbf{v}(\mathbf{x})$ as an n -dimensional ‘‘arrow’’. The terms *velocity field* and *vector field* will be used interchangeably. A *vector-field space* \mathcal{V} is a linear space whose elements are vector fields; $\dim(\mathcal{V})$ may be finite or infinite. Let $\mathcal{V}'_{\Omega, \mathcal{P}}$ and $\mathcal{V}_{\Omega, \mathcal{P}}$ denote the spaces of PA and CPA velocity fields on Ω w.r.t. \mathcal{P} . Note $\mathcal{V}_{\Omega, \mathcal{P}} \subset \mathcal{V}'_{\Omega, \mathcal{P}}$. For short, we will also write \mathcal{V}' and \mathcal{V} , suppressing the dependency on Ω and \mathcal{P} .

Lemma 1. \mathcal{V}' and \mathcal{V} are linear spaces, $D \triangleq \dim(\mathcal{V}') = (n^2 + n) \times N_c$, $d \triangleq \dim(\mathcal{V}) = n \times N_v$, and $d \leq D$ with equality if and only if $N_c = 1$.

See, e.g., Table 1. A generic element of \mathcal{V}' is denoted by \mathbf{v}_A where $A \triangleq (A_1, \dots, A_{N_c})$ groups its N_c associated

N_c	5	25	45	65	85	105
our proposed solver	0.06	0.08	0.08	0.09	0.11	0.12
a generic solver	0.33	0.43	0.41	0.43	0.47	0.48

Table 2: Timing comparison (1D). Results, in [sec], are averages of timings of integration of random CPA fields. The tessellations, of $[0, 1]$, consist of N_c intervals. $N_{\text{pts}} = 10^5$, $N_{\text{steps}} = 100$ and $n_{\text{steps}} = 10$ (see Algorithm 1).

$n \times (n + 1)$ matrices. If $A \in \mathbb{R}^{n \times (n+1)}$, then $\text{vec}(A) \in \mathbb{R}^{n^2+n}$ is its row-by-row flattening to a column vector. Likewise, $\text{vec}(A) \triangleq [(\text{vec}(A_1))^T \dots (\text{vec}(A_{N_c}))^T]^T \in \mathbb{R}^D$. An inner product on \mathcal{V}' is defined by $\langle \mathbf{v}_{A_1}, \mathbf{v}_{A_2} \rangle = \text{vec}(A_1)^T \text{vec}(A_2)$. We now explain how to build an orthonormal basis for \mathcal{V} , how its elements are parametrized by θ , and how PA velocity fields are projected onto \mathcal{V} .

Lemma 2. An element of \mathcal{V} is any $\mathbf{v}_A \in \mathcal{V}'$ such that $\text{vec}(A)$ satisfies a linear system of constraints, denoted by $L \text{vec}(A) = \mathbf{0}$; see **Sup. Mat.** for the construction of L .

Let $B = [B_1 \dots B_d] \in \mathbb{R}^{D \times d}$ denote a specific orthonormal basis of $\text{null}(L)$: the one obtained via SVD of L . Let $\theta = [\theta_1 \dots \theta_d]^T \in \mathbb{R}^d$. If $\text{vec}(A) = B\theta$ then \mathbf{v}_A is CPA. Particularly, $\mathbf{v}_{\text{vec}^{-1}(B_j)}$ is CPA for every $j \in \{1, \dots, d\}$, and $\{\mathbf{v}_{\text{vec}^{-1}(B_j)}\}_{j=1}^d$ is a basis for \mathcal{V} . For a visualization of this basis, see **Sup. Mat.** Regardless of whether \mathbf{v}_A is CPA or not, $\mathbf{v}_{\text{vec}^{-1}(BB^T \text{vec}(A))}$ is CPA. When earlier we denoted a generic element of \mathcal{V} by \mathbf{v}^θ we meant that θ stands for the coefficients w.r.t. the $\{\mathbf{v}_{\text{vec}^{-1}(B_j)}\}_{j=1}^d$ basis:

$$\mathbf{v}^\theta(\mathbf{x}) = A_{\gamma(\mathbf{x}), \theta} \tilde{\mathbf{x}} = \sum_{j=1}^d \theta_j \mathbf{v}_{\text{vec}^{-1}(B_j)}(\mathbf{x}) \quad (5)$$

where $A_\theta \triangleq (A_{1,\theta}, \dots, A_{N_c,\theta}) \triangleq \text{vec}^{-1}(\sum_{j=1}^d \theta_j B_j)$. This basis is orthonormal w.r.t. the inner product $\langle \cdot, \cdot \rangle_B : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$, $(\mathbf{v}^{\theta_1}, \mathbf{v}^{\theta_2}) \mapsto \theta_1^T \theta_2$. Let $\mathbf{v}^\theta_{\text{vert}} \triangleq \{\mathbf{v}^\theta(\xi)\}_{\xi \in \mathcal{P}_{\text{vert}}}$ denote the ordered N_v -tuple of n -dimensional values \mathbf{v}^θ takes at the N_v points in $\mathcal{P}_{\text{vert}}$.

Lemma 3. $L_{\theta \mapsto \mathbf{v}^\theta_{\text{vert}}}$, the map sending θ to $\mathbf{v}^\theta_{\text{vert}}$, is a linear bijection and its associated $d \times d$ matrix is given in closed form. We denote its inverse by $L_{\mathbf{v}^\theta_{\text{vert}} \mapsto \theta}$.

If $\Omega \subsetneq \mathbb{R}^n$, let $\partial\Omega$ denote the boundary of Ω , and let $\mathbf{n} : \partial\Omega \rightarrow S^{n-1}$ denote the unit normal to $\partial\Omega$.

Lemma 4. Optional constraints such as $\langle \mathbf{v}^\theta(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle = 0$ on $\partial\Omega$ and/or $\{\text{tr}(A_c) = 0\}_{c=1}^{N_c}$, are linear and can thus extend L to have more rows. The null space of the extended L is a linear subspace (whose dimension is denoted by some $d' < d$) of the null space of the original L . Reusing the symbol $B = [B_1 \dots B_{d'}] \in \mathbb{R}^{D \times d'}$ to denote the orthonormal basis of this new null space obtained via SVD of the extended L , we get a d' -dimensional basis of CPA velocity fields, $\{\mathbf{v}_{\text{vec}^{-1}(B_j)}\}_{j=1}^{d'}$, that satisfy the new constraint(s).

\mathcal{P}	\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	\mathcal{P}_5
our proposed solver	0.06	0.09	0.15	0.24	0.27
a generic solver	0.22	0.23	0.25	0.26	0.28

Table 3: Timing comparison (2D). Results, in [sec], are averages of timings of integration of random CPA fields. The tessellations are as in Fig. 3, the image size is 512×512 , $N_{\text{steps}} = 100$ and $n_{\text{steps}} = 10$ (see Algorithm 1).

The corresponding subspaces are denoted \mathcal{V}^∂ , \mathcal{V}^{tr} and $\mathcal{V}^{\partial, \text{tr}} = \mathcal{V}^\partial \cap \mathcal{V}^{\text{tr}}$. We still denote an element of any of these by \mathbf{v}^θ , with the understanding that then $\theta \in \mathbb{R}^{d'}$. We also define, similarly to Eqn. (2), subsets of M : $M^\partial \triangleq \exp(\mathcal{V}^\partial)$, $M^{\text{vp}} \triangleq \exp(\mathcal{V}^{\text{tr}})$ and $M^{\partial, \text{vp}} \triangleq \exp(\mathcal{V}^{\partial, \text{tr}})$.

3.2. CPA-Based Transformations

Via Eqn. (1), $\mathbf{v}^\theta \in \mathcal{V}$ implies $T^\theta \in M$, a relation denoted $T^\theta = \exp(\mathbf{v}^\theta)$. The detail alluded to in § 1 is that for Eqn. (1) to be well defined, $\phi^\theta(\mathbf{x}, \tau)$ must always be in Ω , the domain of \mathbf{v}^θ . This holds if $\Omega = \mathbb{R}^n$. It also holds if $\Omega \subsetneq \mathbb{R}^n$ and $\mathbf{v}^\theta \in \mathcal{V}^\partial$. If $\Omega \subsetneq \mathbb{R}^n$ and $\mathbf{v}^\theta \in \mathcal{V} \setminus \mathcal{V}^\partial$, it might not. Also, some of the results below require both the continuity and PA properties of \mathbf{v}^θ . We need a lemma.

Lemma 5. *If $\Omega \subsetneq \mathbb{R}^n$, we can extend \mathcal{P} and constrain the velocity fields such that we obtain a linear subspace of \mathcal{V} whose elements extend to CPA fields on \mathbb{R}^n . In which case, we also redefine Ω to be equal to (the whole of) \mathbb{R}^n .*

Henceforth, if $\Omega \subsetneq \mathbb{R}^n$ and we work with \mathcal{V} (or \mathcal{V}^{tr}), as opposed to \mathcal{V}^∂ (or $\mathcal{V}^{\partial, \text{tr}}$), we will assume the procedure from Lemma 5 has been applied. Thus, T^θ is always an $\Omega \rightarrow \Omega$ map, Eqn. (1) is well defined and \mathbf{v}^θ is indeed CPA. Note that $\forall \mathbf{x} \in \Omega$, $\phi^\theta(\mathbf{x}, 0) = \mathbf{x}$ and that $\{\mathbf{x} : \mathbf{v}^\theta(\mathbf{x}) = 0\}$ are fixed points of T^θ (see Fig. 1a). Let us define a useful building block. If $A \in \mathbb{R}^{n \times (n+1)}$, let $\tilde{A} = \begin{bmatrix} A \\ \mathbf{0}_{1 \times (n+1)} \end{bmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}$. Let \mathbf{expm} denote the matrix exponential. The last row of $\mathbf{expm}(\tilde{A})$ is $[\mathbf{0}_{1 \times n} \ 1]$. Moreover, $\det \mathbf{expm}(\tilde{A}) > 0$. Particularly, $\mathbf{expm}(\tilde{A})$ is invertible. If $n = 1$ or A has a special structure, $\mathbf{expm}(\tilde{A})$ has a closed form. Otherwise, since n is small and due to the structure of \tilde{A} , a generic \mathbf{expm} routine approximates it well. Let $t \in \mathbb{R}$ and define $\psi_{\theta, c}^t : \Omega \rightarrow \mathbb{R}^n$ via

$$\begin{bmatrix} \psi_{\theta, c}^t(\mathbf{x}) \\ 1 \end{bmatrix} \triangleq T_{A_{c, \theta}, t} \tilde{\mathbf{x}}, \quad T_{A_{c, \theta}, t} \triangleq \mathbf{expm}(t \tilde{A}_{c, \theta}) \quad (6)$$

(a solution to an ODE with an affine velocity field, $\xi \mapsto A_{c, \theta} \xi$). We note that the solution to Eqn. (1) is the composition of a finite number, denoted by m , of such solutions:

$$\phi^\theta(\mathbf{x}, t) = (\psi_{\theta, c_m}^{t_m} \circ \dots \circ \psi_{\theta, c_2}^{t_2} \circ \psi_{\theta, c_1}^{t_1})(\mathbf{x}). \quad (7)$$

As mentioned earlier, T^θ is defined via $T^\theta(\mathbf{x}) = \phi^\theta(\mathbf{x}, 1)$. The compact form of Eqn. (7) hides an ugly truth: the number of the trajectory segments, m , their durations, $\{t_i\}_{i=1}^m$,

N_{pts}	N_{steps}	\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	\mathcal{P}_5
64×64	10	.001	.001	.002	.002	.006
64×64	100	.002	.003	.003	.005	.014
256×256	10	.004	.006	.009	.011	.014
256×256	100	.023	.028	.046	.064	.083
512×512	10	.011	.016	.025	.036	.041
512×512	100	.079	.129	.185	.287	.300
1024×1024	10	.031	.051	.083	.146	.143
1024×1024	100	.255	.414	.519	1.005	1.070

Table 4: Timings (in [sec]) for varying N_{pts} and N_{steps} . The setup is as the one described in in Table 3.

and the indices of the cells involved, $\{c_i\}_{i=1}^m$ (where a cell may appear more than once), all depend on \mathbf{x} . Except the first cell, $c_1 = \gamma(\mathbf{x})$, they also depend on θ and t . Thus, a more precise and cumbersome notation would be:

$$(\psi_{\theta, c_m, \mathbf{x}, \theta, t}^{t_m(\mathbf{x}, \theta, t)} \circ \dots \circ \psi_{\theta, c_2, \mathbf{x}, \theta, t}^{t_2(\mathbf{x}, \theta, t)} \circ \psi_{\theta, \gamma(\mathbf{x})}^{t_1(\mathbf{x}, \theta, t)})(\mathbf{x}).$$

Remark 4. While invertible affine matrices form a group, $\phi^\theta(\cdot, t) : \Omega \rightarrow \Omega$ is not affine, exactly because of the \mathbf{x} -dependencies mentioned above. In fact, since the above quantities vary with \mathbf{x} even within a cell, the continuous $\mathbf{x} \mapsto \phi^\theta(\mathbf{x}, t)$ is not PA, hence not CPA.

A map, $T : \Omega \rightarrow \Omega$, is called a diffeomorphism (on Ω) if T^{-1} exists and both T and T^{-1} are differentiable. Let G be the space of (orientation-preserving) diffeomorphisms on Ω . Let $H \subset G$ be its restriction to diffeomorphisms that can be obtained from differentiable stationary velocity fields via integration; i.e., $T(\cdot) = \phi(\cdot, 1)$ where $\phi(\mathbf{x}, t) = \mathbf{x} + \int_0^t \mathbf{v}(\phi(\mathbf{x}, \tau)) d\tau$ with a differentiable $\mathbf{v} : \Omega \rightarrow \Omega$. Both G and H are ∞ -dimensional nonlinear spaces.

Theorem 1. (i) If $\alpha \in \mathbb{R}$ then $\mathbf{v}^\theta = \mathbf{v}^{\alpha\theta}/\alpha$ and $\phi^\theta(\cdot, t) = \phi^{\alpha\theta}(\cdot, t/\alpha)$. (ii) $(T^\theta)^{-1}$ exists, is in M , and equals to $T^{-\theta}$. (iii) $M \subset G$. (iv) If $T^\theta \in M^{\text{vp}}$ then T^θ is volume-preserving (hence the vp). (v) M is a d -dimensional nonlinear space; similar statements hold for M^∂ , M^{vp} and $M^{\partial, \text{vp}}$, with their respective values of d . (vi) If $\mathcal{P}_1 \prec \mathcal{P}_2 \prec \dots$ is a tessellation sequence such that eventually all the cells become arbitrarily small, then $M_{\Omega, \mathcal{P}_m} \xrightarrow{m \rightarrow \infty} H$.

Corollary 1. If $n = 1$, T^θ is increasing. Let J be either \mathbb{R} or an interval. If $F_0 : J \rightarrow \Omega$ is either non-decreasing, increasing, right-continuous, continuous, differentiable, a diffeomorphism, or a step function (7 non-mutually-exclusive cases), then so is $F^\theta \triangleq T^\theta \circ F_0 : J \rightarrow \Omega$. If, in addition, $\Omega = [0, 1]$, $\mathbf{v}^\theta \in \mathcal{V}^\partial$ and F_0 is a CDF, then F^θ is a CDF.

This gives an unconstrained parametrization of large classes of increasing functions and CDFs/histograms via a finite-dimensional linear space (Figs. 4 and 5). In the CDF case, F_0 need have neither a density nor finite moments.

Equation (7) is very useful. It justifies and leads to a simple proof of Theorem 1, provides an insight into the struc-

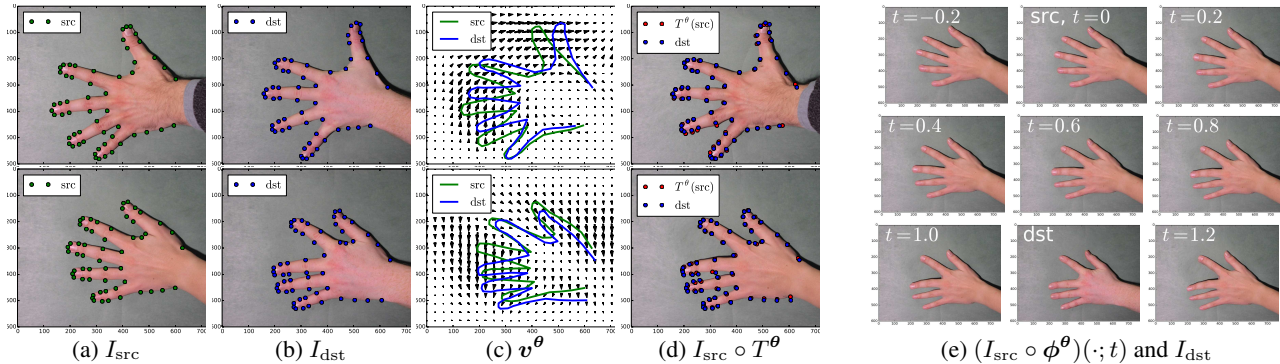


Figure 6: Landmark-based warping. One example in each row. (a) I_{src} (and src landmarks). (b) I_{src} (and src/dst landmarks). (c) The inferred \mathbf{v}^θ . (d) A new image is synthesized by warping I_{src} using T^θ (and dst & $T^\theta(\text{src})$ landmarks). (e) Animation.

ture of $t \mapsto \phi^\theta(\mathbf{x}, t)$, and, when $n = 1$, leads to a closed-form solution for $\mathbf{x} \mapsto T^\theta(\mathbf{x})$. Lastly, when $n \in \{2, 3\}$, it suggests a practical, *essentially*-exact solution.

Theorem 2. *If $n = 1$ then $m_{\mathbf{x}, \theta, t}$, $\{t_i(\mathbf{x}; \theta, t)\}_{i=1}^{m_{\mathbf{x}, \theta, t}}$, and $\{c_i(\mathbf{x}; \theta, t)\}_{i=2}^{m_{\mathbf{x}, \theta, t}}$ have closed forms. Thus, so does $T^\theta(\mathbf{x})$.*

If $n > 1$ then $T^\theta(\mathbf{x})$ is given *essentially* in closed form in the sense that Eqn. (7) still holds, but a generic **expm** is needed (a mild issue) and, more importantly, to find the quantities mentioned in Theorem 2 one needs to solve, for every \mathbf{x} , a number (depending on \mathbf{x}) of sequential problems of the form $\arg \min_{t > 0} \gamma(\mathbf{expm}(t\hat{A})\xi) \neq \gamma(\xi)$ where the pair $(\xi, A) \in \Omega \times \mathbb{R}^{n \times (n+1)}$ changes in each of these problems (while the set of all A 's is shared across different \mathbf{x} 's, we cannot know a-priori which A will appear in each problem). Doing this for $n = 1$ is easy, fast, and accurate. But for $n \geq 2$, partly since $\phi^\theta(\mathbf{x}, \cdot)$ may reenter a cell, this requires tedious bookkeeping, multiple **expm** calls, and multiple invocations of a numerical solver. This leads to slow approximated solutions that are hard to implement in GPU.

Inference over latent transformations often requires evaluating T^θ for multiple values of θ (e.g., 10,000), and a high N_{pts} , where N_{pts} is number of points to be transformed (e.g., the number of pixels in an image). We thus prefer a more practical alternative and propose a specialized solver for integrating CPA fields. This solver, summarized in Algorithm 1, alternates between the analytic solution (with neither bookkeeping nor need to solve explicitly for $m_{\mathbf{x}, \theta, 1}$, $\{c_i(\mathbf{x}; \theta, 1)\}_{i=2}^{m_{\mathbf{x}, \theta, 1}}$ and $\{t_i(\mathbf{x}; \theta, 1)\}_{i=1}^{m_{\mathbf{x}, \theta, 1}}$) and a generic solver. Thus, our solver is both faster and more accurate than a generic solver since most of the trajectory is solved exactly, while only in small portions of the trajectory does our solver resort to the generic one. Importantly, the number of **expm** evaluations required is constant, N_c , and grows with neither N_{pts} nor t . In fact, since the algorithm is highly accurate, fast, and simple, and since usually $N_c \ll N_{\text{pts}}$, we prefer to use it even when $n = 1$.

Construction of Gaussian smoothness priors. It is easy to build a Gaussian smoothness prior on \mathcal{V}' (e.g., we use a $D \times D$ covariance whose correlations decay with inter-cell distances), denoted $\mathcal{N}(\mathbf{0}_{D \times 1}, \Sigma_{\text{PA}})$, and then the $\mathcal{V}' \rightarrow \mathcal{V}$ map, $\mathbf{A} \mapsto \theta = \mathbf{B}^T \text{vec}(\mathbf{A})$, induces a prior on \mathcal{V} : $p(\theta) = \mathcal{N}(\mathbf{0}_{d \times 1}, \Sigma_{\text{CPA}})$ where $\Sigma_{\text{CPA}} = \mathbf{B}^T \Sigma_{\text{PA}} \mathbf{B}$. See Fig. 2 as well as **Sup. Mat.** for samples from $p(\theta)$.

4. Applications and Results

We focus here on 1D/2D; see **Sup. Mat.** for 3D.

Integration Accuracy. By construction, Algorithm 1 is more accurate than any generic solver used as its subroutine. We also verified empirically, by comparing the integration error (that we can compute this error at all, is by the virtue of our closed-form solution) averaged over random CPA velocity fields. See **Sup. Mat.** for more details.

Timings. We timed our implementation on an Nvidia GTX 780 card. Tables 2 and 3 compare our solver and a generic one. Our implementation can be further optimized; e.g., we wrapped CUDA calls via python and not, say, C++. Table 4 shows how N_{steps} and N_{pts} affect the timings. Empirically, $N_{\text{steps}} = 10$ typically sufficed for good results.

Real-Time diffeomorphic image editing. Our code includes a GUI where a user chooses velocities at some/all vertices of \mathcal{P} . Using Lemma 3, the Gaussian priors on \mathbf{v}^θ , and the fast integration, we compute, in real time, the conditional warp, $\exp(E(\mathbf{v}^\theta | \text{user's choice}))$. A video with a demo is included in our **Sup. Mat.**

Expressiveness. While simple, CPAB transformations capture a wide range of diffeomorphisms (obviously, like any other work on diffeomorphisms, we make no claims about capturing non-diffeomorphisms such as occlusions). One way to see this is by inspecting samples from $p(\theta)$. M also contains what is sometimes called large deformations (an overloaded term, used in some texts for elements of G outside its identity component), e.g., a rectangle-to-horseshoe morphing; see **Sup. Mat.** Expressiveness is also

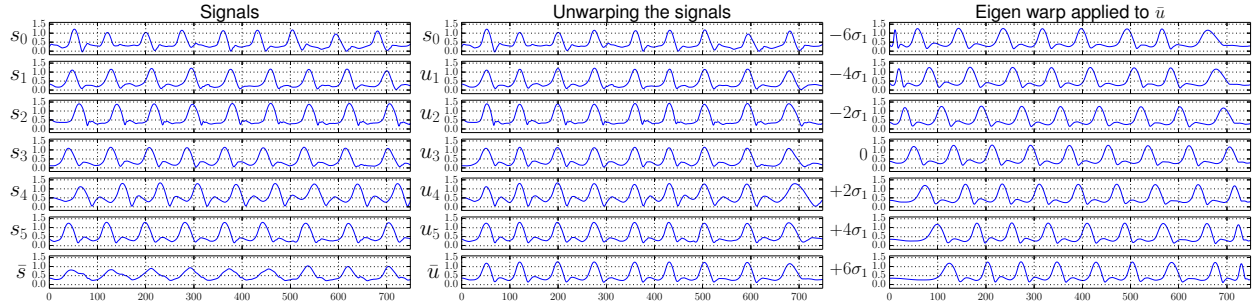


Figure 7: Nonlinear time warping in real MoCap Data. Left: a reference signal, s_0 , 5 other signals, $\{s_i\}_{i=1}^5$, and their mean, $\bar{s} = \frac{1}{5} \sum_{i=1}^5 s_i$. The mean is smeared since $\{s_i\}_{i=1}^5$ are misaligned. Center: having inferred the warps, $\{T^{\theta_i}\}_{i=1}^5$, we unwrap the signals, setting $u_i = s_i \circ T^{-\theta_i}$. Note $\{u_i\}_{i=1}^5$ are better aligned, as is evident by the details preserved in their mean, $\bar{u} = \frac{1}{5} \sum_{i=1}^5 u_i$. Right: warping \bar{u} by $T^{\bar{\theta} + j\sigma_1\xi_1}$ where $\bar{\theta} = \frac{1}{5} \sum_{i=1}^5 \theta_i$, $j \in \{-6, -4, -2, 0, 2, 4, 6\}$, ξ_1 is the first principal component of $\{\theta_i - \bar{\theta}\}_{i=1}^5$, and σ_1 is the corresponding standard deviation.

exemplified in the experiments below (Figs. 6 and 7).

Inferring latent transformations. Reasoning over real-world signals requires three oft-confounded stages: a *representation*; a *model* (e.g., a likelihood); *inference*. While these are related it is crucial to distinguish between them; e.g., Horn and Schunk’s representation+model of optical flow had been long regarded highly inaccurate, till modern inference rendered it almost state-of-the-art [39]. Focusing here on *representations*, we avoid advocating a particular likelihood or inference type, and instead use, in a coarse-to-fine manner, simple choices such as Gaussian likelihood and either conjugate-gradient optimization or the Metropolis algorithm (where, in the proposal, we use either global or local moves, utilizing Lemma 3). Without claiming these choices are optimal, these turned out to be effective.

Latent transformations arise in two typical settings: with and without known correspondences. In the first, also called (exact/inexact) diffeomorphic point matching [24, 18], one has *landmark* pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_{\text{pts}}}$ and seeks $T : \Omega \rightarrow \Omega$ such that $\{T(\mathbf{x}_i) \approx \mathbf{y}_i\}_{i=1}^{N_{\text{pts}}}$. We note that for $n = 1$, by Corollary 1, our ability to easily solve this problem using CPAB transformations means we can solve a *monotonic regression* problem, as we demonstrate in Fig. 4. Particularly, by adding boundary constraints and simulated annealing, we can well fit a CPAB transformation to any reasonable CDF; see also our **Sup. Mat.** More generally, but still for $n = 1$, our method can be used to solve inference/optimization problems over monotonic functions, CDFs and (cumulative sums of) histograms/filters (while respecting bin ordering). The case where $n \in \{2, 3\}$ appears, e.g., in landmark-based warping. As shown in Fig. 6, using real images and landmarks from a hands dataset ([37], www2.imm.dtu.dk/pubdb/p.php?403), we infer a transformation between different hands in different poses. We then apply it to the whole source image, creating new

hands, whose appearance belongs to the source hands, with geometry akin to the destination hands. We also animate the source image by evaluating ϕ^θ for different t ’s (including outside of $[0, 1]$); e.g., Fig. 6e shows animated frames for the example in the second row of Fig. 6a-6d. Our **Sup. Mat.** contains more examples, including ones that show we can handle a very large N_{pts} . In the no-correspondences case, there are two subsets of Ω , denoted U_{src} and U_{dst} , and two feature maps, $I_{\text{src}} : U_{\text{src}} \rightarrow \mathcal{F}$ and $I_{\text{dst}} : U_{\text{dst}} \rightarrow \mathcal{F}$, where \mathcal{F} is a feature space (e.g., color), and one seeks $T : \Omega \rightarrow \Omega$ such that $I_{\text{dst}} \circ T^{-1} \approx I_{\text{src}}$ (or $I_{\text{src}} \circ T \approx I_{\text{dst}}$); e.g., this is the case for time warps and landmark-free image registration. In Fig. 7 we show inferred time warps for MoCap data [22]. A similar process can be applied to the images in Fig. 6, though one may prefer information-theoretic measures to pixelwise likelihood. Lastly, we show an example for statistics on the inferred transformations. Particularly, Fig. 7 (right) shows warping \bar{u} (defined in the figure’s caption) along the 1st eigen warp (computed using PCA on the inferred velocity fields); i.e., we use Euclidean statistics in the linear \mathcal{V} and map the results to the nonlinear M .

5. Conclusion

We proposed new well-behaved transformations that are simple, expressive, efficient, and have many applications. We showed the transformations are computed fast and accurately and that they are highly expressive. Like the space in [3], our spaces lack closure under composition; but since $T^{\theta_2} \circ T^{\theta_1} \in G$, our representation can still be used within frameworks that apply consecutive transformations [43]. While this work focused on representation it was primarily motivated by modeling and inference: e.g., the ability to quickly run MCMC inference over diffeomorphisms is a hallmark of the method, showing that simplicity pays off.

References

- [1] S. Allasonnière, A. Trouvé, and L. Younes. Geodesic shooting and diffeomorphic matching via textured meshes. In *EMMCVPR*. Springer, 2005. 3, 4
- [2] N. Arad, N. Dyn, D. Reifeld, and Y. Yeshurun. Image warping by radial basis functions: Application to facial expressions. *CVGIP*, 1994. 4
- [3] V. Arsigny, O. Commowick, X. Pennec, and N. Ayache. A log-euclidean polyaffine framework for locally rigid or affine registration. In *BIR*. Springer, 2006. 3, 4, 8
- [4] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *IJCV*, 2005. 3
- [5] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *TPAMI*, 1989. 4
- [6] T.-L. Chen and S. Geman. Image warping using radial basis functions. *JAS*, 2014. 4
- [7] T. F. Cootes, C. J. Twining, V. Petrovic, R. Schestowitz, and C. J. Taylor. Groupwise construction of appearance models using piecewise affine deformations. In *BMVC*, 2005. 3
- [8] P. Dupuis, U. Grenander, and M. I. Miller. Variational problems on flows of diffeomorphisms for image matching. *QAM*, 1998. 3
- [9] S. Durrleman, S. Allasonnière, and S. Joshi. Sparse adaptive parameterization of variability in image ensembles. *IJCV*, 2013. 3
- [10] P. Fletcher. Geodesic regression and the theory of least squares on Riemannian manifolds. *IJCV*, 2012. 4
- [11] P. Fletcher, C. Lu, and S. Joshi. Statistics of shape via principal geodesic analysis on Lie groups. In *CVPR*, 2003. 4
- [12] P. Fletcher, C. Lu, S. Pizer, and S. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE TMI*, 2004. 4
- [13] O. Freifeld and M. Black. Lie bodies: A manifold representation of 3D human shape. In *ECCV*. Springer, 2012. 4
- [14] O. Freifeld, S. Hauberg, and M. Black. Model transport: Towards scalable transfer learning on manifolds. In *CVPR*, 2014. 4
- [15] U. Grenander. *General pattern theory: A mathematical study of regular structures*. Clarendon Press, 1993. 3
- [16] U. Grenander and M. Miller. Representations of knowledge in complex systems. *JRSS, Series B*, 1994. 3
- [17] U. Grenander and M. Miller. *Pattern theory: from representation to inference*. Oxford University Press, 2007. 3
- [18] H. Guo, A. Rangarajan, and S. Joshi. Diffeomorphic point matching. In *Handbook of Mathematical Models in Computer Vision*. Springer, 2006. 3, 8
- [19] M. T. Harandi, M. Salzmann, and R. Hartley. From manifold to manifold: geometry-aware dimensionality reduction for SPD matrices. In *ECCV*. Springer, 2014. 4
- [20] S. Hauberg, O. Freifeld, and M. Black. A geometric take on metric learning. In *NIPS*, 2012. 4
- [21] J. Hinkle, P. T. Fletcher, and S. Joshi. Intrinsic polynomials for regression on riemannian manifolds. *JMIV*, 2014. 4
- [22] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 2014. 8
- [23] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. Kernel methods on the riemannian manifold of symmetric positive definite matrices. In *CVPR*, 2013. 4
- [24] S. C. Joshi and M. I. Miller. Landmark matching via large deformation diffeomorphisms. *IEEE TIP*, 2000. 3, 8
- [25] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, 2013. 3
- [26] D. Lin, E. Grimson, and J. Fisher. Modeling and estimating persistent motion with geometric flows. In *CVPR*, 2010. 3
- [27] C. Liu, H.-Y. Shum, and C.-S. Zhang. A two-step approach to hallucinating faces: global parametric model and local nonparametric model. In *CVPR*, 2001. 4
- [28] M. Lorenzi and X. Pennec. Geodesics, parallel transport & one-parameter subgroups for diffeomorphic image registration. *IJCV*, 2012. 4
- [29] E. G. Miller and C. Chefd'hotel. Practical non-parametric density estimation on a transformation group for vision. In *CVPR*, 2003. 4
- [30] D. Mumford and A. Desolneux. *Pattern theory: the stochastic analysis of real-world signals*. AK Peters, 2010. 3
- [31] T. Nir, A. M. Bruckstein, and R. Kimmel. Over-parameterized variational optical flow. *IJCV*, 2008. 2
- [32] X. Pennec. Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. *JMIV*, 2006. 4
- [33] Y. Rubner, C. Tomasi, and L. Guibas. Metric for distributions with applications to image databases. In *CVPR*, 1998. 4
- [34] S. Sommer, F. Lauze, S. Hauberg, and M. Nielsen. Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximation. *ECCV*, 2010. 4
- [35] A. Srivastava, I. Jermyn, and S. Joshi. Riemannian analysis of probability density functions with applications in vision. In *CVPR*, 2007. 4
- [36] A. Srivastava, S. H. Joshi, W. Mio, and X. Liu. Statistical shape analysis: Clustering, learning, and testing. *TPAMI*, 2005. 4
- [37] M. B. Stegmann and D. D. Gomez. A brief introduction to statistical shape analysis, 2002. 8
- [38] R. Subbarao and P. Meer. Nonlinear mean shift over riemannian manifolds. *IJCV*, 2009. 4
- [39] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. 8
- [40] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 2, 3
- [41] S. Taheri, P. Turaga, and R. Chellappa. Towards view-invariant expression analysis using analytic shape manifolds. In *AFGR*, 2011. 4
- [42] M. Vaillant, M. I. Miller, L. Younes, and A. Trouvé. Statistics on diffeomorphisms via tangent space representations. *NeuroImage*, 2004. 3, 4
- [43] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache. Diffeomorphic demons: Efficient non-parametric image registration. *NeuroImage*, 2009. 3, 4, 8
- [44] Q. Xie, S. Kurtek, H. Le, and A. Srivastava. Parallel transport of deformations in shape space of elastic surfaces. In *ICCV*, 2013. 4
- [45] L. Younes. Spaces and manifolds of shapes in computer vision: An overview. *IVC*, 2012. 3