DOI: 10.1111/cgf.70035
EUROGRAPHICS 2025 / A. Bousseau and A. Dai
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 44 (2025), Number 2

Preconditioned Single-step Transforms for Non-rigid ICP

Yucheol Jung† Hyomin Kim† Hyejeong Yoon Seungyong Lee

POSTECH, South Korea

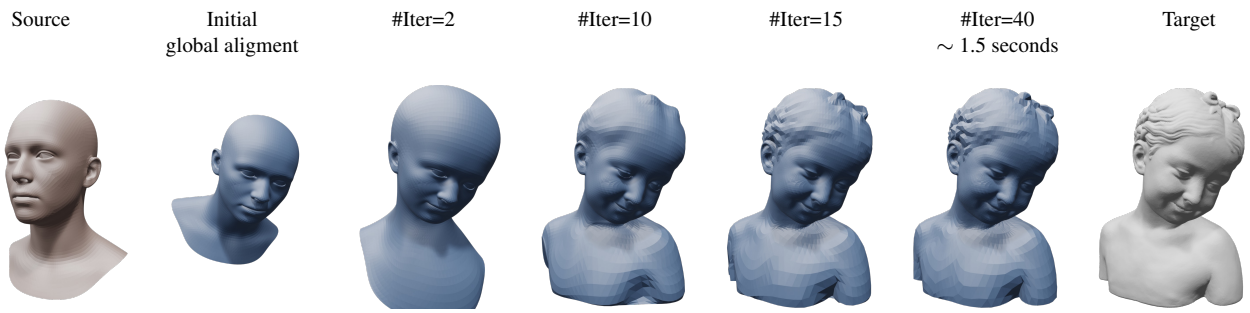

Figure 1: *Our optimizer enables fast and stable non-rigid ICP, even for challenging non-rigid alignment of two vastly different shapes.*

Abstract

Non-rigid iterative closest point (ICP) is a popular framework for shape alignment, typically formulated as alternating iteration of correspondence search and shape transformation. A common approach in the shape transformation stage is to solve a linear least squares problem to find a smoothness-regularized transform that fits the target shape. However, completely solving the linear least squares problem to obtain a transform is wasteful because the correspondences used for constructing the problem are imperfect, especially at early iterations. In this work, we design a novel framework to compute a transform in single step without the exact linear solve. Our key idea is to use only a single step of an iterative linear system solver, conjugate gradient, at each shape transformation stage. For this single-step scheme to be effective, appropriate preconditioning of the linear system is required. We design a novel adaptive Sobolev-Jacobi preconditioning method for our single-step transform to produce a large and regularized shape update suitable for correspondence search in the next iteration. We demonstrate that our preconditioned single-step transform stably accelerates challenging 3D surface registration tasks.

CCS Concepts

• ***Computing methodologies*** → *Mesh models; Reconstruction;* • ***Mathematics of computing*** → *Continuous optimization;*

1. Introduction

Non-rigid registration of 3D surfaces is an important procedure for 3D shape analysis of various deformable shape classes, such as 3D human faces [RBSB18; YZW*20; PZP*24] and the 3D human body [ASK*05; BRLB14; LMR*15], building dense point-wise correspondences between 3D shapes. Non-rigid ICP [HTB03; ARV07] is a popular framework for the registration of 3D shapes. The framework operates by alternating correspondence search and shape transformation to gradually transform one shape (the source) to minimize the error to the target shape while improving the quality of point-wise correspondence. A rich history of literature improving on various aspects of the ICP framework exists, including correspondence search [ACP03; LSP08; IZN*16], fitting error energy design [Rus19; MSC06; Hir22], deformation models [SP04; LH22; GSVS18], and numerical energy optimization [YDXZ20; YDXZ23].

In this work, we focus on the design of the numerical energy optimizer for non-rigid ICP. A popular approach used in many 3D shape dataset processing pipelines [YZW*20; QXQ*21; WZH*23] is to formulate each shape transform in non-rigid ICP as a linear least-squares problem [ARV07]. The solution for this linear least-squares problem can be obtained using a direct solver such as CHOLMOD [CDHR08] or an iterative solver like the preconditioned conjugate gradient (CG) method [CGO76].

† These authors contributed equally to this work.

© 2025 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

delivered by EUROGRAPHICS DIGITAL LIBRARY
www.eg.org diglib.eg.org

We note that exactly obtaining the solution for each shape transformation is inherently a wasteful computation because correspondences in early ICP iterations are of poor quality, and the linear system is changed every ICP iteration after correspondence updates. Therefore, instead of computing the optimum for the system, finding an approximate solution that is good enough for the next correspondence search would be ideal.

From this observation, we design a novel numerical optimization framework for non-rigid ICP. We start by considering the conjugate gradient method [CGO76] as the linear system solver. The conjugate gradient (CG) method can quickly compute an approximate solution of a linear system using a series of line searches. Still, multiple line searches per ICP iteration would be slow. Our key idea is that only one conjugate gradient line search per ICP iteration could be enough if we design an appropriate preconditioner for CG to find a line search direction for obtaining a transform expected to provide improved correspondence quality in the next correspondence search.

For this single-step transformation with a single CG line search to succeed, each line search must provide a stable correspondence search in the next ICP iteration. A traditional approach [ARV07; LSP08; RS12] for stable non-rigid ICP is to use a coarse-to-fine progression: fitting the coarse shape in early iterations and fitting the details in later iterations. While previous methods achieved this progression via smoothness energy, we instead employ a preconditioner for CG to realize the coarse-to-fine optimization.

Our preconditioner is based on the Sobolev preconditioner [AFW97], which was originally designed for accelerating the solver for partial differential equations and has also been used in the optimization of 3D geometry recently [MJBC13; SBI18; NJJ21] due to its bias towards coarse-to-fine optimization. The Sobolev preconditioner provides a strong bias towards a smoother solution for line search by diffusing the displacement of each vertex over the mesh, naturally prioritizing the fitting of a coarse shape while not prohibiting the reconstruction of fine details in the long run. However, unlike traditional preconditioners for CG, such as the Jacobi preconditioner [Wat15], designed to reduce the energy as quickly as possible, we observed that the Sobolev preconditioner does not sufficiently accelerate the optimization for non-rigid ICP.

To enable fast optimization while keeping the natural coarse-to-fine optimization from the Sobolev preconditioning, we design a novel adaptive Sobolev-Jacobi preconditioner. Our idea is to adaptively change the strength of the Sobolev preconditioner according to the number of iterations. Furthermore, we augment the Sobolev preconditioning with the Jacobi preconditioner to lower the condition number so that each line search can decrease the energy quickly.

Our experiments demonstrate that our method provides fast convergence even for challenging registration tasks (Fig. 1). We thoroughly analyze the effect of our single-step transformation and adaptive preconditioner. Compared to state-of-the-art non-rigid registration algorithms, our method provides an improved or equal level of fitting in a much smaller time frame. We further demonstrate applications in various shape domains.

In summary, our main contributions are as follows:

- We propose a novel *single-step transform* scheme to accelerate the computation of shape transformation in non-rigid ICP.
- We propose a novel *adaptive Sobolev-Jacobi preconditioner* that provides fast coarse-to-fine optimization for non-rigid ICP.
- By combining the single-step transform and the adaptive Sobolev-Jacobi preconditioner, we present an efficient numerical optimizer for shape transformation in non-rigid ICP.

2. Related Work

Surface registration methods can be classified into two major approaches: the intrinsic approach, which optimizes the parameterization of the surface, and the extrinsic approach, which optimizes the extrinsic embeddings of surfaces in 3D space. Non-rigid ICP is an extrinsic approach that iteratively deforms the source surface in 3D space to fit the target surface. It is a widely used framework due to its versatility; compared to intrinsic methods working in the 2D parameter domain [AL15; AL16], the non-rigid ICP framework is more generous with the difference in source-target topology and allows direct control over the deformed mesh in 3D space. For a comprehensive discussion of non-rigid ICP and registration in general, refer to the survey paper by Deng *et al.* [DYDZ22]. Here, we focus on the optimizer and preconditioners in non-rigid ICP.

2.1. Numerical Optimizer

The numerical optimizers employed for shape transformation in non-rigid ICP and surface registration were designed in many forms. One of the most widely used non-rigid ICP algorithms [ARV07] employs a direct linear system solver [Dav04] to minimize quadratic energy in a closed form. Similarly, Rouhani and Sappa [RS12] use a direct solver. Direct solvers are also heavily used in calculating transformations of surface maps, e.g., [ESB19; EHA*19].

Many non-rigid ICP frameworks [WJH*07; LSP08; YDXZ20] formulate their energy for shape transformation in the form of non-linear least squares, and their optimization is usually performed via quasi-Newton methods [NW99], such as Gauss-Newton and LBFGS [LN89]. Quasi-Newton methods locally approximate non-linear energy with quadratic energy at each ICP iteration and solve a linear equation for the quadratic energy. Recent works [LH22; JKH*23] also utilized gradient-based optimizers, such as gradient descent and Adam [KB15].

In our work, we consider a numerical optimizer for simple quadratic energy similar to the energy of [ARV07]. We start with a preconditioned conjugate gradient solver [CGO76] and design a preconditioner that is suitable for producing a desirable transform with a single conjugate gradient step.

2.2. Preconditioning

Preconditioning is a commonly adopted technique in iterative optimization to modulate the path of iterative updates. Usually, preconditioners are designed to approximate the inverse of the original system so that the condition number of the system is lowered to achieve a large decrease of the energy in a fewer number of iterations. Commonly adopted types of preconditioners in general

© 2025 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

cases include the Jacobi [NW99], incomplete Cholesky factorization [GV13], and SSOR [You14]. Such general-purpose preconditioners can be employed in non-rigid ICP to accelerate quadratic energy optimization via the conjugate gradient method [CGO76]. Data-driven learning of domain-specific preconditioners has also been explored [LBZ*20].

The Sobolev preconditioner [AFW97] was originally designed as a domain-specific preconditioner to accelerate the solution of partial differential equations. This preconditioner has also been employed in 3D shape deformation. Martin *et al.* [MJBC13] used a set of Sobolev preconditioners for coarse-to-fine optimization of surface smoothing, parameterization, and elasticity simulation. Slavcheva *et al.* [SBI18] applied the Sobolev preconditioner for the fusion of multiple truncated signed distance functions (TSDFs) defined on 3D voxels. Nicolet *et al.* [NJJ21] applied the Sobolev preconditioner in inverse rendering to diffuse sparse gradients from object silhouettes. Jung *et al.* [JKH*23] used the Sobolev preconditioner to regularize the smoothness of a 3D mesh while not conflicting with the goal of vertex density adaptation in inverse rendering and non-rigid registration.

However, there exists only a limited amount of discussion on the Sobolev preconditioning in non-rigid ICP [SBI18; JKH*23]. In this work, we analyze the effect of Sobolev preconditioning in non-rigid ICP of 3D meshes in terms of the condition number. Furthermore, we propose a novel adaptive Sobolev-Jacobi preconditioner to achieve fast coarse-to-fine optimization.

3. Non-rigid ICP via Linear Least Squares

We start by describing a simple formulation for non-rigid ICP based on a series of linear least-squares problems. We then propose our novel optimizer in Sec. 4.

3.1. Overall Framework of Non-rigid ICP

The non-rigid ICP framework to align two 3D shapes is designed as an iterative process. Given a source 3D mesh $\mathcal{S}_0$ and a target 3D mesh $\mathcal{T}$, $\mathcal{S}_0$ is gradually transformed toward $\mathcal{T}$ through a deformed source mesh $\mathcal{S}_i$ at the i-th iteration. Specifically, ICP alternates between 1) point-wise correspondence search between the deformed source $\mathcal{S}_i$ and the target $\mathcal{T}$ and 2) shape transformation $\mathcal{S}_i \rightarrow \mathcal{S}_{i+1}$ for improving the fitting error to $\mathcal{T}$ based on the correspondence.

For the point-wise correspondence search, we employ a simple nearest neighbor search between point samples on the source $\mathcal{S}_i$ and the target $\mathcal{T}$. For each point sample $\mathbf{v}_k \in \mathbb{R}^3$ on the source mesh $\mathcal{S}_i$, the closest point sample $\mathbf{v}'_l \in \mathbb{R}^3$ on the target $\mathcal{T}$ is matched to form a set of source-to-target correspondences $c = \{(k,l) \mid \mathbf{v}'_l \text{ is closest to } \mathbf{v}_k, \forall \mathbf{v}_k\}$. Similarly, we form a set of target-to-source correspondence c' by iterating over $\forall \mathbf{v}'_l$ instead of $\forall \mathbf{v}_k$. We use the union $c_i = c \cup c'$ as the set of point-wise correspondence for $\mathcal{S}_i$. We use triangle centroids of source and target meshes as point samples in the construction of c. For c', we use vertices instead of triangle centroids for source point samples.

We represent the surface $\mathcal{S}_i$ using a triangle mesh with N vertices, so the array of vertex positions $\mathbf{x} \in \mathbb{R}^{3N}$ is the optimized variable in the shape transform stage; the positions $\mathbf{p}$ of point samples is a function of $\mathbf{x}$ based on barycentric coordinates.

3.2. Shape Transform via Linear Least Squares

Our key interest is in the shape transform stage that follows the correspondence search stage. The shape transformation is formulated as an energy optimization problem based on the correspondences c_i. A popular choice for this shape transformation stage [ARV07] is to use a linear least-squares framework.

We build a simple linear least-squares problem using the sum of a point-to-plane error term E_{p2pl}, a landmark error term E_{lmk}, and a regularization term E_{reg}. The point-to-plane error is defined as

$$E_{p2pl}(\mathcal{S}_i, \mathcal{T}) = \frac{1}{|c_i|} \sum_{(k,l)\in c_i} \|(\mathbf{v}_k(\mathcal{S}_i) - \mathbf{v}'_l) \cdot \mathbf{n}'_l\|^2, \tag{1}$$

where $\mathbf{v}(\mathcal{S}_i)$ and $\mathbf{v}'$ are the sets of point samples on the surface $\mathcal{S}_i$ and $\mathcal{T}$, respectively, and $\mathbf{n}'_l$ is the surface normal of $\mathbf{v}'_l$. The landmark term, which enforces user-specified sparse correspondences c_{lmk}, is formulated as

$$E_{lmk}(\mathcal{S}_i, \mathcal{T}) = \frac{1}{|c_{lmk}|} \sum_{(k,l)\in c_{lmk}} \|\mathbf{v}_k(\mathcal{S}_i) - \mathbf{v}'_l\|^2. \tag{2}$$

The regularization energy is defined as the squared mean of Laplacians for vertex positions $\mathbf{x}$:

$$E_{reg}(\mathcal{S}_i) = \frac{1}{N} \sum_{k}^{N} \sum_{m\in\mathcal{N}(k)} \|\mathbf{x}_k(\mathcal{S}_i) - \mathbf{x}_m(\mathcal{S}_i)\|^2, \tag{3}$$

where $\mathcal{N}(k)$ is the set of neighboring vertices for the k-th vertex.

In our method, E_{reg} is not for strong smoothness regularization; such regularization comes from our preconditioner (Sec. 4.5). Instead, E_{reg} prevents occasional triangle flipping at sharp features, e.g., ear tips, and thus should have a low weight. We use a uniformly weighted Laplacian for numerical stability across diverse template meshes. While a high weight on E_{reg} with uniform weighting can distort the mesh, making triangles nearly equilateral, our low weight mitigates this issue. Although we prioritize stability across various meshes, cotangent-weighted Laplacians [PP93] could be beneficial in scenarios where preserving the shapes of triangles in the template is critical. We present an analysis of triangle distortions for both weighting schemes in Sec. 5.5.

Finally, the fitting energy is calculated as a weighted sum:

$$E_{fit}(\mathcal{S}_i, \mathcal{T}) = E_{p2pl}(\mathcal{S}_i, \mathcal{T}) + w_{lmk}E_{lmk}(\mathcal{S}_i, \mathcal{T}) + w_{reg}E_{reg}(\mathcal{S}_i), \tag{4}$$

with weights w_{lmk} and w_{reg}. We deform $\mathcal{S}_i$ to minimize the fitting energy and obtain the deformed shape $\mathcal{S}_{i+1}$, which is used in the next iteration. Unlike [ARV07], which employs dynamic scheduling for the regularization term E_{reg}, we use a fixed weight for the regularization term with a very small value.

The fitting energy E_{fit} is a linear least-squares function with respect to the vertex positions $\mathbf{x}$, and its minimization can be achieved by solving a normal equation [Str12]. We write the normal equation in the form of $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is a symmetric positive semi-definite (SPD) matrix of size $(3N \times 3N)$.

3.3. Motivation for Single-step Transforms

Our central focus of this work is on the numerical solver for the linear equation $\mathbf{Ax} = \mathbf{b}$ in the context of non-rigid ICP. A solu-

© 2025 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

tion for the linear equation can be obtained via direct sparse matrix solvers, such as Cholesky factorization [CDHR08] and LU factorization [Dav04], or iterative solvers like the preconditioned conjugate gradient (CG) method [CGO76]. However, we note that obtaining an exact solution for the linear equation is not necessary in non-rigid ICP, as $\mathbf{A}$ and $\mathbf{b}$ will change every ICP iteration with updated correspondences. Therefore, an approximate solution that can be computed quickly, although not fully satisfying the equation, would be more desirable.

A popular approach for obtaining an approximate solution for a linear equation is to exit the CG iteration early based on a prescribed error tolerance. In Algorithm 1, we outline the pseudocode of CG adopted for non-rigid ICP based on this approach. Starting from vertex positions $\mathbf{x}$ of the deformed source mesh $\mathcal{S}_i$, CG performs a sequence of line searches along a set of conjugate directions to produce the exact solution in $3N$ iterations. This iterative line search is stopped early using error tolerance ϵ to produce an approximate solution quickly. A notable property of the CG framework is that the sequence of search directions is heavily influenced by the preconditioner $\mathbf{M}$. We found that if we design a good preconditioner, a single iteration in CG can provide a good enough approximate solution in the context of non-rigid ICP. In the following section, we discuss the design of our preconditioner to achieve such a single-step solution.

Algorithm 1: ICP with Preconditioned Conjugate Gradient

Input: Initialization $\mathbf{x}_0$, preconditioner $\mathbf{M}$, tolerance ϵ
$\mathbf{x} \leftarrow \mathbf{x}_0$
// ICP loop
while *not converged* **do**
  $\mathbf{A}, \mathbf{b} \leftarrow \mathit{correspondenceSearch}(\mathbf{x}, \mathcal{T})$
  $\mathbf{r} \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}$; $\mathbf{z} \leftarrow \mathbf{M}^{-1}\mathbf{r}$
  $\mathbf{p}_{steepest} \leftarrow \mathbf{z}$
  $\mathbf{p} \leftarrow \mathbf{p}_{steepest}$
  // CG loop
  while $|\mathbf{r}| > \epsilon$ **do**
    // Perform line search
    $\mathbf{x} \leftarrow \mathbf{x} + \frac{\mathbf{r}^T\mathbf{z}}{\mathbf{p}^T\mathbf{A}\mathbf{p}}\mathbf{p}$
    // Find next line search direction $\mathbf{p}$
    $\mathbf{r}' \leftarrow \mathbf{r} - \frac{\mathbf{r}^T\mathbf{z}}{\mathbf{p}^T\mathbf{A}\mathbf{p}}\mathbf{A}\mathbf{p}$; $\mathbf{z}' \leftarrow \mathbf{M}^{-1}\mathbf{r}'$
    $\mathbf{p}_{conj} \leftarrow \mathbf{z} + \frac{\mathbf{r}'^T\mathbf{z}'}{\mathbf{r}^T\mathbf{z}}\mathbf{p}$
    $\mathbf{p} \leftarrow \mathbf{p}_{conj}$
    $\mathbf{r} \leftarrow \mathbf{r}'$; $\mathbf{z} \leftarrow \mathbf{z}'$
  end
end
return x

Algorithm 2: ICP w/ Preconditioned Single-step Transforms

Input: Initialization $\mathbf{x}_0$, preconditioner $\mathbf{M}$, tolerance ϵ, weight α
$\mathbf{A}, \mathbf{b} \leftarrow \mathit{correspondenceSearch}(\mathbf{x}, \mathcal{T})$
$\mathbf{x} \leftarrow \mathbf{x}_0$
$\mathbf{r} \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}$; $\mathbf{z} \leftarrow \mathbf{M}^{-1}\mathbf{r}$
$\mathbf{p} \leftarrow \mathbf{z}$
// ICP loop
while $|\mathbf{r}| > \epsilon$ **do**
  $\mathbf{x} \leftarrow \mathbf{x} + \frac{\mathbf{r}^T\mathbf{z}}{\mathbf{p}^T\mathbf{A}\mathbf{p}}\mathbf{p}$
  $\mathbf{A}, \mathbf{b} \leftarrow \mathit{correspondenceSearch}(\mathbf{x}, \mathcal{T})$
  $\mathbf{r}' \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}$
  $\mathbf{M} \leftarrow \mathit{updatePreconditioner}(\mathbf{A})$
  $\mathbf{z}' \leftarrow \mathbf{M}^{-1}\mathbf{r}'$
  $\mathbf{p}_{steepest} \leftarrow \mathbf{z}'$
  $\mathbf{p}'_{conj} \leftarrow \mathbf{z} + \frac{\mathbf{r}'^T\mathbf{z}'}{\mathbf{r}^T\mathbf{z}}\mathbf{p}$
  $\mathbf{p} \leftarrow (1-\alpha)\mathbf{p}_{steepest} + \alpha\mathbf{p}'_{conj}$
  $\mathbf{r} \leftarrow \mathbf{r}'$; $\mathbf{z} \leftarrow \mathbf{z}'$
end
return x

4. Single-step Transforms for Non-rigid ICP

We outline our overall framework in Algorithm 2. Our algorithm introduces two changes to the original CG algorithm.

First, for each iteration in the ICP loop, instead of running multiple steps from the CG loop, we take only a single CG step, consisting of a line search and a search direction update. Then, the design of the preconditioner $\mathbf{M}$ becomes more important to obtain desirable line search directions for single-step transforms, as the preconditioner is the major factor that affects the computation of the direction $\mathbf{p}$.

Second, since the first iteration of the original CG loop always uses the steepest descent direction $\mathbf{p}_{steepest}$, which is known to exhibit oscillations in the convergence [NW99], we instead use an approximate conjugate direction $\mathbf{p}'_{conj}$ computed assuming the updated matrix $\mathbf{A}$ is analogous to the previous one. However, $\mathbf{p}'_{conj}$ is not exactly conjugate to previous search directions due to the matrix update, and relying only on $\mathbf{p}'_{conj}$ can exhibit slow convergence. To accelerate the convergence while suppressing oscillations, we use a linear blend between $\mathbf{p}_{steepest}$ and $\mathbf{p}'_{conj}$ in Algorithm 2.

The primary factor influencing the effectiveness of Algorithm 2 is the choice of the preconditioner $\mathbf{M}$ that determines the search directions for our single-step transforms. We present the design of a suitable preconditioner in the remainder of this section. In Sec. 5.4, we analyze the effectiveness of the linear blend of search directions through an ablation study.

4.1. Desirable Properties of the Preconditioner

In this section, we discuss desirable properties for the preconditioner that allows successful single-step line searches. We first note that the primary goal of the shape transformation for non-rigid ICP is to provide a better shape for the correspondence search in the next iteration. One way to achieve this property is to use coarse-to-fine optimization [ARV07; LSP08; RS12; WXZ*20; SXZ*20]. Similarly, we wish to fit only the coarse shape in early iterations, then develop the shape details in the late iterations.

In addition, a sufficiently large decrease in the fitting error is also important for fast non-rigid ICP. The speed of energy decrease expected from a conjugate gradient method can be observed from the condition number of the system, which is the proportion of the

© 2025 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

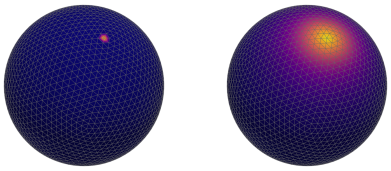

Figure 2: *Illustration of the effect of* $\mathbf{M}_{sob}^{-1}$*. Left: visualization of a function* $\mathbf{u} \in \mathbb{R}^{3N}$ *defined on a mesh. The function value is* $(1,1,1)$ *at one vertex (bright spot) and* $(0,0,0)$ *elsewhere. Right: visualization of* $\mathbf{M}_{sob}^{-1}\mathbf{u}$*. The preconditioner smoothly diffuses the function over the mesh. That is, the effect of a low-pass filter is applied to the function* $\mathbf{u}$*. We use* $t = 10, p = 2$ *for this illustration.*

largest and the smallest eigenvalues of the system. A small value of the condition number allows the conjugate gradient to converge fast [NW99]. Therefore, the preconditioner is desired to lower the condition number of the system.

Finally, since preconditioned conjugate gradient assumes that the preconditioner $\mathbf{M}$ is symmetric positive definite (SPD), we must satisfy SPD in the design of the matrix.

In summary, our design of the preconditioner $\mathbf{M}$ is aimed at providing a coarse-to-fine optimization while lowering the condition number of the system sufficiently. Also, the matrix $\mathbf{M}$ must satisfy the SPD condition.

4.2. Sobolev Preconditioning

In recent works in 3D shape optimization [MJBC13; NJJ21; JKH*23], the Sobolev preconditioner has demonstrated its role as a 3D shape regularizer. We consider the Sobolev preconditioner in the matrix form of

$$\mathbf{M}_{sob} = (\mathbf{I} + t\mathbf{L})^p, \tag{5}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{L} \in \mathbb{R}^{3N \times 3N}$ is the stiffness matrix of a cotangent Laplacian operator [PP93] for the initial source mesh $\mathcal{S}_0$, and $t \in [0, \infty)$, $p \in \mathbb{Z}_+$ are hyperparameters for the strength of the preconditioning. The matrix $\mathbf{M}_{sob}$ is SPD since the stiffness matrix $\mathbf{L}$ is SPD, and the sum of two SPD matrices, $\mathbf{I}$ and $t\mathbf{L}$, is also SPD.

The matrix $\mathbf{M}_{sob}^{-1}$ can be interpreted as an operator that diffuses a function $\mathbf{u} \in \mathbb{R}^{3N}$ defined over the mesh. For example, if we consider

$$\mathbf{y} = \mathbf{M}_{sob}^{-1}\mathbf{u}, \tag{6}$$

$\mathbf{y}$ is the result of applying a low-pass filter to the function $\mathbf{u}$. Fig. 2 illustrates this behavior.

When employed as a preconditioner for conjugate gradients, the matrix affects the update of the search direction $\mathbf{p}$. In Algorithm 2, the search directions are affected by the preconditioner via the equation

$$\mathbf{z} = \mathbf{M}^{-1}\mathbf{r} \tag{7}$$

and similarly via the equation for $\mathbf{z}'$. Since subsequent operations calculating $\mathbf{p}$ use $\mathbf{z}$ instead of the raw residuals $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$, we can observe that the high-frequency components in the residuals $\mathbf{r}$ of the fitting error E_{fit} are attenuated in the calculation of line search direction. As a result, the line search prioritizes minimization of the low-frequency components in the error residuals. Therefore, Sobolev preconditioning provides natural bias towards the coarse fitting, which helps avoid erroneous local correspondences in the initial iterations.

4.3. Sobolev Preconditioning and Condition Number

However, we observed that the Sobolev preconditioner does not significantly accelerate the optimization, as the decrease of errors slows down once coarse fitting has been reached. This behavior can be analyzed in terms of condition number. The condition number of a linear system is defined as the ratio of maximum and minimum eigenvalues $\lambda_{max}/\lambda_{min}$, where λ satisfies the relation

$$\mathbf{A}\mathbf{e} = \lambda\mathbf{e} \tag{8}$$

for an eigenvector $\mathbf{e}$.

To inspect the condition number for the preconditioned system, we examine the condition number of $\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}$ [NW99]. Its maximum and minimum eigenvalues can be obtained via the generalized eigenvalue problem [GV13];

$$\mathbf{A}\mathbf{e} = \lambda\mathbf{M}\mathbf{e}. \tag{9}$$

Usually, preconditioners are designed to approximate the inverse of the matrix $\mathbf{A}$ so that Eq. (9) is approximately transformed to

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{e} \approx \mathbf{I}\mathbf{e} = \lambda\mathbf{e}, \tag{10}$$

whose eigenvalues λ are always 1, resulting in the ideal condition number 1.

However, since the Sobolev preconditioner is constructed independently from the matrix $\mathbf{A}$, we cannot expect the relation $\mathbf{M}^{-1}\mathbf{A} \approx \mathbf{I}$. We empirically observed that Sobolev preconditioning has a limited effect in terms of decreasing the condition number. Fig. 3 shows the comparison of condition numbers of the linear systems with and without the Sobolev preconditioner. To further decrease the condition number, we modify the Sobolev preconditioner to be blended with a traditional inverse-approximating preconditioner.

4.4. Sobolev-Jacobi Preconditioning

The Jacobi preconditioner [NW99] is a simple and commonly used preconditioner for conjugate gradient methods to reduce the condition number of the system. The Jacobi preconditioner is defined as

$$\mathbf{M}_{jac} = \mathrm{diag}(\mathbf{A}). \tag{11}$$

We design the Sobolev-Jacobi preconditioner by combining the Sobolev preconditioner $\mathbf{M}_{sob}$ (Eq. (5)) and the Jacobi preconditioner:

$$\mathbf{M}_{sj} = \mathbf{M}_{sob} + \beta'\mathbf{M}_{jac}, \tag{12}$$

where β' is a scaling factor that equally balances the effects of the two preconditioners; we set $\beta' = \beta/\mathrm{median}(\mathrm{diag}(\mathbf{A}))$ with a hyperparameter β.

© 2025 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

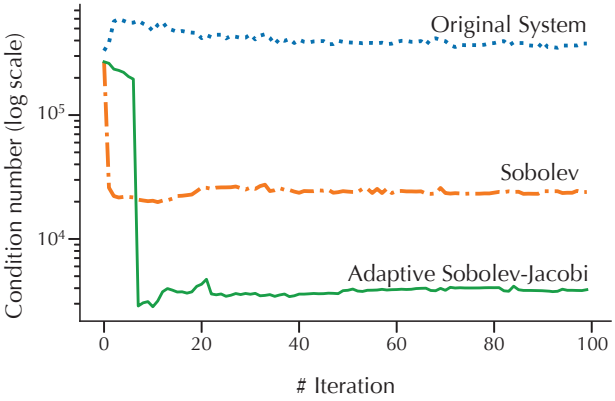

Figure 3: *Comparison of condition numbers of the linear system with different preconditioners over iterations. The plots visualize the condition numbers in the registration of a 3D face. The Sobolev preconditioner, while providing natural coarse-to-fine optimization, shows a limited effect in terms of decreasing the condition number. Our adaptive Sobolev-Jacobi preconditioner accelerates the optimization by drastically lowering the condition number while still providing coarse-to-fine optimization.*

Still, simply adding the two preconditioners is not enough to significantly accelerate the convergence; while $\mathbf{M}_{jac}$ itself approximates the inverse of $\mathbf{A}$, the addition of $\mathbf{M}_{sob}$ limits the decrease of the condition number.

4.5. Adaptive Sobolev-Jacobi Preconditioning

To ensure fast convergence, we dynamically adjust the strengths of $\mathbf{M}_{sob}$ and $\mathbf{M}_{jac}$ through ICP iterations. During coarse alignment, we leverage the strong diffusion of $\mathbf{M}_{sob}$ by assigning high values to t and p. However, once coarse alignment has been achieved, the Sobolev preconditioner exhibits slower error reduction and the diffusion strength should be reduced. Thus, after coarse alignment, we aim to strongly benefit from the influence of $\mathbf{M}_{jac}$.

One possible choice to achieve this goal is to dynamically change the weight β for β' in Eq. (12), but designing such a schedule is non-trivial. Instead, while fixing β, we design dynamically updated preconditioners $\hat{\mathbf{M}}_{sob}$ and $\hat{\mathbf{M}}_{jac}$ to obtain an adaptive preconditioner. That is, we define our adaptive Sobolev-Jacobi preconditioner as

$$\mathbf{M}_{asj} = \hat{\mathbf{M}}_{sob} + \beta' \hat{\mathbf{M}}_{jac}. \tag{13}$$

For $\hat{\mathbf{M}}_{sob}$, we start with a strong diffusion strength and gradually diminish the strength over iterations. We observed $p = 2$ allows providing sufficiently strong diffusion in most cases. Instead of discretely switching $p = 1$ and $p = 2$, we multiply two adaptive Sobolev preconditioners with $p = 1$, each having different decay rates $0 < \gamma_1, \gamma_2 \leq 1$:

$$\hat{\mathbf{M}}_{sob} = (\mathbf{I} + \gamma_1^k t_1 \mathbf{L})(\mathbf{I} + \gamma_2^k t_2 \mathbf{L}), \tag{14}$$

where k is the number of iterations and t_1 and t_2 are hyperparameters. $\hat{\mathbf{M}}_{sob}$ initially acts as a strong preconditioner with $p = 2$ and gradually transforms into a relatively weaker preconditioner with $p = 1$ because the matrix with faster decay rate approaches $\mathbf{I}$.

The adaptive update of our Sobolev preconditioner can be compared to scheduling the regularization weight w_{reg} in traditional non-rigid ICP [ARV07]. Both enable coarse-to-fine optimization, but our method simplifies parameter tuning by not altering the minimum of E_{fit}. Our adaptive preconditioning achieves coarse-to-fine optimization by improving the condition number without compromising the fitting criteria in E_{fit}. In contrast, scheduling w_{reg} shifts the minimum of E_{fit}, requiring careful planning of the desired level of fitting at each iteration for coarse-to-fine optimization.

For $\hat{\mathbf{M}}_{jac}$, since we observed excessive acceleration from the Jacobi preconditioner to be prone to oscillations in early iterations, we control the Jacobi preconditioner to take effect only after coarse alignment using

$$\hat{\mathbf{M}}_{jac} = \mathbb{1}[\gamma_1^k < \omega] \cdot \mathbf{M}_{jac}, \tag{15}$$

where $0 < \omega \leq 1$ is a hyperparameter and $\mathbb{1}[\cdot]$ is an indicator function producing 1 when the condition is true.

We use the matrix $\mathbf{M}_{asj}$ as our final preconditioner. At the beginning, the condition number of $\mathbf{M}_{asj}$ is dominated by strong diffusion of $\hat{\mathbf{M}}_{sob}$, but the effect of $\hat{\mathbf{M}}_{sob}$ is diminished in later iterations, where the effect of $\hat{\mathbf{M}}_{jac}$ becomes stronger, reducing the condition number. Fig. 3 shows that our adaptive preconditioner significantly decreases the condition number compared to the fixed Sobolev preconditioner. The computation required for $\mathbf{M}_{asj}^{-1}$ can be efficiently performed via cuDSS [NVI24] because $\mathbf{M}_{asj}$ produces an identical sparsity pattern in every iteration.

5. Experiments

We evaluate our method on various challenging non-rigid registration tasks, including the registration of 3D cartoon faces with large exaggerations, diverse 3D faces from different sources, and the 3D human body.

In our quantitative evaluation and ablations, we use 3DCaricShop [QXQ*21], which contains highly detailed artist-sculpted 3D caricature meshes, each with 38 landmark annotations. We consider two template meshes to be used as the source mesh in the registration. First, *Face* template is the template mesh with 11K vertices originally used for the registration of 3DCaricShop in [QXQ*21], possessing the shape of a regular 3D face without exaggeration. Second, *Sphere* template is a spherical mesh with 10K vertices. Since this spherical shape requires more complex deformation compared to *Face*, registration using this template is generally more challenging. Both template meshes are annotated with landmarks. For *Face*, we use the template landmark annotations provided in 3DCaricShop. For *Sphere*, we use the landmark annotations provided by [JKH*23].

5.1. Implementation Details

Our prototype is implemented using PyTorch [PGM*19]. We run the evaluations on a desktop PC with NVIDIA RTX4090 and Intel i9-13900K. We use the following hyperparameters unless mentioned otherwise: $w_{lmk} = 1.0$, $w_{reg} = 0.1$ $\epsilon =$ 5e-6, $\alpha = 0.5$, $t_1 = 50$,

© 2025 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

Table 1: *Quantitative comparison with state-of-the-art non-rigid registration methods. For ours, we separately report the times spent on the preparation and the main iterations in the form of (preparation + main iterations).*

	Chamfer	Normal	Time (sec.)
Face → 3DCaricShop			
[ARV07]	1.91e-3	6.09e-2	28.3
[QXQ*21]	3.35e-2	8.80e-2	N/A
[YDXZ20]	1.16e-4	5.51e-2	3.74
[LH22]	2.95e-4	7.43e-2	3.09
[Hir22]	2.81e-4	1.04e-1	10.4
[ZJM*24]	6.19e-5	3.92e-2	45.0
Ours (Balance)	5.53e-5	1.42e-2	**0.35 + 0.65**
Ours (Quality)	**5.48e-5**	**1.24e-2**	0.35 + 3.73
Sphere → 3DCaricShop			
[ARV07]	3.28e-2	2.24e-1	25.2
[YDXZ20]	1.35e-2	9.01e-2	8.26
[LH22]	4.53e-4	8.44e-2	3.94
[Hir22]	5.50e-4	1.40e-1	5.20
[ZJM*24]	9.10e-5	9.28e-2	34.4
[JKH*23]	6.35e-5	1.30e-2	24.8
Ours (Balance)	5.53e-5	1.28e-2	**0.35 + 0.61**
Ours (Quality)	**5.46e-5**	**1.07e-2**	0.35 + 3.2

$t_2 = 10$, $\gamma_1 = 0.9$, $\gamma_2 = 0.99$, $\omega = 0.5$, $\beta = 100$. We set 400 as the maximum number of iterations for our method. Refer to the supplementary document for implementation details of *correspondenceSearch()* and *updatePreconditioner()* in Algorithm 2.

To evaluate the fitting accuracy of registration results, we measure two error metrics: Chamfer distance and surface normal error. We first sample 100,000 points from each of the fitting result and the target mesh. Chamfer distance is defined as the average of the Euclidean distances between the closest fitting-target point pairs. It measures the overall alignment of the fitting and the target mesh. Using the point pairs used for the Chamfer distance, we also measure the discrepancy of the surface normals using cosine distance $1 - \cos(\Delta\theta)$, where $\Delta\theta$ is the angle between fitting and target surface normals. Surface normal error can be high if shape details are not reconstructed faithfully or the fitting exhibits artifacts such as triangle flipping. Lower is better for both error measures. We use the implementation of PyTorch3D [RRN*20] to compute the Chamfer distance and normal error.

All reported time measurements are in the unit of seconds. For ours, we separately report the times spent on preparation and on the main iterations. In the preparation stage, the sparsity pattern of our adaptive preconditioner is analyzed for factorization. This analysis result can be shared across different fitting instances if the same template mesh is used.

5.2. Comparison with State-of-the-art Methods

In Table 1 and Figs. 11 and 12, we compare our method with state-of-the-art non-rigid registrations. The methods were evaluated on 100 artist-sculpted 3D caricatures as the target. We report the average error metric values and time. We evaluated two presets for our method: *Ours (Balance)* and *Ours (Quality)*, with the only difference being the selection of the tolerance parameter ϵ deciding the criterion for convergence. We used ϵ =5e-6 for the Balance preset and ϵ =5e-7 for the Quality preset. Our running time is reported in the format of (preparation + main iterations).

[ARV07] is a widely used method for non-rigid registration of 3D shape datasets [YZW*20; WZH*23] based on a series of linear least-squares problems, similarly to ours. We used an open-source GPU-accelerated implementation [wuh21] for the evaluation. We experienced this method failed to reconstruct strongly exaggerated shapes, such as sharp chins, and showed slow fitting by spending time finding the exact solution for each linear least-squares problem. Our method produced more accurate and fast registration despite using a similar simple correspondence search and linear formulation, thanks to our novel optimizer design.

[QXQ*21] is the template registration result included in the dataset distribution. We noticed the registration results miss many important details from the artist-sculpted 3D faces with fine details and large exaggerations, such as the exaggerated chin shape and the shape of the eyes. Our method produces accurate reconstruction of the targets, including sharp details around the eyes and wrinkles.

[YDXZ20] is one of the state-of-the-art non-rigid registration methods providing robust and fast convergence. We used the official CPP code without GPU acceleration that is open-sourced by the authors. This method provided accurate fitting results for the *Face* template. In contrast, the result using the *Sphere* template produced poor fitting with many missing details. Our method produced accurate fitting for both templates using the same set of hyperparameters.

[LH22] is a state-of-the-art neural framework for non-rigid registration based on a pyramid of neural deformation fields. We used the official PyTorch GPU implementation for evaluation. We did not use landmark inputs in this case because the framework does not support landmarks. While this method performed well in the original demonstration of matching human body poses between different scans, it showed limited performance in our task of 3D cartoon face registration. We observed a loss of details and a strong bias towards local rigidity. Our method faithfully reconstructs shape details.

[Hir22] is one of the state-of-the-art non-rigid registration frameworks based on coherent point drift (CPD) [MS10]. We do not employ landmark inputs since the official CPU implementation was not designed to process landmark inputs. While this method provides a good coarse fitting, its bias towards local smoothness results in a loss of details, producing high normal error. Our method provides accurate fitting in terms of shape details.

[ZJM*24] is the most recent state-of-the-art non-rigid registration method for point clouds. We used the official MATLAB implementation with GPU acceleration. To test this framework for mesh registration, we used the mesh vertices as the input and copied the original meshing to the output. While this method demonstrated accurate fitting in terms of Chamfer distance, it does not consider the meshing of the source shape, resulting in an output mesh with many artifacts and significantly increased normal error. Furthermore, the method showed slow convergence due to the large size of the input. Our method provides a cleaner mesh with faster convergence speed.

© 2025 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

Table 2: *Quantitative comparison with other baseline preconditioners. We separately report the times spent on the preparation and the main iterations.*

	Sob	AdaSob	Jac	Sob-Jac	Ours
Face → 3DCaricShop					
Prep. (sec.)	0.15	0.35	0	0.35	0.35
Main (sec.)	7.10	6.41	0.53	10.42	0.66
# Iter	400	353	28	400	28
Chamfer	5.62e-5	5.43e-5	6.43e-5	5.72e-5	5.48e-5
Normal	1.95e-2	1.27e-2	4.05e-2	2.22e-2	1.49e-2
Sphere → 3DCaricShop					
Prep. (sec.)	0.15	0.35	0	0.35	0.35
Main (sec.)	5.62	3.13	0.62	4.56	0.61
# Iter	322	171	34	182	26
Chamfer	5.49e-5	5.43e-5	5.75e-5	5.60e-5	5.44e-5
Normal	1.47e-2	1.18e-2	2.12e-2	1.74e-2	1.17e-2

[JKH*23] uses the Sobolev preconditioner in non-rigid ICP using the Adam [KB15] optimizer and demonstrates state-of-the-art results in the registration of 3DCaricShop using vertex density adaptation. We used the official PyTorch GPU code in the evaluation. For this method, we report the result for the *Sphere* template only because this framework takes a spherical mesh as the input. Despite GPU acceleration, its convergence is slow due to the use of the Adam optimizer, which is versatile but requires many iterations to converge. Furthermore, with Sobolev preconditioning, the reconstruction of fine details is slow after coarse alignment. In contrast, our optimizer, using preconditioned single-step transforms, produces much faster convergence and more accurate results.

5.3. Evaluation of Adaptive Sobolev-Jacobi Preconditioning

To evaluate our adaptive Sobolev-Jacobi preconditioner, we provide a comparison of different preconditioner choices for our method in Table 2 and Fig. 4. We use 10 artist-sculpted meshes from 3DCaricShop as the target and report the average time and errors. We explain each baseline preconditioner and discuss its property in the following.

Sob is a fixed Sobolev preconditioner $\mathbf{M}_{sob}$ (Eq. (5)) with $t = 10, p = 1$. In this case, as the preconditioner has fewer non-zero elements than ours, it exhibits a lower preparation time. The Sobolev preconditioner has been employed in [JKH*23], but as discussed in the previous section, its convergence rate is slow. We observed that using a fixed Sobolev preconditioner slows down the decrease in energy after reaching coarse alignment. As a result, the iterations reach the maximum iteration of 400 before reaching the tolerance ϵ =5e-6.

AdaSob is the adaptive Sobolev preconditioner $\hat{\mathbf{M}}_{sob}$ (Eq. (14)) with the same hyperparameters as ours. Compared to ours, this baseline does not employ the adaptive Jacobi preconditioner. This baseline, compared to the fixed Sobolev preconditioner, provides faster convergence to some degree, but still shows slow convergence, reaching close to the maximum number of iterations.

Jac is the result of applying the fixed Jacobi preconditioner $\mathbf{M}_{jac}$ (Eq. (11)) without the adaptive Sobolev preconditioner. The Jacobi preconditioner provides fast convergence but suffers from a high normal error, which indicates that the results contain artifacts, including self-intersections and flipped triangles. The primary reason for this failure is the lack of coarse-to-fine optimization, which is crucial for maintaining a stable sequence of ICP correspondences.

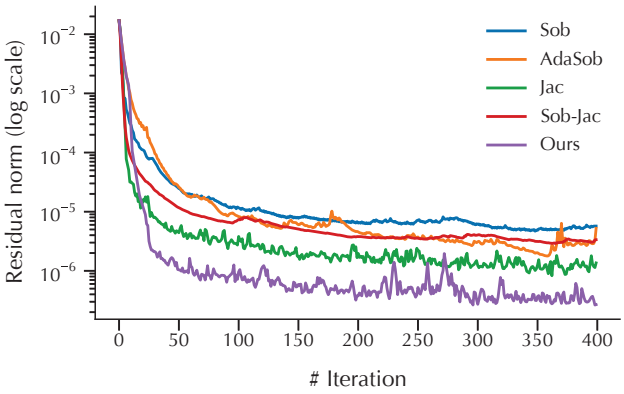

Figure 4: *Comparison of convergence speed with other baseline preconditioners. The norm of the residual $|\mathbf{r}| = |\mathbf{b} - \mathbf{A}\mathbf{x}|$ in the registration for one target shape is presented at different iterations. We use tolerance $\epsilon = 0$ to reach the maximum number of iterations. Sob, AdaSob, and Sob-Jac suffer from slow convergence. Jac shows faster initial decrease in the residual norm, but becomes stuck in a local minimum due to the lack of coarse-to-fine optimization. Our adaptive Sobolev-Jacobi preconditioner produces an accurate solution quickly.*

Sob-Jac is the result of applying the Sobolev-Jacobi preconditioner $\mathbf{M}_{sj}$ (Eq. (12)) without the adaptive update. In this setting, we set $\gamma_1 = 1$, $\gamma_2 = 1$, and $\omega = 0$. As the strong smoothing effect of the Sobolev preconditioner does not diminish over iterations, the addition of the Jacobi preconditioner does not significantly improve the convergence speed.

Ours shows the best result with good reconstruction quality and fast convergence by benefiting from both the coarse-to-fine optimization of the Sobolev preconditioner and the fast convergence from the Jacobi preconditioner.

5.4. Ablation Study

We present an ablation study for our hyperparameters. In this experiment, we use 10 shapes from 3DCaricShop as the targets, and report average time and errors. We use the *Sphere* template as the source shape. We report the running time for the main iterations in the following experiment. The running time for the preparation stage was 0.35 seconds.

Blending weight α for line search directions In Algorithm 2, we employ a linear blend between the steepest descent direction $\mathbf{p}_{steepest}$ and the approximate conjugate gradient direction $\mathbf{p}'_{conj}$ computed assuming the updated matrix $\mathbf{A}$ is analogous to the previous one. We present the ablation study of the blending weight in Table 3. Compared to pure steepest descent ($\alpha = 0$) and pure conjugate gradients ($\alpha = 1$), our choice of $\alpha = 0.5$ produces the fastest convergence with a low normal error.

© 2025 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

Table 3: *Ablation of the search direction blending weight* α

α	0.0	0.1	0.5 (Ours)	0.9	1.0
Main (sec.)	0.67	0.63	**0.60**	0.79	0.80
# Iter	30	28	**26**	36	36
Chamfer	5.45e-5	5.45e-5	5.45e-5	5.44e-5	5.44e-5
Normal	1.22e-2	1.22e-2	1.18e-2	1.15e-2	1.15e-2

Table 4: *Ablation of the tolerance* ϵ

ϵ	5e-6	1e-6	8e-7	5e-7	None
Main (sec.)	0.61	1.47	1.85	3.15	7.46
# Iter	26	74	95	166	400
Chamfer	5.53e-5	5.47e-5	5.47e-5	5.46e-5	5.46e-5
Normal	1.28e-2	1.12e-2	1.10e-2	1.07e-2	1.04e-2

Table 5: *Ablation of the scale of parameters* t_1 *and* t_2 *for the strength of* $\hat{\mathbf{M}}_{sob}$.

Scale	1	5	10 (Ours)	20	100
Main (sec.)	0.52	0.58	0.60	0.63	0.77
# Iter	22	24	26	27	35
Chamfer	5.57e-5	5.45e-5	5.45e-5	5.43e-5	5.46e-5
Normal	1.76e-2	1.26e-2	1.17e-2	1.16e-2	1.25e-2

Tolerance ϵ **for convergence criterion** ϵ acts as an intuitive hyperparameter for the tradeoff between fitting quality and speed. We present the fitting experiments with different tolerance values in Table 4. The number of iterations increases as ϵ decreases. We find 5e-6 provides a good balance, and 5e-7 provides fine detail reconstruction at the small cost of increased iterations.

Parameters t_1, t_2 **for the strength of** $\hat{\mathbf{M}}_{sob}$ t_1 and t_2 affect the degree of diffusion from the Sobolev preconditioning. Table 5 presents fitting results using different scales of the parameters. The scales are multiplied to both $t_1 = 5$ and $t_2 = 1$. Generally, a higher scale provides stronger attenuation of high-frequency transforms in the early iterations and requires more iterations to converge. The results are not highly sensitive to the scale unless given a very low value, e.g., 1.

Parameters ω, β **for** $\hat{\mathbf{M}}_{jac}$ Table 6 presents an ablation of ω, which decides when the Jacobi preconditioning takes effect. ω is not too sensitive to the selection. Table 7 shows the ablation of β, determining the strength of the Jacobi preconditioner. For our experimental results, we used the values that balance between the required iterations and fitting accuracy.

5.5. Distortion Analysis

To analyze distortions in our fitting, we measured the symmetric Dirichlet energy [SPK23], where lower values indicate less distortion. We compare the average symmetric Dirichlet energy of *Ours (Quality)* with those of [QXQ*21] and [LH22], which show visually clean surfaces (see Fig. 11). We also evaluated our method using cotangent weighting [PP93] in the regularization term E_{reg}, referred to as *Ours (Cotan)*. This experiment uses the *Face* template to register 100 inputs from the 3DCaricShop dataset.

On average, our method produced a lower distortion energy (24.8) than the original fittings (40.3) included in the 3DCaricShop dataset [QXQ*21]. While [LH22] demonstrated the minimum energy (1.40), it suffered from severe under-fitting. We observed that many original fittings in the 3DCaricShop dataset exhibited lower distortions than our results, while some exhibited higher distortions. However, these lower distortions come with under-fitting due to smooth deformations. Our method may introduce a modest level of distortion to achieve a tight fit without explicit regularization for distortion. Combining our approach with explicit minimization of distortion energies used in surface maps, e.g., [SPK23], is a promising future research direction.

The use of uniform weighting in the weak regularization term E_{reg} could incur distortions since it favors equilateral triangles, whereas the template mesh may consist of irregular triangles, e.g., the *Face* template. *Ours (Cotan)* produced a lower distortion (7.89) in most cases compared to *Ours (Quality)* with some exceptions. While uniform weighting produces stable gradients regardless of triangle shapes, cotangent weighting for E_{reg} can be preferable if the template consists of irregular triangles. Refer to the supplementary document for a full table of results of each instance in the 100 inputs.

Table 6: *Ablation of the parameter* ω *for the scheduling of* $\hat{\mathbf{M}}_{jac}$.

ω	0.1	0.3	0.5 (Ours)	0.7	0.9
Main (sec.)	0.70	0.61	0.60	0.60	0.60
# Iter	33	27	26	26	25
Chamfer	5.47e-5	5.44e-5	5.45e-5	5.44e-5	5.43e-5
Normal	1.29e-2	1.19e-2	1.18e-2	1.19e-2	1.30e-2

Table 7: *Ablation of the parameter* β *for the strength of* $\hat{\mathbf{M}}_{jac}$.

β	1	10	50	100 (Ours)	1000
Main (sec.)	0.97	0.69	0.62	0.61	0.53
# Iter	46	30	27	26	22
Chamfer	5.43e-5	5.44e-5	5.45e-5	5.44e-5	5.45e-5
Normal	1.21e-2	1.19e-2	1.17e-2	1.18e-2	1.30e-2

5.6. Registration of Noisy Inputs

Our method can handle moderate levels of noise in the input. The Sobolev-Jacobi preconditioner enables coarse-to-fine optimization, naturally rejecting fine-scale noise until the later stages of optimization. When we performed an additional experiment for *Ours (Balance)* with Gaussian noise in the target mesh using the Sphere template, our method still achieved an accurate fitting. In this case, the Chamfer error was 5.73e-5, which is lower than all other methods, and the normal error was 1.77e-2, which is comparable to that of [JKH*23] and lower than those of other methods. We used 10% of the average edge length of the input mesh as the standard deviation for the noise.

When the scale of the noise is large, our registration may not successfully recover a smooth surface from the input. Fig. 5 shows visual results for the registration of noisy inputs with different noise scales. Our method reconstructs a reasonably smooth surface when the standard deviation for the noise is 10% of the average edge length. However, when the standard deviation reaches over 20%,

© 2025 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

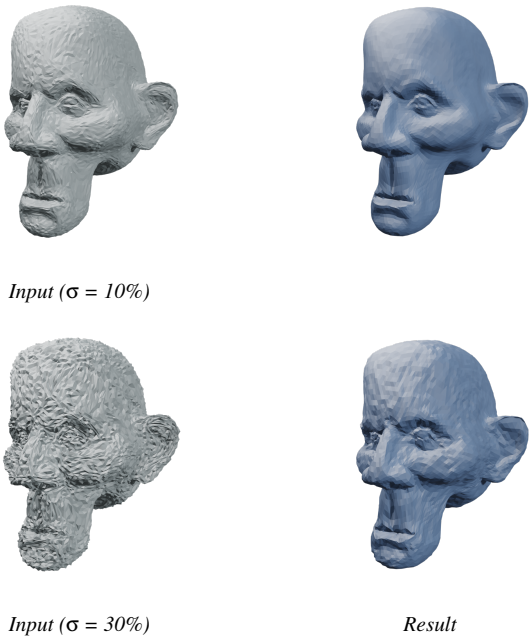

Figure 5: *Registration of inputs with Gaussian noise. Results using the standard deviation σ set to 10% and 30% of the average edge length of the input mesh are presented. Our method produces a smooth surface when the scale of the noise is moderate, e.g., σ=10%.*

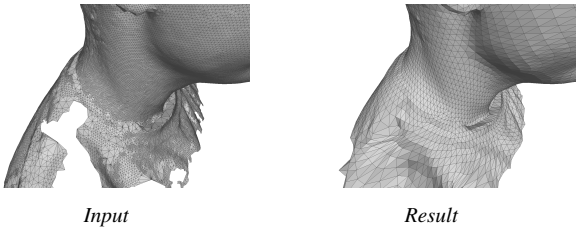

Figure 6: *Fitting result for an input with moderately sized holes. Our method automatically fills moderately sized holes due to the point-to-plane fitting error and smoothness regularization from the preconditioner.*

the registration produces a noisy surface. Handling of large noise and outliers is left as future work.

5.7. Registration of Incomplete Inputs with Holes

In this section, we provide results and analysis for cases where the input contains an incomplete surface, i.e., holes. While the bidirectional calculation of E_{p2pl} assumes bijection, we observed that moderately sized holes in input surfaces are automatically filled via our registration thanks to the point-to-plane error metric and the smoothness regularization from the preconditioner.

The point-to-plane error does not bring two points to the same positions, preventing the source points from being attracted to the target hole boundaries. Fig. 6 visualizes such a case, where the hole on the shoulder is smoothly filled automatically. Still, the filled region exhibits slightly distorted triangles due to weak but persistent attraction to the hole boundaries.

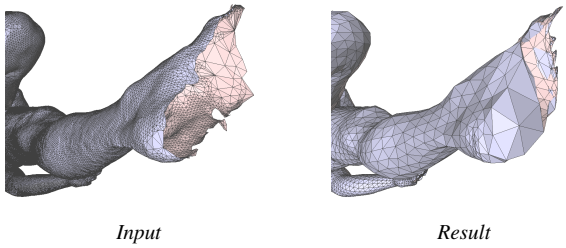

Figure 7: *Fitting result for an input with a large hole. The registration fails because our simple formulation does not account for partial-to-partial mappings in the correspondence search and the fitting error calculation.*

When the hole is too large and the hole boundary is formed along a high-curvature region, the points on the template can be attracted to the boundary of the hole, and the fitting result may not naturally fill the missing surface. Fig. 7 visualizes such a case, where the hole cut along the boundary of the foot is not smoothly filled via our registration. A modification in the correspondence search, e.g., incorporating pruning as in [HAWG08], would be required to fully handle partial-to-partial matching.

5.8. Gallery of Registration Results

Fig. 8 presents results for diverse 3D faces using a single 3D face template mesh as the source. We use 24 landmark annotations to apply an initial global alignment of the source mesh using a rigid transformation [Ume91], followed by landmark-based deformation [LSC*04]. Our method enables stable and accurate reconstruction of diverse facial shapes even when the source mesh is substantially different from the target mesh.

Fig. 9 demonstrates registration between two 3D meshes with a large number of vertices, where both source and target possess approximately 32K vertices. We do not use landmarks in this scenario. The registration converges in 20 iterations, taking 1 second for the preparation and 1.5 seconds for the main iterations. Our method can process dense input meshes efficiently and produce accurate results even without landmarks given a reasonable initial alignment.

Fig. 10 presents the application of our method for the registration of a human body dataset [YZG*21]. We initialize the pose of the source shape for each target shape using SMPL [LMR*15]. The SMPL template mesh was subdivided to generate additional vertices to express surface details. Our method reconstructs fine details in the target shape accurately. Additionally, we present more examples on human body shapes from the SHREC dataset [MMRC20] in the supplementary document.

© 2025 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

Figure 8: *Registration of diverse 3D faces using a single template mesh. The supplementary video includes animated interpolation examples between different shapes.*

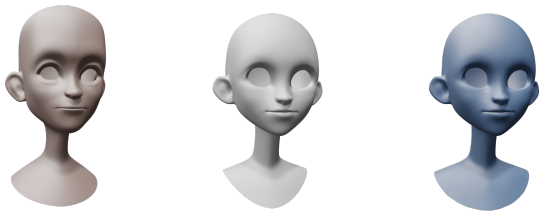

Figure 9: *Registration between 3D meshes with a large number of vertices (32K). Left: template mesh. Middle: target mesh. Right: Fitting result. Our method produces fast and accurate results even without landmarks given a reasonable initial alignment.*

6. Conclusion and Future Work

We designed an optimizer to handle two key traits of non-rigid ICP. First, ICP optimization handles iteratively updated equations based on intermediate solutions. Second, coarse-to-fine optimization is desirable for stable correspondence search in ICP. Our method efficiently handles both traits by providing fast approximate solutions (single-step transforms) designed to improve intermediate solutions in a coarse-to-fine manner (preconditioner). The two traits can also exist in other problems, such as inverse rendering where pixel-surface correspondence is dynamically updated and coarse-to-fine optimization is usually desirable. Extending our framework to such problems is a promising future work.

Application of our preconditioner to a more complex and non-linear energy is an interesting direction for future work. Many non-rigid ICP frameworks are based on quasi-Newton methods using non-linear least squares optimizers like Gauss-Newton. Such optimizers typically operate by iteratively solving a linear equation that minimizes a locally approximated quadratic energy. We can apply our method to handle the quadratic energy at each iteration by modifying the updates of A and b in Algorithm 2.

This work focused on the development of the numerical optimizer in non-rigid ICP while keeping other components as simple as possible. Currently, our simple linear least-squares formulation is not designed for registration requiring large rotations. For example, registration of the human body requires handling of large rotations due to poses of the limbs. Such cases can be handled by incorporating an embedded deformation graph as in [YDXZ20]. In addition to large rotations, non-rigid ICP presents many other challenges, such as partial surfaces, noisy data, differences in topology between source and target, self-intersections, and so on. Exploiting the key idea of our single-step transform scheme, i.e., computing an approximate transform with just enough accuracy needed for the next fitting iteration, to address these challenges is a promising future direction.

Acknowledgements

This work was supported by NRF grants (RS-2023-00280400, RS-2024-00451947) and IITP grants (ICT Research Center, RS-2024-00437866; AI Innovation Hub, RS-2021-II212068; AI Graduate School Program, RS-2019-II191906) funded by the Korean government (MSIT), and Samsung Electronics (IO201209-07904-01).

References

[ACP03] ALLEN, BRETT, CURLESS, BRIAN, and POPOVIĆ, ZORAN. "The space of human body shapes: reconstruction and parameterization from range scans". *ACM Trans. Graph.* 22.3 (2003), 587–594 1.

[AFW97] ARNOLD, DOUGLAS, FALK, RICHARD, and WINTHER, RAGNAR. "Preconditioning in H (div) and applications". *Mathematics of computation* 66.219 (1997), 957–984 2, 3.

[AL15] AIGERMAN, NOAM and LIPMAN, YARON. "Orbifold tutte embeddings." *ACM Trans. Graph.* 34.6 (2015), 190–1 2.

[AL16] AIGERMAN, NOAM and LIPMAN, YARON. "Hyperbolic orbifold tutte embeddings." *ACM Trans. Graph.* 35.6 (2016), 217–1 2.

© 2025 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

Figure 10: *Registration of human body. From back to front: source mesh, target shape, and fitting result. Our registration reconstructs fine details in the target shape accurately.*

© 2025 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

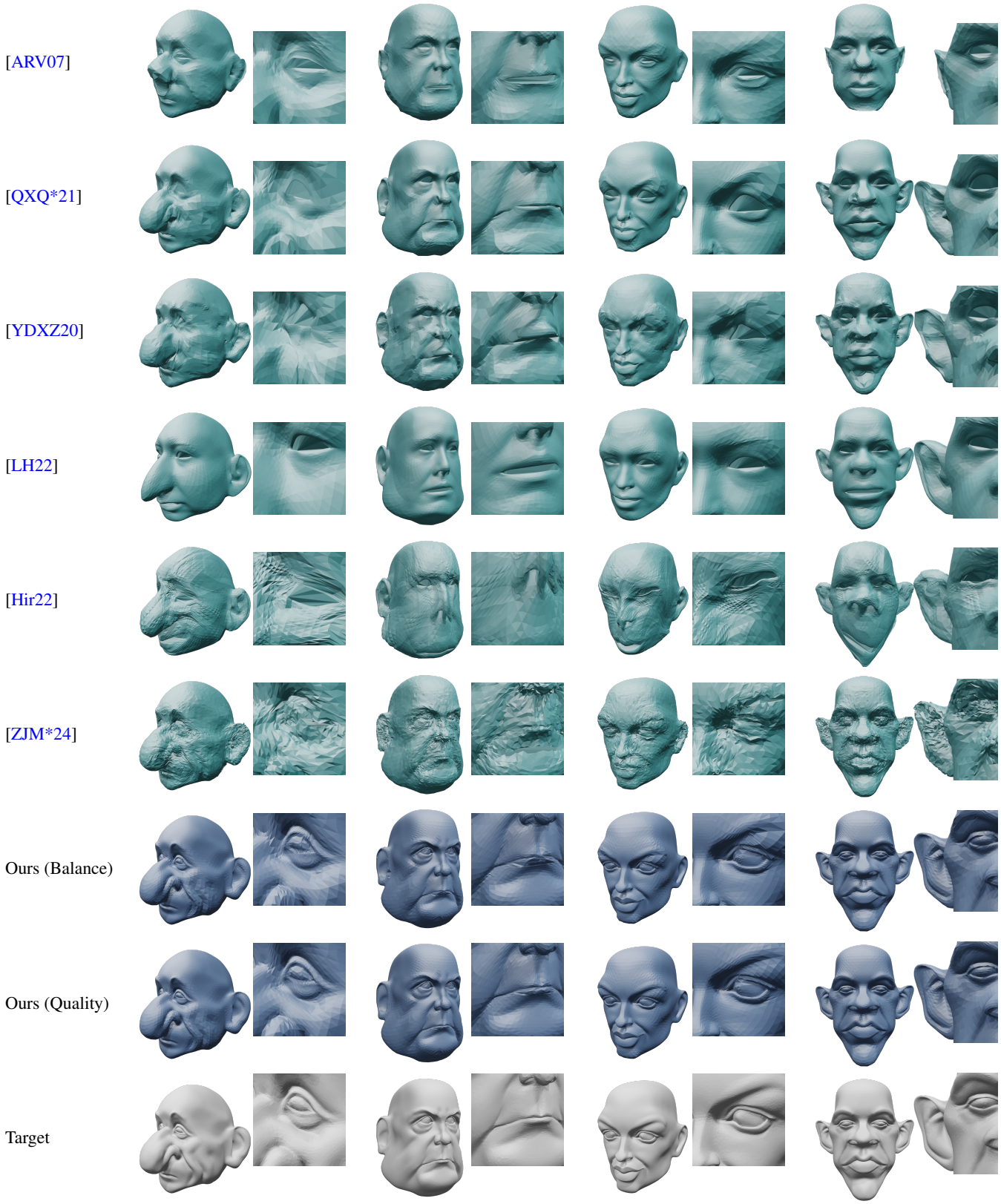

Figure 11: *Qualitative comparison with state-of-the-art non-rigid registration methods using Face template mesh as the source mesh.*

© 2025 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

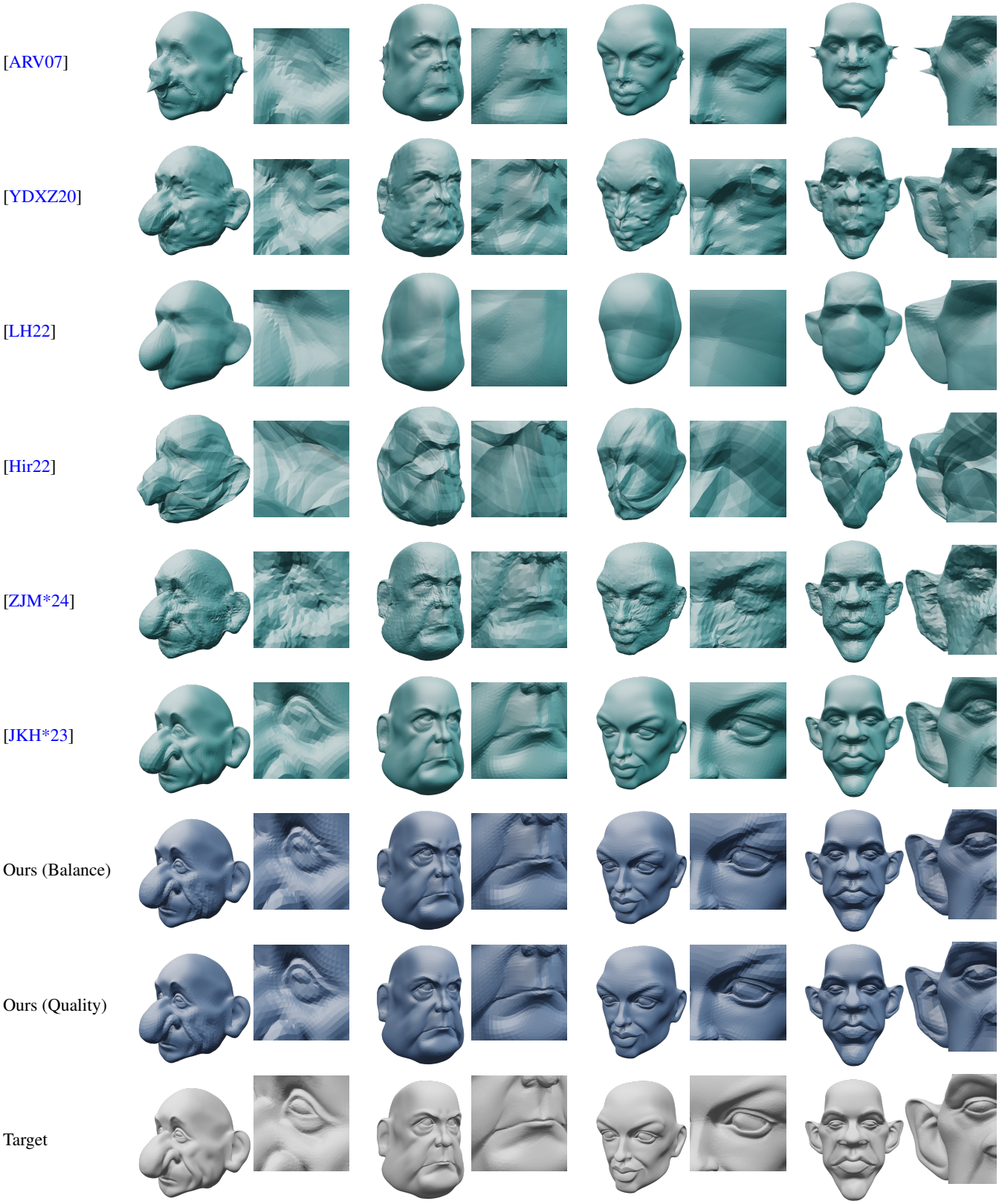

Figure 12: *Qualitative comparison with state-of-the-art non-rigid registration methods using Sphere template mesh as the source mesh.*

© 2025 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

[ARV07] AMBERG, BRIAN, ROMDHANI, SAMI, and VETTER, THOMAS. "Optimal step nonrigid ICP algorithms for surface registration". *Proc. CVPR*. IEEE. 2007 1–4, 6, 7, 13, 14.

[ASK*05] ANGUELOV, DRAGOMIR, SRINIVASAN, PRAVEEN, KOLLER, DAPHNE, et al. "Scape: shape completion and animation of people". *Proc. SIGGRAPH*. 2005 1.

[BRLB14] BOGO, FEDERICA, ROMERO, JAVIER, LOPER, MATTHEW, and BLACK, MICHAEL J. "FAUST: Dataset and evaluation for 3D mesh registration". *Proc. CVPR*. Piscataway, NJ, USA: IEEE, June 2014 1.

[CDHR08] CHEN, YANQING, DAVIS, TIMOTHY A, HAGER, WILLIAM W, and RAJAMANICKAM, SIVASANKARAN. "Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate". *ACM Transactions on Mathematical Software (TOMS)* 35.3 (2008), 1–14 1, 4.

[CGO76] CONCUS, PAUL, GOLUB, GENE H, and O'LEARY, DIANNE P. "A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations". *Sparse matrix computations*. Elsevier, 1976, 309–332 1–4.

[Dav04] DAVIS, TIMOTHY A. "Algorithm 832: UMFPACK V4. 3—an unsymmetric-pattern multifrontal method". *ACM Transactions on Mathematical Software (TOMS)* 30.2 (2004), 196–199 2, 4.

[DYDZ22] DENG, BAILIN, YAO, YUXIN, DYKE, ROBERTO M, and ZHANG, JUYONG. "A Survey of Non-Rigid 3D Registration". *Computer Graphics Forum*. Vol. 41. 2. Wiley Online Library. 2022, 559–589 2.

[EHA*19] EZUZ, DANIELLE, HEEREN, BEHREND, AZENCOT, OMRI, et al. "Elastic correspondence between triangle meshes". *Computer Graphics Forum*. Vol. 38. 2. Wiley Online Library. 2019, 121–134 2.

[ESB19] EZUZ, DANIELLE, SOLOMON, JUSTIN, and BEN-CHEN, MIRELA. "Reversible harmonic maps between discrete surfaces". *ACM Transactions on Graphics (ToG)* 38.2 (2019), 1–12 2.

[GSVS18] GOLYANIK, VLADISLAV, SHIMADA, SOSHI, VARANASI, KIRAN, and STRICKER, DIDIER. "Hdm-net: Monocular non-rigid 3d reconstruction with learned deformation model". *Proc. EuroVR*. Springer. 2018 1.

[GV13] GOLUB, GENE H and VAN LOAN, CHARLES F. *Matrix computations*. JHU press, 2013, 497 3, 5.

[HAWG08] HUANG, QI-XING, ADAMS, BART, WICKE, MARTIN, and GUIBAS, LEONIDAS J. "Non-rigid registration under isometric deformations". *Computer Graphics Forum*. Vol. 27. 5. Wiley Online Library. 2008, 1449–1457 10.

[Hir22] HIROSE, OSAMU. "Geodesic-based bayesian coherent point drift". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.5 (2022), 5816–5832 1, 7, 13, 14.

[HTB03] HAEHNEL, DIRK, THRUN, SEBASTIAN, and BURGARD, WOLFRAM. "An extension of the ICP algorithm for modeling nonrigid objects with mobile robots". *Proc. IJCAI*. Vol. 3. 2003 1.

[IZN*16] INNMANN, MATTHIAS, ZOLLHÖFER, MICHAEL, NIESSNER, MATTHIAS, et al. "Volumedeform: Real-time volumetric non-rigid reconstruction". *Proc. ECCV*. Springer. 2016 1.

[JKH*23] JUNG, YUCHEOL, KIM, HYOMIN, HWANG, GYEONGHA, et al. "Mesh Density Adaptation for Template-based Shape Reconstruction". *Proc. SIGGRAPH*. 2023 2, 3, 5–9, 14.

[KB15] KINGMA, DIEDERIK P and BA, JIMMY. "Adam: A Method for Stochastic Optimization". *Proc. ICLR*. 2015 2, 8.

[LBZ*20] LI, RUILONG, BLADIN, KARL, ZHAO, YAJIE, et al. "Learning formation of physically-based face attributes". *Proc. CVPR*. 2020 3.

[LH22] LI, YANG and HARADA, TATSUYA. "Non-rigid point cloud registration with neural deformation pyramid". *Advances in Neural Information Processing Systems* 35 (2022), 27757–27768 1, 2, 7, 9, 13, 14.

[LMR*15] LOPER, MATTHEW, MAHMOOD, NAUREEN, ROMERO, JAVIER, et al. "SMPL: A Skinned Multi-Person Linear Model". *ACM Trans. Graph.* 34.6 (2015). ISSN: 0730-0301 1, 10.

[LN89] LIU, DONG C and NOCEDAL, JORGE. "On the limited memory BFGS method for large scale optimization". *Mathematical programming* 45.1 (1989), 503–528 2.

[LSC*04] LIPMAN, YARON, SORKINE, OLGA, COHEN-OR, DANIEL, et al. "Differential coordinates for interactive mesh editing". *Proceedings Shape Modeling Applications, 2004*. IEEE. 2004, 181–190 10.

[LSP08] LI, HAO, SUMNER, ROBERT W, and PAULY, MARK. "Global correspondence optimization for non-rigid registration of depth scans". 27.5 (2008), 1421–1430 1, 2, 4.

[MJBC13] MARTIN, TOBIAS, JOSHI, PUSHKAR, BERGOU, MIKLÓS, and CARR, NATHAN. "Efficient Non-linear Optimization via Multi-scale Gradient Filtering". *Computer Graphics Forum*. Vol. 32. 6. Wiley Online Library. 2013, 89–100 2, 3, 5.

[MMRC20] MARIN, RICCARDO, MELZI, SIMONE, RODOLA, EMANUELE, and CASTELLANI, UMBERTO. "Farm: Functional automatic registration method for 3d human bodies". *Computer Graphics Forum*. Vol. 39. 1. Wiley Online Library. 2020, 160–173 10.

[MS10] MYRONENKO, ANDRIY and SONG, XUBO. "Point set registration: Coherent point drift". *IEEE transactions on pattern analysis and machine intelligence* 32.12 (2010), 2262–2275 7.

[MSC06] MYRONENKO, ANDRIY, SONG, XUBO, and CARREIRA-PERPINAN, MIGUEL. "Non-rigid point set registration: Coherent point drift". *Advances in neural information processing systems* 19 (2006) 1.

[NJJ21] NICOLET, BAPTISTE, JACOBSON, ALEC, and JAKOB, WENZEL. "Large steps in inverse rendering of geometry". *ACM Trans. Graph.* 40.6 (2021), 1–13 2, 3, 5.

[NVI24] NVIDIA CORPORATION. *cuDSS*. https://developer.nvidia.com/cudss. Software package. NVIDIA Corporation, 2024 6.

[NW99] NOCEDAL, JORGE and WRIGHT, STEPHEN J. *Numerical optimization*. Springer, 1999, 120–183 2–5.

[PGM*19] PASZKE, ADAM, GROSS, SAM, MASSA, FRANCISCO, et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". (2019), 8024–8035 6.

[PP93] PINKALL, ULRICH and POLTHIER, KONRAD. "Computing discrete minimal surfaces and their conjugates". *Experimental mathematics* 2.1 (1993), 15–36 3, 5, 9.

[PZP*24] PAN, DONGWEI, ZHUO, LONG, PIAO, JINGTAN, et al. "RenderMe-360: A Large Digital Asset Library and Benchmarks Towards High-fidelity Head Avatars". *Advances in Neural Information Processing Systems* 36 (2024) 1.

[QXQ*21] QIU, YUDA, XU, XIAOJIE, QIU, LINGTENG, et al. "3DCaricShop: A Dataset and A Baseline Method for Single-view 3D Caricature Face Reconstruction". *Proc. CVPR*. 2021 1, 6, 7, 9, 13.

[RBSB18] RANJAN, ANURAG, BOLKART, TIMO, SANYAL, SOUBHIK, and BLACK, MICHAEL J. "Generating 3D faces using Convolutional Mesh Autoencoders". *Proc. ECCV*. 2018 1.

[RRN*20] RAVI, NIKHILA, REIZENSTEIN, JEREMY, NOVOTNY, DAVID, et al. "Accelerating 3D Deep Learning with PyTorch3D". *arXiv:2007.08501* (2020) 7.

[RS12] ROUHANI, MOHAMMAD and SAPPA, ANGEL D. "Non-rigid shape registration: A single linear least squares framework". *Proc. ECCV*. Springer. 2012, 264–277 2, 4.

[Rus19] RUSINKIEWICZ, SZYMON. "A symmetric objective function for ICP". *ACM Trans. Graph.* 38.4 (2019), 1–7 1.

[SBI18] SLAVCHEVA, MIROSLAVA, BAUST, MAXIMILIAN, and ILIC, SLOBODAN. "Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion". *Proc. CVPR*. 2018, 2646–2655 2, 3.

[SP04] SUMNER, ROBERT W and POPOVIĆ, JOVAN. "Deformation transfer for triangle meshes". *ACM Trans. Graph.* 23.3 (2004), 399–405 1.

[SPK23] SCHMIDT, PATRICK, PIEPER, DÖRTE, and KOBBELT, LEIF. "Surface Maps via Adaptive Triangulations". *Computer Graphics Forum*. Vol. 42. 2. Wiley Online Library. 2023, 103–117 9.

© 2025 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

[Str12] STRANG, GILBERT. *Linear Algebra and Its Applications 4th ed.* 2012, 181–189 3.

[SXZ*20] SU, ZHUO, XU, LAN, ZHENG, ZERONG, et al. "Robustfusion: Human volumetric capture with data-driven visual cues using a rgbd camera". *Proc. ECCV*. Springer. 2020 4.

[Ume91] UMEYAMA, SHINJI. "Least-squares estimation of transformation parameters between two point patterns". *IEEE Transactions on Pattern Analysis & Machine Intelligence* 13.04 (1991), 376–380 10.

[Wat15] WATHEN, ANDREW J. "Preconditioning". *Acta Numerica* 24 (2015), 329–376 2.

[WJH*07] WAND, MICHAEL, JENKE, PHILIPP, HUANG, QIXING, et al. "Reconstruction of deforming geometry from time-varying point clouds". *Proc. SGP*. 2007, 49–58 2.

[wuh21] WUHAOZHE. *GPU Accelerated Non-rigid ICP for surface registration*. Accessed: 2024. 2021. URL: https://github.com/wuhaozhe/pytorch-nicp 7.

[WXZ*20] WANG, KANGKAN, XIE, JIN, ZHANG, GUOFENG, et al. "Sequential 3D human pose and shape estimation from point clouds". *Proc. CVPR*. 2020 4.

[WZH*23] WU, MENGHUA, ZHU, HAO, HUANG, LINJIA, et al. "High-fidelity 3d face generation from natural language descriptions". *Proc. CVPR*. 2023 1, 7.

[YDXZ20] YAO, YUXIN, DENG, BAILIN, XU, WEIWEI, and ZHANG, JUYONG. "Quasi-Newton Solver for Robust Non-Rigid Registration". *Proc. CVPR*. June 2020 1, 2, 7, 11, 13, 14.

[YDXZ23] YAO, YUXIN, DENG, BAILIN, XU, WEIWEI, and ZHANG, JUYONG. "Fast and robust non-rigid registration using accelerated majorization-minimization". *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023) 1.

[You14] YOUNG, DAVID M. *Iterative solution of large linear systems*. Elsevier, 2014 3.

[YZG*21] YU, TAO, ZHENG, ZERONG, GUO, KAIWEN, et al. "Function4D: Real-time Human Volumetric Capture from Very Sparse Consumer RGBD Sensors". *Proc. CVPR*. June 2021 10.

[YZW*20] YANG, HAOTIAN, ZHU, HAO, WANG, YANRU, et al. "FaceScape: a Large-scale High Quality 3D Face Dataset and Detailed Riggable 3D Face Prediction". *Proc. CVPR*. 2020 1, 7.

[ZJM*24] ZHAO, MINGYANG, JIANG, JINGEN, MA, LEI, et al. "Correspondence-Free Non-Rigid Point Set Registration Using Unsupervised Clustering Analysis". *Proc. CVPR*. 2024 7, 13, 14.

© 2025 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.