Eurographics Symposium on Geometry Processing (2005), pp. 1–11
M. Desbrun, H. Pottmann (Editors)

Global Registration of Multiple 3D Point Sets via Optimization-on-a-Manifold

Shankar Krishnan, Pei Yean Lee, John B Moore, Suresh Venkatasubramanian

Abstract

We propose a novel algorithm to register multiple 3D point sets within a common reference frame using an manifold optimization approach. The point sets are obtained with multiple laser scanners or a mobile scanner. Unlike most prior algorithms, our approach performs an explicit optimization on the manifold of rotations, allowing us to formulate the registration problem as an unconstrained *minimization on a* constrained manifold*. This approach exploits the Lie group structure of SO_3 and the simple representation of its associated Lie algebra $\mathfrak{so}_3$ in terms of $\mathbb{R}^3$.*

Our contributions are threefold. We present a new analytic method based on singular value decompositions that yields a closed-form solution for simultaneous multiview registration in the noise-free scenario. Secondly, we use this method to derive a good initial estimate of a solution in the noise-free case. This initialization step may be of use in any general iterative scheme. Finally, we present an iterative scheme based on Newton's method on SO_3 that has locally quadratic convergence. We demonstrate the efficacy on our scheme on scan data taken both from the Digital Michelangelo project and from scans extracted from models, and compare it to some of the other well known schemes for multiview registration. In all cases, our algorithm converges much faster than the other approaches, (in some cases orders of magnitude faster), and generates consistently higher quality registrations.

1. Introduction

Constructing a 3D computer model of a real object from 3D surface measurement data has various applications in computer graphics, virtual reality, computer vision and reverse engineering. To construct such a model, a single view of the object is often insufficient due to self occlusion, the presence of shadows and limitations of the field of view of the 3D scanner. Multiple partial views of the object from different viewpoints are therefore needed to describe the entire object. Typically the views are obtained from multiple scanners or from a single scanner stationed at different locations and orientation, or even a fixed scanner taking time-sampled images of an object on a moving turntable. The images are often simplified as a set of features such as points and the relative position and orientation (pose) between views are only known imprecisely (if at all). Thus, these partially overlapping views need to be registered within a common reference frame to determine the unknown relative pose.

Two-view (pairwise) registration is a well studied problem in the literature. It is known that a closed-form solution can be obtained in this case; this was shown by Faugeras and Herbert [FH86], Horn [Hor87], and Arun *et al.* [AHB87]. An overview of these techniques can be found in [Kan94] and a comparison of these methods has been presented in [LEF95].

Multiview registration is a more difficult problem. There are two strategies towards solving the problem, local (sequential) registration and global (simultaneous) registration. The sequential registration approach (of which ICP [BM92] is the most well-known) involves the alignment of two overlapping views at a time followed by an integration step to ensure all views are combined. This widely used approach does not give an optimal solution because errors can accumulate and propagate, as other researchers have pointed out. On the other hand, global registration attempts to aligns all scans at the same time by distributing the registration error evenly over all overlapping views.

The particular problem of multiview registration is that the function to be minimized is a nonconvex function of a set of rotations (translations can usually be eliminated, as we shall see). Any algorithm that minimizes this function must also maintain the constraint that the rotations remain so during the course of an iterative procedure[†]. Thus, standard optimization approaches either use Lagrange constraints or have to perform projection steps after each iteration to ensure that this (nonlinear) constraint is maintained.

A different approach that has been considered is to perform the minimization *directly on the constraint manifold*. $\mathbb{R}^n$ is a manifold, albeit of very special type, and by translating the usual notions of derivatives and tangents in their differential-geometric generalizations, it is conceivable that standard numerical methods like Newton/Gauss iterations and conjugate gradients can be translated into their manifold-based analogues. This area has received considerable attention over the past few decades. Much work has gone into both theoretical and practical approaches to manifold-based optimization, and although a detailed review of the literature is beyond the scope of this paper, a good review can be found in the work by Edelman, Arias and Smith [EAS99] on implementing Newton's method and conjugate gradients on the Grassman and Stiefel manifolds.

In graphics, vision, and robotics, the "natural" constraint manifolds arise from transformations groups like SO_3, SE_3 and the like. The group structure allows us to view these manifolds as Lie groups, with associated Lie algebras. For SO_3 in particular, many of the relevant formulae (the exponential map, the logarithmic map, geodesic curves) are easy to write down, making this approach very tractable both mathematically and computationally. There are now several examples of the use of Lie group methods in areas like pose estimation [LM04, Gov04], path planning [Agr05], and animation [Ale02].

1.1. Our Work

In this paper, we consider the simultaneous registration of multiview 3D point sets with known correspondences between overlapping scans.

We address the global registration task as an unconstrained optimization problem on a constraint manifold. Our novel algorithm involves iterative cost function reduction on the smooth manifold formed by the N-fold product of special orthogonal groups. The optimization is based on locally quadratically convergent Newton-type iterations on this constraint manifold. The proposed algorithm is fast, converges at a local quadratic rate, computation per iteration is low since the iteration cost is independent of the number of data points in each view.

In addition, we present a new closed form solution based on singular value decomposition for simultaneous registration of multiple point sets. In the noise free case, it gives correct registrations in a single step. In the presence of noise an additional projection step to the constraint manifold is required. This analytical solution is a useful initial estimate for any iterative algorithm.

Paper Outline We start with a review of prior art in the area in Section 2. After a high level overview of our method (Section 3, we formulate and simplify the problem in Sections 5 and 6, following which we describe our analytic noise-free solution and our noisy initialization steps (Section 7. Following some mathematical background (Section 8, we present our iterative scheme in Sections 9 and 10. Experimental evaluation follows in Section 11. In Section II, global multiview registration of multiple 3D point sets is formulated as an unconstrained optimization-on-a- constraint manifold. In Section III, the derivation of the Hessian and the gradient are presented. Section IV illustrates the proposed algorithm in details. In Section V, we propose a novel analytic solution based on singular value decomposition that gives an exact solution in the noise free case in a single step and can be used as a good initial estimate for any iterative algorithm. Outline of algorithm can be found in Section VI and its convergence properties are analyzed in Section VII. This is followed by simulation results and conclusion. In the Appendix, we provide rigorous mathematical proof of the local quadratic convergence of the algorithm.

2. Related Work

The first work on pairwise scan alignment was done by Faugeras and Herbert [FH86], Horn [Hor87], and Arun*et al.* [AHB87]. In all cases, the authors obtained simple closed form expression for the single transformation minimizing the least squares error between the registered scans. Such pairwise schemes were used as modules in general multiview approaches like ICP [BM92] and the work of Chen and Medioni [CM92]. Simultaneous multiview registration schemes were considered by numerous researchers [MY95],[CM92],[BSGL96], [EFF98],[Pul99],[SLW04],[SBB03]; among the more recent are papers by Benjemaa and Schmitt [BS98] and Williams and Bennamoun [WB01], the former group formulating the optimization in quaternion

[†] Other approaches have been proposed; Pottmann*et al.* [PHYH04] suggest using the underlying affine space, applying the rigidity constraints only towards the end.

space, and the latter deriving a similar approach using matrix representations. A comparative study of simultaneous multiview registration schemes was performed by Cunnington and Stoddart [CS99]; however this comparison predates the work of Williams and Bennamoun.

Registration of corresponding points is not the only approach to solving multiview registration in general. ICP itself uses other heuristics to align surfaces, and in many cases matching a point to a surface can provide a better fit than simple point-point matching [RL01] Due to space limitations, we will not discuss these approaches further.

The most directly relevant prior art is a paper by Adler *et al.* [ADM*02] that considers the problem of spine realignment. There, the problem is to determine correct poses for individual vertebrae on the spinal cord such that misalignment between adjacent vertebrae is minimized and a balance criterion (expressed as an affine condition over the poses) is maintained. They demonstrate that a good solution to this problem closely resembles a healthy spinal alignment. Their approach, like ours, is to view the problem as a minimization over a product manifold of SO_3, and use a Newton-type method to solve it. The specifics of their approach are different in that they derive an iterative scheme from first principles by using the covariant derivative ∇_X on the manifold; our approach uses the Lie-algebraic representation of the tangent space to yield an more direct approach.

3. Overview and Intuition

To help explain our algorithms, we present a brief overview of how to perform Newton-type methods on manifolds. This is intended to capture the intuition behind our methods and is not intended to be mathematically rigorous. A reader familiar with Lie groups and basic differential geometry may go directly to the algorithm description.

A traditional unconstrained or constrained optimization methods performs searches in $\mathbb{R}^N$. Directions of motion are computed using Newton's method (or other approaches) and a small step is made in this direction. The standard iterative step is of the form $x_{k+1} = x_k + a\omega_k$, where x_k is the k^{th} iterate, a is a scalar, and ω_k is a *descent direction*. The descent directions lie in the tangent space of $\mathbb{R}^N$, which is $\mathbb{R}^N$ itself, a crucial fact that allows us to combine the terms x_k and ω_k.

When we move to a general manifold, almost every aspect of the above iteration needs to be reinterpreted. Firstly, the descent direction ω_k lies in the tangent space at x_k, which is in general different to the tangent space at any other point, and is different in general from the manifold itself. Thus, some mapping is needed to *pull back* the tangent to the manifold. Secondly, the operator $+$ is specific to $\mathbb{R}^N$ as a group operator that takes two elements of a group and maps to a third element.

For a Lie group, the tangent space at a point can be expressed in terms of the associated Lie algebra. For SO_3, the associated Lie algebra $\mathfrak{so}_3$ is the space of three dimensional skew-symmetric matrices. Thus the descent direction can be represented by a skew-symmetric matrix. The pull back operator is called the *exponential map*. For matrices, it is in fact the function e^A (see Section 4 for the definition). The operator $+$ is replaced by the group operator $\circ$ of the Lie group (which for SO_3 is matrix multiplication). What we then obtain is an iteration of the form $R_{k+1} = R_k \circ e^{aA}$, where once again, a is a scalar. We will additionally exploit the isomorphism of $\mathfrak{so}_3$ with $\mathbb{R}^3$, allowing us to parametrize the matrix A by coordinates in $\mathbb{R}^3$.

4. Preliminaries

We introduce some common matrix operators that we will use in subsequent sections. If M is an $n \times k$ matrix, then $\text{vec}(M)$ is a $nk \times 1$ vector formed by writing down the columns of M one at a time. The *Kronecker product* or tensor product $A \otimes B$ of two matrices A and B is the matrix formed by replacing each element a_{ij} of A by the matrix $a_{ij}B$. This is different from the direct sum $\oplus$ of matrices, which is equal to a block diagonal matrix with the individual matrices as the diagonal blocks. Let $\text{tr}(A) = \sum_i a_{ii}$ denote the trace of a square matrix A. The following identities are well-known: $\text{tr}(AB) = \text{tr}(BA)$ if A and B are both square, $(X \otimes Y)^\top = X^\top \otimes Y^\top$, $(X \otimes Y)^{-1} = X^{-1} \otimes Y^{-1}$ when the inverses exist, $(X \otimes Y)(A \otimes B) = (XA \otimes YB)$, and $\text{vec}(XYZ) = (Z^\top \otimes X)\,\text{vec}(Y)$. A useful fact is that $\text{tr}(X^\top Y) = \text{tr}(XY^\top) = \text{vec}^\top(x)\,\text{vec}(Y)$, which implies that for vectors u, v, the dot product $u \cdot v = u^\top v$ can be written as $u \cdot v = \text{tr}(uv^\top)$. The *exponential* e^A of a matrix A is defined as $e^A = \sum_i \frac{A^i}{i!}$.

5. The Problem Formulation

Given possibly noisy surface measurements from multiple 3D images and point correspondences among overlapped images, the registration process is to find the rigid body transformations between each image coordinate frame in order to align sets of surface measurements into a reference frame.

5.1. 3D Object Points and Multiple Views

Consider a 3D *object* as a set of 3D points $W := \{w^k \in \mathbb{R}^3 \mid k = 1, 2, \cdots, n\}$ in a 'world' reference frame (Fig. 1(a)). Throughout the paper we indicate the k^{th} point in a set by a superscript k.

Now consider *multiple views* of the object, each view being from a different vantage point and viewing direction and each viewing being of possibly only a subset of the n 3D points. For N views, let us denote the relative rotations and translations as $(R_1, t_1), \cdots, (R_N, t_N)$, that is, relative to the 'world' reference frame, where R_i is a 3×3 rotation matrix, satisfying $R_i^\top R_i = I_3$, $\det(R_i) = +1$, and $t_i \in \mathbb{R}^3$ is a translation vector.

The i^{th} view is limited to n_i points $W_i = \{w_i^k \in \mathbb{R}^3 \mid k = 1, 2, \cdots, n_i\} \subset W$ and is denoted $V_i = \{v_i^k \in \mathbb{R}^3 \mid k = 1, 2, \cdots, n_i\}$ and consists of the images of the n_i points in W_i with relative rotation matrices and translation vectors given by (R_i, t_i). Thus in the noise free case,

$$w_i^k = R_i v_i^k + t_i, \quad k = 1, 2, \cdots, n_i. \tag{1}$$

Let $W_{ij} = W_i \cap W_j$ be the set of n_{ij} points in W_i for which there are corresponding points in W_j, for $i, j = 1, \cdots, N$. That is, $W_{ij} = W_{ji}$ consists of $n_{ij} = n_{ji}$ points $w_{ij}^k = w_{ji}^k \in \mathbb{R}^3$, $k = 1, \cdots, n_{ij}$. In view V_i the set of images of these points is denoted $V_{ij} := \{v_{ij}^k \in \mathbb{R}^3 \mid k = 1, 2, \cdots, n_{ij}\} \subset V_i$ and of course for view V_j it is denoted $V_{ji} := \{v_{ji}^k \in \mathbb{R}^3 \mid k = 1, 2, \cdots, n_{ij}\} \subset V_j$. In the noise free case, it is immediate that

$$\begin{aligned} w_{ij}^k &= R_i v_{ij}^k + t_i = R_j v_{ji}^k + t_j \\ &\forall\, i, j = 1, 2, \cdots, N,\ k = 1, 2, \cdots, n_{ij}, \end{aligned} \tag{2}$$

5.2. Registration Error Cost Function

When there is measurement noise, it makes sense to work with a cost functions that penalizes the error $(R_i v_{ij}^k + t_i) - (R_j v_{ji}^k + t_j)$ for all $i, j = 1, 2, \cdots, N$ and $k = 1, 2, \cdots, n_{ij}$. Trivially the error is zero for $i = j$. The cost index for all the registrations which first comes to mind is given by the sum of the squared Euclidean distances between the corresponding points in all overlaps,

$$\begin{aligned} g(\mathcal{R}, \mathcal{T}) &= \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=1}^{n_{ij}} \|(R_i v_{ij}^k + t_i) - (R_j v_{ji}^k + t_j)\|^2, \\ &= \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=1}^{n_{ij}} (\|R_i v_{ij}^k - R_j v_{ji}^k\|^2 \\ &\quad + 2(t_i - t_j)^\top (R_i v_{ij}^k - R_j v_{ji}^k) + \|t_i - t_j\|^2). \end{aligned} \tag{3}$$

6. A More Compact Reformulation

Let e_i denote the i^{th} column of the $N \times N$ identity matrix I_N and let $e_{ij} := e_i - e_j$. Let

$$\mathcal{R} := \begin{bmatrix} R_1 & R_2 & \cdots & R_N \end{bmatrix} \in \mathbb{R}^{3 \times 3N} \tag{4}$$

and

$$\mathcal{T} := \begin{bmatrix} t_1 & t_2 & \cdots & t_N \end{bmatrix} \in \mathbb{R}^{3 \times N} \tag{5}$$

then we have

$$R_i = \mathcal{R}(e_i^\top \otimes I_3), \quad t_i = \mathcal{T} e_i, \quad t_i - t_j = \mathcal{T} e_{ij}. \tag{6}$$

Let $a_{ij}^k := (e_i \otimes I_3) v_{ij}^k - (e_j \otimes I_3) v_{ji}^k$. Substituting the value of R_i from Eq.(6),

$$R_i v_{ij}^k - R_j v_{ji}^k = \mathcal{R} a_{ij}^k$$

and thus

$$\|R_i v_{ij}^k - R_j v_{ji}^k\|^2 = \mathcal{R} a_{ij}^k \cdot \mathcal{R} a_{ij}^k$$

Similarly substituting the value of t_i, we can rewrite the inner expression of Eq.(3) as

$$\mathcal{R} a_{ij}^k \cdot \mathcal{R} a_{ij}^k + 2\mathcal{T} e_{ij} \cdot \mathcal{R} a_{ij}^k + \mathcal{T} e_{ij} \cdot \mathcal{T} e_{ij}$$

Let

$$\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=1}^{n_{ij}} \begin{bmatrix} a_{ij}^k \\ e_{ij} \end{bmatrix} \begin{bmatrix} a_{ij}^{k\top} & e_{ij}^\top \end{bmatrix} \geq 0 \tag{7}$$

Using the fact that $u \cdot v = \operatorname{tr}(uv^\top)$, we can now rewrite Eq.(3) as

$$\begin{aligned} g(\mathcal{R}, \mathcal{T}) &= \operatorname{tr}(\mathcal{R} A \mathcal{R}^\top + 2\mathcal{R} B \mathcal{T}^\top + \mathcal{T} C \mathcal{T}^\top) \\ &= \operatorname{tr}\left(\begin{bmatrix} \mathcal{R} & \mathcal{T} \end{bmatrix} \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \begin{bmatrix} \mathcal{R}^\top \\ \mathcal{T}^\top \end{bmatrix} \right) \geq 0, \end{aligned} \tag{8}$$

or equivalently, as

$$\begin{aligned} g(\mathcal{R}, \mathcal{T}) &= \operatorname{tr}(\mathcal{R} A \mathcal{R}^\top) + 2 \operatorname{vec}^\top(\mathcal{T}) \operatorname{vec}(\mathcal{R} B) \\ &\quad + \operatorname{vec}^\top(\mathcal{T})(C \otimes I_3) \operatorname{vec}(\mathcal{T}), \end{aligned} \tag{9}$$

since $\operatorname{tr}(XY^\top) = \operatorname{vec}^\top(X) \operatorname{vec}(Y)$.

6.1. Eliminating $\mathcal{T}$

Equation (9) is a quadratic function of $\operatorname{vec}(T)$. This function is convex (and thus has a unique minimum) iff $C \otimes I_3$ is positive definite. An element c_{ii} of C is $\sum_{k \neq i} n_{ik}$ and $c_{ij} = -n_{ij}$ for $j \neq i$. Unfortunately, this implies that C is singular, since $C\mathbf{1}$ (where $\mathbf{1}$ is the all-ones vector) vanishes.

This is a consequence of the fact that we can only recover relative transformations from our input, not absolute transformations. We can fix (say) the first reference frame $(R_i, t_i) = (I_3, \mathbf{0})$, where $\mathbf{0}$ is the zero vector, and eliminate the first row and column from all the matrices. We will abuse notation by continuing to use the same variables for $\mathcal{R}, Tcal$ and other matrices.

Eliminating the first row and column from C leaves a matrix that is symmetric and *strictly diagonally dominant* i.e., each diagonal element is in absolute value strictly larger than the sum of the absolute values of off-diagonal entries in that row. It is a basic property that such matrices are positive definite, which consequently implies that $C \otimes I_3$ is positive definite, and thus $g(\mathcal{R}, \mathcal{T})$ has a unique minimum for fixed $\mathcal{R}$ and varying $\mathcal{T}$. The minimizing value of $\mathcal{T}$ is then

$$\begin{aligned}\operatorname{vec}(\mathcal{T}^*(\mathcal{R})) = -(C^{-1} \otimes I_3)\operatorname{vec}(\mathcal{R}B) = -\operatorname{vec}(\mathcal{R}BC^{-1}) \\ \mathcal{T}^*(\mathcal{R}) = -\mathcal{R}BC^{-1}. \qquad (10)\end{aligned}$$

Substituting $\mathcal{T}^*(\mathcal{R})$ from Eq.(10) into (8) leads to a registration error cost function depending only on rotations,

$$\begin{aligned}f(\mathcal{R}) := g(\mathcal{R}, \mathcal{T}(\mathcal{R})) &= \operatorname{tr}(\mathcal{R}\mathcal{M}\mathcal{R}^\top) \\ &= \operatorname{vec}^\top(\mathcal{R}^\top)(I_3 \otimes \mathcal{M})\operatorname{vec}(\mathcal{R}^\top)\end{aligned} \qquad (11)$$

where $\mathcal{M} := A - BC^{-1}B^\top$.

7. Initialization

Here we present a new closed form solution based on singular value decomposition that simultaneously registers all range images which is used as the initial guess for the proposed iterative algorithm of the previous section. In the noise free case, it gives optimal and thus exact rotation matrices in a single step. In the presence of noise, this step leads to an 'optimal' matrix $\mathcal{R} \in \mathbb{R}^{3\times 3N}$ but such that $R_i \notin SO_3$ for some i typically. Thus, an additional projection step to the manifold is required.

7.1. Noise Free Solution

In the noise free case, for $\mathcal{R} \in SO_3^N$, the optimal value of the cost function (11) is zero, as

$$\begin{aligned}\operatorname{vec}^\top(\mathcal{R}^\top)\operatorname{vec}(\mathcal{M}\mathcal{R}^\top) = 0 \Rightarrow \operatorname{vec}(\mathcal{M}\mathcal{R}^\top) = 0 \\ \Rightarrow \quad \mathcal{M}R^\top = 0.\end{aligned} \qquad (12)$$

Since $\mathcal{M}$ is symmetric, a singular value decomposition gives

$$\begin{aligned}\mathcal{M} =& U\Sigma U^\top = \begin{bmatrix} U_a & U_b \end{bmatrix} \begin{bmatrix} \Sigma_a & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_a^\top \\ U_b^\top \end{bmatrix} \\ &\Rightarrow \mathcal{M}U_b = 0.\end{aligned} \qquad (13)$$

To obtain $\mathcal{R}$ such that $R_1 = I_3$, let $\widehat{U} := \begin{bmatrix} I_3 & 0 \end{bmatrix} U_b$, then the closed form solution is

$$\mathcal{R} = \widehat{U}^{-\top} U_b^\top. \qquad (14)$$

7.2. Initialization in Noisy Case

In the presence of noise, the optimal cost function is no longer equal to zero. In this case, U_b is chosen to be the set of right singular vectors associated with 3 least singular values of $\mathcal{M}$, which may not be zero. These singular vectors might not be on SO_3^N. Thus, an additional projection step is required. Denoting $G_i := \widehat{U}^{-\top} U_b(e_i \otimes I_3)$, we have

$$R_i^{opt} = arg \min_{R_i \in SO_3} \|R_i - G_i\| = arg \max_{R_i \in SO_3} \operatorname{tr}(R_i^\top G_i). \qquad (15)$$

By applying a singular value decomposition on G_i, we obtain

$$G_i = W\Lambda Z^\top, \quad R_i^{opt} = W \begin{bmatrix} I_2 & 0 \\ 0 & \det(WZ^\top) \end{bmatrix} Z^\top, \qquad (16)$$

where $\det(R_i^{opt}) = +1$.

8. The Product Manifold of SO_3

Here we review the geometry of the special orthogonal group and its product manifold. Let SO_3 denote the group of 3×3 orthogonal matrices with determinant $+1$, then $R_i \in SO_3$ for $i = 1, \cdots, N$.

Definition 8.1 (Lie Group) A *Lie group* is a differential manifold $\mathcal{G}$ equipped with a product $\cdot : \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ that satisfies the group axioms of associativity, identity, and has an inverse p^{-1}. Further, the maps $(p, r) \mapsto p \cdot r, p \mapsto p^{-1}$ are smooth functions.

Definition 8.2 (Lie Algebra) A *Lie algebra* $\mathfrak{g}$ is a linear space V equipped with a *Lie bracket*, a bilinear skew-symmetric mapping

$$[\cdot, \cdot] : V \times V \to V$$

that obeys the properties

Skew Symmetry. $[F, G] = -[G, F]$
Scalar Multipliers, $[\alpha F, G] = \alpha [F, G] \; \forall \alpha \in \mathbb{R}$
Bilinearity. $[F + G, H] = [F, H] + [G, H]$
Jacobi's identity. $[F, [G, H]] + [G, [H, F]] + [H, [F, G]] = 0$

A*vector field* on a manifold $\mathcal{G}$ is a function that maps a point p on $\mathcal{G}$ to an element of the tangent space $T_p(\mathcal{G})$. A *left-invariant* vector field X is a vector field such that for all $g \in \mathcal{G}$, $XL_g = L_g X$, where X is viewed as a differential operator and $L_g[f](x) = f(g \cdot x)$. The vector space of all left-invariant vector fields over a Lie group $\mathcal{G}$ is a Lie algebra with the operator $[X, Y] = XY - YX$, and is said to be the *associated Lie algebra* $\mathfrak{g}$.

SO_3 is a Lie group with the group operator being matrix multiplication. Its associated Lie algebra $\mathfrak{so}_3$ is the set of 3×3 skew symmetric matrices of the form,

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \quad (17)$$

There is a well known isomorphism from the Lie algebra $(\mathbb{R}^3, \times)$ to the Lie algebra $(\mathfrak{so}_3, [.,.])$, where $\times$ denotes the cross product and $[.,.]$ denotes the matrix commutator. This allows one to identify $\mathfrak{so}_3$ with $\mathbb{R}^3$ using the mapping in (17), which maps a vector $\omega = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix} \in \mathbb{R}^3$ to a matrix $\Omega \in \mathfrak{so}_3$. Denoting

$$Q_x := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \; Q_y := \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \text{ and} \quad Q_z := \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (18)$$

note that

$$\Omega = \Omega(\omega) = Q_x\omega_x + Q_y\omega_y + Q_z\omega_z. \quad (19)$$

An identity that we will make use of later is:$\mathrm{vec}(\Omega^\top) = Q\omega$. In this paper we will consider the N-fold product manifold of SO_3 which is a smooth manifold of dimension $3N$, given by

$$SO_3^N = \overbrace{SO_3 \times \cdots \times SO_3}^{N \; times}. \quad (20)$$

8.1. Tangent Space of SO_3^N

First recall that the tangent space of SO_3 at R_i is given as $T_{R_i}SO_3 = \{R_i\Omega_i \mid \Omega_i \in \mathfrak{so}_3\}$ and the affine tangent space is $T_{R_i}^{aff}SO_3 = \{R_i + R_i\Omega_i \mid \Omega_i \in \mathfrak{so}_3\}$. Define

$$\widetilde{\Omega} := \Omega_1 \oplus \Omega_2 \oplus \cdots \oplus \Omega_N, \quad \Omega_i \in \mathfrak{so}_3. \quad (21)$$

Due to isomorphism, the tangent space of SO_3^N at $\mathcal{R} = \begin{bmatrix}R_1 & R_2 & \cdots & R_N\end{bmatrix} \in SO_3^N$ can be identified as, $T_{\mathcal{R}}SO_3^N = \mathcal{R}\widetilde{\Omega}$ and the affine tangent space is $T_{\mathcal{R}}^{aff}SO_3^N = \mathcal{R} + \mathcal{R}\widetilde{\Omega}$.

8.2. Local Parameterization of SO_3^N

Let $\mathcal{N}(0) \subset \mathbb{R}^3$ denotes a sufficiently small open neighbourhood of the origin in $\mathbb{R}^3$, and let $R_i \in SO_3$. Then the exponential mapping

$$\mu : \mathcal{N}(0) \subset \mathbb{R}^3 \to SO_3, \;\; \omega_i \mapsto R_i e^{\Omega_i(\omega_i)}, \quad (22)$$

is a local diffeomorphism from $\mathcal{N}(0)$ onto a neighbourhood of R_i in SO_3. Due to isomorphism, the product manifold SO_3^N at $\mathcal{R} \in SO_3^N$ can be locally parameterized by

$$\varphi : \mathcal{N}(0) \times \cdots \times \mathcal{N}(0) \subset \mathbb{R}^{3N} \to SO_3^N,$$
$$\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_N \end{bmatrix} \mapsto \mathcal{R}\left(e^{\Omega(\omega_1)} \oplus e^{\Omega(\omega_2)} \oplus \cdots \oplus e^{\Omega(\omega_N)}\right)$$
$$= \mathcal{R}e^{\widetilde{\Omega}(\omega)} \quad (23)$$

9. Constructing A Local Approximation

We are now ready to present our algorithm. Firstly, we construct a local approximation of f, using a second order Taylor expansion. Instead of differentiating f, we will use the local parametrization of SO_3 described earlier, performing the approximation on the function $f \circ \varphi$, whose domain is $\mathbb{R}^{3N}$. Intuitively, the use of the local parametrization φ ensures that we always stay on the manifold.

The second order Taylor approximation of $f \circ \varphi$ is given by the function $j_0^{(2)}(f \circ \varphi) : \mathbb{R}^{3N} \to \mathbb{R}$,

$$\omega \mapsto \left((f \circ \varphi)(t\omega) + \frac{\mathrm{d}}{\mathrm{d}t}(f \circ \varphi)(t\omega) + \frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}t^2}(f \circ \varphi)(t\omega)\right)\Bigg|_{t=0}. \quad (24)$$

As with a univariate Taylor expansion, the above expression can be written in the form $(f \circ \varphi)(t\omega) + \omega^\top\nabla + \frac{1}{2}\omega^\top H\omega$, where ∇ is the gradient and H is the Hessian of the function $f \circ \varphi$.

The first term in (9) is $(f \circ \varphi)(t\omega)|_{t=0} = \mathrm{tr}(\mathcal{R}\mathcal{M}\mathcal{R}^\top)$. The second term is

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}(f \circ \varphi)(t\omega)\Big|_{t=0} &= 2\,\mathrm{tr}(\mathcal{R}\widetilde{\Omega}\mathcal{M}\mathcal{R}^\top) \\ &= 2\omega^\top\nabla(f \circ \varphi)(0), \end{aligned} \quad (25)$$

Recall that $\mathrm{tr}(\mathcal{R}\widetilde{\Omega}\mathcal{M}\mathcal{R}^\top)$ can be written as $\mathrm{vec}^\top(\widetilde{\Omega}\mathcal{R}^\top)\,\mathrm{vec}(\mathcal{M}\mathcal{R}^\top)$.

$$\begin{aligned} \mathrm{vec}^\top(\widetilde{\Omega}\mathcal{R}^\top) &= [\mathrm{vec}(\widetilde{\Omega}\mathcal{R}^\top)]^\top \\ &= [\mathrm{vec}(I_{3N}\widetilde{\Omega}\mathcal{R}^\top)]^\top \\ &= [(R \otimes I_{3N})\,\mathrm{vec}(\widetilde{\Omega})]^\top \end{aligned} \quad (26)$$

Let $\widetilde{Q} := Q_{e_1} \oplus Q_{e_2} \oplus \cdots \oplus Q_{e_N}, \quad Q_{e_i} := \begin{bmatrix} e_i \otimes Q_x \\ e_i \otimes Q_y \\ e_i \otimes Q_z \end{bmatrix}$.

Then, using (18), we have $\mathrm{vec}(\widetilde{\Omega}) = \widetilde{Q}\omega$, and then (26) can be written as

$$\mathrm{vec}^\top(\widetilde{\Omega}\mathcal{R}^\top) = \omega^\top J$$

where $J := (\mathcal{R} \otimes I_{3N})\widetilde{Q}$. Substituting back into (25),

$$\nabla(f \circ \varphi)(0) = J^\top \operatorname{vec}(\mathcal{M}R^\top) \quad (27)$$

Finaly, the quadratic term in (9) consists of a sum of two terms. The first term is given as

$$\operatorname{tr}(\mathcal{R}\widetilde{\Omega}\mathcal{M}\widetilde{\Omega}^\top\mathcal{R}^\top) = \omega^\top \widehat{H}_{(f\circ\varphi)(0)}\omega, \quad (28)$$

and the second quadratic term is

$$\begin{aligned}\operatorname{tr}(\mathcal{R}\widetilde{\Omega}^2\mathcal{M}\mathcal{R}^\top) &= \operatorname{vec}^\top(\widetilde{\Omega}^\top)\operatorname{vec}(\mathcal{M}\mathcal{R}^\top\mathcal{R}\widetilde{\Omega}) \\ &= \omega^\top \widetilde{H}_{(f\circ\varphi)(0)}\omega\end{aligned} \quad (29)$$

By applying similar methods as above, we obtain the Hessian of $f \circ \varphi$ evaluated at zero:

$$H_{(f\circ\mu)(0)} = \widehat{H}_{(f\circ\mu)(0)} + \widetilde{H}_{(f\circ\mu)(0)}, \quad (30)$$

where

$$\widehat{H}_{(f\circ\varphi)(0)} = J^\top(I_3 \otimes \mathcal{M})J \succeq 0 \quad (31)$$

$$\widetilde{H}_{f\circ\varphi)(0)} = -\widetilde{Q}^\top(I_{3N} \otimes \mathcal{M}\mathcal{R}^\top\mathcal{R})\widetilde{Q}. \quad (32)$$

Note that H is a sum of the positive semidefinite term $\widehat{H}$ and the term $\widetilde{H}$. Since $\widetilde{H}$ is nonzero, we cannot guarantee that f has a global minimum. However, the fact that we can decompose H as a sum of a positive definite term and another term will prove to be useful in the iterative algorithm we present next.

We note that $\widetilde{H}$ vanishes when there are only two views, illustrating the known fact that the two-view registration problem can be solved optimally.

10. The Algorithm

The proposed algorithm consists of the iteration,

$$s = \pi_2 \circ \pi_1 : SO_3^N \to SO_3^N, \quad (33)$$

where π_1 maps a point $\mathcal{R}$ on the product manifold SO_3^N to an element in the affine tangent space that minimizes $j_0^{(2)}(f\circ\varphi)(0)$ and π_2 maps that element back to SO_3^N by means of the parametrization φ. The mapping π_1 is a standard iterative scheme that uses a modified Newton method to determine a descent direction and a line search to move along this direction. In what follows, we describe this approach in brief; the reader is referred to the text by Nocedal and Wright [NW99] for more details.

10.1. Optimization in Local Parameter Space

Optimization in the local parameter space consists of two steps. First we calculate a suitable descent direction, and then we search for a step length that ensures reduction in cost function. These two steps are described by the mapping

$$\pi_1 = \pi_1^b \circ \pi_1^a : SO_3^N \to \mathbb{R}^{3N\times 3N}. \quad (34)$$

In the first step, π_1^a is used to obtain a descent direction,

$$\pi_1^a : SO_3^N \to \mathbb{R}^{3N\times 3N}, \ \ \mathcal{R} \mapsto \mathcal{R} + \mathcal{R}\widetilde{\Omega}(\omega_{opt}),$$

where ω_{opt} is given by the Newton direction

$$\omega_{opt}(\varphi(\omega)) = -H^{-1}_{(f\circ\varphi)(\omega)}\nabla(f\circ\varphi)(\omega), \quad (35)$$

or a Gauss direction

$$\omega_{opt}(\varphi(\omega)) = -\widehat{H}^{-1}_{(f\circ\varphi)(\omega)}\nabla(f\circ\varphi)(\omega). \quad (36)$$

Once an optimal direction is computed, an approximate one dimensional line search is carried out in this direction, denoted by the mapping π_1^b. We proceed with a search that ensures that the cost function is reduced at every step. We use backtracking line search ([NW99]) for this purpose. Since we are using a descent direction, choosing a sufficiently small step size will ensure that the cost function goes downhill. Backtracking line search first tries a step size of 1, and if this is unacceptable, it reduces the step size according to a specific formula ([NW99]), until an acceptable step length is found. Thus,

$$\begin{aligned}\pi_1^b : \mathbb{R}^{3N\times 3N} &\to \mathbb{R}^{3N\times 3N}, \\ \mathcal{R} + \mathcal{R}\widetilde{\Omega}(\omega_{opt}) &\mapsto \mathcal{R} + \mathcal{R}\widetilde{\Omega}(\lambda_{opt}\omega_{opt})\end{aligned} \quad (37)$$

where λ_{opt} is the step length that reduces the cost function in direction ω_{opt}, and is found using the simple backtracking line search.

10.2. Projecting back via parametrization

Once the descent direction and downhill step size is obtained, we map the resulting point back to the manifold via the parametrization $\pi_2 : \mathbb{R}^{3N\times 3N} \to SO_3^N$:

$$\begin{aligned}\mathcal{R} + \widetilde{\Omega}(\lambda_{opt}\omega_{opt}) &\mapsto \mathcal{R}e^{\widetilde{\Omega}(\lambda_{opt}\omega_{opt})} \\ &= \mathcal{R}\left(e^{(\Omega_1(\lambda_{opt}\omega_1^{opt}))} \oplus \cdots \oplus e^{\Omega_N(\lambda_{opt}\omega_N^{opt})}\right)\end{aligned} \quad (38)$$

since $\omega_{opt} = \begin{bmatrix}\omega_1^{opt\top} & \ldots & \omega_N^{opt\top}\end{bmatrix}^\top$.

We summarize the algorithm in Algorithm 10.1:

Theorem 10.1 Consider the iteration $\mathcal{R}_{k+1} = s(\mathcal{R}_k)$ defined by a single step of Algorithm 10.1 and denote $\mathcal{R}_* = \varphi(0)$ as belonging to the set of local minima of $j_0^{(2)}(f\circ\varphi)(\omega)$. Further assume that $\mathcal{R}_*$ is an *isolated minimum* in that $H^{-1}_{(f\circ\varphi)(0)}$ exists. Then s converges locally quadratically to $\mathcal{R}_*$.

We omit a detailed proof. The reader may refer to Lee's thesis ([Lee05]) for more details.

Implementation Notes We use a simple eigenvalue computation to determine whether the Hessian H is positive definite. This is not the most efficient approach; other, more sophisticated numerical methods

Algorithm 10.1 Iterative Algorithm

Initialize $\mathcal{R} = \mathcal{R}_0 = [R_1 \ R_2 \ \cdots \ R_N] \in SO_3^N$ using the initialization procedure described in Section 7.2
repeat
  {/* **Step 1: Carry out optimization** */}
  Compute $\nabla(f \circ \varphi)(0)$, $H_{(f\circ\varphi)(0)}$ via (27), (30) respectively.
  if $H_{(f\circ\varphi)(0)} \succ 0$ **then**
    $\omega_{opt} = H^{-1}_{(f\circ\varphi)(0)} \nabla(f \circ \varphi)(0)$ {Newton step}
  else
    $\omega_{opt} = \widehat{H}^{-1}_{(f\circ\varphi)(0)} \nabla(f \circ \varphi)(0)$ {Gauss step}
  end if
  Compute optimum step size λ_{opt} in direction ω_{opt}.
  Set $\mathcal{R}' \leftarrow \pi_1^b(\mathcal{R}_k)$ (37)

  {/* **Step 2: Map back to manifold** */}
  $\mathcal{R}_{k+1} \leftarrow \pi_2(\mathcal{R}')$ (38)
until $\|\nabla(f \circ \varphi)(0)\| > \epsilon$

can simplify this step, and even avoid computing the Hessian directly. We defer a detailed implementation study to an extended version of this paper. To reduce computational effort per iteration of the algorithm, the sparse matrix J (27) that we use for Hessian and gradient computation can be manipulated further as follows. Recalling Ω from (17),

$$J = \left[(R_1 \otimes I_{3N})Q_{e_1}(R_2 \otimes I_{3N})Q_{e_2} \cdots (R_N \otimes I_{3N})Q_{e_N}\right] = \begin{bmatrix} \Omega(\bar{e}_1^\top R_1) \oplus \Omega(\bar{e}_1^\top R_2) \oplus \cdots \oplus \Omega(\bar{e}_1^\top R_N) \\ \Omega(\bar{e}_2^\top R_1) \oplus \Omega(\bar{e}_2^\top R_2) \oplus \cdots \oplus \Omega(\bar{e}_2^\top R_N) \\ \Omega(\bar{e}_3^\top R_1) \oplus \Omega(\bar{e}_3^\top R_2) \oplus \cdots \oplus \Omega(\bar{e}_3^\top R_N) \end{bmatrix}. \tag{39}$$

In general, determining a suitable modification to a non-positive-definite Hessian to make it positive definite is the core of the modified Newton method that we employ. It is interesting that for this problem, the Hessian decomposes cleanly into positive definite and non-positive-definite portions, and this might be a sign of further structure in the problem that a better iterative scheme might exploit.

11. Experimental Evaluation

We now present an experimental study of our algorithm, focusing primarily on the quality of the registrations it produces, and the convergence rate of the method.

Methods We will compare our algorithm (which we will refer to as **MBR** (Manifold-based registration)) to the schemes proposed by Benjemaa and Schmitt [BS98] (**QUAT**) and Williams and Bennamoun [WB01] (**MAT**). **MBR** and **MAT** are matrix based and are written in `MATLAB`. **MAT**, which uses quaternions in its formulation, is written in C. Our method of comparison will be both visual quality as well as iteration counts and error convergence rates (we will not use clock time).

Data Our first data set consists of actual 3D models from the Stanford 3D Scanning Repository. For each of three models, we generated a collection of views as follows: we first generate a unit vector (representing a view) and extracted the points on all front-facing triangles with respect to this view. Next, each view is randomly rotated and translated into a local coordinate system. Finally, each point in each view is randomly perturbed using a Gaussian noise model. This yields a collection of views that possess a global noisy registration. With this data, we have ground truth (exact correspondences) since we have the original model. Table 1 summarizes the statistics of this data.

Our second data set consists of 3D range scan data from the Digital Michelangelo Project [LPC*00]. The individual scans come with an original alignment (stored in `.xf` files). We perform ICP on pairs of scans, using the routines built into `scanalyze`, and retain all pairs of scans that have at least three points in common as determined by ICP. In each instance, we run ICP five times and take the best alignment thus generated (each instance of ICP runs for ten iterations). The model of correspondence used is point-point.

11.1. 3D Models

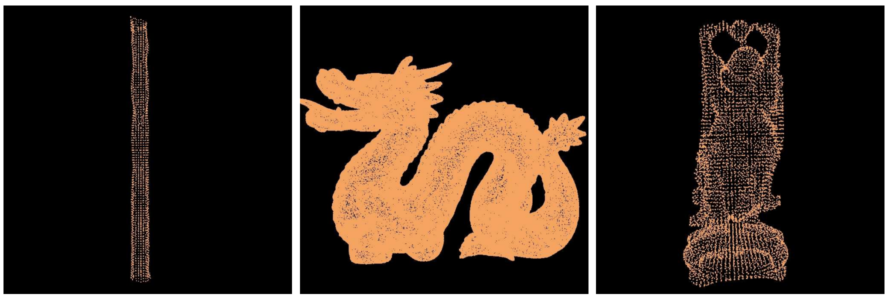
We first ran the three algorithms on the view pairs obtained from the three 3D models. In Figure 1 we show the output registrations obtained by **MBR**. For these examples, the other two schemes produced similar registrations, although with higher error. In Table 2, we compare the performance of the three schemes on the models, in terms of both the number of iterations till convergence, and the final error. What is striking about the numbers is that although in the end the other approaches mostly (except for DRILL) yield comparable error, their iteration counts are orders of magnitude higher than that of our scheme. This is a clear demonstration of locally quadratic convergence properties of our scheme.

Factors that influence iteration counts Since our method converges significantly faster than the other algorithms, we attempted to investigate other factors that might improve their performance. Some of the parameters that influence iteration counts are the density of the correspondence graph i.e how many view pairs are provided, and the strength of match for each pair (average number of points in each view pair).

In all cases, the behaviour of our method was un-

Model	Number of vertices	Number of scans	Total size of all scans	Number of view pairs generated
DRILL	1961	20	23298	77
DRAGON	100250	20	1142487	98
BUDDHA	32328	50	252580	526

Table 1: *Statistics for the 3D models*



(a) Drill (b) Dragon (c) Buddha

Figure 1: *Registrations produced by* ***MBR***

	MBR		**MAT**		**QUAT**	
	Iter.	Error	Iter.	Error	Iter.	Error
Drill	**2**	3.5e-7	47	3.5e-7	48	7e-7
Dragon	**4**	5e-3	200	1e-2	1000	1e-2
Buddha	**2**	e-3	200	2e-3	718	3e-3

Table 2: *Performance of the three methods*

affected. However, we noticed a fairly weak correlation between the density of the correspondence graph and the number of iterations needed; as the graph got denser, implying a more constrained system, the number of iterations needed to converge reduced. For example, on the drill, the iterations for **MAT** and **QUAT** went from 200 (for a sparse graph) to 47 (for a dense graph).

11.2. Range Scan Data

Having evaluated the performance of our scheme in relation to prior art in a controlled setting where ground truth (exact correspondences) are known, we now present the results of running the schemes on range scan data. We focus on the model of David, specifically the views corresponding to the head and bust region. After implementing the view generation procedure described earlier, we obtain a 10-scan instance for the bust and a 38-scan instance for the head. We also use a 21-scan instance that has bad starting alignment.

Figure 2 presents the registrations obtained by **MBR**, **MAT**, and **QUAT**. In all cases, the registration produced by our algorithm is quite plausible. The other methods do not fare so well; a typical problem is that the two halves of David's face do not register properly, creating the false effect of two heads. Table 3 summarizes the performance: we do not report errors since they are dependent on the initial correspondences and are typically quite high.

12. Acknowledgements

We acknowledge Prof. Marc Levoy and the Digital Michelangelo Project at Stanford University for providing access to the raw scan data used in this paper.

References

[ADM*02] Adler R., Dedeiu J.-P., Margulies J. Y., Martens M., Shub M.: Newton's method

	MBR Iter.	**MAT** Iter.	**QUAT** Iter.
Head	48	247	1000
Bust	12	1000	1000
Bust - Bad Alignment	81	1000	1000

Table 3: *The David model: performance of the three methods*

on riemannian manifolds and a geometric model for the human spine. *IMA Journal of Numerical Analysis 22* (2002), 359–390.

[Agr05] Agrawal M. A.: Lie algebraic approach for consistent pose registration for general euclidean motion. In *Proc. IEEE ICRA* (2005).

[AHB87] Arun K., Hwang T. S., Bolstein S.: Least squares fitting of two 3D point sets. *IEEE PAMI.* (1987), 698–700.

[Ale02] Alexa M.: Linear combination of transformations. *ACM Trans. Graph. 21*, 3 (2002), 380–387.

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

[BS98] Benjemaa R., Schmitt F.: A solution for the registration of multiple 3d point sets using unit quaternions. In *Proc. ECCV* (1998), pp. 34–50.

[BSGL96] Bergevin R., Soucy M., Gagnon H., Laurendeau D.: Towards a general multi-view registration technique. *IEEE Trans. Pattern Anal. Mach. Intell. 18*, 5 (1996), 540–547.

[CM92] Chen Y., Medioni G.: Object modelling by registration of multiple range images. *Image Vision Comput. 10*, 3 (1992).

[CS99] Cunnington S., Stoddart A. J.: N-view point set registration: A comparison. In *British Machine Vision Conference* (1999).

[EAS99] Edelman A., Arias T. A., Smith S. T.: The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl. 20*, 2 (1999), 303–353.

[EFF98] Eggert D. W., Fitzgibbon A. W., Fisher R. B.: Simultaneous registration of multiple range views for use in reverse engineering of cad models. *Comput. Vis. Image Underst. 69*, 3 (1998), 253–272.

[FH86] Faugeras O. D., Herbert M.: The representation, recognition, and location of 3d objects. *Intnl. J. Robotics Research 5*, 3 (1986), 27–52.

[Gov04] Govindu V. M.: Lie-algebraic averaging for globally consistent motion estimation. In *Proc. IEEE CVPR* (2004).

[Hor87] Horn K. P.: Closed form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A* (1987).

[Kan94] Kanatani K.: Analysis of 3D rotation fitting. *IEEE PAMI 16*, 5 (1994), 543–549.

[Lee05] Lee P. Y.: *Geometric Optimization for Computer Vision.* PhD thesis, Australian National University, 2005.

[LEF95] Lorusso A., Eggert D. W., Fisher R. B.: A comparison of four algorithms for estimating 3d rigid transformations. In *British Machine Vision Conference* (1995).

[LM04] Lee P. Y., Moore J. B.: Pose estimation via a Gauss-Newton-on-manifold approach. In *16th International Symposium on Mathematical Theory of Network and System (MTNS)* (2004).

[LPC*00] Levoy M., Pulli K., Curless B., Rusinkiewicz S., Koller D., Pereira L., Ginzton M., Anderson S., Davis J., Ginsberg J., Shade J., Fulk D.: The digital Michelangelo project: 3d scanning of large statues. In *Proc. SIGGRAPH* (2000), pp. 131–144.

[MY95] Masuda T., Yokoya N.: A robust method for registration and segmentation of multiple range images. *Computer Vision and Image Understanding 61*, 3 (1995), 295–307.

[NW99] Nocedal J., Wright S. J.: *Numerical Optimization.* Springer, August 1999.

[PHYH04] Pottmann H., Huang Q.-X., Yang Y.-L., Hu S.-M.: *Geometry and convergence analysis of algorithms for registration of 3D shapes.* Tech. Rep. 117, TU Wien, 2004.

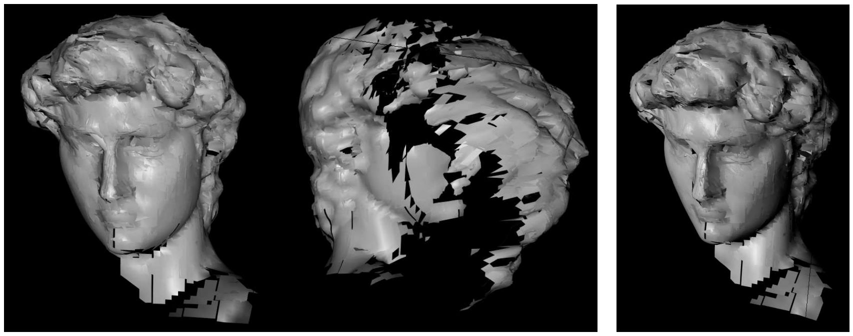
[Pul99] Pulli K.: Multiview registration for large data sets. In *Proc. 3DIM* (1999), pp. 160–168.

[RL01] Rusinkiewicz S., Levoy M.: Efficient variants of the icp algorithm. In *3DIM* (2001).

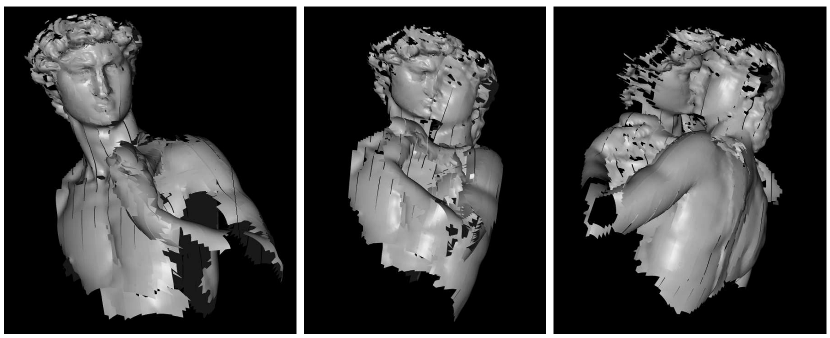
[SBB03] Silva L., Bellon O. R. P., Boyer K. L.: Enhanced, robust genetic algorithms for multiview range image registration. In *Proc. 3DIM* (2003), pp. 268–275.

[SLW04] Sharp G. C., Lee S. W., Wehe D. K.: Multiview registration of 3d scenes by minimizing error between coordinate frames. *IEEE PAMI. 26*, 8 (2004), 1037–1050.

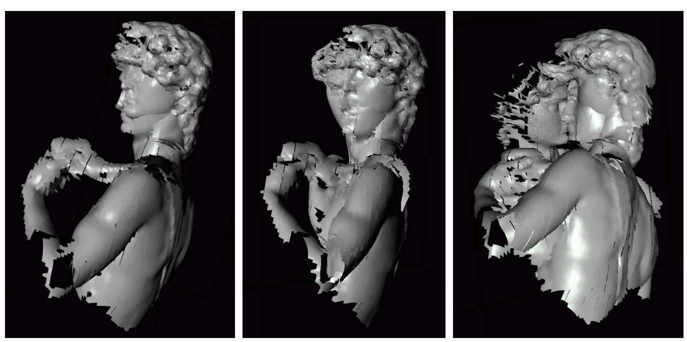
[WB01] Williams J., Bennamoun M.: Simultaneous registration of multiple corresponding point sets. *Comput. Vis. Image Underst. 81*, 1 (2001), 117–142.



(a) The head of David (detailed: 38 scans)



(b) The head and bust of David (10 scans)



(c) Head and bust: Bad initial alignment (21 scans)

Figure 2: *Registrations produced by* ***MBR****,* ***MAT****, and* ***QUAT*** *(from left to right) on different instances of the David model.*