

# Regression of 3D Rigid Transformations on Real-Valued Vectors in Closed Form

Takuya Funatomi<sup>1</sup>, Masaaki Iiyama<sup>2</sup>, Koh Kakusho<sup>3</sup> and Michihiko Minoh<sup>2</sup>

**Abstract**—In this paper, we present a regression for predicting 3D rigid transformations from real-valued vectors. We use a unit dual quaternion to represent the transformation. The regression is formulated as blending unit dual quaternions. To formulate it in a closed form, we introduce an approximation based on error metrics according to geometric algebra. Finally, we take an articulated motion and an elastic deformation as examples to present the descriptive power of our method in modeling the motion and the deformation.

## I. INTRODUCTION AND RELATED WORK

Linear models and linear regression techniques are fundamental statistical tools commonly used in a broad array of fields. Even for state-of-the-art nonlinear techniques, most have their roots in linear models. Linear techniques are still popular since they are simple, interpretable, and solvable in closed form. They also require fewer observations than recent nonlinear techniques, which can present a significant advantage over nonlinear techniques for many applications. For example, deep learning techniques are popular recently but they usually require millions of data for training. Meanwhile, simple linear techniques work well even if much a smaller number of data is available.

Most regression techniques predict a single real number from a vector of real numbers, *i.e.*  $\mathbb{R}^P \mapsto \mathbb{R}$ , where  $P$  denotes the dimension of the vector. Meanwhile, some tasks in robotics field need to process *geometric variables* such as motion. For instance, some techniques have been proposed to predict three-dimensional (3D) rotation [14], [17], [16], [15] and 3D rigid transformations [1], [11], [12]. Further, in [1] and [14], support vector regression (SVR) is generalized to predict 3D rotation and 3D rigid transformations. In [11], [12], Lang *et al.* proposed an extension of Gaussian processes (GPs) to predict 3D rigid transformations from a time series of 3D rigid transformations, *i.e.* a mapping from  $\{SE(3)\}^t$  to  $SE(3)$ . Here, key techniques include defining the kernel function in SVR or the covariance function for GPs such that the distance measure for the two transformations is properly expressed. Although these methods stand on well-known and powerful techniques, as far as we know, no methods have been proposed that correspond to the basis of “linear regression”.

The above geometric variables can be represented in several ways, such as by a unit quaternion or a homogeneous transformation matrix. Although these representations seem to be common multi-dimensional vectors, the elements are mutually dependent, *e.g.*, the norm/determinant are restricted to be 1. When we focus on each component independently, it seems to become nonlinear. However, when we treat all elements simultaneously based on geometric algebra, a linear approach will be still promising.

In this paper, we suppose a problem of 3D motion capture and study a method of predicting 3D rigid transformation  $SE(3)$  from a vector of real numbers  $\mathbb{R}^P$ . In order to utilize geometric algebra, we represent a 3D rigid transformation as a single number, which inherently has the mutual dependencies in its algebra, and propose a regression technique on the number. As the number, we adopt unit dual quaternion  $\hat{\mathbb{H}}$  to represent a 3D rigid transformation, which Kavan *et al.* have adopted in [10], and discuss the formulation of a regression of unit dual quaternions. In our formulation, the independent variable is a real vector  $\boldsymbol{x} \in \mathbb{R}^P$  and the dependent variable is a unit dual quaternion  $\hat{\mathbb{H}}$ . Since the input space is common such as a sensor output, our formulation will be applicable to a large number of problems.

Further, in this paper, we will show the descriptive power of the proposed method by applying it to model an articulated motion and an elastic deformation. As for an articulated motion, we suppose a finger with two rotational joints. As far as the joint angles are given, the regression of the unit dual quaternion can accurately predict the position and orientation of the fingertip (*i.e.* the distal phalanx) relative to its root (*i.e.* the proximal phalanx) without its kinematic model. To construct the regressor, we suppose that a few observations are given as pairs of joint angles  $\boldsymbol{x}$  and 3D rigid transformations  $\hat{\boldsymbol{y}} \in SE(3)$  of the fingertip relative to the root. As for an elastic deformation, we observe 3D rigid transformations of markers on a balloon while it is squashed. In order to model the deformation, we define two and one independent variables for a coordinate on the surface of the balloon and a degree of squashing, respectively. Then, we construct a regressor of 3D rigid transformations of the marker. We will show that this regressor represents a 3D shape of the balloon with its deformation as a result.

The major contribution of our paper is the formulation of a regression model of a unit dual quaternion with a closed-form approximation. This approximation makes the formulation easier to understand, easier to implement, and more computationally efficient. Through our experiments, we present

<sup>1</sup> Takuya Funatomi is with Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma, Nara 630-0192 Japan funatomi@is.naist.jp

<sup>2</sup> Masaaki Iiyama and Michihiko Minoh are with Kyoto University, Yoshida Honmachi, Sakyo-ku, Kyoto 606-8501 Japan

<sup>3</sup> Koh Kakusho is with Kwansei Gakuin University, 2-1 Gakuen, Sanda, Hyogo 669-1337 JAPAN

that our regressor achieves significantly better accuracy and has more descriptive power; further, the approximation is negligible in many cases.

The remainder of our paper is organized as follows. We introduce the unit dual quaternion in Section II and our proposed method in Section III. In Section IV, we present and discuss our experiments and corresponding results. Finally, we provide a conclusion and avenues for future work in Section V.

## II. REPRESENTING A 3D RIGID TRANSFORMATION BY A UNIT DUAL QUATERNION

There are several ways to represent 3D rigid transformations. It is common to use a  $4 \times 4$  homogeneous transformation matrix or decompose such a matrix into rotational and translational components. For the rotational component, there are many choices for representation, including the Euler angle, Cardan angle, Rodrigues formula, and unit quaternion. The translational component is commonly represented as a 3D real vector, but its values depend on the decomposition (e.g.,  $RT$ ,  $TR$ , and  $TRT^{-1}$ , where  $R$  and  $T$  represent rotational and translational transformation, respectively). For a given transformation, we can use any of these for representation and can convert one to another; however, we need to pay close attention to the selected representation when we consider multiple transformations. Even if two transformations are geometrically similar, the distance between them can be large in some representations. For example, “gimbal lock” is a well-known problem in Euler angle representation.

In the context of geometric skinning for a CG character, Kavan *et al.* [10] discussed the representations of 3D rigid transformations and blending, pointing out that the translation should not be treated independently from the rotation. They adopted the unit dual quaternion as a representation and proposed an algorithm that calculates a weighted average of such transformations. Although this paper explains an overview of dual quaternion below, refer [10] for further details on dual quaternion and its connection to  $SE(3)$ .

Originally proposed by Clifford [5] in the nineteenth century, a dual quaternion can be considered as a quaternion whose elements are dual numbers. Similar to complex numbers, any dual number  $\hat{a} \in \hat{\mathbb{R}}$  can be written as  $\hat{a} = a_0 + \epsilon a_\epsilon$ , where  $a_0$  is the non-dual part,  $a_\epsilon$  is the dual part, and  $\epsilon$  is a dual unit satisfying  $\epsilon^2 = 0, \epsilon \neq 0$ . Its conjugate is  $\hat{a}^* = a_0 - \epsilon a_\epsilon$ .

Multiplication of two dual numbers is given as  $\hat{a}\hat{a}' = (a_0 + \epsilon a_\epsilon)(a'_0 + \epsilon a'_\epsilon) = (a_0 a'_0) + \epsilon(a_0 a'_\epsilon + a_\epsilon a'_0)$ . Dual quaternion  $\hat{q} \in \hat{\mathbb{H}}$  can be written as the sum of two ordinary quaternions, i.e.  $\hat{q} = q_0 + \epsilon q_\epsilon$ , where  $q_0, q_\epsilon \in \mathbb{H}$ . This can also be written as  $\hat{q} = \hat{a} + \hat{b}i + \hat{c}j + \hat{d}k$ , where  $\hat{a}, \hat{b}, \hat{c}, \hat{d} \in \hat{\mathbb{R}}$  and  $i, j, k$  are the usual quaternion units.

As arithmetic operations, scalar multiplication, addition, multiplication, exponential and logarithm mapping of a dual

quaternion are given as follows [6].

$$s\hat{q} = sq_0 + \epsilon sq_\epsilon \quad (1)$$

$$\hat{q} + \hat{q}' = (q_0 + q'_0) + \epsilon(q_\epsilon + q'_\epsilon) \quad (2)$$

$$\hat{q}\hat{q}' = (q_0 q'_0) + \epsilon(q_0 q'_\epsilon + q_\epsilon q'_0) \quad (3)$$

$$e^{\hat{q}} = e^{a_0}(1 + \epsilon a_\epsilon) \left( \cos \phi + \frac{\sin \phi}{\phi}(b_0 i + c_0 j + d_0 k) + \epsilon \left( -\frac{\sin \phi}{\phi}\theta + \frac{\sin \phi}{\phi}(b_\epsilon i + c_\epsilon j + d_\epsilon k) + \frac{\cos \phi - \frac{\sin \phi}{\phi}\theta}{\phi^2}(b_0 i + c_0 j + d_0 k) \right) \right) \quad (4)$$

$$\log \hat{q} = \left( \log |q_0| + \frac{\psi}{\phi}(b_0 i + c_0 j + d_0 k) + \epsilon \left( \theta + \frac{a_0 a_\epsilon}{|q_0|^2} + \frac{\psi}{\phi}(b_\epsilon i + c_\epsilon j + d_\epsilon k) + \frac{\theta \frac{a_0 - \frac{\psi}{\phi}|q_0|^2}{\phi^2} - a_\epsilon}{|q_0|^2}(b_0 i + c_0 j + d_0 k) \right) \right), \quad (5)$$

where  $s \in \mathbb{R}$ ,  $\theta = b_0 b_\epsilon + c_0 c_\epsilon + d_0 d_\epsilon$ ,  $\phi = \sqrt{b_0^2 + c_0^2 + d_0^2}$ , and  $\psi = \arctan \frac{\phi}{a_0}$ .

Since the dual quaternion is defined as a combination of dual number and quaternion, there are multiple definitions for the conjugate of a dual quaternion, i.e.  $q_0^* + \epsilon q_\epsilon^*$ ,  $q_0 - \epsilon q_\epsilon$ , and  $q_0^* - \epsilon q_\epsilon^*$ , where  $q_0^*$  and  $q_\epsilon^*$  are the quaternion conjugate of  $q_0$  and  $q_\epsilon$ , respectively. In this paper, we use  $\hat{q}^*$  to denote the first type of conjugate of dual quaternion  $\hat{q}$ . The product  $\hat{q}\hat{q}^*$  generally becomes a dual number with its square root called the *magnitude*,  $\|\hat{q}\| = \sqrt{\hat{q}\hat{q}^*}$ <sup>1</sup>. The dual quaternion that represents any rigid transformation with a magnitude of 1 is called a *unit dual quaternion*. When the unit dual quaternion  $\hat{q}$  represents a rigid transformation,  $\hat{q}^*$  corresponds to the inverse transformation of  $\hat{q}$ . Since the magnitude of any unit dual quaternion equals to 1, it can be intuitively interpreted that the unit dual quaternions form a hypersphere in  $\hat{\mathbb{H}}$ . Then, the logarithm mapping for a unit dual quaternion can be interpreted as the projection onto the tangent space of the hypersphere at identity, which equals to  $1 \in \hat{\mathbb{H}}$ . The exponential mapping corresponds to the back projection as well.

It is well-known that any 3D rotational transformation is represented by a unit quaternion. The dual quaternion can also represent pure rotation in the same manner as a quaternion by setting the dual part to zero. From angle  $\theta$  and unit axis  $\mathbf{n} = (n_x, n_y, n_z)$ , the rotational transformation is represented by  $(\cos \frac{\theta}{2} + \sin \frac{\theta}{2}(n_x i + n_y j + n_z k)) + \epsilon(0 + 0i + 0j + 0k)$ .

As for a pure translation of  $\mathbf{t} = (t_x, t_y, t_z)$  with no rotation, the corresponding dual quaternion is  $(1 + 0i + 0j + 0k) + \epsilon(0 + \frac{t_x}{2}i + \frac{t_y}{2}j + \frac{t_z}{2}k)$ . According to these expressions, a general rigid transformation can be converted into the corresponding dual quaternion by decomposing it into its rotational and translational components. For example,

<sup>1</sup>The square root of dual number  $\hat{\mathbb{R}}$  is  $\sqrt{a_0 + \epsilon a_\epsilon} = \sqrt{a_0} + \epsilon \frac{a_\epsilon}{2\sqrt{a_0}}$ .

suppose that a 3D rigid transformation is decomposed as rotational component  $\hat{r}$  followed by translational component  $\hat{t}$ ; here, the corresponding dual quaternion is represented by their product, *i.e.*  $\hat{t}\hat{r}$ .

### III. REGRESSION OF A UNIT DUAL QUATERNION ON A REAL-VALUED VECTOR

In this paper, we propose a formulation of a regression model for the relationship between a 3D rigid transformation and a real-valued vector. Let  $\mathbf{x} \in \mathbb{R}^P$  and  $\hat{y} \in \hat{\mathbb{H}}$  be a  $P$ -dimensional real vector and the unit dual quaternion of a 3D rigid transformation, respectively. Suppose we have the problem of predicting a 3D rigid transformation  $\hat{y}$  from real-valued independent variables  $\mathbf{x} = (x_1, \dots, x_P)$ , where  $N$  observations of  $\{\hat{y}_n, \mathbf{x}_n\}_{n=1}^N$  are given.

As supposed in the traditional linear regression method, we assume that the dependent variable is predicted as a linear combination of the independent variables with an intercept. In this problem, each of the independent variables is a real value and the dependent variable is a unit dual quaternion. Thus, the linear combination can be considered as a linear blending of  $(P + 1)$  unit dual quaternions as

$$\hat{y}'_n = f(\tilde{\mathbf{x}}_n, \hat{\mathbf{b}}), \quad (6)$$

where  $\hat{y}'_n$  is the predicted dependent variable,  $\tilde{\mathbf{x}}_n = (1, x_{n1}, \dots, x_{nP})^\top \in \mathbb{R}^{P+1}$ ,  $\hat{\mathbf{b}} = (\hat{b}_0, \hat{b}_1, \dots, \hat{b}_P)^\top \in \hat{\mathbb{H}}^{P+1}$  is a vector of unit dual quaternions, which corresponds to the coefficients of the traditional linear regression model, and  $f(\tilde{\mathbf{x}}, \hat{\mathbf{b}}) : \mathbb{R}^{P+1} \times \hat{\mathbb{H}}^{P+1} \mapsto \hat{\mathbb{H}}$  is a blending of unit dual quaternions  $\hat{\mathbf{b}}$  with weights  $\tilde{\mathbf{x}}_n$ . Given this, the problem is then formulated as the problem of obtaining  $\hat{\mathbf{b}}$  that minimizes prediction error  $\sum_n |\hat{y}_n - \hat{y}'_n|$ .

Algorithms for blending dual quaternions, *i.e.* dual quaternion linear blending (DLB) and dual quaternion iterative blending (DIB), have been proposed in [10] as described in Algorithm 1. In brief, DLB computes the linear combination of unit dual quaternions simply, and then normalizes it to ensure the magnitude equals 1. The computation here is simple, but the results may not be theoretically optimal. Conversely, DIB is an iterative algorithm that finds theoretically optimal blending result  $\hat{q}$  with sufficiently small error  $\|\hat{e}\|$  in the same manner as that proposed in [8]. According to geometric algebra, the error is defined as the weighted sum of  $\log(\hat{q}^* \hat{b}_i)$ , which is the distance from  $\hat{q}$  to  $\hat{b}_i$  on the tangent space at  $\hat{q}$  (see Fig.1–a). Since the error here is not defined in a closed form, it is not trivial to solve it analytically. This is why DIB takes an iterative approach in finding the optimal solution numerically.

In [2], Buss *et al.* discussed the convergence rate of such iterative blending algorithms (though not for dual quaternion, rather for points on a  $d$ -dimensional sphere in  $\mathbb{R}^{d+1}$ ), assuming that the blending weight is non-negative and the sum of the weights equals 1. These assumptions are also satisfied in the context of [10]; however, in our case,  $\tilde{\mathbf{x}}_n$  can be an arbitrary  $(P+1)$ -dimensional real vector that does not satisfy these assumptions. This means that there is no guarantee of

---

#### Algorithm 1: Dual quaternion Iterative Blending (DIB) in [10]

---

**Input:** Unit dual quaternions  $\hat{\mathbf{b}} = (\hat{b}_0, \dots, \hat{b}_P)$ , convex weights  $\mathbf{x} = (x_0, \dots, x_P)$ , desired precision  $d$   
**Output:** Blended unit dual quaternion  $\hat{q}$

$$\hat{e} = 0, \hat{q} = DLB(\mathbf{x}, \hat{\mathbf{b}}) = \frac{\sum_{i=0}^P x_i \hat{b}_i}{\|\sum_{i=0}^P x_i \hat{b}_i\|}$$

**repeat**

$$\hat{q} = \hat{q} \exp(\hat{e})$$

$$\hat{e} = \sum_{i=0}^P x_i \log(\hat{q}^* \hat{b}_i)$$

**until**  $\|\hat{e}\| < d$

**return**  $\hat{q}$

---

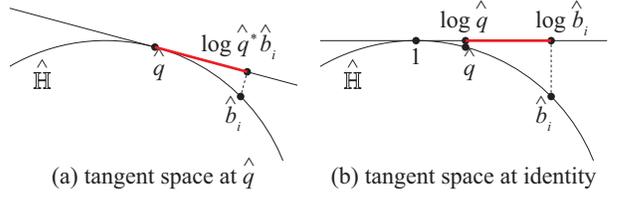


Fig. 1. Distance between unit dual quaternion  $\hat{a}$  and  $\hat{b}$ , with (a) and (b) presenting the mathematically ideal distance adopted by DIB and our relaxation, respectively.

convergence for the iterative algorithm. Hence, we explore another way to approximately solve this problem.

In DIB, error  $\|\hat{e}\|$  was formulated as the norm of the weighted sum of  $\log(\hat{q}^* \hat{b}_i)$ . Since  $\hat{q}$  is unknown, it becomes difficult to evaluate the error in the closed form; thus, the iterative algorithm is adopted to solve the problem. Therefore, we introduce the following two approximations. First, we approximate this error by evaluating the distance from  $\hat{q}$  to the weighted sum of  $\hat{b}_i$  on the tangent space at identity, not at  $\hat{q}$ , as illustrated in Fig.1–b. This approximation is known as the first order approximation through Baker–Campbell–Hausdorff (BCH) formula in Lie algebra [18]. This changes the error in blending to  $\hat{e} = \log \hat{q} - \sum_{i=0}^P x_i \log \hat{b}_i$ . Second, we modify the blending algorithm so as to initialize  $\hat{q}$  by identity, not via DLB, as in algorithm 1. Accordingly, we formulate the blending in linear regression as

$$f(\tilde{\mathbf{x}}, \hat{\mathbf{b}}) := \exp(\tilde{\mathbf{x}}^\top \log \hat{\mathbf{b}}), \quad (7)$$

where  $\log \hat{\mathbf{b}} = (\log \hat{b}_0, \dots, \log \hat{b}_P)^\top$ . This is analytically solvable regardless of whether the assumptions on the weights are satisfied. Equation 6 can then be rewritten as

$$\log \hat{y}'_n = \tilde{\mathbf{x}}_n^\top \log \hat{\mathbf{b}}. \quad (8)$$

The above simply becomes a linear combination through the logarithm map, so we regard this as a usual linear regression problem.

For the given observations,  $\hat{\mathbf{b}}$  is obtained by minimizing the residual  $|\hat{y}_n - \hat{y}'_n|$ . By following the same approximations introduced above, the residual is also defined by evaluating the distance on the tangent space at identity.

Eventually, the error term is formulated as

$$\left\| \hat{\mathbf{e}}(\hat{\mathbf{b}}) \right\| = \sum_n \left\| \log \hat{\mathbf{y}}_n - \tilde{\mathbf{x}}_n^\top \log \hat{\mathbf{b}} \right\| \quad (9)$$

Next, the formulation becomes quite similar to the linear regression of a real variable and can be solved as

$$\log \hat{\mathbf{b}} = \left( \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \right)^{-1} \tilde{\mathbf{X}}^\top \log \hat{\mathbf{y}}, \quad (10)$$

where  $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_N)^\top$ , and  $\log \hat{\mathbf{y}} = (\log \hat{\mathbf{y}}_1, \dots, \log \hat{\mathbf{y}}_N)^\top$ . Using the obtained  $\log \hat{\mathbf{b}}$ , the formula below provides a prediction of the 3D rigid transformation represented by unit dual quaternion  $\hat{\mathbf{y}}'$  for arbitrary independent variables  $\mathbf{x}$ .

$$\hat{\mathbf{y}}' = \exp \left( \tilde{\mathbf{x}}^\top \log \hat{\mathbf{b}} \right) \quad (11)$$

#### IV. EXPERIMENTS

##### A. Modeling Articulated Motion

Here, we present a calibration problem of a simple wearable motion capture device. Suppose a finger with two rotational joints, as illustrated in Fig.2–a and a sensor is attached along the finger to capture the degree of bending. For such a device, the calibration can be achieved via regression analysis where the joint locations are unknown. In particular, we solve the calibration problem that maps the joint angles  $\mathbf{x}$  to a 3D rigid transformations  $\hat{\mathbf{y}} \in SE(3)$  of the end-effector (*i.e.* the distal phalanx) relative to its root (*i.e.* the proximal phalanx) from a few observations  $\{\hat{\mathbf{y}}_n, \mathbf{x}_n\}_{n=1}^N$ . If the joint locations are given, this was easily solvable as forward kinematics. If the end-effector is directly connected to the root via single joint, its location can be estimated with some methods [3], [4]. However, these methods are not applicable since there are multiple joints between the end-effector and the root.

First, we will show the stability and the robustness of the proposed method with synthetic data by controlling the number of training samples and the level of additive noise. Then, we will show the actual use of the proposed method to real data using a simple equipment.

1) *Generating Synthetic Data:* To simulate actual observations, we suppose that the distal phalanx has four markers. For  $M$  kinds of joint angles  $\mathbf{x}_m$ , we generate 3D position of markers  $\mathbf{P}_m = (\mathbf{p}_{1m}, \dots, \mathbf{p}_{4m})$ . We set 26 steps for both  $x_1$  and  $x_2$  independently, so totally  $M = 676$  sets of data are generated. Figure 2–b presents examples of the data.

In this experiment, we choose a subset of the data as observations  $\{\hat{\mathbf{y}}_n, \mathbf{x}_n\}_{n=1}^N$  for training and use whole dataset for accuracy evaluation. We add Gaussian noise on both  $\mathbf{P}_n$  and  $\mathbf{x}_n$  with various noise levels,  $\sigma = 10^{-3}, 10^{-2}, 10^{-1}, 10^0$ . Here, the unit is mm for  $\mathbf{P}_n$ . We define  $\hat{\mathbf{y}}_n$  as the 3D rigid transformation from  $\mathbf{P}_0$  to  $\mathbf{P}_n$  by the algorithm proposed in [9]. Here,  $\mathbf{P}_0$  is the marker position of a *reference posture*, which corresponds to  $\mathbf{x}_0 = (120, 120)$ .

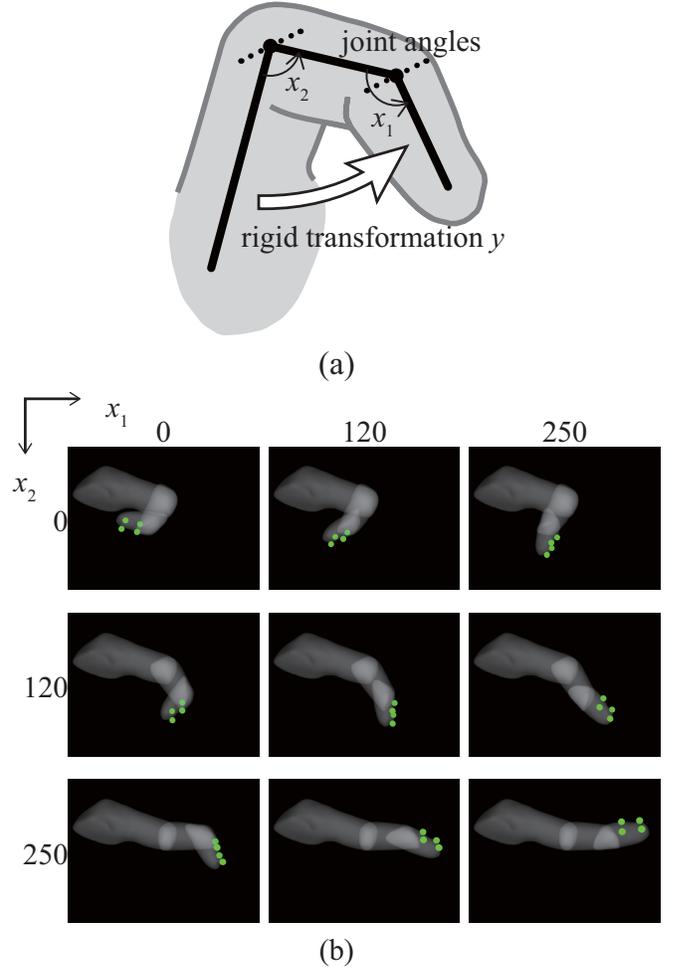


Fig. 2. Synthetic data: (a) Kinematic model of a finger. (b) Examples of synthetic data; although the phalanges are displayed for reference, the 3D position of the markers (displayed as green dot in the images) and corresponding joint angles are the only data provided by the dataset.

2) *Comparisons with Other Methods:* To present the advantages of our proposed method, we compare the accuracy with several methods listed in Table I.

In addition to our proposed method, we evaluate two types of regression methods in our experiments. The first one uses DIB or DLB to substitute for Eq.7 in the regression of a unit dual quaternion, which can be addressed as the direct extension of [10]. For example, a regression with DLB can be formulated as

$$\hat{\mathbf{b}}_i^{\text{DLB}} = \frac{\left\{ \left( \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \right)^{-1} \tilde{\mathbf{X}}^\top \hat{\mathbf{y}} \right\}_i}{\left\| \left\{ \left( \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \right)^{-1} \tilde{\mathbf{X}}^\top \hat{\mathbf{y}} \right\}_i \right\|} \quad (12)$$

$$\hat{\mathbf{y}}^{\text{DLB}} = \frac{\tilde{\mathbf{x}}^\top \hat{\mathbf{b}}^{\text{DLB}}}{\left\| \tilde{\mathbf{x}}^\top \hat{\mathbf{b}}^{\text{DLB}} \right\|}. \quad (13)$$

The second type of method uses multiple real-valued regressors to predict the transformation parameters. There are many choices for representing 3D rigid transformations

TABLE I  
THE METHODS EVALUATED IN THIS EXPERIMENTS.

ID	Regression methods
(a)	the proposed method
(b)	DIB based regression of unit dual quaternion
(c)	DLB based regression of unit dual quaternion
(d)	SVRs for elem. of quaternion and translation
(e)	SVRs for elem. of unit dual quaternion
(f)	Ridge regressions for elem. of quaternion and translation
(g)	Ridge regressions for elem. of unit dual quaternion

and the regressors, including nonlinear approaches. As an example, method (d) represents a 3D rigid transformation as the combination of the unit quaternion and the translation. This method has seven (*i.e.* four for quaternion and three for translation) regressors independently trained from the given observations. At the prediction stage, each element is predicted independently. Since the norm of the quaternion must be 1, the predicted elements of the quaternion are normalized to ensure that they represent a 3D rotational transformation. We also used the same approach for a unit dual quaternion with eight regressors. As for the regression method, we used ridge regression and a support vector regression (SVR) with a radius basis function kernel. Since these methods have some parameters to be tuned, *e.g.*  $\lambda$  for ridge regression, we adopted grid search over a parameter grid.

3) *Experimental Results and Discussion:* We compared the methods via accuracy of the prediction. For independent variables  $\mathbf{x}$ , we used  $(x_1, x_2, x_1x_2)^\top$ . We evaluated the accuracy with various noise levels on different sizes of observations, *i.e.*  $N = \{3^2, 9^2, 26^2\}$ . To evaluate accuracy, we predicted marker positions and compared them with noise-free data by calculating mean distances. Figure 3 shows the results in log scale.

Overall, our proposed method (a) achieved the best accuracy for all cases. Especially when the noise level is low, the accuracy was less than 0.5 mm and significantly better than the others. This indicates that our proposed regression model can describe the given motions well.

Conversely, our results show that method (b) could not model the motions at all since DIB failed to converge despite it has precise error metric. Even method (c) achieved worse results than ours. Although the difference between methods (a) and (c) lies only in blending function  $f(\hat{\mathbf{x}}_n, \hat{\mathbf{b}})$ , the accuracy was significantly different. This supports that our approximation was appropriate.

From the results of methods (d)–(g), it is indicated that the SVRs could model the motions better for closed test ( $N = 676$ ) when the noise level is small. When the number of samples is limited, method (g) seems the best among them. Meanwhile, our proposed method (a) always outperforms the others by successfully handling the motion through geometric algebra.

4) *Limitations due to Approximations:* Our formulation currently approximates the blending function as Eq.7. Obviously, this approximation is based on the assumption that  $\hat{y}$  to be predicted is close to the identity transformation, which

corresponds to no rotation and no translation.

As mentioned in Section IV-A.1, we defined the reference posture, which corresponds to the identity transformation, as the posture of  $\mathbf{x}_0 = (120, 120)$ , where  $x_1, x_2 \in [0, 250]$ . Since  $\mathbf{x}_0$  was defined in the middle of the range, the dependent variable was naturally supposed to have a 3D rigid transformation close to the identity.

To validate this approximation, we attempted to break the above assumption by taking  $\mathbf{x}'_0 = (0, 0)$ , which corresponds to a limit of the range. Figure 4 presents the accuracy comparison among the different settings of the reference posture. Results show that the accuracy of using  $\mathbf{x}'_0$  was significantly degraded because the observations contained large transformations that were rotationally over  $\pi$ . Conversely, when the observations are limited ( $N = 9$ ), the accuracy degradation was small, in which the observations did not contain such large transformations. In the dual quaternion representation, the rotation of angle  $\theta > \pi$  is treated as that of angle  $2\pi - \theta$ , breaking the linear relationship in the regression model; thus, the accuracy was significantly degraded. If such large transformations are included in the observations, additional care for them will be required, which is a goal of our future work. Excepting that, our results show that the approximation slightly affects accuracy.

5) *Experiments using Real Data:* We also demonstrated the given problem using real data in which we predicted the position and orientation of the fingertip from the output of a flex sensor. Although an index finger potentially has 2DOF, the joint angles are strongly correlated and it can be approximately treated as 1DOF [13]. Thus, we used single flex sensor to measure the degree of bending. Figure 5 presents the setup used for this experiment. In general, a flex sensor increases its resistance when it is bent. To observe the degree of flexion of a finger, we attached the sensor along the index finger by a fingerstall and fastener tape. We used this output  $s$  to make the independent variable of regression as  $\mathbf{x} = (s, s^2)^\top$ . We also defined the dependent variable as the 3D rigid transformation of the fingertip relative to the proximal phalanx. To measure it, we placed two AR markers on the distal and proximal phalanges in the observation stage, capturing 3D rigid transformations of them by a camera [7].

In the prediction stage, we removed the marker on the distal phalanx. Using the constructed regressor, we obtained 3D rigid transformation  $\hat{y}'$  from the output of the flex sensor. We measured the AR marker on the proximal phalanx and predicted the position and orientation of the fingertip by multiplying by  $\hat{y}'$ . In Fig.5, the prediction result is illustrated by an arrow in the captured image. Although the flex sensor does not accurately provide the joint angles, instead only showing how the finger flexes, our regressor also achieved good prediction in this experiment from only a few observations.

#### B. Modeling Elastic Shape Deformation using Real Data

1) *Problem Settings:* Next, we present a problem of modeling elastic deformation of a real object. In order to apply the regression approach, we need to suppose some observable

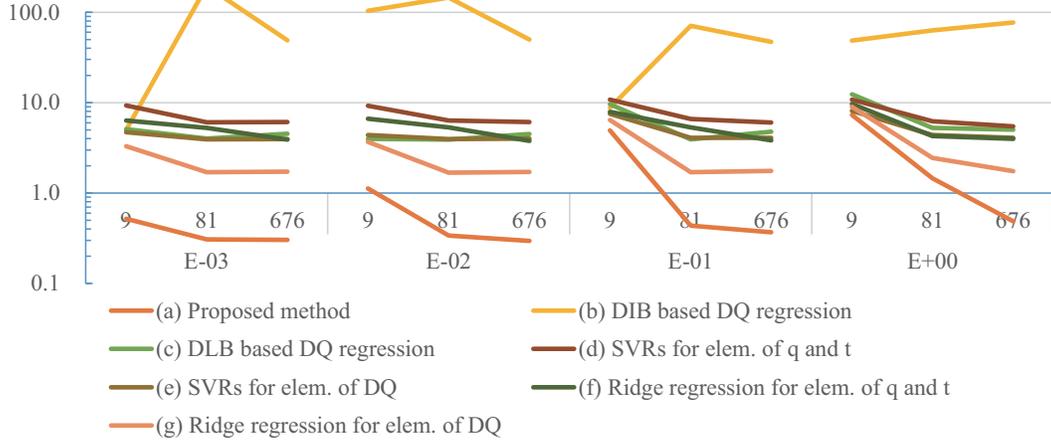


Fig. 3. Accuracy evaluation for various methods with various noise levels and the sizes of observations on the dataset. We add noise with four kind of levels onto both  $P_n$  and  $x_n$  of the dataset. Each chart presents the relationship between the size of observation and the prediction accuracy.

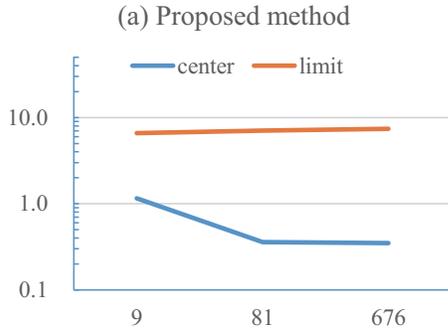


Fig. 4. Comparison between the reference postures, with “center” and “limit” adopting postures  $x_0 = (120, 120)$  and  $x'_0 = (0, 0)$  as reference postures, respectively, with  $i = 1, 2$ .

independent variables that control the deformation. As an example of such deformation, we use a motorized vertical stage to squash a balloon as illustrated in Fig.6–a. Using the apparatus, the deformation is accurately controlled by the height of the stage  $w$ .

Although there are many ways to represent a deformation of an object, we represent the deformation as the 3D rigid transformation of a small patch on the object surface. If we can track the small patches on the balloon surface from the observations while the balloon is squashed, the deformation of the balloon can be acquired. Since the deformation is elastic, the transformation will smoothly varies along the surface. In order to model this variety, we also define local coordinate system  $(u, v)$  along the surface as illustrated in Fig.6–b, and use them as independent variables as well.

In short, the independent variables  $x = (u, v, w)$  and the dependent variable  $\hat{y}$  is defined as the 3D rigid transformation of a small patch on the surface of the balloon. For measuring them, we put some AR markers [7] on the surface of the balloon in a grid. We can assign  $(u, v)$  for each marker according to the grid and track the 3D rigid transformation of them. In order to accurately acquire the transformation, we used two cameras to measure the 3D position of the

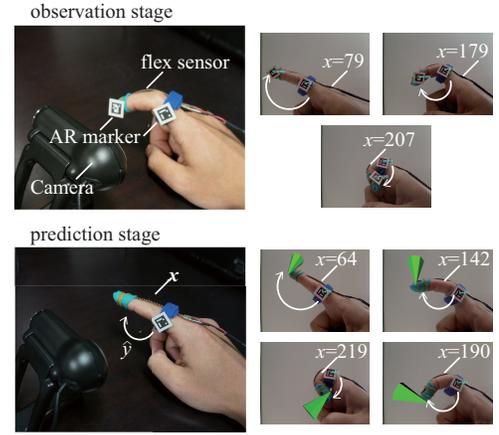


Fig. 5. Settings for the experiment with real data and some results of the proposed method. The green triangles in the prediction stage show the estimated position and orientation of the marker that placed on the fingertip in the observation stage. See the video as well for qualitative evaluation.

corners of the markers using stereo technique. Then, the 3D rigid transformation is calculated by the algorithm proposed in [9] as well as in Section IV-A.

2) *Experimental Results and Discussions:* We built an apparatus Fig.6–a and observed a deformation of a squashed balloon. The size of the balloon was about 200mm in diameter, 20 markers of 15mm square were put on the balloon, and two cameras were set about 1.5m away. The displacement of the vertical stage for squashing the balloon was 140mm. The cameras captured images with an interval of 2.5mm of the displacement. Figure 7–a presents examples of captured image. Although we put 20 markers, only 15 of them were used since the rest could not be detected on one of the cameras. As the result,  $M = 15 \times (140/2.5 + 1) = 855$  samples of  $x_m = (u, v, w)$  and  $\hat{y}_m$  were captured in total. Here, we use the marker of  $x = (50, 50, 50)$  as a reference marker and  $\hat{y}_n$  is defined with respect to it.

As well as Section IV-A, we compare the accuracy of the

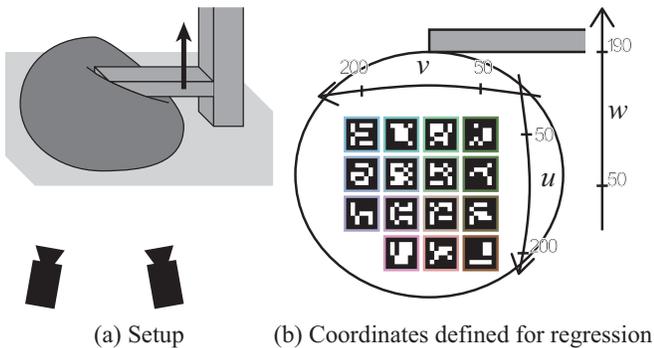


Fig. 6. Experimental setup for modeling elastic deformation by regression using real data. (a) shows the setup for the experiment. Two cameras are used to perform 3D measurement. A motorized vertical stage is used to control the deformation. (b) shows the independent variables  $(u, v, w)$ .

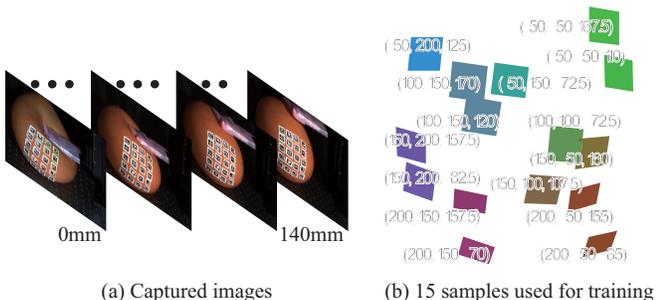


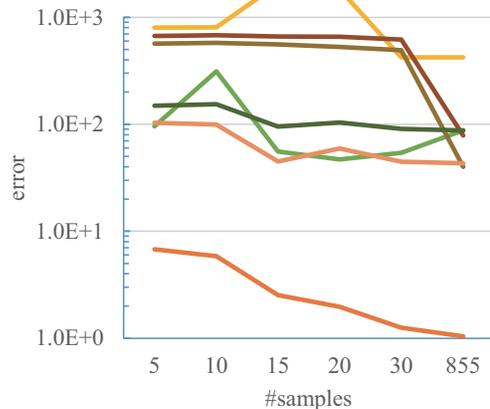
Fig. 7. Observation of a squashed balloon deformation. Totally, 57 frames are captured and 3D position and orientation of 15 markers are measured for each frame. Some input images and corresponding value of  $w$  are presented in (a). For training the regressor, we choose 15 samples from  $57 \times 15 = 855$  samples using k-means clustering.  $(u, v, w)$  and corresponding marker position and orientation is presented in (b).

proposed method with other methods with various number of training samples. For training, we choose  $N$  samples out of  $M$  observations. Here, k-means clustering algorithm is applied for  $x_m$  and the center of each cluster is selected to simply perform uniform sampling in the parameter space of  $x$ . Figure 7–b visualizes the training samples when  $N = 15$ . Each square represents the marker in 3D and  $x_n$  is also indicated.

We made several training datasets of  $N = 5, 10, 15, 20, 30$ , and 855, and trained the regression methods as enumerated in Table I. For (a)–(c) and (f)–(g), we prepared three types of explanatory variables that are the permutations of  $p$  independent vectors  $\tilde{x}$  to emulate polynomial regression<sup>2</sup>. For (a)–(c), we also modified Eq.10 to make it as ridge regression with  $l_2$  regularization. As a result, these methods also have  $p$  and  $\lambda$  as parameters of regressor as well as other methods. Thus, we also adopted grid search over a parameter grid of  $p = \{1, 2, 3\}$  and  $\lambda = \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$  for (a)–(c). For the trained regressor, the prediction accuracy is evaluated using the 3D position of the corners of the markers of  $M$  samples.

Figure 8 presents the results of the methods (a)–(g) for

<sup>2</sup> $p = 1 : (1, u, v, w)$ ,  $p = 2 : (1, u, v, w, uv, vw, wu, u^2, v^2, w^2)$ , and so on.



— (a) Proposed method  
— (b) DIB based DQ regression  
— (c) DLB based DQ regression  
— (d) SVRs for elem. of  $q$  and  $t$   
— (e) SVRs for elem. of DQ  
— (f) Ridge regression for elem. of  $q$  and  $t$   
— (g) Ridge regression for elem. of DQ

Fig. 8. Errors on predicting the corners position of the markers of various methods. Only the proposed method achieved the meaningful results. All of the other methods obviously failed to model the elastic deformation.

various size of training samples in log scale. Similar to the results in Fig.3, the proposed method obviously outperforms the other methods. As the errors imply, only the proposed method could model the deformation well and the results of the other methods were meaningless. We also visualize these results to show it. Figure 9–b are the results of the proposed method trained from 15 samples, which are depicted in Fig.6–b. Here, the viewpoint of this figure is slightly changed from the original image to show that the markers are predicted in 3D. This shows that the proposed method successfully models the changes on the 3D rigid transformation of the markers. Since the proposed method models the deformation along the coordinate  $(u, v)$  continuously defined along the surface, it can also recover the continuous shape. Figure 9–c presents upsampled shapes. On the contrary, even the best results among the others in Fig.8, e.g. (e) and (g) with  $N = 855$ , form completely different shape deformation as shown in Fig.9–d and e, respectively.

## V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a regression method of predicting 3D rigid transformations on real-valued vectors. By introducing an approximation on error metric for the blending of unit dual quaternions, our regression model is formulated in the closed form. This approximation makes the formulation easy to understand, easy to implement, and computationally efficient. We have used articulated motion and elastic deformation as examples and experimentally demonstrated that geometric algebra can help to formulate

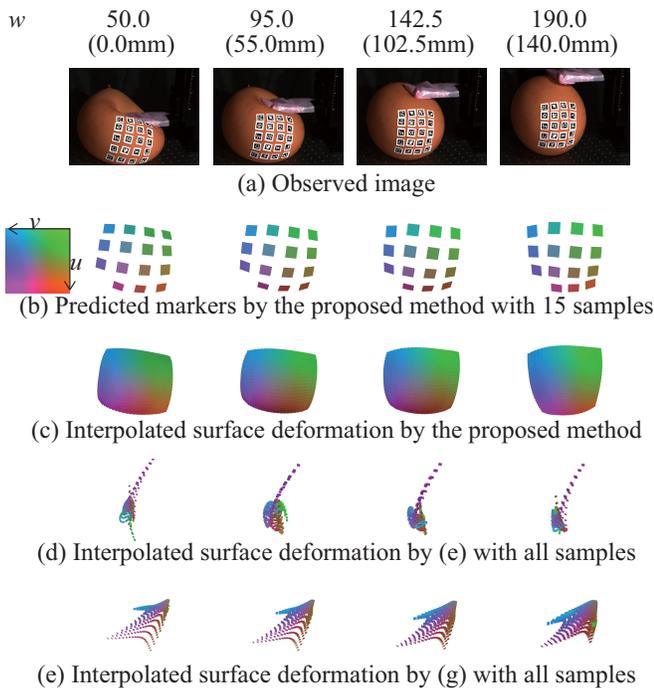


Fig. 9. Visual results of the regression. The proposed method appropriately describe the deformation of the balloon as presented in (b) and (c). The squares in (b) correspond to the markers on the balloon. (c) shows the smooth surface defined by the regressor along  $(u, v)$ . Meanwhile, SVR approaches obviously fail to model the deformation as shown in (d) and (e) even if all the samples are used. Note that the same variables are used as the independent variables for the regressors. See the video as well for qualitative evaluation.

the transformation in simple form even if each parameter of the representation has nonlinearity. Our results have shown that our proposed method is applicable to accurately model both articulated motion and elastic deformation from a small number of observations. Especially, the achievement of the proposed method opens up new possibilities in the modeling 3D shape and its deformation of a real object.

In future work, we will consider more accurate approximation by taking higher order terms in the BCH formula into account. Also, we plan to extend our method to support more advanced statistical methods.

#### ACKNOWLEDGMENT

This work was supported by JSPS Grant-in-Aid for Young Scientists B–23700231, and A–26700013. We are grateful to the following people for resources, discussions and suggestions: Prof. Leonidas J. Guibas (Stanford University), Prof. Masayuki Mukunoki, Haruna Akuzawa, Takuya Yamane (Kyoto University), Akihiro Ono (Kwansei Gakuin University), and Takahiro Kushida (Nara Institute of Science and Technology).

#### REFERENCES

[1] E. Bayro-Corrochano and R. Vallejo, “Svms using geometric algebra for 3d computer vision,” in *Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on*, vol. 2, 2001, pp. 872–877 vol.2.

[2] S. R. Buss and J. P. Fillmore, “Spherical averages and applications to spherical splines and interpolation,” *ACM Trans. Graph.*, vol. 20, no. 2, pp. 95–126, 2001. [Online]. Available: <http://doi.acm.org/10.1145/502122.502124>

[3] L. Y. Chang and N. S. Pollard, “Constrained least-squares optimization for robust estimation of center of rotation,” *Journal of Biomechanics*, vol. 40, no. 6, pp. 1392 – 1400, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021929006001606>

[4] —, “Robust estimation of dominant axis of rotation,” *Journal of Biomechanics*, vol. 40, no. 12, pp. 2707 – 2715, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021929007000437>

[5] W. Clifford, R. Tucker, and H. Smith, *Mathematical Papers*. Macmillan and Company, 1882.

[6] N. Dantam, H. Ben Amor, H. Christensen, and M. Stilman, “Online multi-camera registration for bimanual workspace trajectories,” in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, 2014, pp. 588–593.

[7] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marn-Jimnez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” *Pattern Recognition*, vol. 47, no. 6, pp. 2280 – 2292, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0031320314000235>

[8] V. Govindu, “Lie-algebraic averaging for globally consistent motion estimation,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, 2004, pp. I–684–I–691 Vol.1.

[9] B. K. P. Horn, “Closed-form solution of absolute orientation using unit quaternions,” *J. Opt. Soc. Am. A*, vol. 4, no. 4, pp. 629–642, 1987. [Online]. Available: <http://josaa.osa.org/abstract.cfm?URI=josaa-4-4-629>

[10] L. Kavan, S. Collins, J. Žára, and C. O’Sullivan, “Geometric skinning with approximate dual quaternion blending,” *ACM Trans. Graph.*, vol. 27, no. 4, pp. 105:1–105:23, 2008. [Online]. Available: <http://doi.acm.org/10.1145/1409625.1409627>

[11] M. Lang, O. Dunkley, and S. Hirche, “Gaussian process kernels for rotations and 6d rigid body motions,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, pp. 5165–5170.

[12] —, “Gaussian process dynamical models over dual quaternions,” in *European Control Conference (ECC)*, 2015.

[13] H. Rijkema and M. Girard, “Computer animation of knowledge-based human grasping,” *SIGGRAPH Comput. Graph.*, vol. 25, no. 4, pp. 339–348, 1991. [Online]. Available: <http://doi.acm.org/10.1145/127719.122754>

[14] A. Shilton, D. Lai, and M. Palaniswami, “A division algebraic framework for multidimensional support vector regression,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 2, pp. 517–528, 2010.

[15] P. Thomas Fletcher, “Geodesic regression and the theory of least squares on riemannian manifolds,” *International Journal of Computer Vision*, vol. 105, no. 2, pp. 171–185, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s11263-012-0591-y>

[16] F. Tobar and D. Mandic, “Quaternion reproducing kernel hilbert spaces: Existence and uniqueness conditions,” *Information Theory, IEEE Transactions on*, vol. 60, no. 9, pp. 5736–5749, 2014.

[17] R. Y. Wang, K. Pulli, and J. Popović, “Real-time enveloping with rotational regression,” *ACM Trans. Graph.*, vol. 26, no. 3, 2007. [Online]. Available: <http://doi.acm.org/10.1145/1276377.1276468>

[18] X. Yang, Y. Li, D. Reutens, and T. Jiang, “Diffeomorphic metric landmark mapping using stationary velocity field parameterization,” *International Journal of Computer Vision*, vol. 115, no. 2, pp. 69–86, 2015. [Online]. Available: <http://dx.doi.org/10.1007/s11263-015-0802-4>