# Motion Imitation Based on Sparsely Sampled Correspondence

Shuo Jin[1], Chengkai Dai[1], Yang Liu[2], and Charlie C.L. Wang[3,*]

*Abstract*— **Existing techniques for motion imitation often suffer a certain level of latency due to their computational overhead or a large set of correspondence samples to search. To achieve real-time imitation with small latency, we present a framework in this paper to reconstruct motion on humanoids based on sparsely sampled correspondence. The imitation problem is formulated as finding the projection of a point from the configuration space of a human's poses into the configuration space of a humanoid. An optimal projection is defined as the one that minimizes a back-projected deviation among a group of candidates, which can be determined in a very efficient way. Benefited from this formulation, effective projections can be obtained by using sparsely sampled correspondence, whose generation scheme is also introduced in this paper. Our method is evaluated by applying the human's motion captured by a RGB-D sensor to a humanoid in real time. Continuous motion can be realized and used in the example application of tele-operation.**

*Index Terms*— **motion imitation, configuration projection, sparsely sampled correspondence, tele-operation**

## I. Introduction

Humanoid robots have been widely studied in the research of robotics. With the recent development of motion capture devices such as RGB-D camera (e.g., Kinect) and wearable sensor system (e.g., Xsens MVN), efforts have been made to generate human-like motions for humanoid robots with high degree-of-freedoms. However, directly applying captured poses of human to humanoids is difficult due to the difference in human's and humanoid's kinematics. Therefore, a variety of kinematics based approaches for humanoid imitation have been investigated, which can be classified into two categories. Many of them perform an offline optimization step to compute corresponding configurations that conform to the mechanical structures and kinematics of humanoids from input human data [1]–[6]. It is obvious that the significant computational overhead in those techniques prevents us from applying them to real-time imitation. Methods in the other thread of research compute online imitation following captured human motion [7]–[12].

In this paper, we consider about the problem of realizing real-time human-to-humanoid motion imitation. Unfortunately, it is not an easy task due to:

- full sampling of human-to-humanoid correspondence often leads to large data size;
- high non-linearity of underlying mechanical rules results in significant computational cost;
- how to find the configuration of a humanoid according to the input poses of human in real-time is not intuitive.

Artificial neural networks have been adopted to ease the difficulties, with which a lot of efforts have been made in simulation and for robots with small degree-of-freedoms [13]–[21]. A recent work [17] by Stanton et al. directly introduced neural networks with particle swarm optimization to find the mapping between human movements and joint angle positions of humanoid. However, there is no measurement presented in their work to evaluate the quality of humanoid poses generated by the trained neural system. On the other aspect, our method is also different from this work in terms of the training data set. We use the sparse correspondence instead of the densely recorded raw data, which can help eliminate the redundancy in data set and improve the training speed. Moreover, only requiring a sparse set of correspondence samples leads to a lower barrier of system implementation.

We propose a framework that allows efficient projection of a pose from human's space to the configuration space of humanoid based on sparsely sampled correspondence extracted from recorded raw data, which can be used to realize motion imitation in real time (see Fig. 2). Experimental results show that our framework can be successfully applied in the motion imitation of

[1]S. Jin and C. Dai are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong.

[2]Y. Liu is with Microsoft Research Asia, Beijing, China.

[3]C.C.L. Wang is with the Department of Design Engineering and TU Delft Robotics Institute, Delft University of Technology, The Netherlands.

[*]Corresponding Author. Email:c.c.wang@tudelft.nl

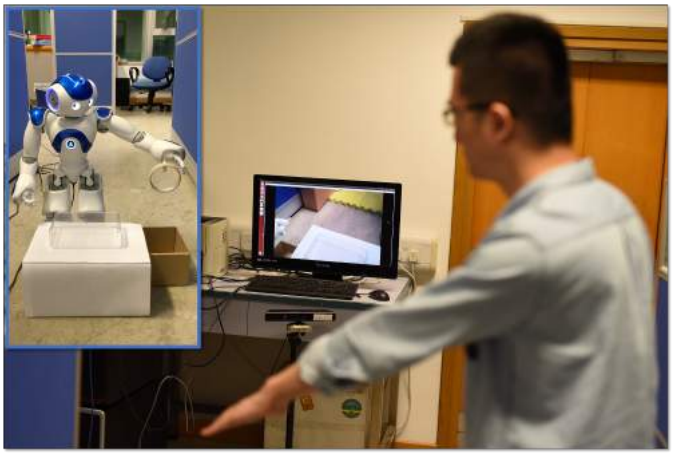

Fig. 1. An example of imitation realized by our framework working with the NAO humanoid.

humanoid (see Fig.1 for an example of tele-operation using a NAO humanoid). Motion control is a very important task in recent popular development of human-robot interaction. With the help of this tool, the job of robot's motion control can be implemented easily by assigning sparse correspondences between the poses of human and humanoid robots.

## II. Framework of Configuration Projection

### A. Problem Definition

A human pose can be uniquely represented as a point (abbreviated as $C$-point) $\mathbf{h} \in \mathbb{R}^m$ in the configuration space (abbreviated as $C$-space – $\mathcal{H}$) of human's motion and its corresponding pose of humanoid can be denoted as a point $\mathbf{r} \in \mathbb{R}^n$ in the $C$-space of humanoid – $\mathcal{R}$. We assume one-to-one correspondence between the poses of human body and humanoid, i.e. the mapping between human and humanoid's $C$-spaces is bijective. A pair of human's and humanoid's configurations is denoted as $(\mathbf{h}, \mathbf{r}) \in \mathbb{R}^{m+n}$. Given stored correspondence pairs $\{(\mathbf{h}, \mathbf{r})\}$ as the known knowledge and a new input pose $\mathbf{h}^* \in \mathbb{R}^m$, the configuration projection $\Omega(\cdot)$ can be defined as finding a corresponding $\mathbf{r}^* \in \mathbb{R}^n$ that satisfies two basic properties:

- **Identity** – for any sample pair $(\mathbf{h}_i, \mathbf{r}_i)$ in the data-set, it should have
$$\Omega(\mathbf{h}_i) = \mathbf{r}_i.$$
- **Similarity** – for an input $C$-point of human $\mathbf{h}^*$, if
$$\max\{\min_i \|\mathbf{h}_i - \mathbf{h}^*\|\} < \delta$$
then it should have
$$\|\Omega(\mathbf{h}^*) - \tilde{\mathbf{r}}(\mathbf{h}^*)\| < \epsilon,$$
where $\delta$ and $\epsilon$ are two constant values, and $\tilde{\mathbf{r}}(\mathbf{h}^*)$ is a $C$-point of humanoid that can be obtained by more accurate but computational intensive methods (e.g., inverse kinematics) as the ground truth.

All sample pairs should be repeated with the projection $\Omega(\cdot)$ according to the property of *identity*. The demand on *similarity* indicates that if a new input is close to the known samples, its projected result should not deviate too much from its corresponding ground truth.

The main difficulty of finding the projection $\mathbf{r}^*$ lies in the lack of explicit functions to determine the mapping between two $C$-spaces with different dimensions (i.e., degree-of-freedoms). Given sparsely aligned pairs of poses as samples, we try to solve this problem by proposing a strategy of kernel-based projection to find a good approximation for $\mathbf{r}^*$.

### B. Data Pre-processing

The knowledge of correspondence $\{(\mathbf{h}, \mathbf{r})\}$ can be established through experiments. Although aligning a pose of human body with a corresponding pose of humanoid can be taken manually, it is a task almost impossible if thousands of such correspondence samples need to be specified. Therefore, in our experiments, we first capture continuous motions of human bodies by using a motion capture system. The data-set obtained in this way often results in large size and redundancy. To resolve this problem, we perform a pre-processing step to extract marker poses from the raw data-set recorded from human's motion. Specifically, mean shift clustering [22] is employed to generate the marker set denoted as $\mathcal{H}$. For each sample $\hat{\mathbf{h}} \in \mathcal{H}$, its corresponding pose $\hat{\mathbf{r}}$ in the configuration space of humanoid can be either specified manually (when the number of samples in $\mathcal{H}$ is small) or generated automatically by a sophisticated method (e.g., the inverse kinematics methods). The pairs of correspondence, $\{(\hat{\mathbf{h}}_i, \hat{\mathbf{r}}_i)\}_{i=1,\cdots,N}$, extracted in this way is treated as landmarks to be used in our framework.

### C. ELM Based Kernels

As the configuration pairs of marker data-set are discrete in space, we define a kernel $\kappa(\cdot)$ on each marker configuration $\hat{\mathbf{h}}_i$ and $\hat{\mathbf{r}}_i$ as a local spatial descriptor using the technique of *Extreme Learning Machine* (ELM) [23]. ELM method has been widely used in regression and classification problems as a *single hidden layer feed-forward network* (SLFN) with its advantageous properties of fast training speed, tuning-free neurons and easiness in implementation (ref. [24]). Basically, the training formula of ELM can be expressed as $\mathbf{Hb} = \mathbf{T}$, where $\mathbf{H}$ is the hidden layer output matrix of SLFN, $\mathbf{b}$

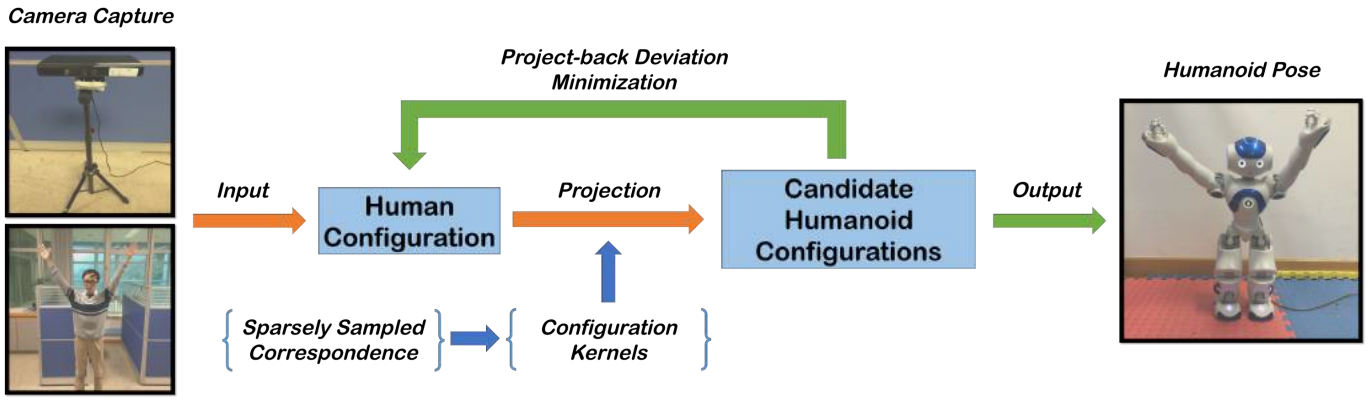


Fig. 2. An illustration of our framework for motion imitation using configuration projection.

is the output weight vector to be computed, and $\mathbf{T}$ is the target feature vector.

Given a new input $\mathbf{x}$, the prediction function of ELM is $\mathbf{f}(\mathbf{x}) = \mathbf{Q}(\mathbf{x})\mathbf{b}$, where the $\mathbf{Q}(\mathbf{x})$ is the hidden layer feature mapping of $\mathbf{x}$. It has been pointed out in [23] that the training errors will be eliminated if the number of hidden nodes is not less than the number of training samples, indicating the trained ELM can be used as a fitting function that interpolates all training samples

$$\mathbf{Q}(\hat{\mathbf{h}}_i)\mathbf{b} = \hat{\mathbf{r}}_i, \quad (i = 1, \cdots, N).$$

In this case, the output weight vector is computed as

$$\mathbf{b} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{T},$$

where $\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}$ is the Moore-Penrose generalized inverse of $\mathbf{H}$. Regularized ELM is proposed in [25] to improve its numerical stability, leading to the following training formula with $\lambda$ (a very small value in practice) as the regularization factor

$$\mathbf{b} = \mathbf{H}^T(\lambda + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{T}.$$

With the help of ELM, a kernel $\kappa_i^h(\cdot) \in \mathbb{R}^n$ for a human's landmark point $\hat{\mathbf{h}}_i$ can be built with its nearest neighbors. Specifically, we find $k$ spatial nearest neighbors of $\hat{\mathbf{h}}_i$ in the set of human's landmarks as $\{\hat{\mathbf{h}}_j\}_{j\in\mathcal{N}(\hat{\mathbf{h}}_i)}$, where $\mathcal{N}(\cdot)$ denotes the set of nearest neighbors. Then, the ELM kernel of $\kappa_i^h(\cdot)$ is trained using $\{(\hat{\mathbf{h}}_j, \hat{\mathbf{r}}_j)\}_{j\in\mathcal{N}(\hat{\mathbf{h}}_i)}$, which is regarded as an approximate local descriptor of the nearby mapping of $\hat{\mathbf{h}}_i$: $\mathcal{H} \mapsto \mathcal{R}$. When inputting a new human pose $\mathbf{h}^* \in \mathbb{R}^m$, a local estimation of mapping with reference to this kernel can be represented as

$$\kappa_i^h(\mathbf{h}^*) = \mathbf{Q}(\mathbf{h}^*)\mathbf{b}.$$

This function is called a *forward* kernel. Similarly, for each $C$-point ${r^m}_i$ of a humanoid, an ELM based kernel $\kappa_i^r(\cdot) \in \mathbb{R}^m$ can be constructed in the same way for the inverse mapping: $\mathcal{R} \mapsto \mathcal{H}$. $\kappa_i^r(\cdot)$ is called a *backward* kernel. These two types of kernel functions will be used in our framework for realizing the projection.

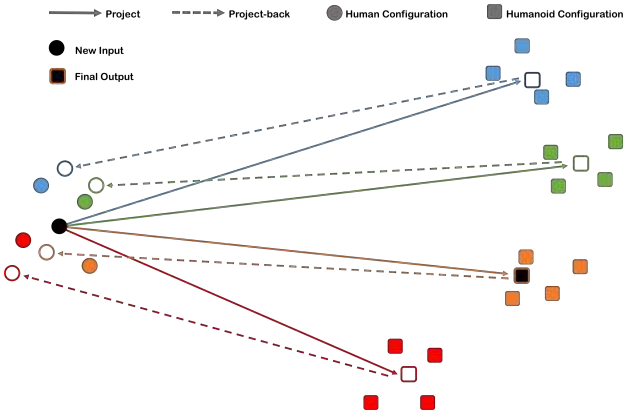


Fig. 3. An illustration of finding an optimal point that minimizes a back-projected deviation (with $L = M = 4$).

### D. Projection

For an input pose $\mathbf{h}^* \in \mathbb{R}^m$, the point determined by the ELM kernel function, $\kappa_i^h(\mathbf{h}^*)$, is not guaranteed to satisfy the requirement of bijective mapping (i.e., $\kappa_i^r(\kappa_i^h(\mathbf{h}^*)) \neq \mathbf{h}^*$). To improve the bijection of mapping, the projection of a human's $C$-point is formulated as determining an optimal point from all candidates generated from different forward kernels.

First of all, $L$ nearest neighbors of $\mathbf{h}^*$ are retrieved in $\mathcal{H}$ as $\{\hat{\mathbf{h}}_j\}$ $(j = 1, \cdots, L)$. From the forward kernel associated with each of these $L$ points in $\mathcal{H}$, a candidate point in $\mathcal{R}$ can be determined by $\mathbf{r}_j^c = \kappa_j^h(\mathbf{h}^*)$. For each $\mathbf{r}_j^c$, we search for its $M$ nearest neighbors in $\mathcal{R}$ as $\mathcal{N}(\mathbf{r}_j^c) = \{\hat{\mathbf{r}}_{j,k}\}$ $(k = 1, \cdots, M)$. In other words, there are $M$ backward kernels associated with $\mathbf{r}_j^c$, which are $\{\kappa_{j,k}^r\}$. In each cluster of backward kernels, we determine a set of weights $w_{j,k}$ that leads to a point

formed as the convex combination of $\{\hat{\mathbf{r}}_{j,k}\}$

$$\tilde{\mathbf{r}}_j^c = \sum_k w_{j,k}\mathbf{r}_{j,k}^c.$$

An optimal point $\tilde{\mathbf{r}}_j^c$ minimizes the deviation of back-projection with regard to the cluster of kernels $\{\kappa_{j,k}^r(\cdot)\}_k$ is defined as

$$\begin{aligned} &\min_{w_{j,k}}\{\|\kappa_{j,k}^r(\sum_k w_{j,k}\mathbf{r}_{j,k}^c) - \mathbf{h}^*\|\}_k, \\ s.t. \quad &\sum_{k=1}^{M} w_{j,k} = 1, \quad w_{j,k} \geq 0. \end{aligned} \tag{1}$$

The final projected point $\mathbf{r}^*$ is then defined as

$$\mathbf{r}^* = \sum_k w_{l,k}\mathbf{r}_{l,k}^c \tag{2}$$

according to the cluster of $\mathcal{N}(\mathbf{r}_l^c)$ that gives the minimal back-projected deviation, which is a solution of

$$\begin{aligned} &\min_j \left\{ \min_{w_{j,k}}\{\|\kappa_{j,k}^r(\sum_k w_{j,k}\mathbf{r}_{j,k}^c) - \mathbf{h}^*\|\}_k \right\}, \\ s.t. \quad &\sum_{k=1}^{M} w_{j,k} = 1, \quad w_{j,k} \geq 0. \end{aligned} \tag{3}$$

The computation for solving above optimization problem can be slow in many cases. Therefore, we propose a sub-optimal objective function as a relaxation of Eq.(3) to be used in real-time applications (e.g., the tele-operation shown in Fig.1). The problem is relaxed to

$$\min_j \left\{ \min_k \{\|\kappa_{j,k}^r(\mathbf{r}_j^c) - \mathbf{h}^*\|\}_k \right\}, \tag{4}$$

the solution of which can be acquired very efficiently by checking each candidate $\mathbf{r}_j^c$ with regard to all its $M$ reference backward kernels. Figure 3 gives an illustration for the evaluation of back-projected deviation.

**Motion Smoothing:** A dynamic motion is processed as a sequence of continuous poses in our system, where the projected poses in the configuration space of humanoid are generated separately. To avoid the generation of jerky motion, we use a method modified from the double exponential smoothing [26] to post-process the projected poses. Given a projected pose $\mathbf{r}_t$ at time frame $t$, the update rules of a smoothed pose $\mathbf{s}_t$ are defined as

$$\begin{aligned} &\mathbf{s}_t = \alpha y_t + (1-\alpha)(\mathbf{s}_{t-1} + \mathbf{b}_{t-1}), \quad 0 \leq \alpha \leq 1 \\ &\mathbf{b}_t = \gamma(\mathbf{s}_t - \mathbf{s}_{t-1}) + (1-\gamma)\mathbf{b}_{t-1}, \quad 0 \leq \gamma \leq 1 \\ &\mathbf{s}_t = \mathbf{s}_{t-1}, \text{ if } \|\mathbf{s}_t - \mathbf{s}_{t-1}\| < \eta \end{aligned} \tag{5}$$

$\alpha$, $\gamma$ and $\eta$ are parameters to control the effectiveness of smoothing, where $\alpha = 0.75$, $\gamma = 0.3$ and $\eta = 0.15$ are used to give satisfactory results in our practice.

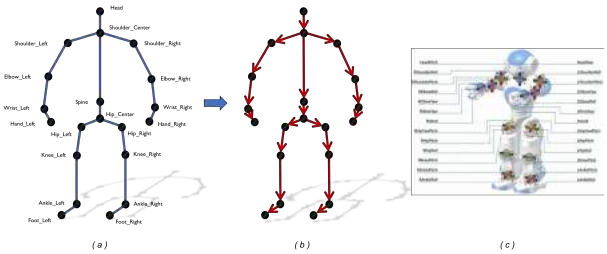


Fig. 4. Feature vectors of human and humanoid: (a) the human skeleton from a Kinect sensor, (b) the corresponding pose descriptor of a human body consists of 19 unit vectors, and (c) the pose descriptor for a NAO humanoid formed by all DOFs on its joints (source: http//:www.ez-robot.com).

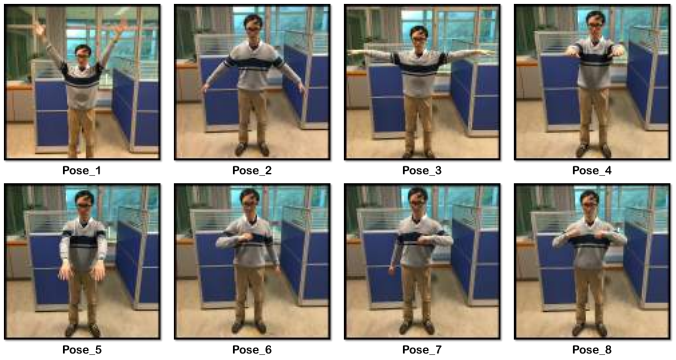


Fig. 5. Basic poses serve as benchmarks for similarity evaluation.

**Remark:** It must be clarified the *Identity* property introduced in Section II-A is relaxed to $\Omega(h_i) \approx r_i$ in practice due to the following reasons:

- Regularized ELM method is employed to construct the kernels, which changes the corresponding energy function where a regularization term is added to improve its numerical stability.
- Double exponential smoothing is applied for smoothing a motion, which introduces minor adjustments on the output values.

## III. Real-time Projection on Humanoids

Our framework is testified on real-time motion imitation of humanoids with a Kinect RGB-D camera as the device to capture the motion of human. The numerical tests are taken on a NAO humanoid robot as a benchmark, and it is also applied to a lab-made Poppy humanoid [27] for further verification.

### A. Human-to-humanoid Motion Imitation

The human skeleton provided by a Kinect sensor is a set of line segments based on predefined key joints as shown in Fig.4(a). We define an abstraction consisting of 19 unit vectors for a pose as illustrated in Fig.4(b), which

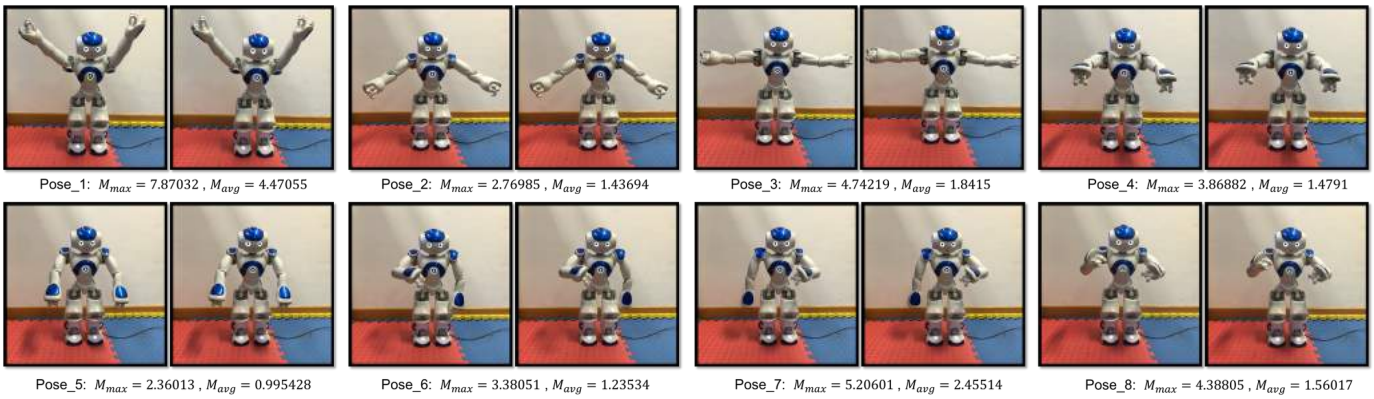


Fig. 6. Eight basic poses are reconstructed by our method (left of each pair) and compared with the ground truth (right of each pair). The similarity metrics, $M_{max}$ and $M_{avg}$, of each pair are also reported. The evaluation is taken on a projection defined by using $1,644$ landmark pairs.

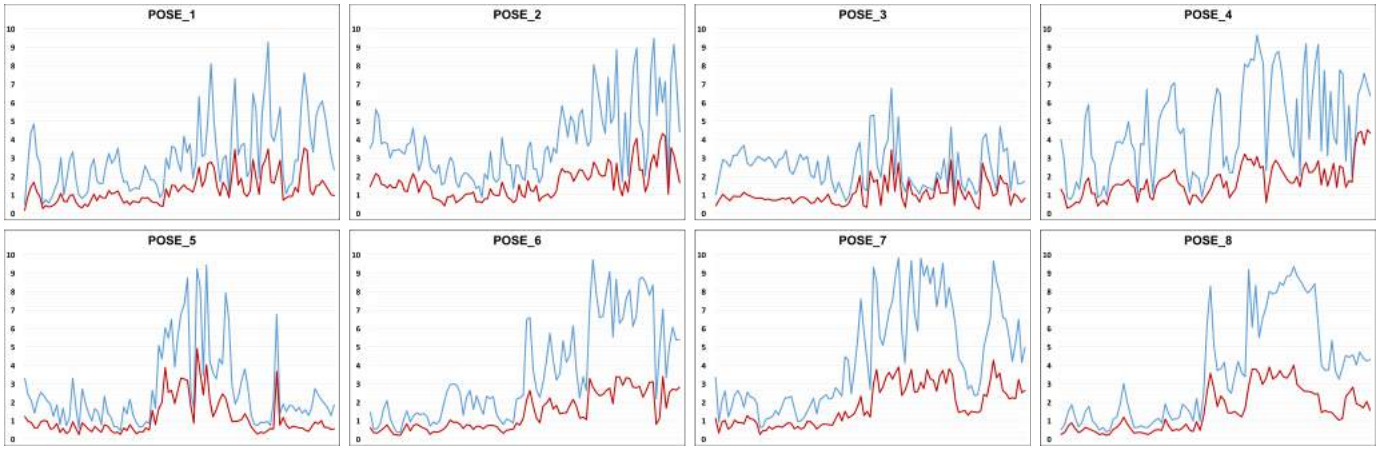


Fig. 7. Statistics in eight motions for the change of two metrics in degree: $M_{max}$ (blue) and $M_{avg}$ (red). The evaluation is also taken on a projection with $1,644$ landmark pairs.

is independent different body dimensions. It should be pointed out that it is unnecessary to always use the full set of unit vectors unless full body motion must be sensed. The NAO humanoid robot has 26 degree-of-freedoms, including the roll, pitch, and yaw of all its joints (see Fig.4(c)). Posing a NAO humanoid can be executed by specifying the values of all its degree-of-freedoms.

To collect the data-set of human's motion, a user is asked to do arbitrary motion in front of a Kinect camera. Meanwhile, we have implemented a straightforward *inverse kinematics* (IK) based scheme for upper-body motion on NAO. The roll, pitch, and yaw of every joint can be computed directly by the unit vectors of a human's skeleton model. After using mean shift to extract the landmarks of motion from raw data set, their corresponding landmark poses in the $C$-space of humanoid can be generated by this IK. Besides, we also define *eight* basic poses (see Fig.5) which play a critical role when evaluating the similarity between projected poses of humanoid and poses of human.

Using the landmark poses defined in this way, human-to-humanoid motion imitation has been implemented by a single-core C++ program. All the tests below are taken on a personal computer with Intel Core i7-3770 3.4GHz and 8GB RAM memory.

### B. Experimental Results

We evaluate our method mainly from three perspectives, including the computational efficiency of projection, the quality of reconstructed motion, and the influence by the size of landmark set. All are tested on the platform of NAO humanoid.

**Efficiency of Projection:** From Section II-D, we know that the complexity for computing projection depends on the size of neighbors (i.e., $L$ and $M$). The cost of computation increases with larger $L$ and $M$ as more

candidates and more reference kernels will be involved. In all our experiments, we use $L = M = 10$ and the average time for making a configuration projection is $0.00273ms$. When increasing to $L = M = 50$, the average time cost is still only $0.0201ms$. By contrast, the rough time cost of offline optimization based techniques (e.g., [1]–[6]) for each computational step ranges from tens of milliseconds to several seconds. Online methods generally require at least several milliseconds to compute each status update as reported in [7]–[12]. Comparatively, the overhead of our method for motion imitation is very light – i.e., it fits well for different real-time applications.

**Quality of Reconstruction:** Two metrics are used in our experiments to estimate the quality of a projected configuration $\mathbf{r}^* \in \mathbb{R}^n$ referring to its corresponding ground truth value $\mathbf{r}_{gt}$ – the maximum absolute deviation in degree as

$$M_{max} = \frac{180^\circ}{\pi}\|r^* - r_{gt}\|_\infty,$$

and the average absolute deviation in degree as

$$M_{avg} = \frac{1}{n}\left(\frac{180^\circ}{\pi}\|r^* - r_{gt}\|_1\right).$$

The evaluation is taken with a set holding $1,644$ configuration pairs as landmarks. All those eight poses shown in Fig.5 are tested, and the results are shown in Fig.6. The results of comparison (in terms of $M_{max}$ and $M_{avg}$) indicates that the poses generated by our method share good similarity with the ground truths. Besides of static poses, we also evaluate the quality of reconstructed motion in the $C$-space of humanoid as a sequence of poses. We define eight basic motion sequences, each of which starts from the rest pose and ends at one of the basic poses. The complete human motions are recorded for the reconstruction using our projection in the $C$-space of humanoid. The projected poses are compared with the poses generated by IK, serving as the ground truths. The values of $M_{max}$ and $M_{avg}$ in these eight motions are shown in Fig.7. It is easy to find that the errors are bounded to less than $10^\circ$ in all motions.

**Size of Landmarks:** As presented in Section II-B, the correspondence samples used to formulate projection in our framework is extracted from the captured motions. In our implementation, it is generated by a user moving in front of a Kinect sensor for 5 minutes. Then, three sets with different number of landmarks ($1,644$, $961$, and $86$ respectively) are extracted. The corresponding pairs of poses are then constructed with the help of IK. The 8-th pose in Fig.5 – POSE_8 and the motion from the rest pose to POSE_8 are constructed from the projections

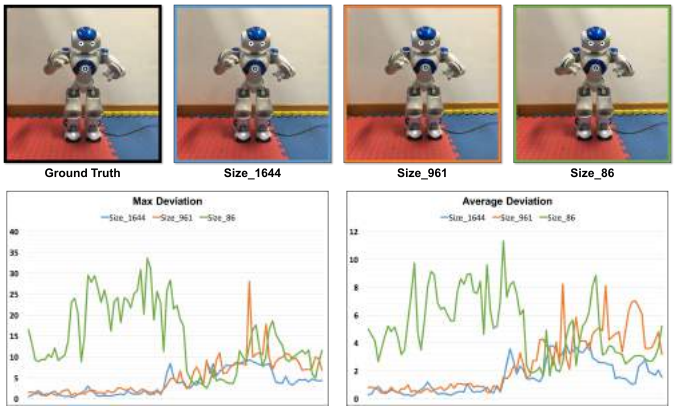


Fig. 8. To reconstruct motion using landmark sets with different number of corresponding samples, statistics of $M_{max}$ and $M_{avg}$ in degree indicate that more landmark pairs lead to better results.

defined on the sets with different number of landmarks. From the statistics and comparisons shown in Fig.8, it is easy to conclude that our projection based formulation converges when the number of landmarks increases. In other words, more landmarks result in a more accurate projection. However, it should also be noted that the projection from the smaller set may still be useful in some applications with low requirement on quality but having more restrictions on speed and memory usage.

### C. Application of Tele-operation

We have tested the motion imitation realized by our method in an application of tele-operation using a NAO humanoid. As illustrated in Fig.1, a user can remotely control the motion of a NAO robot to grasp an object and put it into a box. The scene that can be seen from the camera of NAO is displayed on a screen placed in front of the user as the visual feedback. The imitation realized by our system has good accuracy. As a result, the tele-operation can be performed very smoothly. Another operation is also demonstrated in our supplementary video that an object can be lifted up by the NAO robot through the tele-operation setup based on our approach (see also Fig.9). Functionality of the proposed approach can also be observed from our supplementary video [28] – `https://youtu.be/ok3uFYFEU0I`.

### D. Imitation with Multi-robots

The experiment is extended to further apply the proposed method to a lab-made Poppy humanoid [27], which has 23 degree-of-freedoms. Similar to realizing the imitation on NAO, an IK program is implemented here for whole body motion on the Poppy humanoid. Then the correspondence between specific poses of human and Poppy can be aligned with the help of this

Fig. 9. Application of tele-operation using NAO humanoid: (left) picking up a ring and putting it into a box, and (right) lifting up a poster by two hands.

Fig. 10. Tests on the lab-made Poppy humanoid: (left) full body motion and (right) simultaneous imitation in the heterogeneous environment with a NAO and a Poppy.

IK. The ELM based configuration projection is applied with the sparsely aligned correspondence. Full human body motion has been tested with our setup (see the left of Fig.10). Moreover, we also set up a heterogeneous environment with a NAO and a Poppy, and conduct imitation on them simultaneously (see the right of Fig.10). Note that the NAO humanoid and the Poppy have different degree-of-freedoms, but the imitation with high similarity can be achieved in real time according to our supplementary video [28].

## IV. Conclusion & Discussion

In this paper, we have proposed a framework to realize motion imitation. Different from conventional methods, our method is based on a novel formulation of projection between two configuration spaces with different dimensions. Given a new input pose of human, its projection in the configuration space of humanoid is defined as finding the optimal $C$-point that minimizes a back-projection deviation referring to pre-built kernels. We have validated our idea by reconstructing humanoid motion on a NAO robot and a lab-made Poppy robot. The experimental results are encouraging and motions of good quality can be realized very efficiently.

There are several potential improvements that can be made to our method. The ELM based kernels currently used in our framework do not have an explicit bound for prediction with a new input. Finding kernel functions that can provide a numerical bound on prediction could be an interesting future work. Besides, we are also interested in exploring more applications beyond tele-operation. From the aspect of input devices, a single RGB-D camera is used in our experimental tests. It will be interesting to see how the performance of tele-operation can be improved if a more precise input can be provided (e.g., the dual RGB-D camera system [29]). Vision based motion capture system is always limited by the lighting condition as well as its portability. Another possible future work is to take a direct mapping from measured angular information (e.g., the gyroscope based sensors [30]) to the motion on robots.

## Acknowledgment

This work is supported by Hong Kong ITC Innovation and Technology Fund (ITS/065/14), and Chengkai Dai is also partially supported by Hong Kong RGC General Research Fund (CUHK/14207414). C.C.L. Wang would like to acknowledge the support from the Open Research Fund of Key Laboratory of High Performance Complex Manufacturing at Central South University, China.

## References

[1] W. Suleiman, E. Yoshida, F. Kanehiro, J.-P. Laumond, and A. Monin, "On human motion imitation by humanoid robot," in *IEEE International Conference on Robotics and Automation*, 2008, pp. 2697–2704.

[2] R. Chalodhorn, D. B. Grimes, K. Grochow, and R. P. Rao, "Learning to walk through imitation," in *IJCAI*, vol. 7, 2007, pp. 2084–2090.

[3] S. Nakaoka, A. Nakazawa, F. Kanehiro, K. Kaneko, M. Morisawa, H. Hirukawa, and K. Ikeuchi, "Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances," *The International Journal of Robotics Research*, vol. 26, no. 8, pp. 829–844, 2007.

[4] A. Ude, C. G. Atkeson, and M. Riley, "Programming full-body movements for humanoid robots by observation," *Robotics and autonomous systems*, vol. 47, no. 2, pp. 93–108, 2004.

[5] S. Kim, C. Kim, B. You, and S. Oh, "Stable whole-body motion generation for humanoid robots to imitate human motions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 2518–2524.

[6] A. Safonova, N. Pollard, and J. K. Hodgins, "Optimizing human motion for the control of a humanoid robot," *Proceedings of Applied Mathematics and Applications of Mathematics*, vol. 78, 2003.

[7] C. Ott, D. Lee, and Y. Nakamura, "Motion capture based human motion recognition and imitation by direct marker control," in *8th IEEE-RAS International Conference on Humanoid Robots*, 2008, pp. 399–405.

[8] B. Dariush, M. Gienger, A. Arumbakkam, Y. Zhu, B. Jian, K. Fujimura, and C. Goerick, "Online transfer of human motion to humanoids," *International Journal of Humanoid Robotics*, vol. 6, no. 02, pp. 265–289, 2009.

[9] M. Do, P. Azad, T. Asfour, and R. Dillmann, "Imitation of human motion on a humanoid robot using non-linear optimization," in *8th IEEE-RAS International Conference on Humanoid Robots*, 2008, pp. 545–552.

[10] K. Yamane, S. O. Anderson, and J. K. Hodgins, "Controlling humanoid robots with human motion data: Experimental validation," in *10th IEEE-RAS International Conference on Humanoid Robots*, 2010, pp. 504–510.

[11] J. Koenemann and M. Bennewitz, "Whole-body imitation of human motions with a NAO humanoid," in *7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2012, pp. 425–425.

[12] J. Koenemann, F. Burget, and M. Bennewitz, "Real-time imitation of human whole-body motions by humanoids," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2806–2812.

[13] A. S. Morris and A. Mansor, "Finding the inverse kinematics of manipulator arm using artificial neural network with lookup table," *Robotica*, vol. 15, no. 06, pp. 617–625, 1997.

[14] J. Aleotti, A. Skoglund, and T. Duckett, "Position teaching of a robot arm by demonstration with a wearable input device," in *International Conference on Intelligent Manipulation and Grasping (IMG04)*, 2004.

[15] P. Neto, J. N. Pires, and A. P. Moreira, "Accelerometer-based control of an industrial robotic arm," in *The 18th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2009, pp. 1192–1197.

[16] P. Neto, J. Norberto Pires, and A. Paulo Moreira, "High-level programming and control for industrial robotics: using a hand-held accelerometer-based input device for gesture and posture recognition," *Industrial Robot: An International Journal*, vol. 37, no. 2, pp. 137–147, 2010.

[17] C. Stanton, A. Bogdanovych, and E. Ratanasena, "Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning," in *Proceedings of Australasian Conference on Robotics and Automation*, 2012.

[18] P. Van der Smagt and K. Schulten, "Control of pneumatic robot arm dynamics by a neural network," in *Proc. of the 1993 World Congress on Neural Networks*, vol. 3, pp. 180–183.

[19] S. Jung and T. Hsia, "Neural network reference compensation technique for position control of robot manipulators," in *IEEE International Conference on Neural Networks*, vol. 3, 1996, pp. 1765–1770.

[20] J. C. Larsen and N. J. Ferrier, "A case study in vision based neural network training for control of a planar, large deflection, flexible robot manipulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 2004, pp. 2924–2929.

[21] D. Wang and Y. Bai, "Improving position accuracy of robot manipulators using neural networks," in *Proceedings of the IEEE Instrumentation and Measurement Technology Conference*, vol. 2, 2005, pp. 1524–1526.

[22] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[23] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[24] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.

[25] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine," in *IEEE Symposium on Computational Intelligence and Data Mining*, 2009, pp. 389–395.

[26] J. J. LaViola, "Double exponential smoothing: an alternative to kalman filter-based predictive tracking," in *Proceedings of the workshop on Virtual environments 2003*. ACM, 2003, pp. 199–206.

[27] "Open source platform for the creation, use and sharing of interactive 3D printed robots," https://www.poppy-project.org.

[28] S. Jin, C. Dai, Y. Liu, and C. C. L. Wang, "Motion imitation based on sparsely sampled correspondence," Supplementary Video, https://youtu.be/ok3uFYFEU0I.

[29] K.-Y. Yeung, T.-H. Kwok, and C. C. Wang, "Improved skeleton tracking by duplex kinects: A practical approach for real-time applications," *Journal of Computing and Information Science in Engineering*, vol. 13, no. 4, p. 041007, 2013.

[30] Y. Zheng, K. C. Chan, and C. C. L. Wang, "Pedalvatar: An imu-based real-time body motion capture system using foot rooted kinematic model," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 4130–4135.