



HAL
open science

Optimization-based Motion Retargeting Integrating Spatial and Dynamic Constraints

Thomas Moulard, Eiichi Yoshida, Shin'Ichiro Nakaoka

► **To cite this version:**

Thomas Moulard, Eiichi Yoshida, Shin'Ichiro Nakaoka. Optimization-based Motion Retargeting Integrating Spatial and Dynamic Constraints. The 44th Symposium on Robotics, Oct 2013, Seoul, South Korea. hal-00908568

HAL Id: hal-00908568

<https://hal.science/hal-00908568>

Submitted on 25 Nov 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimization-based Motion Retargeting Integrating Spatial and Dynamic Constraints

Thomas Moulard¹, Eiichi Yoshida¹ and Shin'ichiro Nakaoka²

Abstract—In this paper, we present an optimization-based retargeting method for precise reproduction of captured human motions by a humanoid robot. We take into account two important aspects of retargeting simultaneously: spatial relationship and robot dynamics model. The former takes care of the spatial relationship between the body parts based on “interaction mesh” to follow the human motion in a natural manner, whereas the latter adapts the resulting motion in such a way that the dynamic constraints such as torque limit or dynamic balance are being satisfied. We have integrated the interaction mesh and the dynamic constraints in a unified optimization framework, which is advantageous for generation of natural motions by a humanoid robot compared to previous work that performs those processes separately. We have validated the basic effectiveness of the proposed method with a sequence of postures converted from captured human data to a humanoid robot.

I. INTRODUCTION

One of the advantages of human-size humanoid robots is its ability to generate whole-body motions maintaining similar dynamics to humans. This ability allows a humanoid robot to serve as an entertainer like a dancer of an actor [1], or also to use various machines and devices designed for humans [2]. As an extension of the latter use, a new application has recently been studied: a humanoid robot as an evaluator of human assistive devices [3], [4]. If a humanoid reproduces human motions faithfully, it can be used to test the devices instead of human subjects. This brings several benefits such as no need for ethical process, repeated test with exactly the same motions under the same conditions, and qualitative evaluation through sensory measurement like torque and force. It has been demonstrated that the human-size humanoid HRP-4C [5] can evaluate the effect of load reduction quantitatively by estimating motor torque [4], taking an example of a supportive wear called “Smart Suit Lite” [6] designed to reduce load at the lower back with embedded elastic bands.

The important issue in those applications is how to generate natural motions of a humanoid robot. There have been a number of studies on “motion retargeting” techniques in order to generate humanoid motions based on those of a human measured by a motion capture system. Retargeting captured motion to humanoids has been actively studied during the last decade, thanks to the progress of their dynamic capability. The work of Pollard [7] is one of the pioneering studies that

enable reproduction of human motions by a humanoid, in this case the upper body of Sarcos humanoid robot, by taking into account various constraints. Nakaoka et al. developed a technique to transfer Japanese traditional dancing motions to a humanoid by introducing a notion of leg task model [8], [9]. Miura et al. [10] devised a walking pattern generator that allows the humanoid robot HRP-4C to walk in a manner extremely close to humans, including stretched knees, swing-leg trajectory and single support phase on the toe. Other imitation methods have been proposed based on a dynamic controller [11], [12], motion recognition and primitives [13] and extraction of upper-body motion from markerless motion input [14], [15].

On the other hand, motion retargeting has been investigated intensively in computer graphics domain, typically to generate motions for new characters based on motion capture data using space-time constraints solver [16]. Recently, Ho et al. proposed a new retargeting method called “interaction mesh” that preserves the spatial relationship between closely interacting body parts and objects in the environments [17]. Nakaoka and Komura extended this method for retargeting to a humanoid robot by taking advantage of its capacity to adapt motions to a character with highly different physical properties [18]. Usage of interaction mesh brings natural following of original human motion and self-collision avoidance. Although this method includes balance consideration by shifting the waist, this approach remains specific instead of general whole-body motion optimization and does not deal with dynamic constraints such as torque limits. In addition, those constraints are treated separately after generating re-targeted motion to adapt to the humanoid.

Another related domain is the optimization technique that is more and more employed to generate robot trajectories minimizing certain cost function under mechanical or dynamic constraints. Miossec et al. applied nonlinear optimization to dynamic whole-body motion like a kicking motion of a human-size humanoid [19]. Recently the optimization is utilized for generation of multi-contact dynamic motion through modeling of dynamic constraints using Taylor expansion [20]. For dynamic trajectory optimization for digital human, an efficient algorithm has been proposed [21], [22]. Suleiman et al. proposed another trajectory optimization technique based on Lie algebra that allows efficient computation through analytic integration of dynamics [23] and applied it to human motion imitation [24]. The latter research aims at optimizing the humanoid trajectory to be as close as human motion, but self-collision avoidance is incorporated as a post-processing to the optimized motion like previous work

*This work was not supported by any organization

¹T. Moulard and E. Yoshida are with CNRS-AIST JRL (Joint Robotics Laboratory), UMI3218/CRT, and ¹S. Nakaoka are with Humanoid Research Group, both National Institute of Advanced Industrial Science and Technology (AIST) Tsukuba Central 2, 1-1-1 Umezono, Ibaraki 305-8568 Japan {thomas.moulard, e.yoshida, s.nakaoka}@aist.go.jp

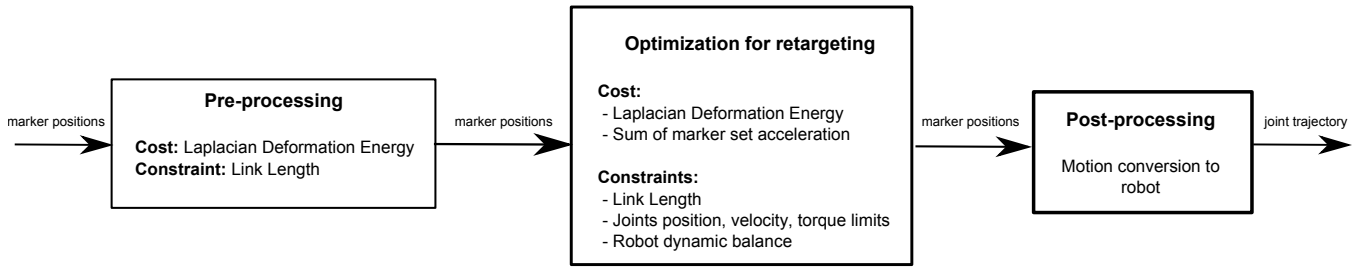


Fig. 1. The two step retargeting process. The final inverse kinematics step is only here to convert the final motion to joints trajectory.

[18]. This is disadvantageous because separate application of collision avoidance may lead to unnatural motions.

Methods in previous research are, therefore, still lacking the capability to optimize humanoid motions by taking into account the associated problems for retargeting in a unified manner. The main contribution of the paper is the optimization process integrating retargeting, dynamics constraints and self-collision at the same time, in order to create the humanoid motion as close to human motions as possible. In this paper, we address this retargeting by formulating it as a nonlinear optimization problem under spatial and dynamic constraints. First, the captured motion is pre-processed to provide better initial guess for the optimization. Then the full optimization problem is solved considering the spatial relationship and dynamic robotic constraints simultaneously. The spatial relationship between body parts of captured motion is reserved by using interaction mesh as introduced by [17], which achieves self-collision avoidance in consequence.

This paper is organized as follows. Section II describes the overall method and in particular how the retargeting is performed. Section III describes the details of each step of the optimization-based retargeting method. Section IV introduces RobOptim, an optimization framework for robotics used to implement the proposed method. Section V presents the results of retargeting with validation of dynamic simulations, before concluding the paper.

II. METHOD OVERVIEW

The proposed retargeting framework consists of three steps as shown in Fig. 1 and takes a time series of captured marker positions to generate a retargeted robot joint trajectory. The main part of the retargeting is the second step, which is the main optimization process whereas the first and last steps are additional data processing for efficient optimization and motion conversion respectively.

The first pre-processing takes care of initial guess for the optimization. It takes care of retargeting problem only to fit the marker positions to the target robot structure, to obtain the initial state that helps the optimization process converge quickly. This geometric problem can be represented as a linear problem using the interaction mesh [17] with a quadratic cost rendering its resolution extremely efficient.

The main optimization process is initialized using the result of the first step and then takes the full problem into consideration. This optimization incorporates robotic

constraints such as motion balance and torque constraints together with the spatial relationship modeled also by interaction mesh. The previous work [18] adopts a two-step approach for this optimization: first the markers were optimized before adapting the motion by optimizing joint angles for the robot in order to satisfy robotic constraints. In contrast, in the proposed method the optimization variables are the marker position throughout the retargeting process until it is converted to joint angles at the last step. This is advantageous to make the constraint matrix sparse [17] and also to maintain a global optimization framework unifying all the constraints simultaneously. Inverse kinematics computation of the target robot is employed in order to estimate the nearest robot configuration from the marker positions at each time step. This allows evaluating robotic constraints such as dynamic balance or joint torque limit by using deduced joint velocity and acceleration. The robotic constraints being non-linear, this main optimization problem is much more computationally intensive.

The third post-processing step is conversion of the resultant marker positions into the robot joint configurations that is done in a straightforward manner using inverse kinematics.

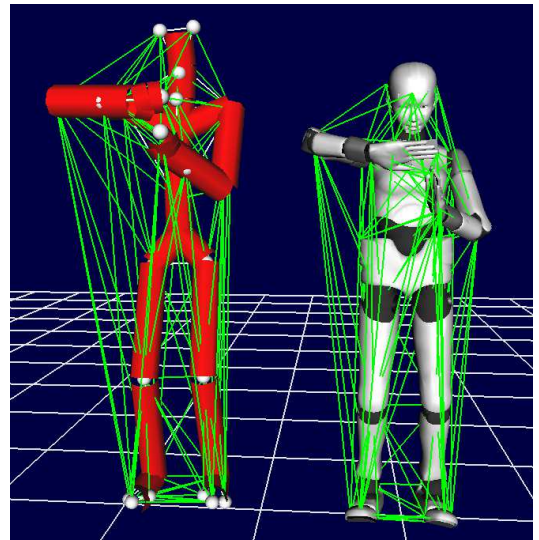


Fig. 2. Examples of interaction mesh representing the spatial relationship between body parts, applied to a digital character and a humanoid [18].

III. OPTIMIZATION-BASED RETARGETING METHOD

As described in the previous section, our retargeting method has two main components: motion retargeting and robotic motion generation that are integrated in the optimization. This section will detail their computational aspects.

A. Motion retargeting

The retargeting algorithm employed in the first two steps in the framework of Fig. 1 relies on the notion of ‘‘interaction mesh’’ to ensure that spatial relationship between bodies is preserved. We will briefly describe how it is incorporated in the optimization based on previous work by Ho et al [17].

By applying Delaunay Tetrahedralization [25] on the marker set, one can generate a mesh which is parameterized by the marker positions $\mathbf{V}_i = (\mathbf{p}_1^i \cdots \mathbf{p}_m^i)$ where $1 \leq i \leq n$. n denotes here the number of frame composing the motion and m the number of markers in each frame. \mathbf{p}_1^i represents the position of the first marker in the i -th frame.

Given a particular interaction mesh, one can compute the ‘‘Laplacian Coordinate’’ of one marker as follows:

$$L(\mathbf{p}_j^i) = \mathbf{p}_i^j - \sum_{l \in N_j} w_l^j \mathbf{p}_l^i \quad (1)$$

In Eq. (1), N_j is the one-hop neighborhood of the marker j in the interaction mesh and w_l^j is the weight of the marker l when computing the Laplacian Coordinate of marker j . This weight is inversely proportional to the distance between j and l .

Considering these two notions, it is possible to introduce the ‘‘Laplacian Deformation Energy’’ associated to a marker set which serve as a cost function in this problem:

$$E_L(\mathbf{V}'_i) = \sum_j \|L(\mathbf{p}_j^i) - L(\mathbf{p}_j^{i'})\|^2 \quad (2)$$

The Laplacian Deformation Energy is the square of the norm of the difference between the Laplacian Coordinates of the original marker set and the updated marker set $\mathbf{V}'_i = (\mathbf{p}_1^i \cdots \mathbf{p}_m^i)$.

In practice, this cost function penalizes motion of highly-connected markers whereas isolated ones will move for a lower cost.

A second cost function is added to the first one to smooth the motion. To achieve this goal, the marker set acceleration energy is considered:

$$E_A(\mathbf{V}'_{i-1}, \mathbf{V}'_i, \mathbf{V}'_{i+1}) = \frac{1}{2} \|\mathbf{V}'_{i-1} - 2\mathbf{V}'_i + \mathbf{V}'_{i+1}\|^2 \quad (3)$$

$\mathbf{V}'_{i-1}, \mathbf{V}'_i, \mathbf{V}'_{i+1}$ being the new marker set position for the frame $i-1, i$ and $i+1$. Acceleration energy is always null for first and last frame.

The final cost function C is by consequence expressed by:

$$C(\mathbf{V}'_{i-1}, \mathbf{V}'_i, \mathbf{V}'_{i+1}) = E_L(\mathbf{V}'_i) + \alpha E_A(\mathbf{V}'_{i-1}, \mathbf{V}'_i, \mathbf{V}'_{i+1}) \quad (4)$$

where α is the weighting coefficient. Additionally, a link length constraint is defined. This constraint aims at retargeting the motion so that it fits the robot morphology. It is defined as follows:

$$\|\mathbf{p}_e^1 - \mathbf{p}_e^2\|^2 = l_e \quad (5)$$

Whenever there is a need for some part of the robot body to stay fixed, an optional positional constraint is also provided through an equality constraint:

$$\mathbf{V}'_i = \mathbf{P}_i \quad (6)$$

With $0 \leq i \leq m$.

The quadratic problem is then solved to generate a new set of marker positions. The goal of the first pre-processing step in Fig. 1 is to obtain the marker positions that fit the robot structure sufficiently well. Of course, the resulting motion may still not be feasible due to physical constraints. The output marker positions from the pre-processing is used as the initial guess for the main optimization process considering robotic constraints and retargeting at the same time, as explained in the next subsection.

B. Robotics motion generation

We here define a non-linear optimization problem using the motion derived with the pre-processing. The same optimization variables, namely the set of marker positions, are used also with the cost function and the constraints introduced in the previous section. By keeping the previous constraints, one can make sure that the good properties ensured by the previous pre-processing are kept during this optimization step. Moreover, the constraint matrix can be kept sparse by using the marker positions as optimization variables [17], which makes the optimization computation efficient, whereas the joint Jacobian matrix for the robot is dense.

To ensure that the motion is feasible by the robot, two additional constraints are added in the main optimization process:

- Dynamic balance constraint.
- Joint angle, velocity and torque constraints.

We here need robot configuration \mathbf{q} from the marker positions to calculate the following constraints with dynamic property like mass and moment of inertia of each link. Since the output marker positions from the pre-processing are close enough to the feasible robot configuration, we can benefit from the joint angle fitting using damped least square method introduced in [18]. Figure 3 shows the kinematic structure of the humanoid HRP-4C [5] the proposed retargeting is applied to. Once the robot configurations are obtained, the following constraints are evaluated on the numerical basis, by using joint velocity and acceleration computed with finite difference.

The first constraint constrains the ZMP (zero moment point) position so that it stays into the robot support polygon:

$$\begin{aligned} x_{ZMP} &= x_G - \frac{\dot{\sigma}_y + m z_G \ddot{x}_G}{m (\ddot{z}_G + g)} \\ y_{ZMP} &= y_G + \frac{\dot{\sigma}_x - m z_G \ddot{y}_G}{m (\ddot{z}_G + g)} \end{aligned} \quad (7)$$

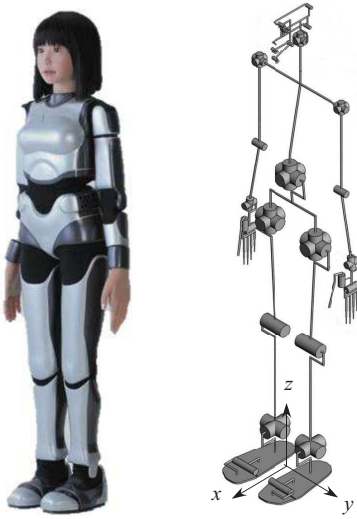


Fig. 3. Humanoid robot HRP-4C and its structure.

where (x_{ZMP}, y_{ZMP}) is the ZMP position on the ground with the coordinate system shown in Fig. 3, $\dot{\sigma}$ the variation of the kinetic momentum around the center of mass, and (x_G, y_G, z_G) the center of mass position. The gravitational constant is denoted by g . Here we assume that the robot is moving on a flat floor. The ZMP acts as a criterion that allows deciding whether a motion can be executed stably or not. As long as it stays inside the convex hull of the robot contact points with the floor, the motion is dynamically stable. Knowing which foot of the robot is in contact with the floor and the foot geometry, it is possible to insert this as an inequality constraint

$$(x_{ZMP}, y_{ZMP}) \subset \mathcal{S}(x_{rfoot}, y_{lfoot}) \quad (8)$$

to make sure that the ZMP is staying inside the current support polygon, denoted by $\mathcal{S}(x_{rfoot}, y_{lfoot})$ that is determined by the right and left positions (x_{rfoot}, y_{lfoot}) .

The second robotic constraint we take into account in this problem is the joint limitations. Given $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$, the set of torques $\tau = (\tau_1, \dots, \tau_o)$ applied to each robot joint can be computed using the classical equation of motion:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \tau \quad (9)$$

where $\mathbf{M}(\mathbf{q})$ is the system mass matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the vector of Coriolis and centrifugal forces and $\mathbf{G}(\mathbf{q})$ the vector of gravitational forces. Other robotic constraints for limits of joint rotation, velocity and torque are expressed as:

$$\begin{aligned} \mathbf{q}_{min} &\leq \mathbf{q} \leq \mathbf{q}_{max} \\ \mathbf{d}\mathbf{q}_{min} &\leq \dot{\mathbf{q}} \leq \mathbf{d}\mathbf{q}_{max} \\ \tau_{min} &\leq \tau \leq \tau_{max} \end{aligned} \quad (10)$$

As mentioned earlier, the output of the main optimization process is the set of marker positions. As we have already applied the robotic constraints, the resultant motion can be easily converted to the joint trajectory to be executed by the robot through inverse kinematics.

IV. IMPLEMENTATION OF THE OPTIMIZATION PROBLEM USING ROBOPTIM

RobOptim is a general framework to assist the development and resolution of optimization problems applied to robotics. The optimization is more and more applied to robotics field to solve complex problems of motion planning and generation with many constraints, as mentioned earlier. Although number of state-of-the-art optimization solver tools and libraries are now available, they are not necessarily ready for immediate use for robotics.

RobOptim has been developed to allow roboticists to prototype their optimization applications easily by providing necessary interfaces specific to robotic problems in the form of C++ libraries. It has a three-layer architecture: the core, the solver and the application layers. The core layer provides a way to define mathematical function and their associated derivatives, while the solver layer encapsulates different state-of-the-art solvers so that they can solve problems defined using the representation proposed by the core layer. The application layer contains dedicated mathematical functions which can be embedded into different optimization problems. An overview of the framework architecture is shown in Fig. (4).

The core layer offers some useful higher-level tools that help users define the functions such as costs and constraints introduced in the previous section. By implementing those functions inherited from the basic mathematical functions of the core layer, RobOptim ensures the compatibility with a number of state-of-the-art solvers whose plug-ins are provided in the solver layer. Those tools include function definition itself and also differentiation of the functions that can be computed analytically, or numerically when no analytical gradient is provided by the user. Even if the gradient is provided, numerical differentiation can be used to ensure the computation correctness. The user can benefit from the transparency of those higher level tools to prototype their problem without thinking about individual solvers used

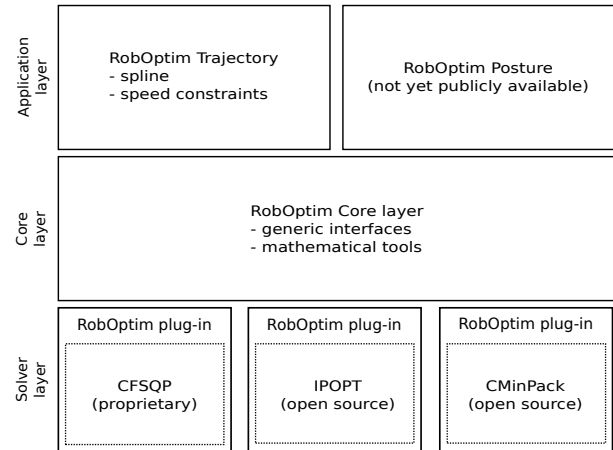


Fig. 4. RobOptim architecture.

for optimization.

Although not really utilized in our problem of retargeting, the application layer is convenient especially for planning purpose. One of the practical tools is the trajectory toolbox, which allows representation of robot motion using B-spline. Since these are associated with mathematical functions of the core layer and others like of minimal-time optimization.

RobOptim is distributed as an open-source library (LGPL-3) through its website: <http://www.roboptim.net/>

V. RETARGETING RESULTS

The proposed algorithm has been applied to a whole body motion taken from CMU motion capture database [26]. We have integrated a RobOptim plug-in for nonlinear optimization tools of NAG optimization library [27] to solve the problem while using sparse matrices computation. This is particularly useful for trajectory optimization where large matrices are involved as some constraints are only considering one frame and thus associated Jacobian are containing a large proportion of null values.

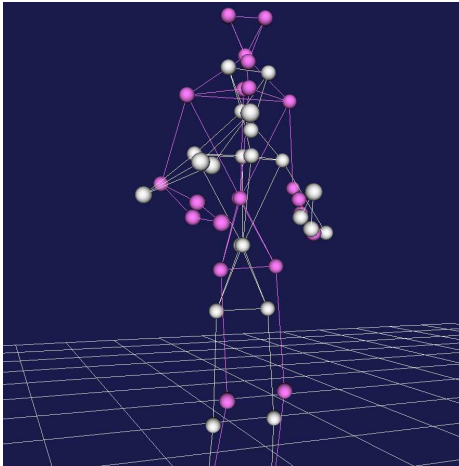


Fig. 5. The marker set before (purple markers) and after (white markers) the initial preprocessing phase.

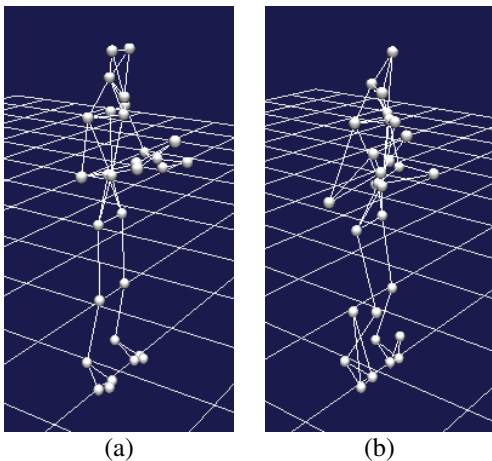


Fig. 6. Optimized posture for HRP-4C before (a) and after (b) the optimization.

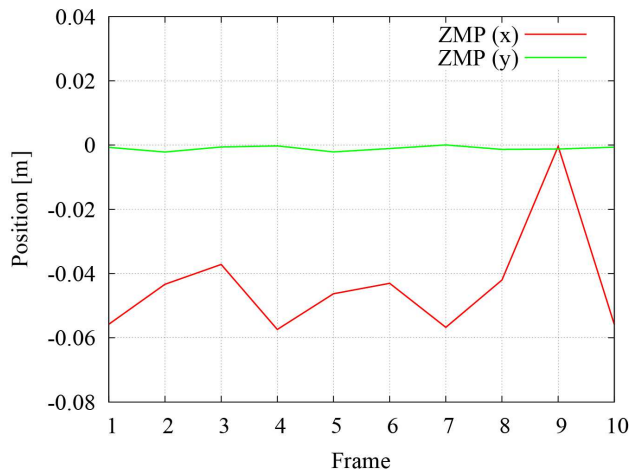


Fig. 7. ZMP values during the motion sequence of 10 frames.

In order to validate its basic capacity of dynamic constraint consideration, we have applied the proposed method to a sequence of several human postures to be converted into stable humanoid configurations. In this validation we use the Laplacian Deformation Energy E_L in Eq. (2) only with $\alpha = 0$ in Eq. (4), with dynamic balance constraint with ZMP of Eq. (7). The support polygon $\mathcal{S}(x_{rfoot}, y_{lfoot})$ for the stability is the square of 15 cm around the center of feet.

The result after the optimization process is illustrated by Fig. 5 (pre-processing) and Fig. 6 (final result). In Fig. 5 we can observe that the markers positions are displaced to fit to the robot whose size is much smaller than the digital character. In the optimized posture in Fig. 6(b), the configuration is modified so that robotic constraints such as joint limits or link length can be satisfied. We also verified the dynamic balance during the motion sequence by computing the resultant ZMP as shown in Fig. 7. As can be seen, the ZMP stays inside the support polygon of the area ± 7.5 cm along each x and y axis as specified in the constraints. Although the profile in Fig. 7 is a bit shaky as we have not yet integrated Acceleration Energy E_A in Eq. (3), the balance constraint itself is satisfied. We therefore believe that more smooth trajectory will be obtained as the optimized trajectory by taking account Acceleration Energy.

VI. CONCLUSION

This paper presented a unified approach combining retargeting and robotics constraints into one single nonlinear optimization problem. For efficient computation, a three-step approach is adopted including pre- and post-processing of the motion. After obtaining pre-processed marker positions approximately fitted to the robot structure, the main optimization process generates also marker motions that respect the original spatial relationship of body parts as much as possible based on interaction mesh, by satisfying the robotic dynamic constraints throughout the motion. The optimization output motion can be converted to the robot trajectory in a straightforward manner by the last step. The fundamental

effectiveness of the proposed method has been validated by converting a sequence of human postures into humanoid configurations by minimizing the cost of spatial relationship with robotic constraints.

Future work includes extensions to take into account different cost functions such as human-likeness, or additional constraints such as collision avoidance or others depending on the task of the humanoid. Application to walking motion retargeting will also be addressed in future work.

ACKNOWLEDGMENT

This research was partially supported by the Japan Society for the Promotion of Science (JSPS; Grant-in-Aid for JSPS Fellows P12803).

REFERENCES

- [1] S. Nakaoka, S. Kajita, and K. Yokoi, "Intuitive and Flexible User Interface for Creating Whole Body Motions of Biped Humanoid Robots," in *Proc. on IEEE Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 1675–1682.
- [2] K. Yokoi, K. Nakashima, M. Kobayashi, H. Mihune, H. Hasunuma, Y. Yanagihara, T. Ueno, T. Gokuyuu, and K. Endou, "A tele-operated humanoid robot drives a backhoe," in *Proc. 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2003, pp. 1117–1122.
- [3] Y. Ogura, H. Aikawa, K. Shimomura, H. Kondo, A. Morishima, H. ok Lim, and A. Takinishi, "Development of a new humanoid robot WABIAN-2," in *Proc. of 2006 IEEE Int. Conf. on Robotics and Automation*, 2006, pp. 76–81.
- [4] K. Miura, E. Yoshida, Y. Kobayashi, Y. Endo, F. Kanehiro, K. Homma, I. Kajitani, Y. Matsumoto, and T. Tanaka, "Humanoid robot as an evaluator of assistive devices," in *Proc. 2013 IEEE Int. Conf. Robotics and Automation*, 2013, pp. 671–677.
- [5] K. Kaneko, F. Kanehiro, M. Morisawa, K. Miura, S. Nakaoka, and K. Yokoi, "Cybernetic human HRP-4C," in *Proc. 2009 IEEE-RAS Int. Conf. on Humanoid Robots*, 2009, pp. 7–14.
- [6] Y. Imamura, T. Tanaka, Y. Suzuki, K. Takizawa, and M. Yamanaka, "Motion-based-design of elastic material for passive assistive device using musculoskeletal model," *Journal of Robotics and Mechatronics*, vol. 23, no. 6, pp. 978–990, 2011.
- [7] N. Pollard, M. R. Jessica Hodgins, and C. Atkeson, "Adapting human motion for the control of a humanoid robot," in *Proc. of 2002 IEEE Int. Conf. on Robotics and Automation*, 2002, pp. 1390–1397.
- [8] S. Nakaoka, A. Nakazawa, K. Yokoi, and K. Ikeuchi, "Leg Motion Primitives for a Dancing Humanoid Robot," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2004, pp. 610–615.
- [9] 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," *Int. J. of Robotics Research*, vol. 26, pp. 829–844, 2007.
- [10] K. Miura, M. Morisawa, F. Kanehiro, S. Kajita, K. Kaneko, and K. Yokoi, "Human-like walking with toe supporting for humanoids," in *Proc. 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2011, pp. 4428–4435.
- [11] K. Yamane, S. O. Anderson, and J. K. Hodgins, "Controlling humanoid robots with human motion data: Experimental validation," in *Proc. 2011 IEEE-RAS Int. Conf. on Humanoid Robots*, 2011, pp. 504–510.
- [12] O. E. Ramos, L. Saab, S. Hak, and N. Mansard, "Dynamic motion capture and edition using a stack of tasks," in *Proc. 2011 IEEE-RAS Int. Conf. on Humanoid Robots*, 2011, pp. 224–230.
- [13] C. Ott, D. Lee, and Y. Nakamura, "Motion capture based human motion recognition and imitation by direct marker control," in *Proc. 2008 IEEE-RAS Int. Conf. on Humanoid Robots*, 2008, pp. 399–405.
- [14] B. Dariush, M. Gienger, A. Arumbakkam, C. Goerick, Y. Zhu, and K. Fujimura, "Online and markerless motion retargeting with kinematic constraints," in *Proc. 2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 191–198.
- [15] M. Do, P. Azad, T. Asfour, and R. Dillmann, "Imitation of human motion on a humanoid robot using non-linear optimization," in *Proc. 2008 IEEE-RAS Int. Conf. on Humanoid Robots*, 2008, pp. 545–552.
- [16] M. Gleicher, "Retargetting motion to new characters," in *Proc. GGRAPH '98*, 1998, pp. 33–42.
- [17] E. S. Ho, T. Komura, and C.-L. Tai, "Spatial relationship preserving character motion adaptation," *ACM Trans. on Graphics*, vol. 29, no. 4, pp. 33:2–8, 2010.
- [18] S. Nakaoka and T. Komura, "Interaction mesh based motion adaptation for biped humanoid robots," in *Proc. 2012 IEEE-RAS Int. Conf. on Humanoid Robots*, 2012, pp. 625–631.
- [19] S. Miossec, K. Yokoi, and A. Kheddar, "Development of a software for motion optimization of robots - application to the kick motion of the hrp-2 robot," in *Proc. 2006 IEEE Int. Conf. on Robotics and Biomimetics*, 2006, pp. 299–304.
- [20] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *International Journal of Robotics Research*, 2013, to appear.
- [21] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *Proc. 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 4906–4913.
- [22] T. Erez and E. Todorov, "Trajectory optimization for domains with contacts using inverse dynamics," in *Proc. 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 4914–4919.
- [23] W. Suleiman, E. Yoshida, J.-P. Laumond, and A. Monin, "On humanoid motion optimization," in *Proceedings of 7th IEEE-RAS International Conference on Humanoid Robots*, December 2007, pp. 180–187.
- [24] W. Suleiman, E. Yoshida, F. Kanehiro, J.-P. Laumond, and A. Monin, "On human motion imitation by humanoid robot," in *Proc. 2008 IEEE Int. Conf. Robotics and Automation*, 2008, pp. 2697–2704.
- [25] H. Si and K. Gärtner, "Meshing piecewise linear complexes by constrained delaunay tetrahedralizations," in *Proc. 14th International Meshing Roundtable*, 2005, pp. 147–163.
- [26] "Carnegie mellon university motion capture database," <http://mocap.cs.cmu.edu/>.
- [27] "Numerical algorithms group," <http://www.nag.co.uk/>.