Avatar Tracking Control with Generations of Physically Natural Responses on Contact to Reduce Performers' Loads

Ken Sugimori
sugimori@gs.haselab.net
Tokyo Institute of Technology
Kanagawa-ken, Japan

Hironori Mitake
mitake@haselab.net
Tokyo Institute of Technology
Kanagawa-ken, Japan

Hirohito Sato
hirohitosatoh@gs.haselab.net
Tokyo Institute of Technology
Kanagawa-ken, Japan

Kensho Oguri
k.oguri@gs.haselab.net
Tokyo Institute of Technology
Kanagawa-ken, Japan

Shoichi Hasegawa
hasevr@haselab.net
Tokyo Institute of Technology
Kanagawa-ken, Japan

ABSTRACT

The real-time performance of motion-captured avatars in virtual space is becoming increasingly popular, especially within applications including social virtual realities (VRs), virtual performers (e.g., virtual YouTubers), and VR games. Such applications often include contact between multiple avatars or between avatars and objects as communication or gameplay. However, most current applications do not solve the effects of contact for avatars, causing penetration or unnatural behavior to occur. In reality, no contact with the player's body occurs; nevertheless, the player must perform as if contact occurred. While physics simulation can solve the contact issue, the naive use of physics simulation causes tracking delay. We propose a novel avatar tracking controller with feedforward control. Our method enables quick, accurate tracking and flexible motion in response to contacts. Furthermore, the technique frees avatar performers from the loads of performing as if contact occurred. We implemented our method and experimentally evaluated the naturalness of the resulting motions and our approach's effectiveness in reducing performers' loads.

CCS CONCEPTS

• **Computing methodologies → Motion processing**.

KEYWORDS

avatar, character, motion capture, physics simulation

ACM Reference Format:

Ken Sugimori, Hironori Mitake, Hirohito Sato, Kensho Oguri, and Shoichi Hasegawa. 2021. Avatar Tracking Control with Generations of Physically Natural Responses on Contact to Reduce Performers' Loads. In *27th ACM Symposium on Virtual Reality Software and Technology (VRST '21), December 8–10, 2021, Osaka, Japan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3489849.3489859

VRST '21, December 8–10, 2021, Osaka, Japan

ACM ISBN 978-1-4503-9092-7/21/12…$15.00
https://doi.org/10.1145/3489849.3489859

1 BACKGROUND AND PURPOSE

With the spread of motion capture, *avatars*, which represent a player's body, are commonly used in real-time motion capture. Prominent examples include virtual YouTubers (VTubers) and virtual idols that present characters' appearances. These are controlled by human performers and streamed as videos; social VRs such as VRChat, where avatars can communicate with each other in VR space; and VR games that allow multiple players to play online. In such avatar applications, contact-based communication (e.g., stroking one's head and hugging) occurs more freely than it does between real people. However, the person or object to be contacted often does not exist in the actual space in which the player operates the avatar. Therefore, unnatural behavior results, including penetration or no reaction.

Through physics simulations, it is possible to automatically execute physical actions that consider the effects of contact and collision. However, if we use a physics simulation naively when the avatar and the object collide, only the virtual object bounces back, which is unnatural. Moreover, penetration is inevitable when avatars collide with each other. Achieving natural motion necessitates performing as if contact and collision occurred. However, this requires a high level of skill and cannot be expected of many performers. One solution is proportional derivative (PD) control [5]. However, the avatar will lag or move too far when PD gains are set to low values for the avatar's flexible response.

This paper proposes an avatar control method that can express natural contact with objects and other avatars, regardless of the performer's skill. To solve the problems mentioned earlier, we propose the following three requirements for our avatar control method.

Visually natural contact Create natural motions during and after contact by automatically performing actions that consider the effects of contact.

Reducing performers' loads Reduce the performers' loads arising from performing as if contact occurred.

Quick tracking Track the performer's performance quickly, except at the point of impact, to avoid affecting performance other than at the time of contact.

2 RELATED RESEARCH

This section first introduces reactive motion generation and then focuses on performance support and tracking delay.

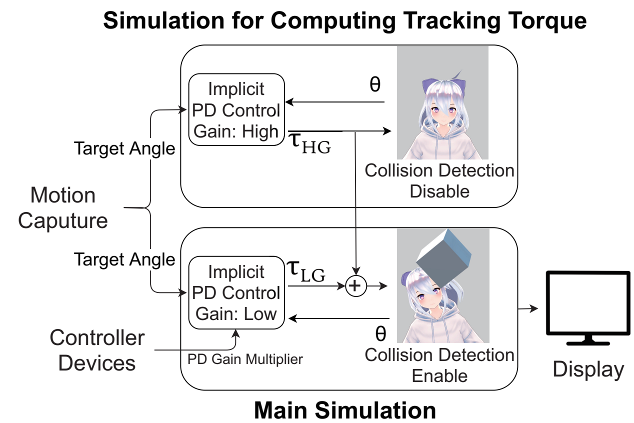

Figure 1: System structure.

Several studies have used physics simulations to automatically generate a character's reaction behavior upon contact. For example, switching between kinematics control and physical control when contact occurs was proposed [6, 10, 15]. Instead of switching, continuously applying physical control by weakening the PD gains of the joints when contact occurs achieves both tracking and responsive motion [12, 14]. Furthermore, Oshita [8] proposed a method to compute the angular acceleration of the joints directly from motion data and external forces. However, these studies involved targeted characters that moved with animation data, not real-time avatars.

Avatar performance that is driven by motion capture is supported by physical computation [3, 4]. Liang et al. [4] uses kinematic and dynamic constraints to generate responsive character animation with an incomplete motion input without considering trackability. Ishigaki et al. [3] automatically generates parts of movements that cannot be performed in reality when the virtual world and the real world do not match and focuses on moving the center of gravity using the environment. In this study, we aim to achieve both trackability and the influence of contact for detailed performance.

In a study focusing on realizing physical reactions of an avatar on contact, Nguyen et al. [5] attempted to automatically generate a character's reaction behavior upon contact using a physics simulation. In Nguyen at al.'s [5] method, PD gains must be low to generate reactive, flexible motions for collisions. However, lower PD gains cause longer delays in tracking captured motions.

A method combining feedforward and feedback controls has been suggested to solve the delay in tracking animation when PD gains are low [7, 11, 13]. In these studies, the target is a character moved via animation data. In the present study, we use the same method as Rubens et al. [7] and Tokizaki et al. [11], applying it to avatars that are manipulated in real-time using motion capture to *reduce performers' loads.* Moreover, Rubens et al. [7] and Tokizaki et al. [11] focused on proposing methods and did not measure the delay or evaluate the generated motions using subject experiments. Therefore, in this study, we confirm whether the generated motions meet the requirements of *visually natural contact* via experimentation on subjects and evaluate whether the generated motions meet the requirements of *quick tracking* by measuring their delays.

3 PROPOSED METHOD

Our proposed method allows the avatar to automatically perform actions that consider the effects of contact using a rigid-body physics simulation that detects collisions and computes contact forces. Thus, the method resolves unnatural motion output and reduces the loads on performers by eliminating the needs to avoid penetration and to react to contact. Moreover, we use feedforward control to solve the problem of tracking delay.

The configuration of our proposed system is shown in Figure 1. We use two physics simulations: the main simulation and the simulation for computing the tracking torque, which is similar to Rubens et al.'s [7] and Tokizaki et al.'s [11] method. In these simulations, we use PD control of the joint angles to make the avatar track the motion capture. These simulations employ different PD gains to achieve both a soft response during contact and quick tracking during non-contact. In the simulation for computing the tracking torque, high PD gains compute the joint torques required for quick tracking of the input motion. This simulation does not consider the effect of contact. In the main simulation, low PD gains are performed while adding the tracking torque. As a result, an avatar tracks without delay when not in contact. Moreover, the avatar deforms softly when in contact and slowly returns to the performer's pose after the contact ends.

We also made it possible to adjust the PD gains in the main simulator by using the performer's analog buttons of the Oculus Touch. The buttons change k_{stiff} to uniformly multiply the avatar's PD gains by a value from 1 to 9. This is a useful function for a performer who wishes to voluntarily control an avatar's stiffness. Moreover, we use implicit [1] rather than explicit PD control, enabling high PD gains for the computation of tracking torques (i.e., quicker tracking) with more stability.

4 IMPLEMENTATION

The avatar's physical model is a rigid-body joint model for physics simulation based on the avatar's bone structure. For the shapes of the collision detection, we used geometric primitives and adjusted them to avoid their protrusion from the avatar's outline as much as possible. The avatar's total weight was set to 50 kg, and the mass ratio of each part of the avatar's body was based on the actual person.

The avatar's root bone (the rigid body of the waist) is PD-controlled over its position and angle. The other bones are PD-controlled over only their angles. We set the P gains for the PD controls to be proportional to the moment of inertia of the body part moved by its joint at T pose, referring to a prior study [5]. We use P_{ratio} as a uniform multiplier applied to the P gains of all of the avatar's joints. Since the P gain is sometimes too small for a particular joint, if $P_i \leq P_{min}$, then $P_i = P_{min}$ is used instead. In addition, the D gains are determined by the critical damping ratio equation.

The values of the PD gains in the example implementation are described below. The PD gains for the main simulation were computed as $P_{ratio} = 30$ and $P_{min} = 10$ for each joint. On the other hand, the PD gains for the waist were set to high values so as not to cause unnatural foot slipping (see Section 7 for details). The PD gains for the position were 50000[N/m] and 5000[N $\cdot$ s/m], and the PD gains for the angle were 5000[N $\cdot$ m/rad] and 500[N $\cdot$ m $\cdot$ s/rad].

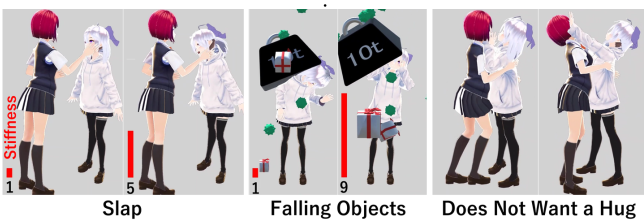

Figure 2: Avatar's generated motions.

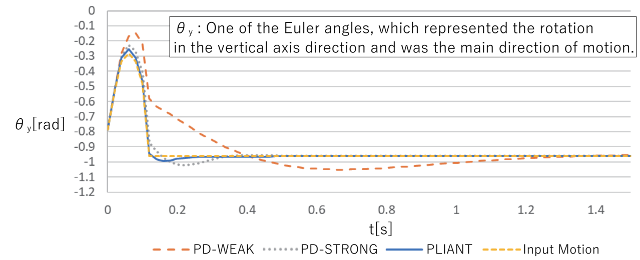

Figure 3: Input/Output angles of the right shoulder joint.

As for the simulation for computing the tracking torque, the P and D gains were 1,000 times higher than those used in the main simulation.

We developed our system on Unity, a widely used game engine. For the physics simulation, we used Springhead [2], which has implicit PD control as a standard feature.

We used an Oculus Rift and an Oculus Touch as input devices to detect the position and angle of the performer's head and hands. We used the estimated joint angles by computing inverse kinematics [9] as motion capture input.

5 AVATAR'S GENERATED MOTIONS

We created motions by performing with the proposed method. The results are shown in Figure 2 and in the supplemental video (1m 17s). The stiffness of the body is indicated as k_{stiff} at the bottom of each video. When the stiffness is not displayed, $k_{stiff} = 1$.

All of the motions are natural reactions without penetration. Furthermore, in **Slap**, **Falling Objects**, the performer could generate various reactions by adjusting the stiffness of the avatar's body.

6 EVALUATION

In this section, we evaluate that our proposed method satisfies the three requirements described in Section 1 by comparing it against other methods. We assess our proposed method against the requirements of *quick tracking* in Section 6.1, *visually natural contact* in Section 6.2, and *reducing performers' loads* in Section 6.3.

We compared four methods of moving an avatar by motion capture using physics simulation.

ONEWAY The motion capture is directly applied to the avatar, and the avatar is not affected when objects collide.

PD-WEAK Each joint is PD-controlled with low gains ($P_{ratio} = 30, P_{min} = 10$).

PD-STRONG Each joint is PD-controlled with high gains ($P_{ratio} = 2100, P_{min} = 10$).

PLIANT The proposed method.

ONEWAY represents a widely used method in current avatar applications in which no penetration occurs when avatars collide with objects, but penetration occurs when avatars collide with each other. The PD gains of PD-STRONG are set to values that are high enough to achieve quick tracking. The PD gains of PD-WEAK are set to the same values as that of the main simulation of PLIANT to obtain the same reaction as is obtained in PLIANT when contact occurs.

6.1 Evaluation of Quick Tracking

We measured the input and output joint angles. We created input motion by moving the right hand from right to front at a constant speed (5 m/s). The right hand is held stationary before and after the movement.

Figure 3 shows the typical delay on the angles for a joint. From Figure 3 and the supplemental video (2m 12s), we can see that PLIANT provided the best tracking performance, followed, in order, by PD-STRONG and PD-WEAK. Furthermore, in PD-WEAK, the avatar's right hand does not stop at the target position and overshoots. In this situation, the performance is hampered. In PLIANT, the delay is almost zero, and the overshoot is only about 0.2 seconds. Therefore, PLIANT satisfies the requirement of *quick tracking* listed in Section 1.

6.2 Evaluation of Visually Natural Contact

This section shows comparisons of *visually natural contact* as listed in Section 1.

6.2.1 Method. We prepared videos featuring each method, showed them to subjects, and asked the subjects to choose which method was more natural through a nine-grade evaluation. The video used for the questionnaire (the supplemental video 2m 18s) depicts the following six situations. **9kg** is to hit a bunch (9kg) of bottles. **1.5kg** is to hit a bottle (1.5kg). **Hit** is to hit a face. **Slap** is to slap someone on the back. **Stroke** is to stroke a head. **Hug** is to hug someone. In all situations, we operated the red-haired avatar. The white-haired character was in a static pose or a small fixed standby motion to match the conditions. In **Hug** only, we used additional positional and rotational PD control of the lower leg, as the avatar's feet slipped when its hips moved (see Section 7 for details).

The subjects included 11 males aged from 20 to 30 years old, 2 females aged from 20 to 30 years old, 1 female aged from 40 to 50 years old, and 1 male aged from 60 to 70 years old. Note that we instructed the subjects to pay attention to the contact, not to the inevitable, slight differences in human performance.

6.2.2 Results and Discussion. Table 1 shows the results of the questionnaire. The naturalness scores are averages calculated from the results of a nine-grade evaluation from -4 to +4 in which positive scores mean that PLIANT was more natural. The compared method means the method compared to PLIANT.

Table 1 shows that **9kg**, **1.5kg**, **Hit**, and **Slap** were significantly more natural when using PLIANT than when using the compared

Table 1: Naturalness scores of the evaluation of *visually natural contact*, **:$p < 0.01$ (Wilcoxon's signed-rank test).

Situation	9kg**	1.5kg**	Hit**	Slap**
Naturalness Score	3.1	2.3	2.9	2.7
Compared Method	PD-STRONG			

Situation	**Stroke**	**Hug**	**Stroke**	**Hug**
Naturalness Score	0.6	-0.13	0.2	0.53
Compared Method	PD-STRONG		ONEWAY	

methods. There was no significant difference observed between the methods for **Stroke** and **Hug**. The cause of this result is whether the movement was large or small.

6.3 Evaluation of Reducing Performers' Loads

We also evaluated whether the requirement of *reducing performers' loads* listed in Section 1 was met. We compared PLIANT to ONEWAY, which requires performing as if contact occurred.

6.3.1 Method. We asked the subjects to perform the following three movements: **Bump**, in which a cube bumps into the avatar; **Stroke**, in which the avatar strokes a moving character; and **Hit**, in which the avatar quickly hits a panel that has changed color.

Regarding the experimental procedure, we first showed a model video (the supplemental video 3m 22s). Second, we provided a simple explanation of the avatar control method. Third, we asked the subject to perform and recorded an avatar video. Fourth, the subject determined whether the avatar video was natural. If unnatural, the experiment continued with the third procedure, and if natural, the experiment ended.

To quantify the ease of performance, we measured the number of takes subjects executed the third step. After the experiment, we recorded their impressions. Since the movements in this experiment did not require changing the avatar's stiffness in Section 3, we excluded the function and fixed it to a value of $k_{stiff} = 1$ to prevent the subjects from accidentally changing it. Since order effects were expected in the experiment, we randomly decided whether ONEWAY or PLIANT would be used first. The subjects included 4 males aged from 20 to 30 years old and 1 male aged from 30 to 40 years old. All subjects had experience using an Oculus Rift.

About Naturalness. Each experiment procedure ended when the subject decided that the performance was at least as natural as the model video. Moreover, we told the subjects, "Ignore the hips and legs because they are not motion-captured." In **Hit**, we told the subjects, "The head may not follow well; please ignore this." (see Section 7 for details).

6.3.2 Results and Discussion. Figure 4 shows that a lesser or equal number of takes was required with PLIANT as compared to ONEWAY. The subjects commented that "ONEWAY is very tiring because I have to anticipate the opponent's movements not to penetrate while performing."

Penetrations were observed only in ONEWAY (the supplemental video 3m 22s). As a result of being aware of penetration, the movement comes to be cautious, as seen in Subject C's **Hit** use of ONEWAY.

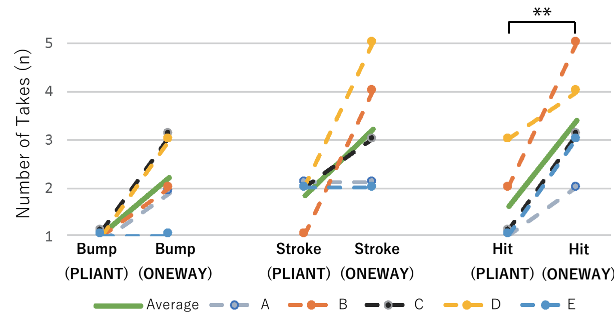

Figure 4: Number of takes for the evaluation of *reducing performers' loads*, **:$p < 0.01$.

This evaluation indicates that PLIANT meets the requirements of *reducing performers' loads* better than ONEWAY.

7 LIMITATIONS

In the proposed system, the performer does not notice the avatar's generated movement. For example, when the avatar pushes against a fixed wall, (the supplemental video 4m 36s), the chest joint bends, and the avatar bends backward. However, the performer does not notice that the avatar is bending backward. This behavior of the avatar pushed back against the wall is physically correct. Nevertheless, if the performer wants to push the wall strongly, the behavior is contrary to the performer's intention.

The generated motion suffered from foot slips, (the supplemental video 4m 41s). The avatar's waist was fixed by PD control in position. If the waist moved, the avatar's whole body moved, and the legs slipped.

8 CONCLUSION

In this paper, we proposed a new avatar technology for performing contact interaction in VR space. Our purpose was to reduce performers' loads of performing as if contact occurred. For this purpose, we automatically generated the effect of the contact. We applied a feed-forward control-based prior method to a real-time motion-captured avatar, combining two types of PD control. Moreover, we used implicit PD control to enable quick and stable tracking. Our method achieves both quick tracking and flexible reaction simultaneously.

We implemented the proposed method and performed various performances. We easily obtained natural motions not only in the effects of objects hitting the avatar but also in social touches, such as stroking one's head and hugging. In addition, evaluation experiments verified that *quick tracking*, *visually natural contact*, and *reducing performers' loads* were achieved. In Section 6.3, some situations required the subject to perform carefully so as not to cause penetration in a conventional method. In contrast, with our method, the subject did not hesitate to move quickly. This result suggests that our method not only reduces the performers' loads but also enables them to perform what they originally intended.

On the other hand, there is a limitation that the automatically generated movements via physics simulation employed by our proposed method may not always meet a performer's intention.

REFERENCES

[1] Mihai Anitescu and Florian A. Potra. 2002. A time-stepping method for stiff multibody dynamics with contact and friction. *Internat. J. Numer. Methods Engrg.* 55, 7 (2002), 753–784. https://doi.org/10.1002/nme.512 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.512

[2] Shoichi Hasegawa, Naoki Okada, Jiro Baba, Yuuichi Tazaki, Hiroshi Ichikawa, Akihiko Shirai, Yasuharu Koike, and Makoto Sato. 2004. Springhead: Open source haptic software for virtual worlds with dynamics simulations. In *Proceedings of the 4th International Conference EuroHaptics 2004*. 385–386.

[3] Satoru Ishigaki, Timothy White, Victor B. Zordan, and C. Karen Liu. 2009. Performance-Based Control Interface for Character Animation. *ACM Trans. Graph.* 28, 3, Article 61 (July 2009), 8 pages. https://doi.org/10.1145/1531326.1531367

[4] Xiubo Liang, Ludovic Hoyet, Weidong Geng, and Franck Multon. 2010. Responsive Action Generation by Physically-Based Motion Retrieval and Adaptation. In *Motion in Games*, Ronan Boulic, Yiorgos Chrysanthou, and Taku Komura (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 313–324.

[5] Nam Nguyen, Nkenge Wheatland, David Brown, Brian Parise, C. Karen Liu, and Victor Zordan. 2010. Performance Capture with Physical Interaction. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Madrid, Spain) *(SCA '10)*. Eurographics Association, Goslar, DEU, 189–195.

[6] Nam H. Nguyen, Raul Arista, C. Karen Liu, and Victor Zordan. 2012. Adaptive Dynamics with Hybrid Response. In *SIGGRAPH Asia 2012 Technical Briefs* (Singapore, Singapore) *(SA '12)*. Association for Computing Machinery, New York, NY, USA, Article 5, 4 pages. https://doi.org/10.1145/2407746.2407751

[7] Rubens F. Nunes, Creto A. Vidal, Joaquim B. Cavalcante-Neto, and Victor B. Zordan. 2008. Simple Feedforward Control for Responsive Motion Capture-Driven Simulations. In *Advances in Visual Computing*, George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Paolo Remagnino, Fatih Porikli, Jörg Peters, James Klosowski, Laura Arns, Yu Ka Chun, Theresa-Marie Rhyne, and Laura Monroe (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 488–497.

[8] Masaki Oshita and Akifumi Makinouchi. 2001. A Dynamic Motion Control Technique for Human-like Articulated Figures. *Computer Graphics Forum* 20, 3 (2001), 192–203. https://doi.org/10.1111/1467-8659.00512 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-8659.00512

[9] RootMotion. 2014. Final IK. https://assetstore.unity.com/packages/tools/animation/final-ik-14290 2020/1/25 browsed.

[10] Ari Shapiro, Fred Pighin, and Petros Faloutsos. 2003. Hybrid Control for Interactive Character Animation. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications (PG '03)*. IEEE Computer Society, USA, 455.

[11] Takashi Tokizaki, Yuuichi Tazaki, Hironori Mitake, and Shoichi Hasegawa. 2009. Pliant Motion: Integration of Virtual Trajectory Control into LCP Based Physics Engines. In *SIGGRAPH '09: Posters* (New Orleans, Louisiana) *(SIGGRAPH '09)*. Association for Computing Machinery, New York, NY, USA, Article 10, 1 pages. https://doi.org/10.1145/1599301.1599311

[12] Pawel Wrotek, Odest Chadwicke Jenkins, and Morgan McGuire. 2006. Dynamo: Dynamic, Data-Driven Character Control with Adjustable Balance. In *Proceedings of the 2006 ACM SIGGRAPH Symposium on Videogames* (Boston, Massachusetts) *(Sandbox '06)*. Association for Computing Machinery, New York, NY, USA, 61–70. https://doi.org/10.1145/1183316.1183325

[13] KangKang Yin, Michael B. Cline, and Dinesh K. Pai. 2003. Motion Perturbation Based on Simple Neuromotor Control Models. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications (PG '03)*. IEEE Computer Society, USA, 445.

[14] Victor Brian Zordan and Jessica K. Hodgins. 2002. Motion Capture-Driven Simulations That Hit and React. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Antonio, Texas) *(SCA '02)*. Association for Computing Machinery, New York, NY, USA, 89–96. https://doi.org/10.1145/545261.545276

[15] Victor Brian Zordan, Anna Majkowska, Bill Chiu, and Matthew Fast. 2005. Dynamic Response for Motion Capture Animation. *ACM Trans. Graph.* 24, 3 (July 2005), 697–701. https://doi.org/10.1145/1073204.1073249