Example Guided Inverse Kinematics
측정 데이타에 기반한 향상된 역 운동학

탁세윤, 고형석
SNU 휴먼 애니메이션 센터
서울대학교 전기공학부
E-mail: {tseyoon,ko}@graphics.snu.ac.kr

Abstract

This paper proposes *example guided inverse kinematics* (EGIK) which extends and enhances existing inverse kinematics technique. In conventional inverse kinematics, redundancy in the model produces an infinite number of solutions. The motion could be jerky depending on the choice of solutions at each frame. EGIK exploits the redundancy for imitating an example motion (a pre-measured motion data) so that a unique solution is chosen. To minimize the gap between the goal and current end-effector position and imitate the original motion at the same time, nonlinear optimization technique is employed. So, the resulting motion resembles the original one in an optimal sense. Experiments prove that the method is a robust and effective technique to animate high DOF articulated models from an example motion.

초록

Example guided inverse kinematics (EGIK)는 기존의 역운동학 (inverse kinematics) 을 확장, 향상시키는 알고리즘이다. 기존의 역운동학은 모델자체의 redundancy에 기인하여 jerky한 동작을 만들어내는 단점이 있다. EGIK에서는 미리 측정된 동작데이타를 효과적으로 이용하여 이러한 redundancy문제를 해결한다. 이때 end-effector 위치 차이와 관절각의 차이를 동시에 최소화하기 위해서 비선형 최적화 기술이 사용된다. 우리는 결과로서의 동작이 원래의 동작의 특성을 잘 보존하는 자연스러운 것임을 보인다. EGIK는 기존의 동작 데이타를 이용하여 다관절체의 동작을 생성해내는 효과적인 기술이다.

제 1 절 Introduction

Animating human motion has been a great challenge. The task may appear easy at the first look since we can completely command an articulated figure by supplying joint angles. However, the difficulty stems from the fact that there are too many things to control. Human body has 206 bones and hundreds of muscles. A reasonable model of it can easily have 40 degrees of freedom. *Computing* such number of joint angles so that the resulting motion resembles that of a real human is not a trivial task. Among diverse approaches to solve this problem, inverse kinematics and motion capture are just two. The algorithm proposed in this paper is about the half way between these two approaches.

Inverse kinematics was originated from robotics field [2]. It computes joint angles that position the end-effector at a desired location. In robotics, major interest has been on six DOF robots. Since end-effector has six DOFs in general (three for position, and the other three for orientation), inverse kinematics on a six DOF robot gives a unique or at most four different solutions. However, if a model has 40 DOFs, there exist an infinite number of solutions (actually, the dimension of the solution space is 34), and only one of them is selected for the frame.

Because the selection is purely up to the numerical process employed, even though the end-effector follows anticipated trajectory, joint angles can make abrupt changes. Therefore neighboring frames won't have *coherence*, and simple replay of those frames may result in a jerky animation. Usually the numerical process picks a configuration that is *reasonably close* to the previous configuration. Therefore, in interactive demonstration, many times the *lacking coherence* is overlooked. When the result is recorded into a video disk and replayed at a normal speed, however, the reasonable closeness is not acceptable to human eyes; it can produce unpleasant artifacts.

On the other hand, motion capture is an effective technique to measure and copy the complex motion of articulated characters. However, this technique at its current state has two major drawbacks. First, the measurement errors are far from negligible. Without elaborate manual processing the resulting animation looks shaky or unrealistic. The other drawback, which is directly relevant to this paper, is that the data is for a specific subject in performing a specific motion. Obviously, the anthropometric scale between the measured subject and to-be-animated figure will be different. Also, the target motion to be animated might be slightly different from the measured motion. To overcome partly this poor generality of motion capture data, a variety of motion-reusing techniques [4, 13, 12, 10] were proposed.

As an effective solution to animate high DOF articulated models from a limited set of motion capture data, we propose *example guided inverse kinematics* (EGIK), which combines inverse kinematics with motion capture. A set of motion capture data is used as an example to be imitated. Before starting the inverse kinematics, the objective function[1] is augmented with several extra terms that will drive the inverse kinematics solution to imitate the example

[1]Nonlinear programming is widely used to solve inverse kinematics

configuration at that moment. The minimization of the objective function will do two things at the same time: minimizing the gap between the goal and current end-effector position, and imitating the original motion.

EGIK can improve both inverse kinematics and motion capture. As pointed earlier, in the conventional inverse kinematics, the redundancy in the model causes the lacking coherence problem. In EGIK, however, the surplus DOFs are involved in imitating the example motion. Therefore the selection among the multiple choices is not arbitrary. This new inverse kinematics provides a single choice which is smooth as long as the original motion was smooth (Section 5). Also, EGIK extends the usability of motion capture data. From a given motion data virtually any number of variation is possible. Still the resulting variations all resemble the original motion in an optimal sense.

The next section relates our work to some notable previous work. Section 3discusses different foci in imitating example motion. Section 4presents EGIK algorithm. We show several experimental results in Section 5. Section 6concludes the paper.

제 2 절 Related Work

Inverse kinematics has been studied in great detail in robotics. But most robot manipulators are relatively simple, so the robotics literature seldom addresses techniques for massively redundant mechanisms such as human figure. Baker and Wampler [2] developed a method to utilize the redundancy in achieving functional goals such as collision avoidance or mechanical soundness. They did not consider the realism of motion.

In computer graphics, many approaches were proposed to solve inverse kinematics problem in highly articulated figures. Zhao and Badler [14] formulated inverse kinematics as an optimization problem, and solved it using nonlinear programming. Their approach was successful in positioning end-effectors to satisfy multiple constraints for complex postures. However, the objective function considered only the positional goal, so could not guarantee motion coherence in animation.

Boulic and Thalmann proposed a hybrid form of direct and inverse kinematic control for articulated figure motion editing [3]. They specified the desired end-effector trajectories with several half-spaces that define the boundaries, and then applied the coach-trainee metaphor whenever end-effector tries to go out of the specified region. Their algorithm for controlling the surplus degree of freedom was based on mathematical null space search.

Rose et al. [11] developed an algorithm for generating transitions between basis motions using a combination of spacetime constraints and inverse kinematics constraints. They used B-spline functions for joint angle representation to guarantee a smooth motion. Their major goal was to generate seamless transitions between two motion segments, by enforcing inverse kinematics constraint. Therefore, it could modify detailed characteristics of the original motion.

Recently, Gleicher [6, 7], Lee and Shin [8], Choi and Ko [5] proposed methods for retargeting a motion to different characters. Gleicher [6, 7] presented a method for editing a motion while preserving as much of the original quality as possible. Lee and Shin [8] enhanced Gleicher's work by employing a hierarchical curve fitting technique and human-specific inverse kinematics solver. Choi and Ko [5] developed the on-line motion retargeting technique which enables captured motion to be retargeted in real-time while the motion is being captured.

In the above work for retargeting, to preserve the qualities of the original motion, they minimized the difference between the source and destination motions under the given end-effector constraints. Since the end-effector constraints were *hard* constraints, the joint angles may have to be significantly modified to satisfy the constraints. Therefore the retargeted result could be troublesome at or near singular configurations. (Singularity occurs when the end-effector is near or at the boundary of the workspace. In such a case, to produce a small change of end-effector position, joints may have to rotate large angles. As a result the motion can be jerky.)

In our example guided inverse kinematics, however, we consider both the end-effector constraint and joint angle imitation goal in one objective function (Section 4.2). It means that end-effector constraint is not a hard constraint any more. Both end-effector discrepancy and joint angle difference are penalized. The ratio between them can be controlled by giving different weights. Differently from the above algorithms [7, 8, 5], as long as we use a non-zero weight for the joint angle difference term, our algorithm produces a smooth motion even near a singular configuration. One might think our algorithm can sacrifice the end-effector goal significantly. An interesting finding of this paper is that by controlling the weights, end-effector error can be reduced to a negligible level. The details are presented in Section 4.2.

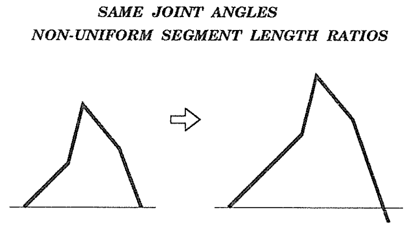


그림 1. Joint angle copying can fail when there is a closed loop.

제 3 절 Focus of Imitation

Example guided inverse kinematics is based on motion imitation. Imitating a motion needs to have a focus. In general, people perceive that two motions look similar if the angles are kept the same at the corresponding joints. We can easily satisfy such condition when the ratios between the corresponding links are uniform.

When the anthropometric scale of the two articulated figures is not uniform, however, the above condition does not have much sense due to the following two reasons; (1) When there exists a

problem of highly redundant articulated figures. The objective function, which is the gap between the goal and current end-effector position in the conventional inverse kinematics, is the quantity that should be minimized by the programming.

closed loop as shown in Figure 1, using the identical joint angles may violate important constraints. (2) If the end-effector trajectory is the focus of imitation (e.g., when a person write the letter 'A' with his finger tip, and another person is imitating it), simple joint angle copying may not imitate the end-effector motion accurately.

Because it is highly probable that the anthropometry of the measured subject is not proportional to that of animated figure, in most cases the joint angle imitation goal should be compromised with the end-effector imitation goal. There might be many other imitation foci. But in this work, we consider only the following two imitation foci:

- **The joint angle pattern (A-pattern)**
- **The end-effector motion pattern (E-pattern)**

In the following section we will define the new objective function in which the above two types of imitation effort can be amalgamated.

제 4 절 Example Guided Inverse Kinematics (EGIK)

4.1 Overview

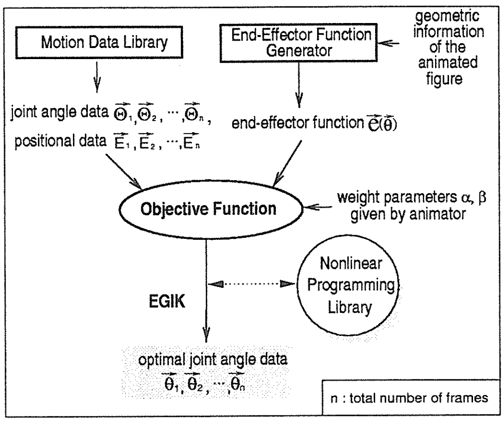


그림 2. Overview of EGIK

Figure 2is a diagram showing how EGIK is carried out. It consists of the following steps.

1. **Selection from the motion data library:**
   The motion data library contains a number of example data sets. A data set consists of the joint angle data $\vec{\Theta}_1, \vec{\Theta}_2, ..., \vec{\Theta}_n$ and the end-effector position data $\vec{E}_1, \vec{E}_2, ..., \vec{E}_n$ at each frame of the motion. Here n is the number of frames in the motion. The joint angle data $\vec{\Theta}_i$ is J-tuple vector $(\Theta_i^1, ..., \Theta_i^J)$ where J is the number of joints, and the end-effector position data $\vec{E}_i$ is three dimensional vector (E_i^x, E_i^y, E_i^z).

2. **End-effector function construction:**
   From the geometric information of the animated figure, the end-effector function $\vec{e}(\vec{\theta})$ is constructed using forward kinematics. The end-effector function $\vec{e}(\vec{\theta})$, which is dependent on the joint variable $\vec{\theta} = (\theta^1, ..., \theta^J)$, gives the global position of the end-effector.

3. **Objective function construction using weight parameters α and β:**
   The objective function for EGIK is formulated. The weight α and β for E-pattern and A-pattern, respectively, participate in forming the function. We present the details in the next section.

4. **Solving nonlinear programming:**
   We minimize the objective function using nonlinear programming library, and obtain the optimal joint angle data $\vec{\theta}_1, \vec{\theta}_2, ..., \vec{\theta}_n$.

4.2 Objective Function for EGIK

In EGIK, we allow a small fraction of error in end-effector position in order to imitate the joint angle pattern. Let us consider two error terms, *E-error* and *A-error*. E-error is the end-effector position error $||\vec{e}(\vec{\theta}) - \vec{E}||$, thus a small E-error means E-pattern is accurately transferred. A-error is the joint angle error $||\vec{\theta} - \vec{\Theta}||$, thus a small A-error means A-pattern is accurately transferred. The objective function of conventional inverse kinematics consists of a single term, $||\vec{e}(\vec{\theta}) - \vec{E}||$. In our EGIK, the objective function $G_n(\vec{\theta_n})$ is augmented with A-error as in:

$$G_n(\vec{\theta_n}) = \alpha\, G_n^E(\vec{\theta_n}) + \beta\, G_n^A(\vec{\theta_n}) \tag{1}$$

with

$$\begin{aligned} G_n^E(\vec{\theta}_n) &= ||\vec{e}_n(\vec{\theta}_n) - \vec{E}_n||^2 + \\ &\quad k_v^E ||\dot{\vec{e}}_n(\vec{\theta}_n) - \dot{\vec{E}}_n||^2 + k_a^E ||\ddot{\vec{e}}_n(\vec{\theta}_n) - \ddot{\vec{E}}_n||^2 \end{aligned} \tag{2}$$

$$G_n^A(\vec{\theta}_n) = ||\vec{\theta}_n - \vec{\Theta}_n||^2 + k_v^A ||\dot{\vec{\theta}}_n - \dot{\vec{\Theta}}_n||^2 + k_a^A ||\ddot{\vec{\theta}}_n - \ddot{\vec{\Theta}}_n||^2 \tag{3}$$

where $\vec{\theta}_n$, $\vec{e}(\vec{\theta}_n)$, $\vec{\Theta}_n$, and $\vec{E}_n$ are joint angle variable, end-effector function, joint angle data, and end-effector position data at n-*th* frame, respectively. $k_v^E, k_a^E, k_v^A, k_a^A$ are coefficients of velocity and acceleration error terms. $G_n^E(\vec{\theta}_n)$ and $G_n^A(\vec{\theta}_n)$ are E-error and A-error terms, respectively. Since velocity and acceleration are important elements of motion, Equations (2) and (3) include the terms for imitating the velocity and acceleration of original motion. Actually, we assign small values to the coefficients $k_v^E, k_a^E, k_v^A, k_a^A$, because positional imitation is the primary goal.

By Equation (1), each frame of the target motion tries to imitate both E-pattern and A-pattern of the source motion. Since imitation is done at each frame independently, consecutive frames may not be coherent. So we augment the Equation (1) with an additional penalty term, which represents the joint angle differences between the current frame and the previous frame. This term prevents one specific joint angle from changing abruptly. The final objective function for EGIK is

$$G_n(\vec{\theta_n}) = \alpha\, G_n^E(\vec{\theta_n}) + \beta\, G_n^A(\vec{\theta_n}) + \|\vec{\theta}_n - \vec{\theta}_{n-1}\|_M^2 \quad (4)$$

$$M = \begin{bmatrix} m_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & m_{JJ} \end{bmatrix}, \quad (5)$$

where $\|\cdot\|_M$ is a matrix M-norm, which is defined by $\|x\|_M = x^t M x$ for a vector $x \in R^J$ with a positive-definite matrix M. The matrix M in Equation (5) represents the stiffness of the joints. It is typically diagonal if the coupling factors among joints do not exist. Note that we can assign different weight values to the diagonal elements of M to control the stiffnesses. A large value makes the joint stiff.

In Equations (1) and (4), α and β are weights specified by the animator interactively. For example, α=1,β=0 is the case of pure inverse kinematics, α=0,β=1 is the case of pure joint angle copying, and α=0.5,β=0.5 is the case in which E-pattern and A-pattern are considered with equal weights. In inverse kinematics, achieving the end-effector position is the primary goal, so we must assign a much larger value to α than β. The reader might think a nonzero value of β causes failure in achieving the end-effector goal. However, it was one of our interesting findings that a sufficiently small value of β (e.g., α=1, β=10^{-5}) could preserve the motion characteristics quite well, while the error in end-effector positioning was negligible. We show the quantitative analysis in Section 5

4.3 Solving Nonlinear Programming

We minimize the objective function using nonlinear programming (NLP). Among diverse approaches for the problem, Lagrange method and SQP(*Sequential Quadratic Programming*) are two popular methods [9]. It is unrealistic to expect to find one general NLP code that works for every kind of nonlinear model. Instead, one should try to select a code that fits the problem. In this work, to see how much our solution depends on different NLP algorithms, we have used two optimization packages; LANCELOT [1] and DONLP2. LANCELOT is a large-scale implementation of the augmented Lagrangian approach[2], and DONLP2 is an implementation of SQP method[3]. We found that in most cases the two solvers produce almost equivalent results.

A scaling factor must be considered in Equation (4), since the unit of end-effector position data is *centimeter (cm)* and the unit of joint angle data is *radian*. From our experiments, we found that it is a good approximation to multiply *the square of total-link-length* to A-error term. The theoretical foundation is as follows; when the serial chain with the total link length of *L cm* rotates one full cycle about the robot base, the base joint rotates 2π *radian* and the end-effector moves $2\pi L$ *cm*. Considering the squared error terms of $G(\vec{\theta})$, $(2\pi L/2\pi)^2 = L^2$ should be the scaling factor between the A-error and E-error terms.

4.4 Discussion

In other approaches [7, 8] for retargeting problem, the objective function is the time integral of the errors between the source and target motions.

$$G(m) = \int (m(t) - m_{src}(t))^2 dt. \quad (6)$$

It is computationally expensive, thus has to be done in off-line. We can distinguish our solution from the above approaches by the following.

- **EGIK performs the optimization at each frame independently.** Therefore it involves much less computation, and can process motions of virtually unlimited lengths.
- **Optimization at individual frames can increase retargeting quality.** Spacetime constraint based approaches [7] obtain the target motion by minimizing a single scalar quantity, whereas our EGIK minimizes a scalar for each frame. Such frame-by-frame optimization can increase the fidelity of motion retargeting.
- **EGIK is free from the problems due to singularities.** Retargeting algorithms with hard end-effector constraints can produce jerky result at or near the singular configurations. In EGIK, both end-effector constraints and joint angle imitation goal are considered in one objective function. It can effectively prevent a jerky motion at a singular configuration by sacrificing a small (negligible) fraction of end-effector goal achievement to follow the joint angle pattern.

제 5 절 Experiments

EGIK was implemented on Silicon Graphics Octane MXI workstation.

[2] The augmented Lagrangian algorithm is based on successive minimization of the augmented Lagrangian.

[3] The sequential quadratic programming algorithm is a generalization of Newton's method for unconstrained optimization in that it finds a step away from the current point by minimizing a quadratic model of the problem. In its purest form, the SQP algorithm replaces the objective function with the quadratic approximation.

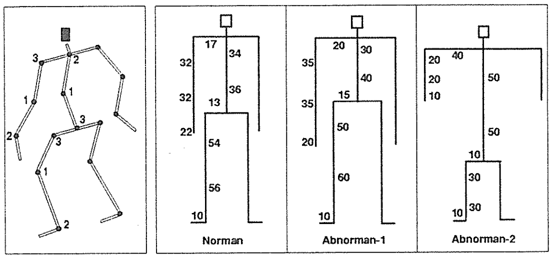
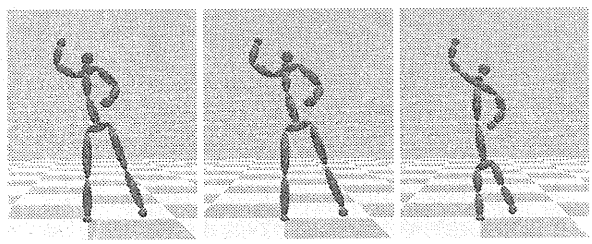
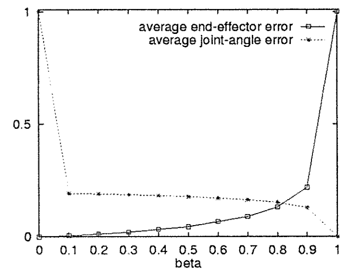


그림 3. Human models. (The numbers in the leftmost box represent the DOFs at the joints, and the numbers in the other boxes represent link lengths.)

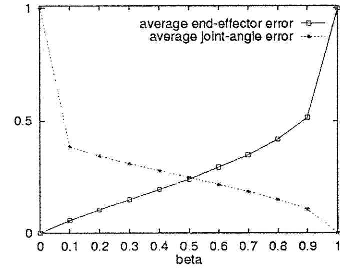


(a) Norman (b) Abnorman-1 (c) Abnorman-2

그림 4. Snapshots from the throwing motions of Norman and Abnormans.



(a) Abnorman-1



(b) Abnorman-2

그림 5. Average errors in the motion of Abnormans 1 and 2. For the end-effector error, the average was taken over all frames. For the joint angle error, the average was taken over all joints over all frames. The averages were normalized between [0,1]. In (a), actual end-effector error at $\beta = 1$ was 14 cm, and joint angle error at $\beta = 0$ was 12 degrees. In (b), they were 43 cm and 27 degrees, respectively.

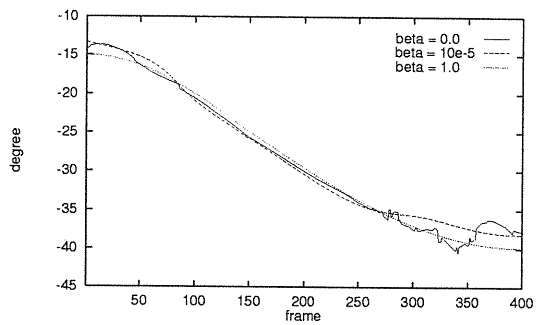
The following discussion refers to the animation clips put on "http://graphics.snu.ac.kr/research/egik/index.html", which were produced using EGIK algorithm.

We modeled three articulated human figures. The first one, called **Norman**, is the subject whose motion was measured. The second and third ones are called **Abnorman-1** and **Abnorman-2**, respectively. As shown in Figure 3, Abnorman-1 has longer limbs and shorter torso than Norman, while Abnorman-2 has excessively shorter limbs and longer torso. Those two figures are out of proportion on purpose, in order to demonstrate that our algorithm is capable of producing similar motions in spite of the anthropometric differences.

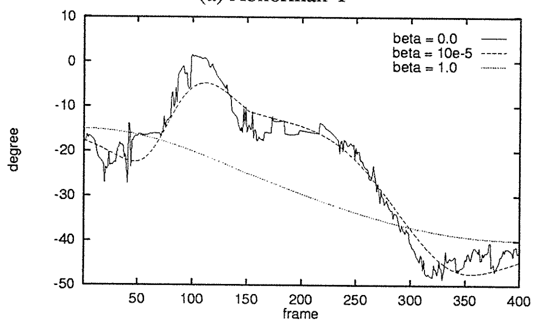
The throwing motion of Norman (Figure 4) was retargeted to Abnormans 1 and 2 using different values of α,β. Figure 4shows snapshots during the process. In animating Abnormans, it might be reasonable to scale the end-effector trajectories proportional to the *size* of animated figures. However, we did not scale the end-effector trajectories in order to demonstrate the adaptability of our algorithm.

Figure 5shows E-error and A-error of Abnormans at different values of α and β ($\alpha = 1 - \beta$). As expected, using a larger β value increases the end-effector errors but reduces the joint-angle errors. In (a), the errors of Abnorman-1 are reduced rapidly even with a small value of α or β. In (b), the errors of Abnorman-2 drop slower than those of Abnorman-1. It is predictable; the anthropometric discrepancy of Abnorman-2 is a lot more excessive than that of Abnorman-1.

The graphs in Figure 6plot the shoulder angles of Abnormans 1 and 2 during the throwing motion. The fluctuating solid curve is for the case when $\beta = 0$, the fine-dotted curve is for the case when $\beta = 1$, and the dashed curve is for the case of EGIK($\beta = 10^{-5}$). Note that in the case of EGIK, joint motions follow the original joint angle pattern quite accurately if the anthropometric difference is not excessive, while the end-effector error was negligible (10^{-14}). In Figure 6(b), due to the excessive anthropometric difference, a



(a) Abnorman-1



(b) Abnorman-2

그림 6. Shoulder angle in Abnormans' motion. Solid, dashed, dotted curves represent conventional IK (β=0.0), EGIK (β=10^{-5}), and pure joint angle copying (β=1.0), respectively.

small value of β could not imitate Norman's joint angle pattern well. But the animated result still seems to preserve the original motion characteristics. Refer to the animation put on the web. Table 1summarizes the end-effector errors at a few β values.

제 6 절 Conclusion and Future Work

We have presented a new algorithm to solve inverse kinematics problem by imitating an example motion. The algorithm can be also used to retarget a motion to different characters. It is an improvement over the previous inverse kinematics or motion retargeting algorithms in the following aspects:

- **Coherence between frames:** it guarantees coherence between frames.
- **Robustness in motion retargeting:** by imposing a soft constraint on the end-effector position, the algorithm can produce a natural motion even for the motions with singular configurations.
- **Increased fidelity in retargeted motions:** the algorithm minimizes a scalar quantity at every frame, which increases the fidelity of retargeting compared to the spacetime constraint based methods.
- **Computational efficiency and ability to process long sequence of motion:** since it performs optimization at individual frames independently, it involves less computation than the spacetime constraint based methods, and can process motions of virtually unlimited length.

As a future work, dynamic soundness should be considered. Kinematically generated motion may not be feasible in the physical world due to the difference in the strength of the two bodies or some other body conditions. Therefore the result of our example guided inverse kinematics should be further adjusted to account for dynamic balance or limitations of the body.

Acknowledgments

This paper was supported by Creative Research Initiatives of the Korean Ministry of Science and Technology. This work was partially supported by ASRI (Automation and Systems Research Institute), Seoul National University.

참고 서적

[1] *LANCELOT: a Fortran Package for Large-Scale Nonlinear Optimization (Release A).* Springer Series in Computational Mathematics, Volume 17, Springer Verlag (Heidelberg, New York), 1992.

[2] D. Baker and C. Wampler. On the inverse kinematics of redundant manipulators. *The International Journal of Robotics Research*, 7(2):3–21, March/April 1988.

[3] Ronan Boulic and Daniel Thalmann. Combined direct and inverse kinematic control for articulated

β	Abnorman-1	Abnorman-2	result
1.0	0.048	0.17	smooth
0.5	0.0017	0.047	smooth
10^{-5}	$\approx 0(10^{-14})$	$\approx 0(10^{-12})$	smooth
0.0	0.0	0.0	jerky

표 1. Average end-effector error. (normalized with respect to the total-link-length of each figure.)

figure motion editing. *Computer Graphics Forum*, 11(4):189–202, 1992.

[4] Armin Bruderlin and Lance Williams. Motion signal processing. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 97–104, August 1995.

[5] Kwang-Jin Choi and Hyeong-Seok Ko. On-line motion retargetting. In Bob Werner, editor, *Pacific Graphics '99 Proceedings*, pages 32–42, October 1999.

[6] Michael Gleicher. Motion editing with spacetime constraints. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, pages 139–148, 1997.

[7] Michael Gleicher. Retargeting motion to new characters. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 33–42. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.

[8] Jehee Lee and Sung Young Shin. A hierarchical approach to interactive motion editing for human-like figures. In *SIGGRAPH 99 Conference Proceedings*, Annual Conference Series. ACM SIGGRAPH, Addison Wesley, August 1999.

[9] David G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley Publishing Company, 1984.

[10] Ken Perlin. Real time responsive animation with personality. In *IEEE Transactions on Visualization and Computer Graphics*, pages 5–15, March 1995.

[11] Charles Rose, Brian Guenter, Bobby Bodenheimer, and Michael F. Cohen. Efficient generation of motion transitions using spacetime constraints. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 147–154, August 1996.

[12] Munetoshi Unuma, Ken Anjyo, and Ryozo Takeuchi. Fourier principles for emotion-based human figure animation. In Robert Cook, editor, *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 91–96, August 1995. ACM-0-89791-701-4.

[13] Andrew Witkin and Zoran Popovic. Motion warping. In Robert Cook, editor, *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 105–107, August 1995.

[14] Jianmin Zhao and Norman I. Badler. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics*, 13(4):313–336, October 1994.