

운동학적 접근 방법을 사용한 복잡한 인간 동작 질의 시스템[☆]

A Kinematic Approach to Answering Similarity Queries on Complex Human Motion Data

한 혁* 김 신 규** 정 형 수*** 엄 현 영****
Hyuck Han Shingyu Kim Hyungsoo Jung Heon Y. Yeom

요 약

대규모 인간 동작 데이터베이스에서 고차원의 데이터를 처리하는 것이 큰 비용을 요구하기에, 최근 데이터베이스 및 그래픽스 학계는 인간 동작 데이터 질의 및 접근에 큰 관심을 가지게 되었다. 특히, 인간 동작 데이터를 위한 효과적인 유사도(거리) 측정 방법이나 질의 처리는 여전히 많은 연구진들이 도전하고 있는 문제이다. 이에, 본 연구진은 SMOFinder라고 명명한 동작 질의 처리 시스템을 제안한다. SMOFinder는 새롭게 고안된 운동학적 거리 측정 그리고 적응적 프레임 세그멘테이션에 기반하는 효율적인 인덱싱을 사용하여 동작 질의를 처리한다.

이를 위해, SMOFinder에서는 인간 동작을 다연결 물리 운동으로 간주하고 새로운 가중치 Minkowski 함수를 정의했다. 또한, 효율적인 인덱싱을 위해 모든 프레임을 저장하지 않고 유사한 프레임들 중에서 대표 프레임을 뽑아서 저장하는 적응적 세그멘테이션을 고안했다. 그리고, 효율적인 검색을 위해 이들 대표 프레임들만 가지고 k-근접 이웃 질의를 수행하는 새로운 방법을 제안한다. 마지막으로, SMOFinder가 데이터베이스 용량이 크게 줄지만(1/25배), 검색 능력은 다른 시스템과 동일하거나 우월하다는 것을 실험을 통해 보여주하고자 한다.

Abstract

Recently there has arisen concern in both the database community and the graphics society about data retrieval from large motion databases because the high dimensionality of motion data implies high costs. In this circumstance, finding an effective distance measure and an efficient query processing method for such data is a challenging problem. This paper presents an elaborate motion query processing system, SMOFinder (*Similar Motion Finder*), which incorporates a novel kinematic distance measure and an efficient indexing strategy via adaptive frame segmentation.

To this end, we regard human motions as multi-linkage kinematics and propose the weighted Minkowski distance metric. For efficient indexing, we devise a new adaptive segmentation method that chooses representative frames among similar frames and stores chosen frames instead of all frames. For efficient search, we propose a new search method that processes k-nearest neighbors queries over only representative frames. Our experimental results show that the size of motion databases is reduced greatly (x1/25) but the search capability of SMOFinder is equal to or superior to that of other systems.

☞ keyword : Motion Retrieval, multi-linkage kinematics, Weighted Minkowski, distance, M-tree, KNN Search
모션 검색, 다연결 운동, 가중치 Minkowski 거리, M-나무, k 근접 이웃 검색

* 정 회 원 : 서울대학교 컴퓨터공학부 박사과정
hhyuck@dclslab.snu.ac.kr

** 정 회 원 : 서울대학교 컴퓨터공학부 박사과정
sgkim@dclslab.snu.ac.kr

*** 정 회 원 : 서울대학교 컴퓨터공학부 박사과정

jhs@dclslab.snu.ac.kr

**** 정 회 원 : 서울대학교 컴퓨터공학부 교수
yeom@dclslab.snu.ac.kr

[2008/08/13 투고 - 2008/08/26 심사 - 2009/01/30 심사완료]

☆ A preliminary version of this paper[1] was presented at ICCT 2008, Busan, Korea.

1. Introduction

Recently motion retrieval from large human motion databases has gained growing attention from database researchers due to the challenging task of indexing and querying high dimensional human motion data. Motion data is generated using the motion capture (mocap) technique, and a massive amount of mocap data is accessible to those who want to index, analyze, and query these datasets. Building a system for querying complex human motions would provide significant benefit in this area.

Much research work [12, 16] deals with this problem by viewing human motion as high dimensional trajectories. They view each segment or frame as a numerical vector; as a result, an entire set of continuous frames is represented by multidimensional time series. Existing distance measures [5, 11] on time series data can be applied to these converted datasets. The drawback of using these distance measures is inaccuracy, due to the high dimensionality of motion data.

We develop SMoFinder, a system for querying complex human motions using a kinematic approach. In our system, we tackle the problem of indexing and querying human motion data from a kinematic viewpoint, instead of a categorical approach [13]. The kinematic approach matches perfectly to an existing mocap technique: all movements of human bones are represented by a 4-dimensional motion matrix. Considering human skeleton (or kinematic) chains, we can see that chains of bones are not completely independent; i.e., all human bones are (uni/bi)laterally linked to other bones. Using this fact, we define a feature vector for each frame, and design a distance

measure that can capture motion similarity more accurately.

We represent each frame of human motion as a multidimensional numeric vector whose values are angles between a given reference vector and a vector representing each bone with respect to modeling coordinates. The reference vector is a perpendicular vector to the plane created by three body points, which constitute a rigid body. Our feature vector is basically a vector of relative angle movements to the incident bone. Unlike the naive method of extracting a feature vector from human motion by using the 4-dimensional coordinates, our method can capture the similar human motions accurately, irrespective of the size of a human actor¹⁾.

To capture similar motion in a more fine-grained way, we then define a similarity measure. The similarity measure of our system exploits the aforementioned observation. The basic principle of our distance function is based on the Minkowski distance. For a point (x_1, x_2, \dots, x_n) and a point (y_1, y_2, \dots, y_n) , the Minkowski distance of order p (p -norm distance) is defined as : the L1 distance (1-norm distance or the Minkowski distance of order 1) is $\sum_{i=1}^n |x_i - y_i|$, the L2 distance (2-norm distance or the Minkowski distance of order 2) is $\sqrt{\sum_{i=1}^n |x_i - y_i|^2}$, and p -norm distance is $\sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p}$. Our approach is based on the L2 distance.

However, the L2 distance between two equally-weighted feature vectors cannot be directly

1) This is a scale-free property.

applied to measuring the difference between the L2 distance generates false positive similar frames. We therefore adapt the L2 distance in order to capture semantically correct similar motions while reducing false positive frames. This is done by using the characteristics of linked human skeleton chains. We use the weighted L2 distance $wL_2 = \sqrt{\sum w_i |x_i - y_i|^2}$ where $\sum w_i = 1$ that will be explained in the Section 3.2. The intuition behind the wL2 distance measure is that a bone with a longer kinematic chain has a larger effect on the change of an entire pose than a bone with a shorter kinematic chain, as a consequence, we give higher weight to bones with a longer kinematic chain.

Due to the high dimensionality of a feature vector, SMoFinder cannot exploit existing indexing techniques (e.g. R-tree or its variants) for the dynamic time warping (DTW) or the discrete Fourier transform (DFT). However, because our similarity measure is based on the L2 distance, we can adopt an indexing technique based on the arbitrary metric. Moreover, we devise the compression strategy to use the storage efficiently and process a motion query quickly.

The graphical user interface (GUI) provides an effective way to generate a motion query. Generally, motion retrieval systems provide two types of query models: textual descriptions such as "a kick of the right foot" and short motion clips[13]. We focus on the second model. The GUI requests the process of the generated motion query and lists results of the query, and play the similar part of the selected clip to the query clip.

In this paper, we present SMoFinder, a novel motion retrieval system that finds similar or

matching motions to a query motion regardless of the complexity of the query motion. Contrary to [6, 13], our work takes into account kinematic elements for the feature extraction, and processes an approximate k-nearest neighbor (kNN) instead of an exact kNN query²⁾.

The rest of the paper is organized as follows. We review related work in Section 2. In Section 3 we present the architecture of the proposed system: feature extraction, distance measure, adaptive segmentation, indexing, and query processing. Section 4 contains the evaluation of SMoFinder. Finally, We conclude this paper in Section 5.

2. Related Work

Large motion capture data (e.g. [2]) reinforces the demand for efficient indexing and retrieval methods. Such methods are necessary to efficiently retrieve logically related motions, which can then be processed via editing and morphing techniques.

Motion retrieval is closely related to similarity search in time series databases, which have been studied for over a decade. Given a distance metric and a query time sequence, the task is to search a database for time sequences whose distance to the query is either below a threshold ϵ or among the k smallest. Most proposed indexing approaches use the original GEMINI framework[8] proposed by Faloutsos et al, but suggest a different approach to the dimensionality reduction stage. First, a low-dimensional approximation is extracted

2) If the distance measure captures the similarity accurately, small difference in the distance is tolerable. In particular, smaller distance from the query motion does not mean more similar result always.

from each time series in the database. Example approximations include the Discrete Fourier Transform (DFT)[4], several kinds of Wavelets (DWT)[14], the average values in adjacent windows[10], and bounding boxes[17]. Next, a distance metric is defined between the time series. Finally, the approximated signals are stored in a spatial data structure such as an R-tree[9].

Here, possible distortions of the time axis constitute a major problem in comparing related time series, usually solved by means of dynamic time warping (DTW). DTW is a technique that is traditionally associated with speech recognition. DTW, however, is cost-intensive in computing time and memory. To address this problem, we use a mixture of a segmentation technique and a dimensionality reduction method.

In [6] and [13], discretization of time series has been studied and one high dimensional categorical feature vector is extracted from one frame of human motion data by thresholding the logic-oriented geometric relations. A human motion is defined by a sequence of geometric relations. For example, the distance between two legs is used to define a "walking" motion. A motion can be defined as a walking motion when a sequence of distance between two legs is (-1m, 0.5m, 0m, 0.5m, 1m). The problem of this approach is that all motions used in applications must be pre-defined as a form of geometric relations. For example, if only walking and dancing motions are defined, applications can not search "dancing" motions. Therefore, general relations between any two bones such angle or distance are needed, and this point is our motivation of our feature vector extraction scheme.

Due to high dimensionality of our feature

vector, SMoFinder adopts an M-tree[7] for indexing motions instead of an R-tree³⁾. The M-tree is a balanced tree that can deal with dynamic and high-dimensional data efficiently. If data in the M-tree meet properties of symmetry, non negativity, and triangle inequality, the M-tree can efficiently process range or k-NN query with novel pruning techniques. Since our feature vector is based on the Minkowski distance, our feature vector meets the three properties, and SMoFinder can exploit the k-NN query processing scheme of the M-tree to find similar motions to a query motion.

3. Motion Search System

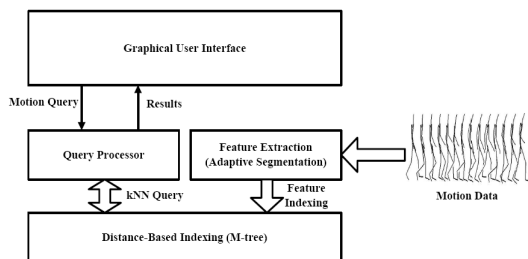
Figure 1 shows the overview of SMoFinder. There are two main parts, indexing and query processing. Firstly, high-dimensional feature vectors are extracted from motion databases, and adaptively segmented using the weighted Minkowski distance measure wL_2 . In SMoFinder, the segmented feature vectors are indexed by M-tree [7] that is efficient for comparing distances within motion data. Users then send motion query requests to the query processor, and M-tree finds the approximate k nearest neighbors from M-tree based index. All these components are explained in the following subsections.

3.1 Feature Vector

In this section, we define a feature vector describing complex motions effectively. Previous work [6, 13] described motions by using binary

3) The problem of the R-tree related to high-dimensionality is known as the curse of dimensionality.

features, which are simple and efficient for processing queries but are not able to depict complex motions.



(Figure 1) Overview of SMOFinder

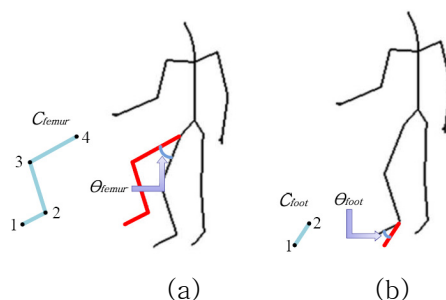
$FV = (\theta_1, \theta_2, \dots, \theta_n)$ where n = number of bones

Our feature vector (FV) consists of angles between each bone and a reference (or root) vector; θ_i represents an angle between a bone i and a reference vector. Figure 2 illustrates how to define the reference plane and feature vectors. The reference plane in Figure 2(b) is defined by three points, which are the root joint, the left hip joint, and the right hip joint. Because their lengths and the angle between the left and right hip joints are fixed, these three points constitute a rigid plane. The reference vector \vec{R} is now created by applying the outer product to two vectors \vec{B} lhipjoint and \vec{B} rhipjoint from these points, so it is perpendicular to the reference plane. An angle θ_i in a feature vector FV can be obtained from the one between the reference vector \vec{R} and the vectors of each bone (e.g. θ_{elbow} between \vec{R} and \vec{B} elbow in Figure 2(c)). In SMOFinder, these angles have radian values, and the feature vectors have twenty eight dimensions. The intuition behind the usage of a relative angle for a feature vector is to resolve the scale-free property. If two

different actors take the same pose, the geometrical coordinates of these motion frames have completely different values. These frames, however, can have a very similar angle θ_i for each bone in our feature vector framework.

3.2 Distance Measure and Adaptive Segmentation

Once we define a feature vector for each frame, we need to define a proper distance metric to measure the similarity between two high dimensional motion frames. A naive way of measuring a similarity between two vectors can be done by using the L_2 norm. The L_2 norm, however, has a deficiency in detecting false positive similar motion frames. Figure 3 shows two motion frames that change two different joints: a right femur (the pose to raise a leg (Figure 3(a))) and a right foot (the pose to reach a foot (Figure 3(b))). Even though these two frames look significantly different, the L_2 norm may say that these frames have the same distance value to the reference frame.



(Figure 3) The motivation of the wL_2 distance.

It, therefore, is very important and difficult to find an effective distance measure for high-

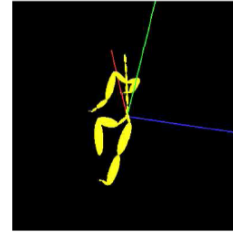
dimensional motion data. To capture these types of hard-to-catch differences in motion frames, we introduce weighting factors into the L2 distance measure:

$$D(FV_1(x_1, \dots, x_n), FV_2(y_1, \dots, y_n)) = \sqrt{\sum_{i=1}^n w_i |x_i - y_i|^2}$$

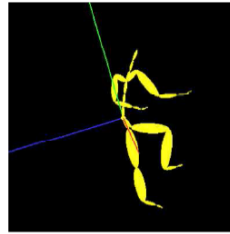
$$\text{where } \sum_{i=1}^n w_i = 1 \text{ and } w_i = \frac{|C_i|}{\sum_{i=1}^n |C_i|}$$

FV_1 and FV_2 are the two feature vectors, and w_i is a weighting factor. C_i and $|C_i|$ denote a kinematic chain starting from a joint i and its length respectively. To measure the importance of each bone, we assign w_i to each bone i , and all weighting factors are normalized. $|C_i|$ indeed represents the relative impact on the change of an entire human pose. As Figure 3 shows, θ_{femur} makes a bigger change to the overall pose than θ_{root} . To this end, the bone which has a longer kinematic chain has a larger weighting factor. A kinematic chain means a sequence of bones that starts from end bone (i.e., right/left toes, right/left fingers, and head) to the reference bone. For example, a kinematic chain for the left femur and left foot are "left toe - left foot - left tibia - left femur" and "left toe - left foot", respectively. Both chain sizes are 4 and 1. Therefore, the weighting factor for the left femur is larger than that for the left foot.

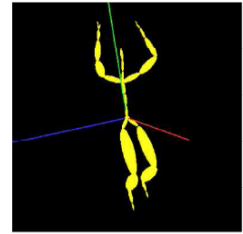
Figure 4 shows weighted L2 distances from the reference frame (Figure 4(a)). Figure 4(a) and 4(b) show the pose to climb up (similar frames), and Figure 4(c) to jump up (different frame). Each frame comes from different motion data. As shown above, our weighted L2 distance measure effectively reveals the differences between motions. The distance of Figure 4(a) and 4(c) is



(a) reference frame



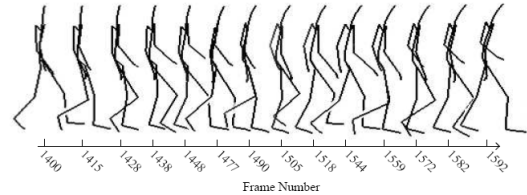
(b) distance : 0.07



(c) distance : 0.22

(Figure 4) Examples wL_2 values

three times larger than that of Figure 4(a) and 4(b).



(Figure 5) Segmentation of walking motion. Distance threshold(δ) = 0.1

Since consecutive frames are very similar, the distance between their feature vectors is very small. A good way to improve the efficiency of similarity search is to choose representative frames from each motion and store the selected frames, LandMark frames, instead of all frames. Once a frame is chosen to be a LandMark frame, the frame is used to represent all successive frames for which the distance from the LandMark frame is less than a user-defined threshold (δ). The LandMark frame is similar to the convex in [6].

Figure 5 shows an example of our segmentation technique with $\delta = 0.1$. Once the 1400th frame in the figure is chosen to be a LandMark frame, the LandMark frame represents successive frames of which the frame number ranges from 1401 to 1414 since the L2 distance between the 1400th frame and the successive frames is less than 0.1. Then, the 1415th frame becomes a new LandMark frame since the L2 distance between the 1400th frame and the 1415th frame is more than 0.1.

3.3 Indexing

We use the M-tree[7] data structure for indexing LandMark frames. An M-tree, a dynamic version of Vp-tree[18], is an index structure for the efficient resolution of similarity queries on complex objects. Query processing over an M-tree is based on an arbitrary metric, i.e. a distance function that satisfies the positivity, symmetry, and triangle inequality properties. Since our distance function satisfies these properties, an M-tree is an appropriate data structure for indexing frames. A 28-dimensional feature vector of a frame and the combinational form of the motion id and frame number are used as the key and a value of each M-tree entry. M-tree leaves for two consecutive frames are linked to each other, which link makes SMoFinder efficiently process a special type of motion query such as "display next(/previous) 200 frames after(/before) the query clip."

3.4 Searching Motions

Searching motions means to find a time series that is similar to the time series of the query clip. SMoFinder provides the approximate nearest neighbor (ANN) query or the approximate k-NN

query. Due to the properties of high dimensionality[6] and our index structure, an indexing technique on DTW or DFT distances cannot be applied to SMoFinder.

Instead we use the dynamic programming method. First, SMoFinder transforms a query clip (t frames) into m LandMark frames with the same threshold (δ). It then generates the kNN query for the first and the last LandMark frames, and processes k-NN queries with the user-defined threshold ε ($>\delta$). When processing the queries, it applies a pruning technique to search similar frames efficiently; a node f_m is pruned if $D(f_m, f_q) > 2 \times \delta$. Since one LandMark frame represents all consecutive frames with $D(f, f_{LandMark}) < \delta$, the constraint is $2 * \delta$ instead of δ .

After processing the kNN queries of the first and the last LandMark frames, SMoFinder retrieves motions that contains frames (t_f and t_l frames) that are similar to the first and the last LandMark frames of the query⁴⁾. FastDTW[15] which is an $O(N)$ algorithm calculates the distance between the query clip and result clips, and we have approximate kNN results. In SMoFinder the distance between two consecutive frames is constant.

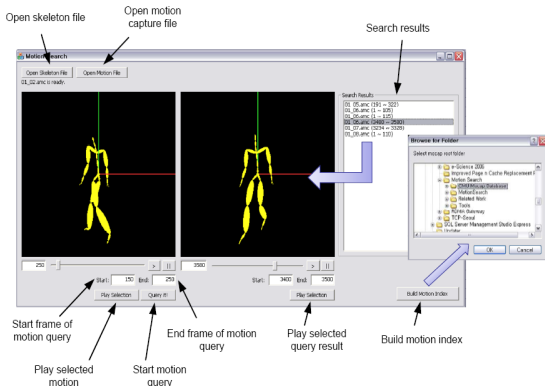
4. Evaluation

In this section, we will show the efficient and effective motion retrieval of our SMoFinder system with data from the CMU Motion Capture Database[2]. SMoFinder is built with Visual C++ and OpenGL. The number of motions is 2434 and the raw motion data size is over 2GB.

4) Distance between t_f (t_l) and the first(last) LandMark frame is less than ε .

4.1 User Interface and Motion Query Generation

Figure 6 shows the user interface of SMoFinder. The major functionalities of SMoFinder are to build the motion index and to search similar motions. The motion index is built by just selecting a directory which contains motion data files. Currently, SMoFinder supports only Acclaim's ASF/AMC format[3]. Building index is a time consuming job. But this task is executed only once at the initialization time and need not be repeated for every retrieval.



(Figure 6) Graphical User Interface of SMoFinder

SMoFinder provides an easy tool to make motion queries. Motion queries are created by extracting specific ranges of motion files. This method might limit the contents of queries within the scope of existing motions. However, making queries from empty space is more difficult than this method due to the high dimensionality of motion data. If someone writes 10 frames of twenty eight dimensional motion queries by hands, it might take over an hour, including revising motions naturally. Thus, we can say that the

method which SMoFinder provides is more user-friendly.

4.2 Effect of Adaptive Segmentation

From Section 3.2, we can see that the space requirement for a LandMark-based time series is less than that for the original motion sequence. Table 1 shows the total index size of feature vectors with or without the adaptive segmentation. In spite of the significant decrease in data size, SMoFinder can keep high search quality. In fact, the difference between one frame and the frame after the next is similar to that between two consecutive categorical features in [5, 12]; this observation validates that our segmentation technique is more fine-grained than the categorical feature extraction.

(Table 1) Index size regarding to adaptive segmentation

	With segmentation	Without segmentation
size	44.7MB	1112MB

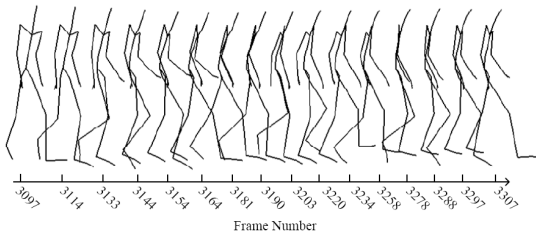
4.3 Similar Motion Search

In this section, we present complex motion search experimental results with two queries. The first query shows the ability to distinguish between walking and running. The relative positions of legs and arms are very similar in these two motions. To this end, a fine-grained way of modeling motions is essential. Figure 7 shows one of the results of searching motions with the query of Figure 5. As it shows, the motion of Figure 7 is very similar to that of Figure 5.

The second query is more complex than the

first one. The second query includes acrobatics, sports, and dances. These motions are hard to describe accurately, and it is much harder to measure distances from such motions. If these motions are not managed in a fine-grained way, many false positive cases can be retrieved. SMoFinder and its feature vectors are enough to retrieve complex motions effectively.

However, content-based indexing approaches such as [6] and [13] can not process these queries if feature vectors for these motions are not defined in the system. Our search performance is not as good as results in dimensionality reduction based techniques such as [5] and [10]. However, they generate more false positives than our approach.



(Figure 7) Result motion (Query motion : Figure 5)

4. Conclusion

In this paper we presented SMoFinder, a novel system for similarity search in human motions. Our system is based on the feature extraction and the approximate kNN query processing engine. In order to extract features from human motions, our system takes into account kinematic elements. We use segmentation and M-tree to index features from human motions. Our query processing engine is based on the weighted Minkowski distance between LandMark frames. SMoFinder benefits

from these characteristics in terms of storage efficiency and effectiveness of query processing. In the near future, we will extend our system for mining motifs and unusual motions.

Acknowledgement

This study was supported by the Seoul R&BD Program (10561), Seoul, Korea. The ICT at Seoul National University provides research facilities for this study.

참 고 문 헌

- [1] Hyungsoo Jung, Hyuck Han, Shingyu Kim, Heon Y. Yoem. SMoFinder: A System for Querying Complex Human Motions Using a Kinematic Approach. In Proceedings of the 2008 Third International Conference on Convergence and Hybrid Information Technology, pages 640~643, Busan, Korea, 2008.
- [2] CMU graphics lab motion capture database. <http://mocap.cs.cmu.edu/>.
- [3] Acclaim. ASF/AMC file specifications. <http://www.darwin3d.com/game/dev/acclaim.zip>.
- [4] R. Agrawal, C. Faloutsos, and A. N. Swami. Efficient similarity search in sequence databases. In Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms, pages 69 - 84, London, UK, 1993. Springer-Verlag.
- [5] L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pages 491 - 502, New York, NY, USA, 2005. ACM.

- [6] Y. Chen, S. Jiang, B. C. Ooi, and A. K. H. Tung. Querying complex spatio-temporal sequences in human motion databases. In ICDE, pages 90 - 99, 2008.
- [7] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *The VLDB Journal*, pages 426 - 435, 1997.
- [8] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *Proceedings of the 1994 ACM SIGMOD international conference on Management of data*, pages 419 - 429, New York, NY, USA, 1994. ACM.
- [9] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pages 47 - 57, New York, NY, USA, 1984. ACM.
- [10] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Locally adaptive dimensionality reduction for indexing large time series databases. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 151 - 162, New York, NY, USA, 2001. ACM.
- [11] E. Keogh, T. Palpanas, V. B. Zordan, D. Gunopulos, and M. Cardle. Indexing large human-motion databases. In *Proceedings of the 30th international conference on Very large data bases*, pages 780 - 791. VLDB Endowment, 2004.
- [12] L. Kovar and M. Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph.*, 23(3):559 - 568, 2004.
- [13] M. Müller, T. Röder, and M. Clausen. Efficient content-based retrieval of motion capture data. *ACM Trans. Graph.*, 24(3):677 - 685, 2005.
- [14] K. pong Chan and A. W.-C. Fu. Efficient time series matching by wavelets. In ICDE, pages 126 - 133, 1999.
- [15] S. Salvador and P. Chan. Fastdtw: Toward accurate dynamic time warping in linear time and space. In *KDD Workshop on Mining Temporal and Sequential Data*, 2004.
- [16] H. T. Shen, B. C. Ooi, and X. Zhou. Towards effective indexing for very large video sequence database. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 730 - 741, New York, NY, USA, 2005. ACM.
- [17] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh. Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 216 - 225, New York, NY, USA, 2003. ACM.
- [18] P. N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the 4th annual ACM/SIAM Symposium on Discrete algorithms*, pages 311 - 321, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.

● 저 자 소 개 ●



한 혁

2003년 서울대학교 컴퓨터공학부(공학사)
2006년 서울대학교 대학원 컴퓨터공학부(공학석사)
2006년~현재 서울대학교 대학원 컴퓨터공학부 박사과정
관심분야 : 데이터베이스, 분산시스템, 운영체제.
E-mail : hhyuck@dcslab.snu.ac.kr



김 신 규

2006년 서울대학교 컴퓨터공학부(공학사)
2008년 서울대학교 대학원 컴퓨터공학부(공학석사)
2008년~현재 서울대학교 대학원 컴퓨터공학부 박사과정
관심분야 : 데이터베이스, 네트워크, 운영체제.
E-mail : sgkim@dcslab.snu.ac.kr



정 형 수

2002년 고려대학교 기계공학과(공학사)
2004년 서울대학교 대학원 컴퓨터공학부(공학석사)
2004년~현재 서울대학교 대학원 컴퓨터공학부 박사과정
관심분야 : 데이터베이스, 분산시스템, 운영체제, 네트워크 etc.
E-mail : jhs@dcslab.snu.ac.kr



염 현 영

1984년 서울대학교 전산학과(이학사)
1986년 Texas A&M University 대학원 전산학과(이학석사)
1992년 Texas A&M University 대학원 전산학과(이학박사)
1993년~현재 서울대학교 컴퓨터공학부 교수
관심분야 : 데이터베이스, 분산시스템, 멀티미디어 시스템, 트랜잭션 처리 etc.
E-mail : yeom@dcslab.snu.ac.kr