

Performance Analysis on an Efficient Human Motion Database with Various Motion Representations

S. M. Ashik Eftakhar, Joo Kooi Tan, Hyoungeop Kim, and Seiji Ishikawa

Department of Control Engineering, Kyushu Institute of Technology, Japan
E-mail: {ashik, etheltan, ishikawa}@ss10.cntl.kyutech.ac.jp

Abstract — In this paper, our proposed structured human motion database is adopted for different motion representations. The motions are first represented as a sequence of frames of 2D images, which were compressed using three recognized motion representation techniques: Exclusive-OR, MEI (Motion Energy Image), and MHI (Motion History Images). The representation is a 2D feature image. The feature image is compressed by characterizing the eigenvectors. A complete vector space called an Eigenspace is constructed that represents the image feature vectors for the feature image. The motions are indexed using the projections onto the eigenspace. For the purpose of efficient searching within the database, our proposed B-Tree Motion Database is created and maintained. The comparative performance evaluations for the aforesaid representations were investigated and satisfactory performances (about 90% recognition rate and smaller searching time) were realized for all of the cases using our proposed motion database structure.

Index Terms — Motion database, motion representation, motion compression, eigenspace, indexing, B-Tree.

I. INTRODUCTION

A motion is, in a concise sense, any kind of activity performed by any subject. The activity might include any task, or simply any movement of the object. Human motion signifies the actions or movements performed by any human being. The researches involved in the recognition of human motion are considered to be a highly challenging task in computer- and robot-vision areas. With the development of computer vision system, the task of human motion/ activity recognition is getting huge significance in recent times. The motivation behind this kind of researches is the variety of applications in surveillance, discovering persons' physical problems in a medical system, aerobics, video conferencing, motion understanding, man-machine interfaces like robotics, biomechanics, virtual/ mixed reality, and in many other real-life applications[1][2]. There are a wide variety of human motion analysis techniques which are beyond this context. Aggarwal and Cai highlighted various forms of human motion representation and recognition [2]. Analyzing the techniques mentioned in [2], the field of motion representation and recognition can be briefly classified into two classes of methods: model-based and appearance-based methods [3]. Model-based methods

basically deal with the development of human 3D model for the task of representation. 3D human models are generally represented in terms of generalized cones, elliptical cylinders, and spheres [4]. For these geometric modeling a number of parameters is needed to be computed. For the complexity of computation, model-based methods are less preferable than appearance-based methods because of the simple modeling involved in appearance-based method. It deals with the 2D representation of motion rather than 3D. The motions are supposed to be composed of postures (appearances) that are somehow transformed into a suitable form or model for representation and recognition. Some recognized representations include Hidden Markov Model [5], Eigenspace [6], Body parts-based modeling [7], Motion Energy Images and Motion History Images [8], Exclusive-OR Technique [14], Directional MHI [9], and many others. The representation of eigenspace technique is much compact, as well as, suitable and applicable representation. It is identified as an image compression or coding technique. It is based on Principle Component Analysis (PCA), deviation of Karhunen-Loeve Transform [10]-[11]. Until now this technique was adopted successfully in many posture, gait, and motion recognition researches [3][6][12]. Different researchers have adopted different concepts of representation and recognition. For the reliable and real-life application of these representations and recognition strategies, the efficiency of those strategies in terms of recognition rate, as well as, searching complexity is a major issue. Recognition rate is simply the rate of recognized motions, whereas searching complexity is considered as the time that is employed for searching similar motions within the motion database.

However, the motion recognition strategies adopted earlier used to maintain a simple database capable of linear searching within that. But such searching is a brute-force searching approach that takes much time for searching in case of enormous motion data registered in the database. At this point, it is recommended to deduce some technique to reduce the searching by structurizing the searching technique. But in order to structurize the searching, construction of structured database is mandatory. Emphasizing on this point, a structured database is also proposed recently with human poses and motions [13]-

[14]. But the registration of huge amount of data, and high-speed searching with excellent performances is not yet been analyzed fruitfully. In [15]-[16], the balanced tree structure called a B-Tree was proposed as the structured database structure. Moreover, the B-Tree structure was successfully employed in many other researches, e.g., [15]-[16]. The reason behind the employment of B-Tree structure is its data arrangement. However, other database structure was also adopted in the motion analysis researches [17]-[18]; most of those were computationally complex to handle the database management.

In this research, successful analysis of motion database is accomplished by proposing a database capable of handling huge data with the ability of high recognition performance. The performance of the proposed technique is evaluated suitably and lucratively. Various representations are used to validate the performances and flexibility of our proposed method. The database development flowchart of the system is shown in Fig. 1.

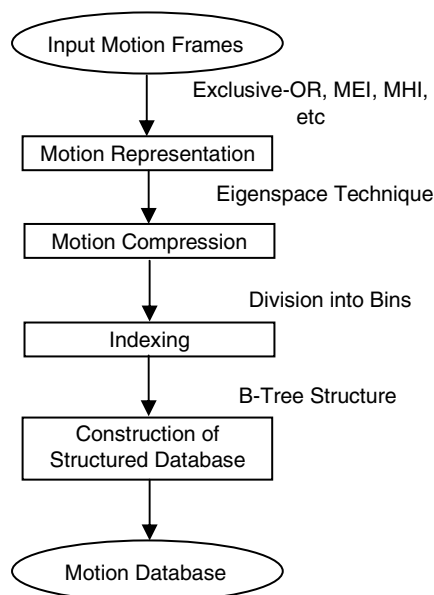


Fig.1. Database Development Flowchart.

II. MOTION REPRESENTATION

Motion representation is a form of characterizing a motion for the computer to understand and to use the motion for recognition. In this sense it is a very crucial task. However, among many motion representations we have chosen three standard representations: Exclusive-OR, Motion Energy Image (MEI), and Motion History Image (MHI). The overview of these representations will be discussed in this section.

A. Exclusive-OR (XOR) Representation

Rather taking all the motion frames into account, the Exclusive-OR (XOR) operations are performed between consecutive frames and the cumulative XORed form of the frames are referred to as XOR Image [14]. This is simple, effective, and fast generating motion representation method. Considering m, h, c representing motion, persons, camera-directions, and f, U denoting original motion frame and XOR frame, then (1) presents the complete XOR operation on the motion frames.

$$\begin{aligned}
 U_c^{m,h}(2) &= f_c^{m,h}(1) \text{ XOR } f_c^{m,h}(2), \\
 U_c^{m,h}(r) &= U_c^{m,h}(r-1) \text{ XOR } f_c^{m,h}(r), \\
 U_c^{m,h} &\equiv U_c^{m,h}(R).
 \end{aligned} \quad (1)$$

B. Motion Energy Image (MEI)

Bobbic and Davis [8] represented *Motion Energy Image (MEI)* as the region *where* there is motion. This information can be used to identify the motion occurrence and viewing condition. It is defined as the cumulative binary image of the motion regions extracted between the consecutive motion frames. Let, $I(x, y, t)$ be an image sequence and $D(x, y, t)$ be a binary difference image. Then binary MEI $E(x, y, t)$ is defined by (2).

$$E_\tau = \bigcup_{i=0}^{\tau-1} D(x, y, t-i) \quad (2)$$

Here, τ is the temporal extent which is critical to define. But for the flexibility of the value of τ , it can be taken as the maximum gray level pixel value 255 [9].

C. Motion History Image (MHI)

Bobbic and Davis [8], like MEI, also represented *Motion History Image (MHI)* as a frame-based temporal template for human motions. As the name implies, this form of motion representation keeps track of the motion history, i.e. representing *how* the motion is moving along a certain period of time. Let H be the pixel intensity function of the temporal history of motion at a particular point. The function is represented in a simple way in (3).

$$H_\tau(x, y, t) = \begin{cases} \tau & \text{if } D(x, y, t) = 1 \\ \max(0, H(x, y, t-1) - 1) & \text{Otherwise} \end{cases} \quad (3)$$

The function returns a scalar value, and according to the function, in the generated image the more recently moving pixels are brighter than past moving pixels. However, both the MEIs and MHIs are represented as vector-images (where and how motion is moving) that can be matched against stored representations of known movements.

In this research, we employed the aforesaid three representations for our proposed technique. However, the motion that is represented as a sequence of 2D frames is transformed into a suitable form to signify the region of interest as well as to eliminate unnecessary and unwanted noise elements. Therefore, the motionless or static region of the motion frames is excluded from each frame. Static region may include motionless part, noise, uneven edge pixels, etc. This portion can be treated as ignorable portion; so it is primarily removed from each motion frames using (4).

$$I_h(n) = \begin{cases} I_h(n) & \text{if } \exists k I_h(n) \neq I_k(n), h \neq k \\ 0 & \text{Otherwise} \end{cases} \quad (4)$$

Here, $I_h(n)$ and $I_k(n)$ are the n -th pixel value of h -th and k -th frame, respectively; $n \in N$ and $k \in F$, where N is the total number of pixels and F is the total number of motion frames.

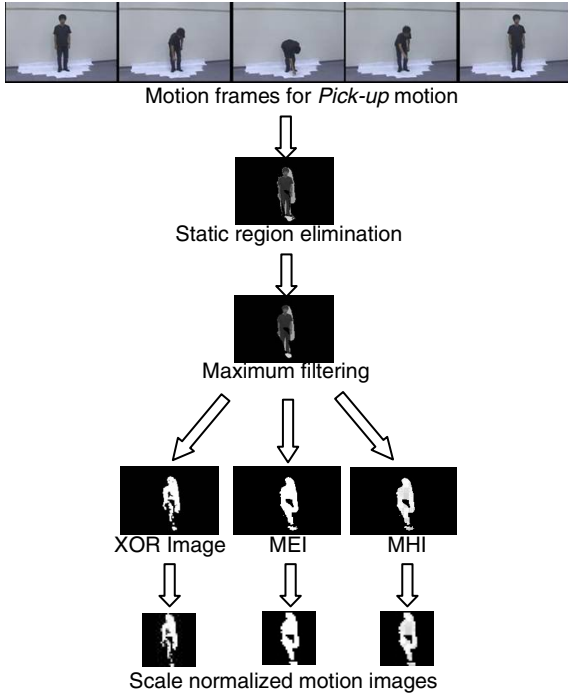


Fig. 2. Generation of different motion Representations.

After the elimination of static portion, the images are filtered using maximum filter to take significant regions into account. Then motion representations are adopted and the resultant motion images are generated step-by-step (Fig. 2). Next, the motion images generated from the input motion frames are *compressed* by projecting the motion images onto the *eigenspaces*.

III. MOTION COMPRESSION

Motion compression is an important step in the human motion recognition system. It refers to either motion coding, or compressed data that are extracted from the motion. For example, extracting the features and parameter values from the motion are forms of motion compression. The parametric eigenspace representation is considered as an excellent motion compression technique. An *Eigenspace* is a hyperspace or a parametric feature space representing an image as a point on the hyperspace. It is a modified form of Karhunen-Loeve Transform which is basically used to derive relationship among different random variables. It has been used as a standard compression technique for its simplicity of use. In practice, a large set of learning motions is needed to be projected onto the parametric eigenspace by finding featuring eigenvectors. For each *motion image* (*XOR image*, *MEI*, or *MHI*) I_m ($m = 1, 2, \dots, M$), an image matrix $\hat{x}_m = (x_1, x_2, \dots, x_N)$ is defined and normalized as $\|x_m\| = 1$. A data matrix X is obtained by subtracting the average image c from each motion image set as follows,

$$X = (x_1 - c, x_2 - c, \dots, x_M - c) \quad (5)$$

The average image c is defined by,

$$c = \frac{1}{M} \sum_{m=1}^M x_m \quad (6)$$

The image matrix X is $N \times M$, where M is the total number of motion images, and N is the total number of pixels in each image. To compute eigenvectors of the image set (motion set), the *covariance matrix* Q is defined as (7).

$$Q = XX^T \quad (7)$$

$$\kappa = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^N \lambda_i} \quad (8)$$

The eigenvalues and corresponding eigenvectors are calculated from the $N \times N$ covariance matrix Q by solving the eigenvalue problem. However, Principal Component Analysis (PCA) strategy can be used here to reduce the dimensions of the eigenspace. Among N eigenvectors, k most prominent eigenvectors (e_1, e_2, \dots, e_k) are chosen to create an eigenspace ES consisting of the learning motions using the metric expressed in (8). In (8), the value of κ is taken to be greater than or equal to 0.80. The eigenvectors for which the variances are more, those are chosen as the prominent eigenvectors. Each learning motion is projected onto the eigenspace ES as a point g_m by,

$$g_m = (e_1, e_2, \dots, e_k)^T x_m \quad (9)$$

For each camera direction, separate eigenspaces are created by projecting corresponding directional motions onto the eigenspace using (9). However, a *global eigenspace* is also constructed with all the learning motions. It confirms the decision about the most similar motions.

IV. MOTION INDEXING

Indexing is the processing denoting data in a suitable form so that it can be easily allocated into the database for further query. However, in this context, indexing involves the task of generating index from the points inside the eigenspace corresponding to each motion. The indexes are stored within the database afterwards. For the task of indexing the dimensions of the eigenspaces is taken as an important cue. The eigenspaces are uniformly divided into several subdivisions. Each eigen-axis e_k ($k = 1, 2, \dots, K$) is separated into S sections each having equal length L . A hypercube with length L along each eigen-axis of an eigenspace is referred to as a *bin*. Each bin is assigned a number from 0 to $S-1$. Along each eigen-axis, each motion point is assigned a number between 0 and $S-1$. Thus each motion point in the eigenspace is represented as K -digit S -nary number which is termed as an *index* (Fig. 3). With large number of motions, if the number of bins is small, there is higher possibility that more motion points will have an identical index and need much time to search similar motions. Conversely, if the number of bins is large, less motion points will be sought. The reason behind it is: each bin covers less number of motion points at the time of higher number of bins.

Digit-($N-1$)	Digit-2	Digit-1	Digit-0
($0 \sim K-1$)	($0 \sim K-1$)	($0 \sim K-1$)	($0 \sim K-1$)

Fig. 3. Index of N -digit K -ary number.

V. CONSTRUCTION OF STRUCTURED DATABASE

In this paper, a database is constructed capable of structured organization and fast query. As the large motion-database structure, B-Tree [19] data structure is employed in our proposed method. A B-Tree is a balanced tree that is capable of fast external memory searching. The B-Tree consisting of m descendents with height h is defined by $\tau(m, h)$ and each page can hold up to m keys. It has the following properties:

- (a) Each path from the root to leaf has the same length h .
- (b) Root is a leaf or has 2 to m descendents.
- (c) Each node has $\lfloor m/2 \rfloor$ to m descendents, except the root and leaves.

Within each page of the B-Tree, there are two kinds of data: a pointer and a key (Fig. 4). Pointer points to another page within the B-Tree, whereas the key stores the main information, i.e., the index. Considering the above properties, a B-Tree is constructed with the indexes. The generated index is stored in the B-Tree systematically. Those indexes are used for recognizing unknown motions in the recognition phase.

So far discussed database construction steps can be summarized as follows: First, for each camera direction, separate eigenspaces are created, then indexes are generated, and those are stored in a corresponding B-tree. The *global eigenspace* consisting of all the motions is used at the recognition stage. Separate eigenspaces arrange the motion data in an organized way and store much data within each B-Tree, at the same time we can still keep our searching faster. The structure of the page of B-Tree is shown in Fig. 5.

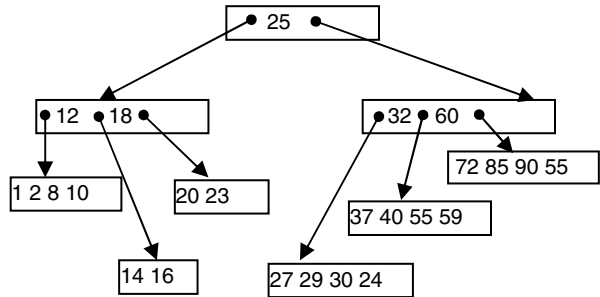


Fig. 4. A B-Tree structure in $(4, 3)$.

p_0	x_1	p_1	x_2	p_2	Unused Space
-------	-------	-------	-------	-------	-------	--------------

Fig. 5. Organization of a page of a B-Tree.

VI. SEARCHING STRATEGY

The most significant part for developing a recognition system is the searching strategy. Both excellent and faster recognition is the concern here. As the B-Tree is a well-structured database structure, the searching is simpler and less time-consuming. The following conditions [19] maintained by the B-Tree structure guarantees efficient searching within the database:

$$\begin{aligned}
 &(\forall y \in Keys(p_0))(y < x_1), \\
 &(\forall y \in Keys(p_i))(x_i < y < x_{i+1}); \quad i = 1, 2, \dots, l-1, \quad (10) \\
 &(\forall y \in Keys(p_l))(x_l < y).
 \end{aligned}$$

Here, the B-Tree assumes l keys and $l+1$ pointers within the page.

With the above structural arrangement, an index is searched within the B-Tree. If the query index resides within the database, it is found by simple *retrieval algorithm*. But for the case of the index not residing within the database, the most similar index corresponding to the most similar motion within the database is found. The pseudo code for finding the most similar index within the B-Tree after using the retrieval algorithm is as follows:

STEP-1: If query index $y > x_i$ and $Keys(p_i) = \text{NULL}$ and $Keys(p_{i,j}) = \text{NULL}$, calculate $\text{MinDist}(x_{i,j}, y)$ and $\text{MinDist}(x_{i+1}, y)$. Then return the minimum distance index.

STEP-2: If query index $y > x_i$ and $Keys(p_i) = \text{NULL}$ and $Keys(p_{i,j}) \neq \text{NULL}$, select the last index from $Keys(p_{i,j})$. Then calculate $\text{MinDist}(x_{i+1}, y)$ and $\text{MinDist}(x_{p_{i-1,j}}, y)$, and return the minimum distance index.

Besides the above, no other condition holds. The above algorithm finds the most similar index from the B-Tree.

VII. RECOGNITION

The recognition strategy is also simple, but effective. An unknown motion is tested for recognition. When an unknown motion comes, it is first represented as a sequence of image frames. Then *XOR image*, *MEI*, or *MHI* is obtained from the motion frames, and then projected onto each camera-directional eigenspace. An index is generated from the unknown motion for each eigenspace representing motion identity within the directional database. For each camera direction, the number of similar motions is obtained by searching the corresponding B-Tree by calculating Euclidian distances among the indexes. Thus we get several *candidate* motions. Those candidate motions are then projected onto the global eigenspace as g_m ($r=1,2,\dots,D$), where D is the number of directions. The unknown is projected as g_m . The most similar motion is calculated within the large motion database using (11):

$$d_m = \min_r \|g_{m_r} - g_m\| \quad (11)$$

Finally, the global eigenspace is used for finding the most similar motion.

VIII. PERFORMANCE ANALYSIS

The performance of our proposed structure was analyzed with three different motion representations: Exclusive-OR, MEI, and MHI. The leave-one-out cross-validation method was used with a set of 60 motions, including 4 persons, 5 motions (*Pick-up*, *Carry*, *Walking*, *Headache*, and *Stomachache*) and 3 camera-directions

(*Left*, *Right*, *front*). The concept of bins is introduced here to reduce the amount of searching within the database. It can be assumed that the more the number of bins, the less the amount of searching. Different number of bins along each eigen-axis was taken into account for performance evaluation.

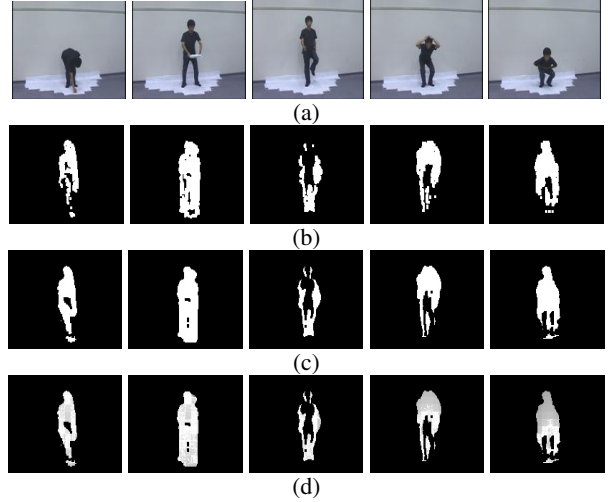


Fig. 6. Motion representations (frontal view) (a) Original motion frames, (b) XOR Image, (c) MEI, (d) MHI.

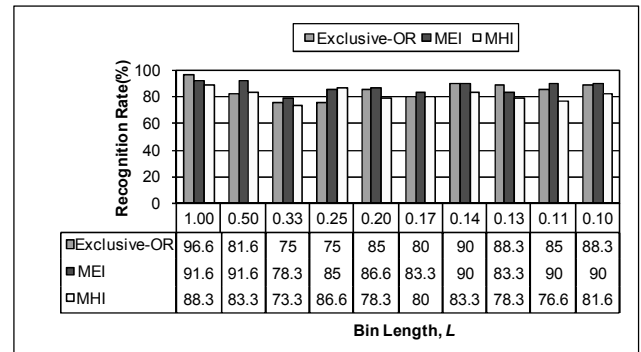


Fig. 7. Performance analysis using recognition rate.

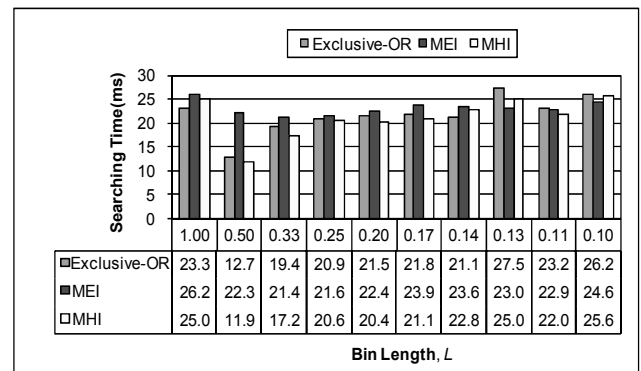


Fig. 8. Performance analysis using searching time.

For each motion representations, 60 motion images were generated with original size 160X107 pixels (Fig. 6) and were scale normalized to 32X32 pixels. Then the eigenspace was created for each camera direction by calculating eigenvalues and eigenvectors using SVD (Singular Value Decomposition) method. The performance was evaluated for each representation. The motions used for learning and testing were quite noisy; even for a single bin the recognition rate is not 100%. Still our proposed method shows excellent result with increased number of bins (Fig. 7). At bin length 0.14, 90% recognition rate was achieved. However, if the database searching time is considered, we see that at bin length 0.14 or less, the searching time is near about 22ms that is surely better than the brute-force search within a single bin (Fig. 8). The average recognition rate and searching time for Exclusive-OR, MEI and MHI are 84.5%, 87%, 81%, and 21.8ms, 23.2ms, 21.2ms, respectively. MEI shows the best recognition rate whereas MHI shows the least searching time with our experimental motions. The experiment was performed by taking the B-Tree parameter values $m=4$, $h=2$ operated on the PENTIUM IV, 2.8 GHz processor, 384 MB RAM Computer.

IX. CONCLUSIONS

The main contribution of this research is the achievement of an excellent recognition rate with large number of motions in the motion database keeping searching time shorter. The flexibility of our database structure is also considered by employing different motion representations. For simple motions MEI can show satisfactory results, whereas incorporating MEI and MHI together, the performance will truly be improved [8]. Other representation and compression methods can be fruitfully employed by creating index, and storing into the B-Tree database for recognition by fast searching with accommodating huge number of motions. The proposed method has, indeed, much potential to be applicable in the real-time human motion recognition applications.

ACKNOWLEDGEMENT

This work was partly supported by KAKENHI under grant 19700175, which is greatly acknowledged.

REFERENCES

- [1] D. Gavriila, "The visual analysis of human movement: a survey", *Computer Vision and Image Understanding*, Vol. 73, pp. 82-98, 1999.
- [2] J. K. Aggarwal and Q. Cai, "Human motion analysis: a review", *Computer Vision and Image Understanding*, Vol. 73, pp. 428-440, 1999.
- [3] T. Ogata, J. K. Tan, and S. Ishikawa, "High-speed human motion recognition based on a motion history image and an eigenspace", *IEICE Transactions on Information and Systems*, Vol. 89, Issue D(1), pp. 281-289, 2006.
- [4] K. Rohr, "Towards model-based recognition of human movements in image sequences", *CVGIP: Image Understanding*, Vol. 59, No. 1, pp. 94 – 115, 1994.
- [5] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using Hidden Markov Model", *In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 379-385, June, 1992.
- [6] H. Murase and S. K. Nayar, "Learning and recognition of 3D objects from appearance", *IEEE Qualitative Vision Workshop New York*, pp. 39-50, 1993.
- [7] Chih-Chang Yu, Jenq-Neng Hwang, Gang-Feng Ho, and Chaur-Heh Hsieh, "Automatic human body tracking and modeling from monocular video sequence", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 917-920, 2007.
- [8] A. F. Bobick, and J. W. Davis, "The recognition of human movement using Temporal Templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 3, pp. 257-267, 2001.
- [9] Md. Atiqur Rahman Ahad, T. Ogata, J.K. Tan, H.S. Kim, and S. Ishikawa., "Performance of multi-directional MHI for human motion recognition in the presence of outliers", *In Proc. Annual Conf. of IEEE Industrial Electronics Society (IECON)*, pp. 2366-2370, Nov. 2007.
- [10] E. Oja, *Subspace Methods of Pattern Recognition*, Research Studies Press, Hertfordshire, 1983.
- [11] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, London, 1990.
- [12] M. M. Masudur Rahman and Seiji Ishikawa, "Representing human postures/ motions using an eigenspace technique", *Proc. of International Conf. on Artificial Intelligence in Engineering & Technology*, pp. 232-236, 2002.
- [13] K. Kouno, J. K. Tan, and S. Ishikawa, "High-speed data retrieval in an eigenspace employing a B-tree structure", *Proc. of SICE-ACCAS International Joint Conference*, pp. 2717-2720, 2006.
- [14] J. K. Tan, K. Kouno, S. Ishikawa, H. S. Kim, and T. Shinomiya, "High speed human motion recognition employing a motion database", *Journal of Image & Electronic Society*, Vol. 36, No. 6, pp. 110-118, 2007.
- [15] T. Arndt, "A survey of recent research in image database management", *Proceedings of the IEEE Workshop on Visual Languages*, Vol. 4, Issue 6, pp. 92-97, 1990.
- [16] E. G. M. Petrakis and C. Faloutsos, "Similarity searching in medical image databases", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9, pp. 435-447, 1997.
- [17] C. Li and B. Prabhakaran, "Indexing of motion capture data for efficient and fast similarity search", *Journal of Computers*, Vol. 1, No. 3, pp. 35-42, June 2006.
- [18] F. Liu, Y. Zhang., F. Wu, and Y. Pan, "3D motion retrieval with motion index tree", *Computer Vision and Image Understanding*, Vol. 92, Issue 2-3, pp. 265-284, 2003.
- [19] R. Bayer and E. McCreight, "Organization and maintenance of large ordered indexes", *Acta Informatica*, Vol. 1, No. 3, pp. 173-189, 1972.