Online Action Recognition with Wrapped Boosting

Yu Nejigane, Masamichi Shimosaka, Taketoshi Mori, and Tomomasa Sato
Graduate School of Information Science and Technology
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
{nejigane,simosaka,tmori,tomo}@ics.t.u-tokyo.ac.jp

***Abstract*— In this paper, we propose wrapped boosting that is extension of boosting algorithm for robust online action recognition. Boosting algorithm is one of ensemble learning algorithm and is also known as a feature selector. In our previous work utilizing boosting, we achieved automatic feature selection and robust model-based action classifiers which had very small calculation cost based on posture information of human body joints. However, which joints we should allocate posture sensors to must be given by humans in advance. Our new learning framework of wrapped boosting provides not only automatic feature selection but also automatic sensor allocation to proper joints of humans for target actions. We evaluated our algorithm targeting gait motion based on motion data fetched by motion capturing system. In consequence, wrapped boosting was able to select proper joints to which limited sensors should be attached, and to construct more robust classifiers compared to constructing classifiers with all joints available. Classifiers constructed only with existing boosting algorithm were subject to overfitting to training data.**

I. Introduction

Recognizing human actions online is siginificant for realization of supporting humans by robots and of surveillance system about suspicious individuals. In recent years, a number of researches on action recognition[1], [2], [3] have been conducted introducing machine learning techniques such as hidden Markov model (HMM)[4], support vector machine (SVM)[5], conditional random fields (CRF)[6] in order to realize robust action recognition.

These methods with machine learning techniques have high recognition performance, but generally need many parameters to enhance recognition performance. And these parameters must be usually given by humans in advance. In addition, calculation cost for recognition is large, and then it is difficult to realize online motion recognition. Moreover, the methods make humans design important features for action recognition despite the designing such features is difficult and bothers humans.

In order to solve these problems, we adopted boosting algorithm[7] in our previous work[8]. Boosting generates simple classifiers called weak learners in stages which trained with the data. Each of weak learners is simple and low performance for recognition. However, boosting algorithm construct a robust classifier by joining together weak learners. In the fields of image processing and natural language processing, boosting algorithm has been introduced and has made succesful results[9], [10] in terms of cognitive performance and calculation cost in recent years. In our previous work, online action recognition is constructed with model-based method. Inspired by the above works of boosting, we design elemental classifiers to classify by threshold processing of certain motion features such as posture of human body joints. Thereby each stage of the boosting process that constructs a new elemental classifer can be viewed as a feature selection process. That is to say, important features for recognition are automatically selected in boosting process. In addition, each elemental classifier has very small calculation cost because it classifies by threshold processing .

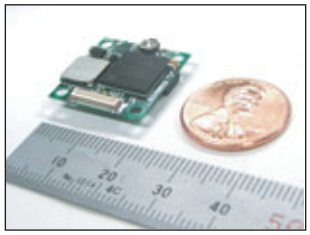

Fig. 1. The wireless posture sensor device. The size is 20 x 20 x 5 [mm]. The weight is 1.5 [g]

In the above method, we assume that posture information of human body joints is fetched from posture sensors. And we have been developing tiny wireless posture sensor devices[11] as shown in Fig. 1 and are planning to applicate the method in the future. Sensor-attaching-based recognition is free from problems of occlusion and changes of illumination environment which become serious issues in appearance-based method with camera vision. However, attaching sensors to all body joints as shown in Fig. 2 (an example of wearing wired sensors) can cause strong

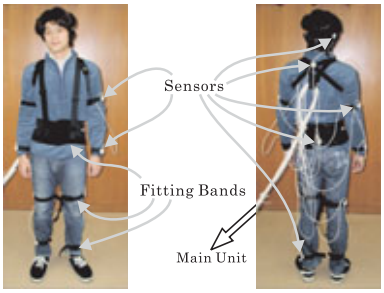

Fig. 2. An example for wearing wired sensors of motion capture system

constraint on daily life activities. Therefore the number of sensors attached to human body should be small to reduce the constraint. But it is challenging to decide automatically which joints we should attach limited sensors to only with existing boosting algorithm in our previous method.

Based on above discussion, we extend existing boosting algorithm, and propose new learning framework *wrapped boosting* for action recognition in this paper. Utilizing motion data fetched from sensors on all joints as training data, allocation of predetermined number of sensors is optimized automatically in the proposed framework. Consequently, the framework enables both automatic feature selection and automatic sensor allocation for constructing robust action classifiers.

This paper is organized as follows. Section II outlines action recognition based on wrapped boosting algorithm. Section III details construction of action classifiers with boosting. Section IV explains about realizing automatic sensor allocation with wrapped boosting framework. Section V describes some experimental results, including a detailed description of our experimental methodology. Finally section VI contains a discussion of the proposed method and future works.

II. Outline of Online Action Recognition with Wrapped Boosting

A. Input and Output for Action Recognition

Daily life actions have a different characteristic from other actions, such as gesture and sign language. Actions of gesture and sign language are exclusive in relationship among these actions, that is, such actions hardly occur simultaneously. However, daily life actions are not always exclusive in relationship among these actions, in turn, several actions may occur simultaneously. For example, humans can recognize the two actions involved when observing someone is *standing* and *walking*.

In order to realize the simultaneous recognition, our method constructs multiple binary classifiers, each of which is assigned to classify one specific action. The process of each classifier runs in parallel with and independent of the others.

Fig. 3 shows an example of input and output in our method. Classifiers receive motion data $\boldsymbol{x}_t$ as an input at a time t, and output action labels $(y_t^{(1)}, y_t^{(2)}, \cdots, y_t^{(M)})$ for

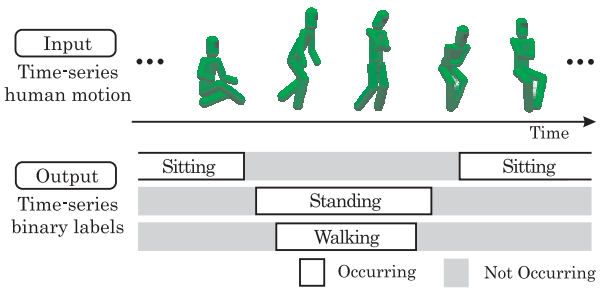

Fig. 3. Input and output in proposed method

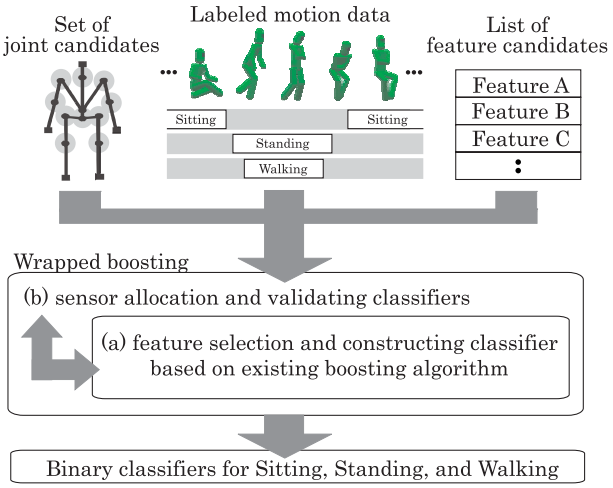

Fig. 4. Proposed framework for constructing binary action classfier

each of M kinds of actions. An action label $y_t^{(i)}$ denotes a label of i-th action at a time t, and $y_t^{(i)} = +1$ indicates that the action is occurring, and $y_t^{(i)} = -1$ indicates not occurring.

B. Framework for Constructing Binary Action Classifiers

In wrapped boosting framework, binary action classifiers are constructed based on: 1) set of joint candidates to attach posture sensors (ex. elbow, shin, shoulder), 2) motion data labeled for each of target actions, 3) list of motion feature candidates to be utilized for constructing classifiers (ex. each component of posture matrix for each joint).(See Fig. 4.) Joint candidates, motion data, and motion feature candidates are common among all target actions.

Roughly speaking, there are two phases inside the framework as follows.

- **changing joints to be attached with limited sensors**
  In this phase, classifiers which are constructed in the phase below are validated. Depending on the validation results, joints which we should attach sensors are changed. (At first, the framework starts with certain joint being attached with a sensor and go to the phase below.)
- **constructing binary classifiers for each action**
  In this phase, classifiers are constructed with normal boosting algrithm (ex. adaBoost[7], logitBoost[12], madaBoost[13]) based on motion features related to the selected joints in the above phase.

Iterating two phases above alternately realizes automaic feature selection and automatic allocation of limited posture sensors. (Automatic allocation of sensors is achieved with the first phase, and automaic feature selection is enabled with boosting algorithm in the second phase.)

As you may see, this framework *wrappes* existing boosing algorithm. Therefore our framework is meta-extension of exisiting boosting algorithm. Note that achieving automatic decision about which joints we should attach limited sensors is very difficult only with existing boosting algorithm.

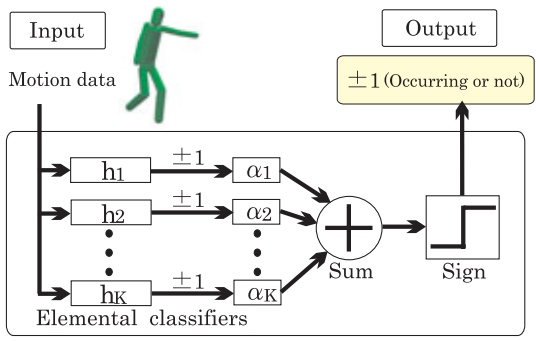

Fig. 5. Configuration of binary action classifier

III. Construction of Binary Action Classifiers with Boosting Algorithm

In this section, we describe how to construct binary action classifiers with boosting algorithm based on motion features related to limited posture sensors, corresponding to the part (a) in Fig. 4.

A. Boosting Algorithm for Action Recognition

Boosting is one of ensemble learing algorithm[14], which is utilized in the fieled of natural language processing and image processing in recent years. The goal of the learning algorithm for the i-th action is to construct a binary classifier $H^{(i)}(\boldsymbol{x})$ based on given training data as below.

$$H^{(i)}(\boldsymbol{x}) = \mathrm{sgn}\Big(\sum_{k=1}^{K} \alpha_k^{(i)} h_k^{(i)}(\boldsymbol{x})\Big)$$

where $h_k^{(i)}(\boldsymbol{x})$ is a classifier producing ± 1 and $\alpha_k^{(i)}$ is a constant. The classifiers $h_k^{(i)}(\boldsymbol{x})$ which we call *elemental classifiers* are trained one-by-one on weighted training data. The training procedure gives higher weight to data that are currently misclassified. The suffix k denotes what number the elemental classifier is constructed. And the constant $\alpha_k^{(i)}$ is recognition confidence of k-th elemental classifier.

Then the final classifier $H^{(i)}(\boldsymbol{x})$ is defined to be sign of linear combination of the elemental classifiers from each stage (See Fig. 5). In this paper, $\alpha_k^{(i)}$ is decided with adaBoost leaning algorithm[7], which is widely used and typical algorithm in boosting algorithm. A detailed description of adaBoost for constructing action classifier is given in Table I. In the rest of this paper, the term "boosting" represents adaBoost learning algorithm.

B. Design of Elemental Classifiers for Action Recognition

In order to lessen calculation cost for recognition and to avoid overfitting to trainig data, we should design elemental classifiers to be simple. Therefore, we decide that elemental classifiers classify by threshold processing of a scalar motion feature. The elemental classifier for i-th action is described as follows.

$$h_k^{(i)}(\boldsymbol{x}) = \mathrm{sgn}(\phi_k^{(i)}(\boldsymbol{x}) - \gamma_k^{(i)})$$

TABLE I

AdaBoost algorithm for constructing each action classifier

0 Given as training data:
$(\boldsymbol{x}_1^{(i)}, y_1^{(i)}), \ldots, (\boldsymbol{x}_n^{(i)}, y_n^{(i)}), \ldots, (\boldsymbol{x}_N^{(i)}, y_N^{(i)})$;
$\boldsymbol{x}_n^{(i)} \in X$, $y_n^{(i)} \in \{+1, -1\}$
if action occur, $y_n^{(i)} = +1$ otherwise $y_n^{(i)} = -1$

1 Initialize: $D_1^{(i)}(n) = \frac{1}{N}$

2 For $k = 1, \ldots, K$:

- Optimize elemental classifier $h_k^{(i)}$ which minimizes the error rate

$$\epsilon_k^{(i)} = \sum_{n: h_k^{(i)}(\boldsymbol{x}_n^{(i)}) \neq y_n^{(i)}}^{N} D_k^{(i)}(n)$$

- Update the weights:

$$D_{k+1}^{(i)}(n) = \frac{D_k^{(i)}(n) \exp(-\alpha_k^{(i)} y_n^{(i)} h_k^{(i)}(\boldsymbol{x}_n^{(i)}))}{Z_k^{(i)}},$$

where $Z_k^{(i)}$ is a normalization factor.

3 Output the classifier:

$$H^{(i)}(\boldsymbol{x}) = \mathrm{sgn}\Big(\sum_{k=1}^{K} \alpha_k^{(i)} h_k^{(i)}(\boldsymbol{x})\Big),$$

where $\alpha_k^{(i)} = \frac{1}{2} \log\Big(\frac{1-\epsilon_k^{(i)}}{\epsilon_k^{(i)}}\Big)$.

where $\phi_k^{(i)}$ is the function which extract a scalar motion feature from motion data $\boldsymbol{x}_t$ according to the given list of feature candidates as mentioned above (See Fig. 4). And $\gamma_k^{(i)}$ is threshold of k-th elemental classifier of i-th action. $\phi_k^{(i)}$ and $\gamma_k^{(i)}$ are optimized to minimize error rate $\epsilon_k^{(i)}$ at k-th round of learning process. This type of classifer is called *decision stump*[15] which is one-level decision tree.

This design of elemental classifiers allows action classifiers to select automatically scalar motion features and its threshold $\gamma_k^{(i)}$ in learning process. This learning process can be viewed as the process of feature selection. In addition, the parameter given by humans is only the number K which denotes what number elemental classifiers are contained in each action classifier.

Here lists the merits of our elemental classifiers: 1) calculation cost for classification is very small. 2) boosting process automatically selects important motion features for classification. 3) the parameter given by humans is only the number of elemental classifiers contained in each action classifier.

IV. Automatic Sensor Allocation with Wrapped Boosting

In this section, we describe about the detail of wrapped boosting algorithm, including the part (b) as well as (a) in Fig. 4.

TABLE II

WRAPPED BOOSTING ALGORITHM FOR ACTION RECOGNITION

0 Given as joint candidates:
$(1, j_1), \ldots, (l, j_l) \ldots, (L, j_L)$; $j_l \in \{+1, 0\}$
if a sensor is allocated to joint l, $j_l = +1$
otherwise, $j_l = 0$
Partition given training data $\{(\boldsymbol{x}, y)\}$ into validation data and sub-training data

1 Initialize: $j_l = 0$

2 For $s = 1, \ldots, S$:

- Set $\Theta = \{l | j_l = +1\}$, $\hat{\Theta} = \{l | j_l = 0\}$
- Based on joints of Θ and $\exists\ \hat{l} \in \hat{\Theta}$, construct $\hat{H}^{(i)}(\boldsymbol{x})$ via boosting algorithm using sub-training data for each action
- Choose $\hat{l}$ which shows the most improvement on mean cognitive performance of all $\hat{H}^{(i)}(\boldsymbol{x})$ for validation data, and fix $j_{\hat{l}} = +1$

3 Set $\Theta = \{l | j_l = +1\}$
Based on joints of Θ, construct $H^{(i)}(\boldsymbol{x})$ using original training data for each action

4 Output the classifier $H^{(i)}(\boldsymbol{x})$ for each action

Wrapped boosting algorithm is inspired by *wrapper method* which is known as one of feature subset selection methods. Our algorithm starts with no joint allocated a sensor to. The algorithm proceeds allocating sensors to the joints one-by-one and running normal boosting algorithm with the allocated joints for all target actions. Then fixed is the one sensor which raise mean cognitive performance most for validation data at current stage. This process repeats until the number of joints to which sensors are allocated reaches the number S given in advanced by humans. The algorithm is detailed in Table II. And as an example, Fig. 6 shows a sensor-allocating lattice for four joints in wrapped boosting. Each node in the figure represents which joints are allocated sensors to.

We will expain the process of sensor allocation using an example. Assume that our goal is to allocate two sensors

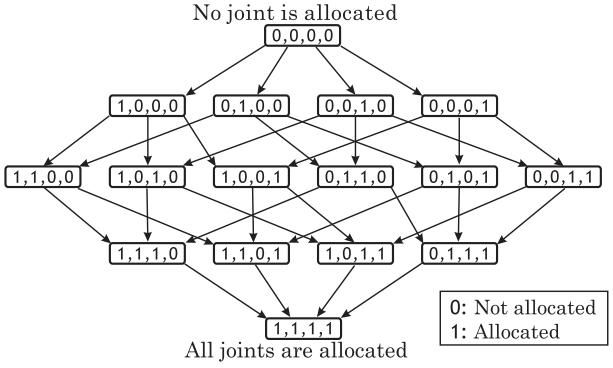

Fig. 6. A lattice for four joints in wrapped boosting

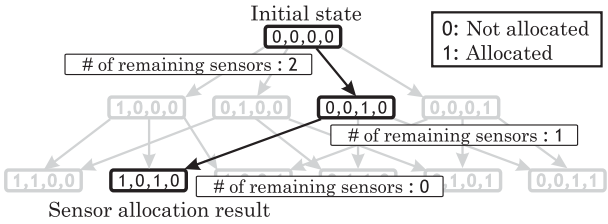

Fig. 7. An example of sensor allocation process

to some appropriate joints. At first, the process starts from the top node in Fig. 7. Then according to wrapped boosting algorithm, 4 types of binary classifiers are constructed. They are constructed via boosting algorithm based on motion features related to any one of four joints. In this example, the binary classifiers based on the third joint makes the most improvement on cognitive performance for validation data, so the thrid joint is chosen to be attached with a sensor. At this point, the number of remaining sensors to allocate is 1. Subsequnetly, 3 types of binary classifiers with the third joint and any one of three joints other than the third one are constructed. Since the classifiers based on the first and third joints are the most robust for validation data, the first joint is chosen. Consequently, the first and third joints are selected as a proper ones to be attached a sensor because the number of chosen joints reaches the number 2 given preliminarily.

Wrapped boosting process selects joints which cause the highest mean cognitive performance for validation data. And classifiers for all target actions are constructed with any motion features related only to the selected joints. In the above exapmle, all classifiers are constructed with features realted only to the first and third joints eventually.

Of course it is conceivable that predetermining the number S may be difficult when various kinds of actions are targeted. We assume that we should target about 20 to 30 kinds of actions in the future to realize supports for human daily life activities by robots. To deal with that case, it is another possible solution that we determin a goal performance value for intended application instead of the number S, and wrapped boosting process continues until the mean cognitive performance for validation data reaches the goal value.

V. EVALUATION EXPERIMENT

This section describes evaluation experiments for our wrapped boosting algorithm. We demonstrate effectiveness of the algorithm by comparing cognitive performance with the case that we do not limit number of sensors and do allocate sensors to all joints. Additionaly, we show an example of classification result with the proposed algorithm.

A. Target Actions to be Classified

We selected *walking* and *running* as target actions to be classified in the experiments. Daily life actions contain actions without movements such as *lying* and *sitting*, and actions with movements such as *walking* and *running*. However we target only actions with movements in the experiments. This is because actions without movements have innate poses

TABLE III

MOTION DATA FOR EVALUATION EXPERIMENTS

# of frames	set1	set2	set3	set4
total	2182	1930	2101	2058
Walking	529	421	494	441
Running	416	317	299	448

and actions with movements vary according to time and then have not innate poses, and then we expect that actions without movement is easy to be classified compared to classification of actions with movements. Hence, if classifiers based on the proposed method classify actions with movements robustly, actions without movements are also classified robustly.

B. Motion Data

The measured motion data for the experiments are sequential human motion data fetched by a magnetic motion capturing system at 30[Hz]. The format of the data file is BVH, a de-fact standard moition file format by Biovision Coroporation. A BVH file contatins the structure of a human as a linked joint model figure (See Fig. 8). We give the joints of the model to wrapped boosting algorithm as candidates for sensor allocation.

The actions included in the motion capture files are *walking*, *running*, *standing still*, and transition from an action to another one. We annotate motion data per frame with *walking* or *not walking*, and with *running* or with *not running*.

The motion data is divided into 4 sets and Table III shows the number of frames in each set. 3 of 4 sets are utilized as training data and 1 of 4 sets is utilized as test data, and we evaluate the proposed algorithm by cross validation. As described in section IV, training data sets are divided into 2 sub-training data sets and 1 validation data set.

C. Candidates of Motion Feaures

In the experiments, we give the following scalar motion features of each joint to wrapped boosting framework:

- Vertical components of the joint's posture matrix in world coordinate sysytem
- Every component of relative posture matrix between two joints

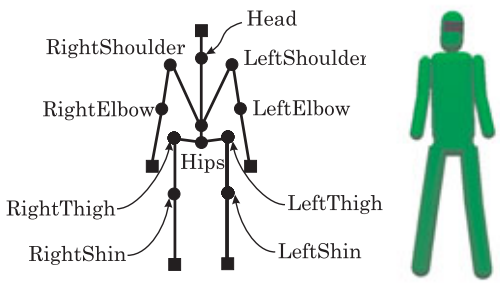

Fig. 8. Human model utilized in the experiments

TABLE IV

EXPERIMENTAL RESULTS OF WRAPPED BOOSITNG

test set	F-measure (%)		selected joint candidates
	Walking	Running	
set1	93.2	96.2	both thighs, both shins
set2	93.2	93.4	both thighs, both shins
set3	95.0	94.5	both thighs, both shins
set4	95.6	95.2	both thighs, both shins
average	94.3	94.8	—

TABLE V

EXPERIMENTAL RESULTS OF EXISTING BOOSTING ALGORITHM

test set	F-measure (%)	
	Walking	Running
set1	92.0	96.8
set2	94.6	97.2
set3	83.5	95.5
set4	74.5	91.9
average	86.2	95.4

- Temporal subtraction of each component listed above

As you can see, we utilize the motion features only about posture of joints, not about position. This is because we are planning to run our boosting framework with three-axis acceleration sensors and three-axis geomagnetic sensors in the future.

D. Evaluation Measure

For cognitive performance measure, we use F-measure. F-measure is a harmonic average of recall rate rate and precision rate. R, P, F denotes recall rate, precision rate, and F-measure respectively. Then they can be defined as follows:

$$\mathrm{R} = \frac{\mathrm{N_c}}{\mathrm{N_a}}, \quad \mathrm{P} = \frac{\mathrm{N_c}}{\mathrm{N_m}}, \quad \mathrm{F} = \frac{2\mathrm{RP}}{\mathrm{R}+\mathrm{P}}$$

where $\mathrm{N_a}$ indicates the number of frames which are annotated as $y_t^{(i)} = +1$ by humans, $\mathrm{N_m}$ denotes the number of frames which are labeled as $y_t^{(i)} = +1$ by classifiers, and $\mathrm{N_c}$ denotes the number of frames labeled as $y_t^{(i)} = +1$ correctly by classifiers.

E. Experimental Results

We run wrapped boosting to select 4 from 10 joints which are corresponding to Head, Hips, RightShoulder, LeftShoulder, RightElbow, LeftElbow, RightThigh, LeftThigh, RightShin, LeftShin in Fig. 8. Each action classifier is constructed with 100 elemental classifiers in this experiment. Considering lateral symmetric property of the target actions, we make a condition that right-and-left joints (ex. RightThigh and LeftThigh) must be selected or not selected all together.

Table IV shows the f-measure and selected 4 joints for each test data set. For all the test data sets, wrapped boosting select both thighs and both shins, and can construct robust classifiers for both actions and every data set.

By way of comparison, we also evaluated classifiers constructed only with exisiting boosting algorithm based on motion features of all 10 joints. The results is shown in Table V. While some classifiers are more robust than ones constructed with wrapped boosting, we can find much less robust classifiers. Existing boosting algorithm caused overfitting to training data due to utilizing motion feature of all the joins.

Both tables tell that our framework can allocate sensors properly and is capable of constructing robust classifiers more reliably.

F. Example of Classification with the Proposed Method

Fig. 9 and Fig. 10 show an example of recognition result with wrapped boosting. Fig. 9 represents thumbnails of human figures fetched every 5 frames (about 0.17 seconds) in targeted sequential motion data file. In this motion data, the subject stands still at first, then start running, and switch to walking finally.

Fig. 10 represents recognition results by classifiers constructed with wrapped boosting. In the graph, horizontal thick solid lines indicate the estimated action labels and vertical lines indicate start or finish of actions. We used smoothing filter for output labels of the classifers in this example. The lines which have arrows on their edges indicates action labels $y_t^{(i)} = +1$ annotated by humans. Fig. 10 shows that the proposed framework can construct robust action classifiers realizing automatic sensor allocation and automatic feature selection.

VI. Conclusion

In this paper, we proposed wrapped boosting algorithm to construct robust action classifiers for online daily life action recognition. We premised the model-based action classifiers working independently and in parallel, and these classifiers were constructed based on boosting which is an ensemble learning algorithm and is also known as a feature selector.

Extending existing boosting algorithm, we proposed a learninig framework which provide not only automatic feature selection but also automatic sensor allocation to proper joints of humans for target actions.

We evaluated our algorithm by applying it to recognition for gait motion; *walking* and *running*. The motion data was fetched by motion capturing system. In consequence, wrapped boosting was able to select proper joints to which limited sensors should be attached, and to construct more robust classifiers compared to constructing classifiers with all joints available. Classifiers constructed only with existing boosting algorithm were subject to overfitting to training data.

Future works are 1) to propose the method that takes account of interdependencies of output action labels in order to construct classifiers which are strong for noise and lack of motion data and 2) to realize online action recognition utilizing wireless sensor devices which consist of acceleration sensors and geomagnetic sensors applicating wrapped boosting algorithm.

References

[1] J. Yamato, J. Ohya, and K. Ishii: "Recognizing Human Action in Time-Sequential Images Using Hidden Markov Model," Proceedings of the 1992 IEEE Conference on Computer Vision and Pattern Recognition, p.379-385, 1992.

[2] D. Cao, O. Masoud, D. Boley, and N. Papanikolopoulos: "Online Motion Classification Using Support Vector Machine," Proceedings of the 2004 IEEE International Conference on Robotics and Automation, vol.3, p.2291-2296, 2004.

[3] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas: "Conditional Models for Contextual Human Motion Recognition," Proceedings of the Tenth IEEE International Conference on Computer Vision, vol.2, p.1808-1815, 2005.

[4] L. Rabiner: "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proceedings of the IEEE, vol.77, p.257-285, 1989.

[5] B. Schölkopf and A. Smola: "Learning with Kernels," MIT Press, 2002.

[6] J. Lafferty, A. McCallum, and F. Pereira: "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," Proceedings of the 18th International Conference on Machine Learning, p.282-289, 2001.

[7] R. Schapire and Y. Singer: "Improved Boosting Using Confidence-rated Predictions," Machine Learning, vol.37, no.3, p.297-336, 1999.

[8] M. Shimosaka, T. Nishimura, Y. Nejigane, T. Mori, and T. Sato: "Fast Online Action Recognition with Boosted Combinational Motion Features," Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, p.5851-5858, 2006.

[9] P. Viola, M. Jones, and D. Snow: "Detecting Pedestrians Using Patterns of Motion and Appearance," Proceedings of the 9th IEEE International Conference on Computer Vision, vol. 2, p.734-741, 2003.

[10] T. Kudo and Y. Matsumoto: "A Boosting Algorithm for Classification of Semi-structured Text," Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, p.301-308, 2004.

[11] T. Harada, T. Gyota, T. Mori, and T. Sato: "Construction of Flexibile Wireless Network System and Tiny Network Device for Wearable and Environmental Sensor Data," Proceedings of the 3rd International Conference on Networked Sensing Systems, p.115–118, 2006.

[12] J. Friedman, T. Hastie, and R. Tibshirani: "Additive Logistic Regression: a Statistical View of Boosting," Technical Report, Stanford University, 1998.

[13] C. Domingo and O. Watanabe: "Madaboost: a Modification of AdaBoost," Proceedings of the 13th Annual Conference on Computer Learning Theory, p.180–189, 2000.

[14] E. Bauer and R. Kohavi: "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants," Machine Learning, vol.36, no.1-2,p.105-139, 1999.

[15] W. Iba and P. Langley: "Induction of One-Level Decision Trees," Proceedings of the 9th International Conference on Machine Learning, p.233-240, 1992.

[16] R. Kohavi and G. John: "Wrappers for Feature Subset Selection," Artificial Intelligence, vol.97, no.1-2, p.273-324, 1997.

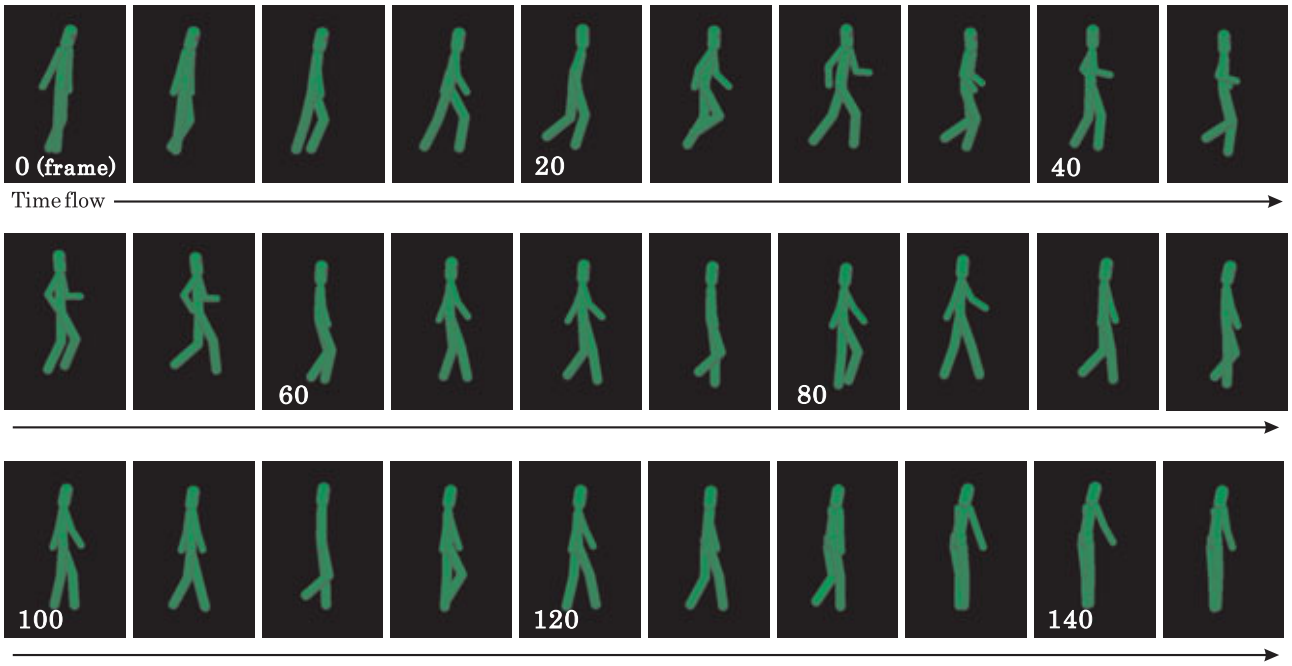

Fig. 9. Thumbnails of the motion data example

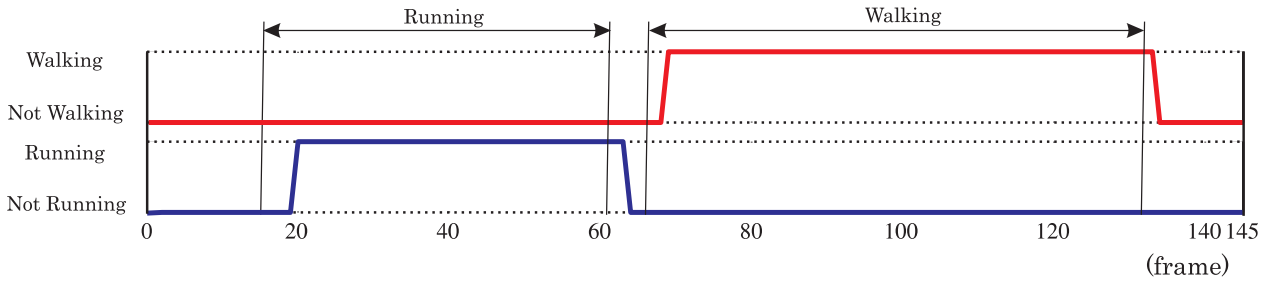

Fig. 10. Recognition results for the motion data example