Journal of Information Processing Vol.22 No.1 56–66 (Jan. 2014)

[DOI: 10.2197/ipsjjip.22.56]

Regular Paper

Summarizing Nursery School Surveillance Videos by Distance Metric Learning

Yu Wang[1,a)] Jien Kato[1,b)] Kenji Mase[1,c)]

Received: January 12, 2013, Accepted: September 13, 2013

Abstract: We propose a novel video summarization approach that takes the mass quantity of nursery school surveillance videos as input and produces short daily video digests for children. The proposed approach makes full use of a distance metric, which is learned using a novel learning algorithm called the adaptive large margin nearest neighbor (ALMNN), and can properly measure the similarity between video clips. The learned distance metric is combined with supervised classification and unsupervised clustering to categorize daily raw surveillance videos into individual event categories. The final digest is constructed by selecting representative video clips that belong to individual event categories. Digests generated using our approach cover and reflect the various activities of children in nursery schools. They are of interest to parents, and they also enable easy access to mass quantities of daily surveillance video data. We implemented the approach as a practical system in a real nursery school environment and assessed its performance.

Keywords: nursery school surveillance videos, video summarization, event classification, distance metric learning

1. Introduction

Many children spend their early years in nursery schools or kindergartens. Most parents want to monitor the progress on a daily basis. To meet these needs, some nursery schools in Japan have introduced remote surveillance camera systems [1]. Such systems not only provide the chance for parents to check events inside the nursery school online, but also allow them to watch the accumulated videos over the Internet later. However, due to the use of many surveillance cameras, a mass quantity of raw videos is accumulated every day. As a result, it is very difficult for parents to browse these videos and find their desired portions.

In addition to watching, these surveillance videos are also valuable for other purposes. For example, because they record the progress of children, long-term behavior statistics can be extracted from the videos by adapting computer-vision-based behavior analysis techniques [2]. Such statistics are a boon for research on child development; for example, analyzing interactions among children in group activities can provide extremely valuable information for early diagnosis of autism [3]. However, because raw video data are very large and behavior analysis is typically conducted on a relatively small portion of such data showing specific activities, raw video data are not useful unless they are appropriately structuralized.

To make it possible to watch and analyze such large volumes of video data, we devise a method to summarize them into compact video digests. Every day, in nursery schools, children participate in different events (see **Fig. 1** for concrete examples). They participate in these events in their own ways, and their performance reflects their daily life and growth. A video digest that covers and reflects the children's performance in different events over a day would be ideal for reviewing their daily activities. It would satisfy the parents' interest in knowing what their child does in the nursery school, and it could also be used as a visible index to facilitate the selection of video materials for further analysis.

Fig. 1 In nursery schools, children participate in different kinds of events and perform them in their own way.

In order to summarize raw video materials into such digests, there are mainly two technical issues that need to be addressed: (1) the categorization of raw video materials into different events and (2) the selection of video clips that could well reflect each event. For the first problem, supervised event classification [2], [4], which learns a classifier for each predefined event category from the training data, is a typical solution. However, when dealing with videos of daily life, it is difficult to pre-

[1] Graduate School of Information Science, Nagoya University, Nagoya, Aichi 464–8603, Japan
[a)] ywang@nagoya-u.jp
[b)] jien@is.nagoya-u.ac.jp
[c)] mase@nagoya-u.jp

define all possible event categories (there are too many kinds of events) and provide sufficient training data for each category (some events do not occur frequently). Therefore, it is not suitable to adopt only this method to our summarizing task. For the second problem, many existing studies on video summarization [6], [7] use motion intensity as a measurement for selecting representative video clips. This mechanism is applicable when dealing with videos on sports or cooking, however, it does not suffice for defining "representativeness" in videos of daily life, because more visual aspects should be taken into consideration.

In this paper, we propose a novel approach for summarizing nursery school surveillance videos. The approach simultaneously solves the previous two problems by learning a distance metric from training data, which can properly measure the similarity between video clips. The distance metric is learned efficiently in a mixed, noisy feature space using a novel learning algorithm known as the adaptive large margin nearest neighbor (ALMNN). It is combined with supervised classification and unsupervised clustering, to categorize raw video materials initially into predefined chief event classes; then, inside each class, it divides them into more detailed event clusters. The representative video clip for each individual event cluster is selected as the centroid using a combination of multiple features. With our approach, only a limited amount of labeling effort is required; additionally, it produces digests that are representative in multiple visual aspects.

The main contribution of this paper is two-fold: (1) we propose an efficient distance-metric-learning-based video summarization approach and implement it as a practical system in a real nursery school environment; and (2) we propose a distance metric learning algorithm that is robust for dealing with mixed, noisy feature vectors. The practical performance of the summarization approach and the learning algorithm are confirmed through quantitative experiments on real-world data and questionnaires.

2. Related Works

There are many successful studies in the field of video summarization. Most earlier works require meta-data (i.e., textual indices) as necessary input. For example, in Ref. [5], Hashimoto et al. developed a system for summarizing TV programs by matching the keywords (specified by users) to TV program subtitles. Later in Ref. [6], Takahashi et al. proposed to summarize baseball game by substituting the corresponding textual index to a predefined baseball game structure. Both these studies demonstrate that meta-data can greatly help in the production of comprehensive digests. However, because meta-data are not available for many kinds of videos (such as personal recordings and surveillance videos), these methods are limited in their usage.

There are also some studies that utilize the inherent rules of video contents for summarization. In order to make video digests for cooking show, Miura et al. [7] devised to follow the cooking process to extract hand shot videos showing both food and cooking activities. Similarly in Ref. [8], Bach et al. learned a Hidden Markov Model to represent the baseball game process, and used baseball games' inherent rules to select highlight scenes for digest producing. These methods work well in practice, however, because they are designed under specific rules (a kind of domain knowledge), it is difficult to generalize them for video contents with other rules or without clear rules.

Recently, an increasing number of studies have tried to analyze video contents directly without using meta-data or domain knowledge. Ren et al. [9] proposed the utilization of unsupervised clustering to summarize rush videos (short videos). Their method groups similar frames into clusters, and construct the digest by selecting frames from individual clusters. Similarly, Tavanapong et al. [10] adopted clustering to create icons for video contents to facilitate faster browsing. In these works, clustering is used to group similar neighboring frames. It works well for short videos because the visual variance between neighboring frames is relatively small, and a simple feature set and Euclidean-distance-based similarity measurement can provide promising clustering results. However, for videos with large data content, such clustering is not suitable because a simple feature set cannot capture various visual aspects. On the other hand, if we use a mixture of multiple features to capture more visual aspects, the Euclidean distance metric will lose its power because it assumes that every feature dimension weighs the same in measuring the similarity.

In addition, some studies have used supervised learning to analyze video contents. Rodriguez [4] learned action classifiers for a number of predefined actions of interest from the labeled data. He used the classifiers to locate video portions containing certain actions to construct a digest. In Refs. [11] and [12], the authors exploited a similar concept in summarizing nursery school videos. The obvious flaw in these methods is that they require a training set for every activity category of interest; this implies that the training set could be very large and very difficult to prepare.

In this study, the data we deal with has its unique characteristics: (1) because it is captured from multiple surveillance cameras, the quantity is huge; (2) unlike many videos such as those on sport or cooking, the contents of the videos, which record the daily lives of children, do not have clear inherent rules, and the events cannot be predefined totally; (3) the data are usually very noisy. Hence, the existing approaches for video summarization are inadoptable for our problem. To deal with these kinds of issues, we explore a novel distance-metric-learning-based approach.

Different from classifier learning which learns how to separate data points into classes, distance metric learning learns how to measure the similarity between data points. Since the learned distance metrics are flexible for use, such an approach has been introduced to more and more vision tasks recently. In many works, the learned distance metrics are directly used to power the k-nearest-neighbor rule. For example, Tran et al. [24] proposed to integrate the learned metric with 1-nearest-neighbor for action recognition, and Guillaumin et al. [21] devised to use the learned metric with weighted-nearest-neighbor to conduct image annotation. In some other works, the learned distance metrics are considered as general knowledge of the data. In Ref. [23], Zhang et al. proposed a cost function which is defined over the metric that learned from a large dataset, for the people re-identification task. In another work, Mensink et al. [20] learns a metric from a large number of labelled images, and use it to classify new image categories without additional learning. In our work, the basic

idea of using distance metric learning is close to the later manners, in which the metric is considered as a general knowledge of the data. However, the usages of the learned metric is designed for practical video summarization task and is unique to existing works.

Similar with many existing works [20], [23], [24], distance metric learning in our work is conducted based on the large margin nearest neighbor (LMNN) algorithm [14]. LMNN is known as one of the best distance metric learning algorithm because of its state-of-the-art performance. It also has successful extensions. In Ref. [22], Kumar et al. proposed an extension which makes feature vectors become invariant to known multivariate polynomial transformations. In another work [20], Mensink et al. devised to dynamically update the target neighbour membership during learning in order to improve the learning performance. In our work, we also introduce an extension of the LMNN algorithm. Different from existing extensions, the proposed ALMNN explicitly integrates feature selection. Though it is a straight forward extension of LMNN, it significantly improves the robustness when dealing with mixed, noisy feature vectors.

3. System Overview

Our objective is to develop an efficient method for summarizing nursery school surveillance videos. For this, we propose a distance-metric-learning-based summarizing approach and implement it as a practical system in a real nursery school environment. In addition, considering that users (parents) would particularly care about a specific child (their own child), we paid special attention to the data acquisition and preprocessing of the system.

3.1 Video Data Acquisition and Preprocessing

Figure 2 summarizes our approach for video data acquisition and preprocessing. The raw videos are captured by surveillance cameras set up throughout the nursery school. In the nursery school that we cooperated with, there are seven network cameras installed in nursery rooms (4), resting room (1), entrance (1), and passageway (1). These cameras are connected to a storage server in the local network, and record over 80 hours' worth of videos every day. When we record the videos, since data transfer over the network and video encoding are sometimes unstable, we segregate the video into small clips of one minute each. This also prevents file corruption as large files are more prone to corruption.

Beside cameras, we also utilize a radio frequency identification (RFID) system to collect information on the locations of children. We set one RFID receiver beside each camera and connect all cameras and receivers to the storage server. Every morning, we give an RFID tag to each child (that they keep in their pockets) when they enter the nursery school. During the day, these tags continuously send out the tag IDs, and the nearby receivers capture them. Once a specific receiver captures a signal from a specific tag at a specific time, it will leave a record on the server. These records are then separated into log files according to the tag IDs. In this way, each log file collects thousands of records that register the locations where a specific child has been during that day. We consider one of these records as evidence for the target child appearing in the video feed of a specific camera at a specific time.

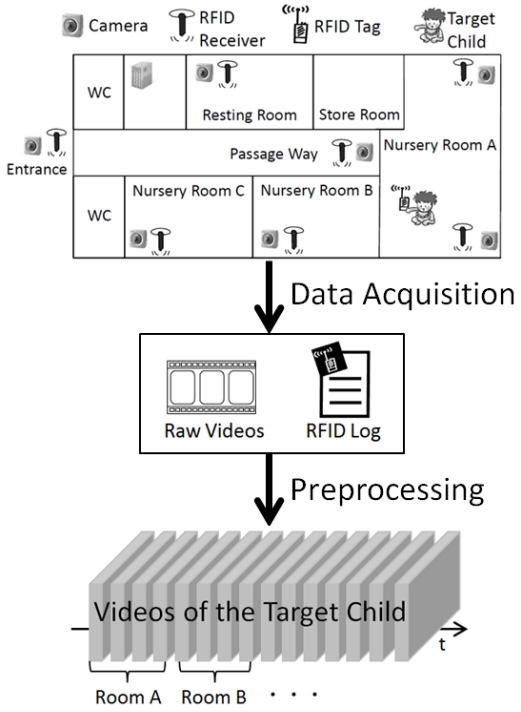


Fig. 2 Video data acquisition and preprocessing of the system.

Although we try to set RFID receivers far from each other and try to avoid signal interference, the signals sent from one tag could sometimes be captured by more than one receiver, and sometimes could be missed by all receivers. However, it is certainly true that signals are more frequently captured by a nearby receiver than a distant one. This is, although a single record may not always be reliable enough, its statistic over a period of time can provide more reliable evidence. We, therefore, calculate the statistic of the records within a unit time interval (in order to conveniently synchronize with the video, we use one-minute as the unit time here as well), and use this information to preprocess the raw video from all cameras into the tracing video of a target child.

The preprocessing is relatively simple yet efficient. We divide the whole day into unit time intervals. For each interval, we collect the records from the log file of a target child, and identify the receiver with the maximum number of records. If the number exceeds a threshold, we take the corresponding camera's video; otherwise, we do not take any video for that interval. The threshold is used to guarantee the reliability of the appearance of the target child in the selected video segment. In case the target child is far away from the camera, or moving between different rooms, the maximum number of the records will be small and no video will be selected. We repeat the video selection for all time intervals, and the selected videos are collected together as the tracing video of the target child.

3.2 Main Summarization Pipeline

The main summarization pipeline is outlined in **Fig. 3**. It can be divided into the learning phase, where the distance metric is learned from the training data, and the summarizing phase, where the learned distance metric is used to summarize the tracing video of the target child into a short video digest.

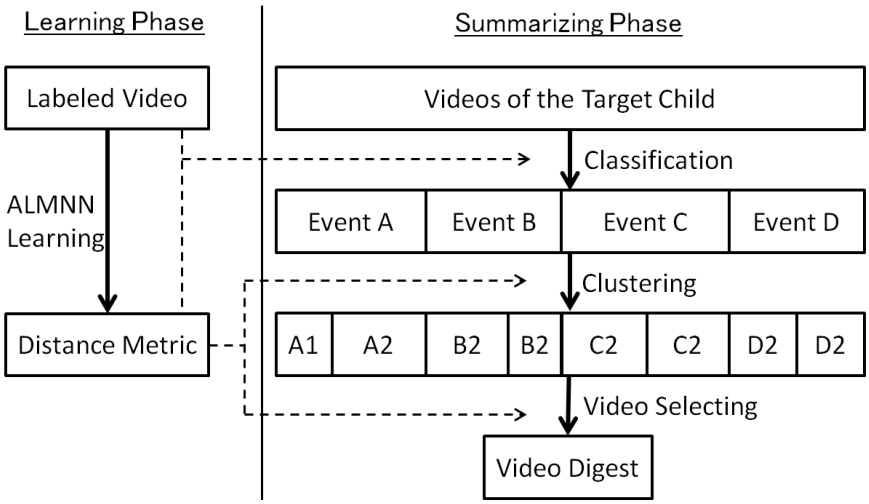


Fig. 3 Main summarization pipeline.

In the learning phase, we manually collect a set of video data from a real nursery school environment. Each individual data point is a one-minute video clip. According to the different activities performed by children, we coarsely divide these data points into a small number of chief event categories which correspond to relatively abstract concepts.

In practice, we defined four chief event categories after discussing with the nursery school teachers. They are "group activity," "play," "sleep," and "meal." "Sleep" and "meal" have small motions and are characteristic in their appearance ("sleep" has the appearance of bedding, "meal" has the appearance of tables). These two types of activities together occupy nearly one-third of a day's duration in the nursery school. The other two types of events, "play" and "group activity," occupy the remaining two-thirds of that day's duration. In "group activity," many children (more than five children) participate in the same activity; whereas in "play," a small number of children (equal to or less than five children) play. These two categories differ in both appearance and motion.

The collected data points are encoded into vectors of motion and appearance features. Then, the distance metric is learned from them with the goal that data points in the same chief event category will be separated by small distances, whereas data points in different categories will be separated by large distances. In our work, we proposed the ALMNN algorithm to learn this metric. It will be detailed in Section 5.

In the summarizing phase, the system makes full use of the learned distance metric. The tracing video of the target child is taken as input, and the digest is produced in three steps. First, the learned distance metric together with the training data are used to conduct k-nearest-neighbor (kNN) classification [13] to classify the videos into one of the predefined chief event categories. Then, within the videos of each chief category, the learned distance metric is used to conduct agglomerative clustering. This clustering step helps to discover visually different event clusters which correspond to more concrete individual events (e.g., within the chief event category "meal," there exist individual events such as "lunch" and "snack"). Finally, within individual event clusters, videos in the centroids are selected as representative videos and are used to construct the output video digest. In the following sections, we will give a detailed description on the main summarizing pipeline and the ALMNN algorithm.

4. Summarization with a Distance Metric

In our proposed approach, summarization is conducted through classification and clustering. Both the steps make full use of a learned distance metric, which makes it possible to predefine only a small number of chief event categories and prepare much less training data than conventional video- analysis-based summarization methods. This leads to an easy practical implementation.

4.1 Feature Extraction

Raw video materials are firstly encoded into feature vectors. In this work, we only utilize visual features. Though time stamp or room ID can also provide cues for event classification, they are weak in nursery school environment because in nursery school, each room has multiple functions and the schedule of each event is only loosely defined. Additionally, since it is difficult to integrate the time stamp and the room ID together, or integrate them with other features (the time stamp is a finite variable and the room ID is an orderless discrete variable), using them in the proposed summarization framework is not feasible.

In nursery school surveillance videos, visual characters that relate to the event identity are mainly the appearance and the motion. Because the same events could occur at different times in different rooms, the features should be robust to the illumination changes and independent to the room environment. Additionally, since the quantity of the surveillance videos is extremely large and the quality is relatively low, the features should also be computationally efficient and robust to sensor noise. From this viewpoint, we compute four kinds of basis features for the one-minute video clips. They are the edge response, the color histogram, the inter-frame subtraction and the histogram of gradient (HOG) distance. An example of these features is shown in **Fig. 4**.

We use the edge response because it is good for representing the global appearance of the scene and robust to changes in il-

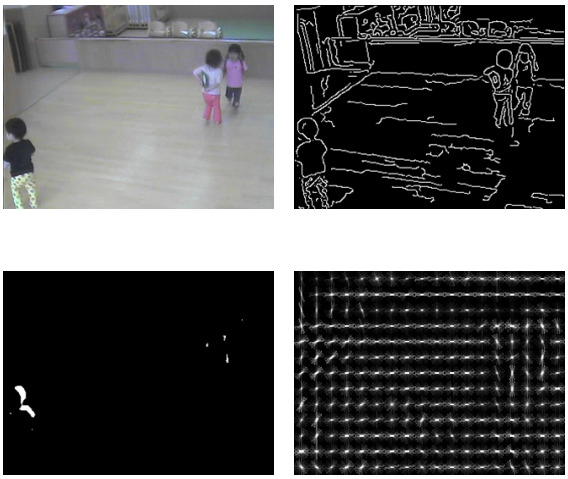


Fig. 4 Visualization of the features that are used in the system. From left-top to right-bottom, they are the original image, the edge response, the inter-frame subtraction, and the HOG feature.

lumination. To obtain the edge response, we apply the Canny edge detector on each frame in the one-minute video clip. We divide each frame into 8×6 blocks and calculate the number of edge points within each block (this enables the feature vectors to reflect the spatial arrangement of these edge points). The calculation result is a 48-dimensional feature vector f_e for each frame.

Color histograms in the 3D RGB space are used to capture the color appearance of the scene. We quantize each color channel into 8 bins, and then, the entire color space is divided into $8\times8\times8$ 3D blocks. Each pixel is assigned to one of the 512 blocks, and could be represented as an integer from 1 to 512. The color feature f_c for each frame is represented as a 512-dimensional histogram by counting the number of pixels for each integer.

Inter-frame subtraction is used to capture the global motion intensity between two frames. We subtract each frame from the previous one, and binarize the difference image using Otsu's method [16], which realizes binarization by identifying the threshold that minimizes the within-class variance between changed and unchanged pixels, and thus is robust to changes in illumination. We also adapt erosion [18] to remove bridges, branches, and noises in the binarized image. The final output is a binary image in which white pixels denote changed pixels. We divide the image again into 8×6 blocks and calculate the white pixels within each block. The resulting feature vector f_i is a 48-dimensional vector.

Though the inter-frame subtraction is good for capturing the global character of the motion, it does not capture the local motion feature. For this reason, we include the HOG distance as another feature. We compute the 36-dimensional HOG feature [17] of each frame with a bin size of 16. In our case, the computation uses a 320×240 pixel image as input and outputs a $18 \times 13 \times 36$-dimensional HOG feature. The HOG distance for each frame is calculated as the bin-wise Euclidean distance between its own HOG feature and that of the previous frame. The result is a 18×13 feature. Because the HOG distance captures the changes in the edge directions, it provides a good supplement for the inter-frame subtraction feature. The HOG distance for each frame is a 234-dimensional feature vector f_h.

We collect these four features for each frame and take the average vectors $(f_{ea}, f_{ca}, f_{ia}, f_{ha})^T$ as well as the deviation vectors $(f_{ed}, f_{cd}, f_{id}, f_{hd})^T$ of all the frames within each video clip. The final feature f for each unit video clip becomes a concatenation of them $(f_{ea}, f_{ed}, f_{ca}, f_{cd}, f_{ia}, f_{id}, f_{ha}, f_{hd})^T$, i.e., a 1684-dimensional feature vector.

4.2 Supervised Chief Event Classification

We utilize supervised classification to assign chief event labels to video materials. These labels not only provide conceptual information on the videos but also make it easy to customize the digests. For example, many parents may like to see less video of sleeping but more of playing. This could be done easily if we know what video portion includes the sleeping scenes and what includes the playing scenes.

However, as mentioned previously, supervised classification requires predefined event categories and corresponding training data, which are not easy to do for the videos of chronicling daily life. Therefore, instead of defining every possible event, we only coarsely predefine a small number of chief events. This leads to a less ambiguous definition of event categories and an easier preparation of the training data.

We adapt the kNN rule [13] for the chief event classification. Despite its simplicity, the kNN rule often yields competitive results when combined with prior knowledge. We manually provide a training set $D = \{(f_i, t_i)\}_{i=1}^n$ that comprises n labeled unit video clips. Here, f_i means the feature vector of the ith segment, and t_i indicates its corresponding chief event label. For a new video clip with a feature vector f_j, we compute its pair-wise distance with all f_i in D using a distance function $Dst(f_i, f_j)$ and find its kNNs. The majority label of the kNNs is chosen as the label of f_j.

In kNN, the performance depends crucially on the distance function $Dst(f_m, f_n)$. Without any knowledge of the data, most kNN implementations utilize the Euclidean distance. In our work, in order to guarantee the accuracy of classification, we learn the metric from the training data. With the proposed ALMNN algorithm, the learned $Dst(f_m, f_n)$ properly reflects the characteristics of the data and yields a promising classification accuracy.

4.3 Unsupervised Event Clustering

Through the chief event classification, raw video materials are divided into a small number of predefined chief categories. However, they are too coarse for the video summarization purpose. Within each chief event category, there still exist many visually different individual events. It is thus necessary to discover these individual events further in order to generate the digests that cover the various activities of children.

We collect all the video clips that share the same chief event label and perform agglomerative clustering on them without assuming the number of individual events. Initially, every video clip forms an individual cluster. We start by computing the pair-wise distance of every two video clips using the learned distance metric $Dst(f_m, f_n)$. Because $Dst(f_m, f_n)$ that learned by the ALMNN algorithm preserves the local compactness within each chief event class, using it can separate the visually different video clips into

more detailed clusters. In each cycle of the clustering, two clusters are combined into a single cluster, if they have the smallest average inter-group distance. The iteration goes on until the distance exceeds a predefined cut-off threshold. This resulted in a set of visually different individual event clusters.

After clustering is completed, the remaining tasks are relatively straightforward. In case of generating a n minutes digest, we pick out the n largest individual event clusters, and select one representative video clip for each cluster. The representative video clip is chosen as the one that has the minimum within-cluster distance to other video clips. Such a choice considers various visual aspects and is suitable for picking the "representative" videos of the child's daily life. The selected representative video clips are collected together and linked in a periodically linear order to become the final digest.

5. Distance Metric Learning

In both classification and clustering described in the previous section, the distance metric plays the most important role. In order to properly measure the similarity between video segments, we learn it from labeled training data. The learning is conducted using our proposed ALMNN algorithm, which originates from the existing large margin nearest neighbor (LMNN) algorithm [14]; however, comparatively, our algorithm is much more robust to mixed, noisy feature vectors. In the following section, we first briefly review the LMNN algorithm, and then introduce our extended ALMNN algorithm.

5.1 LMNN Algorithm

Let $D = \{(f_i, t_i)\}_{i=1}^{n}$ denote a training set that comprises n labeled unit video clips with feature vectors f_i and event labels t_i. We formulate the similarity of two feature vectors f_n and f_m as a Mahanalobis distance function:

$$Dst(f_n, f_m) = (f_n - f_m)^T M(f_n - f_m). \quad (1)$$

Here, M is a symmetric, positive definite matrix that completely parameterizes this distance function. The objective of distance metric learning is to learn M from D with the goal being that if $t_n = t_m$, $Dst(f_n, f_m)$ attains a small value, otherwise, $Dst(f_n, f_m)$ attains a large value. LMNN is one of the state-of-the-art algorithms for such a learning task [15]; it learns M through two steps: (1) for each data point f_i, it uses the Euclidean distance to select its k nearest data points (that have the same label as f_i) as target neighbors, and (2) it estimates M by minimizing a cost function defined as:

$$L(M) = \sum_{i,j} Dst_M(f_i, f_j) + C \sum_{i,j,l} h(1 + Dst_M(f_i, f_j) - Dst_M(f_i, f_l)). \quad (2)$$

The first term penalizes a large distance between data point i and its target neighbors j, whereas the second term penalizes a small distance between i and all imposter points l that have class labels different from t_i. In particular, the second term is designed to enforce the distance between i and l such that it becomes one unit further than the distance between i and j. C is a predefined positive constant, and $h(z) = max(z, 0)$ is the standard hinge loss function that makes the cost function convex. Given the target neighbor membership, M can be solved using the semi-definitive programming (SDP) algorithm.

In LMNN, the learning attempts to minimize the local distances between data points and their target neighbors. Compared to global methods that minimize distances between all pairs of data points of the same labels, LMNN can work out the solution more efficiently [15]. Additionally, in global methods, since the distances between all the same data points are treated equally, each class is actually considered as a uni-modal distribution. However, in LMNN, because data points are trained to be only close to their target neighbors, each class is actually considered as a multi-modal distribution. The metric learned by LMNN preserves the local compactness of the data points, therefore, it is also able to separate the data points into more detailed classes.

However, such a property makes LMNN sensitive in the learning phase. Since selecting the target neighbor is critical for the success of training, if the wrong target neighbors are initialized, the cost function will become unreasonable, and this will eventually result in a poor distance function which cannot properly measure the similarity.

5.2 ALMNN Algorithm

The initialization issue of the original algorithm is very likely to happen when dealing with mixed, noisy feature vectors such as those used in this work. The reason is two-fold: (1) different features usually have different scales; and (2) some dimensions of the feature vectors are sometimes too noisy for training. If the noisy dimensions are weighted too heavy in determining the target neighbors, the initialization will fail.

To deal with this issue, we devise an extension of LMNN known as the Adaptive LMNN. The algorithm relies on an additionally introduced binary feature mask $B = (b_1, ..., b_d)^T$, which has the same length as the feature vector (in our case $d = 1684$). This mask is used to turn on the ith dimension of the feature vector by setting $b_i = 1$, and turn off the ith dimension of the feature vector by setting $b_i = 0$. With this mask, we define a new cost function:

$$L(M, B) = \sum_{i,j} Dst_M(B \otimes f_i, B \otimes f_j) + C \sum_{i,j,l} h(1 + Dst_M(B \otimes f_i, B \otimes f_j) - Dst_M(B \otimes f_i, B \otimes f_l)), \quad (3)$$

where $\otimes$ specifies dimension element-wise multiplication of two vectors. This cost function is derived from the intuition that some dimensions of the feature vector may be very noisy, and the remaining dimensions should work well. The goal of this learning becomes to find a pair of M and B that minimizes $L(M, B)$.

Given a fixed B, we can prove $L(M, B)$ is also convex. Therefore, the SDP algorithm still can be used to find M in the same way as in the original LMNN [14]. However, because the number of possible values of B is very large (in our case 2^{1684}), it is not feasible to estimate M for every possible B and select the best pair. Instead, we propose a feature-selection-like learning algorithm to conduct the learning.

The learning algorithm is implemented in two steps. In the first

step, the complete data set D is used to estimate a discriminability list $P = (p_1, ..., p_d)$ $(d = 1,684)$. Each p_i specifies how well the ith feature dimension can separate the videos into pre-defined categories: the higher the value is, the more discriminative the corresponding dimension should be. The value of p_i is computed by taking the ratio of inter-class variance to the intra-class variance according to the Fisher' criterion. Algorithm 1 summarizes the details of computing P from D.

Algorithm 1 Compute the discriminability list P

Require: the training data: $D = \{(f_i, t_i)\}_{i=1}^{n}$, $t_i \in \{\omega_1, ..., \omega_C\}$
Require: the number of data points in ω_j: n_j
Require: the element-wise multiplication: $\otimes$
Require: the element-wise division: $\oslash$
compute the total mean of all data points: $m = \frac{1}{n}\sum_{i=1}^{n} f_i$
compute the total variance of all data points: $S_T = \sum_{i=1}^{n}(f_i - m) \otimes (f_i - m)$
for $j = 1$ to C **do**
compute the mean for class j: $m_j = \frac{1}{n_j}\sum_{i \in \omega_j} f_i$
compute the within-class variance for class j: $\sigma_j = \sum_{i \in \omega_j}(f_i - m_j) \otimes (f_i - m_j)$
end for
compute the total within-class variance: $S_W = \sum_{j=1}^{c} \sigma_j$
compute the total between-class variance: $S_B = S_T - S_W$
compute P: $P = S_B \oslash S_W$

In the second step, B and M are learned by using P to continuously update B. At first, B is initialized as an all 0 vector. In each iteration τ, one dimension of the feature vector is turned on by updating the corresponding dimension in B to 1 (the order for updating depends on the value in P, from high to low). The resulted B^τ is then substituted into Eq. (3), and the SDP algorithm is used to work out its corresponding distance metric M^τ which has the minimum cost. The iteration continues until all dimensions of B become 1. Next, we compare the costs of all pairs of M and B, and select the pair M^* and B^* which has the minimum cost. The learning algorithm is summarized in Algorithm 2.

Algorithm 2 Learn B^* and M^*

Require: the training data: $D = \{(f_i, t_i)\}_{i=1}^{n}$
Require: the discriminability list: $P = (p_1, ..., p_d)$
Require: the feature mask at time τ: $B^\tau = (b_1, ..., b_d)^T$
$B^0 = 0$ {initialization}
for $\tau = 1$ to d **do**
$a = \arg\max_j(p_j)(j = 1, ..., d)$ {find the index of the strongest feature}
$p_a = 0$ {remove the feature from the list}
$b_a = 1$ {turn on the ath feature}
$M^\tau = \arg\min_M L(M, B^\tau)$ {given B^τ, solve M^τ with SDP}
$C^\tau = L(M^\tau, B^\tau)$ {compute the cost for M^τ and B^τ}
end for
$v = \arg\min_\tau C^\tau(\tau = 1, ..., d)$
$B^* = B^v$, $M^* = M^v$

In Algorithm 2, learning is conducted by slowly adding feature dimensions. The learned B^* specifies only a subset of all feature dimensions, and the learned B^* and M^* have the best combinational performance in minimizing the learning cost.

5.3 Dimension Reduction

In LMNN, when dealing with feature vectors of a high dimensionality, the learning becomes untractable due to a high computational complexity. Weinberger et al. [14] suggest to adopt principal component analysis (PCA) to reduce the feature vectors to a lower dimension before learning. PCA finds a low dimensional subspace which preserves most information and captures major variations of the original high dimensional feature vectors. It is used as a standard preprocessing step of LMNN leaning in many works [14], [20], [23], [24].

The proposed ALMNN also suffers when the feature vectors are of a high dimensionality. We follow [14] to also utilize PCA to do the preprocessing. In ALMNN, the subspace is learned using original unmasked raw feature vectors. During ALMNN learning, at each time τ in Algorithm 2, training vectors are firstly masked by the updated B^τ, then projected to low dimensional feature vectors in the learned subspace. The low dimensional feature vectors are then used to work out the M^τ that minimizes the cost defined in Eq. (3). Note that, in ALMNN, in each learning iteration, we do not use diffident projections that are learned every time after the feature vectors are masked, but use the same projection which is learned from unmasked feature vectors. This makes the PCA be only used to bridge the original high dimensional feature space and its subspace. The projection is not affected by the feature mask which changes dynamically.

PCA helps to reduce the dimensionality, thus, simplifies the computation. However, because it is an unsupervised processing, it does not reduce the noise in the original high dimensional feature vectors. Beside PCA, there also exist supervised dimension reduction approaches, such as linear discriminant analysis (LDA) [19]. LDA finds a low dimensional subspace by maximizing the between-class scatter while minimizing the within-class scatter. Using LDA to conduct the dimension reduction sometimes could lead to less noisy low dimensional feature vectors. However, as a dimension reduction tool, LDA has two limitations: 1) comparing to PCA, using LDA will result in more loss of variance and information of the original data; 2) when the densities of the classes are not distributed in multivariate Gaussian, LDA may extract spurious features that are poor for classification [14]. In our work, because ALMNN has already integrated feature selection and the distribution of event categories is complicated, we do not adopt LDA for dimension reduction.

6. Experimental Results

We implemented the system in a real nursery school environment. The cameras we used are Techno One DTC-301, which record videos with a resolution of 320 × 240, at 4 frames per second. The RFID solution is provided by Megras. It uses the RFT15-05 tags which transmit signals in every 1.1 seconds. We conducted two experiments to evaluate the system. The first experiment evaluated the learned distance metric; the second one evaluated the quality of the automatically generated digest.

6.1 Evaluation of Distance Metric Learning

The distance metric plays a crucial role in our system. In this experiment, we evaluate its working in the chief event clas-

predicted class

actual class	play	group	sleep	meal
play	522	22	2	3
group	30	167	2	13
sleep	2	3	238	3
meal	10	6	0	139

(a) ALMNN

predicted class

actual class	play	group	sleep	meal
play	442	63	15	29
group	84	74	13	41
sleep	23	10	202	11
meal	15	22	8	110

(b) LMNN

Fig. 5 Confusion matrices of the classification result of different classification methods.

Table 1 Accuracy of the three classification methods.

	ALMNN	LMNN	AdaBoost
Paly	95.1%	80.5%	91.6%
Group	78.8%	34.9%	71.7%
Sleep	96.7%	82.1%	95.1%
Meal	89.7%	71.0%	84.5%
Total	91.7%	71.3%	87.8%

sification. To do so, we collected a data set comprising 4,648 one-minute video clips from the surveillance cameras. These video clips were taken from the seven different cameras within a week. We manually provided the chief event labels to these videos as ground truth, and randomly separate them into a training set (play: 1639, group: 642, sleep: 734, meal: 471) and a testing set (play: 549, group: 212, sleep: 246, meal: 155).

We learned the Mahanalobis distance metric with ALMNN algorithm. Because the feature vectors are of high dimension, we used PCA for dimension reduction (according to Section 5.3). We first conducted PCA on the training set feature vectors, and selected the top 80 eigenvectors for projection. Then, the projection is used during learning: in Algorithm 2, at each time τ, after the feature mask is applied, training vectors are projected to the linear subspace spanned by the selected 80 eigenvectors. We used the 80-d subspace because it makes the training complexity acceptable, and the top 80 eigenvectors capture almost all the information of the feature vectors. We have also tried subspaces with higher and lower dimensionality: from 10-d to 100-d with a step of 10-d. We observed that ALMNN's performances are the same for subspaces higher or equal to 40-d, and the performances show obvious decreasing for subspaces lower than 40-d.

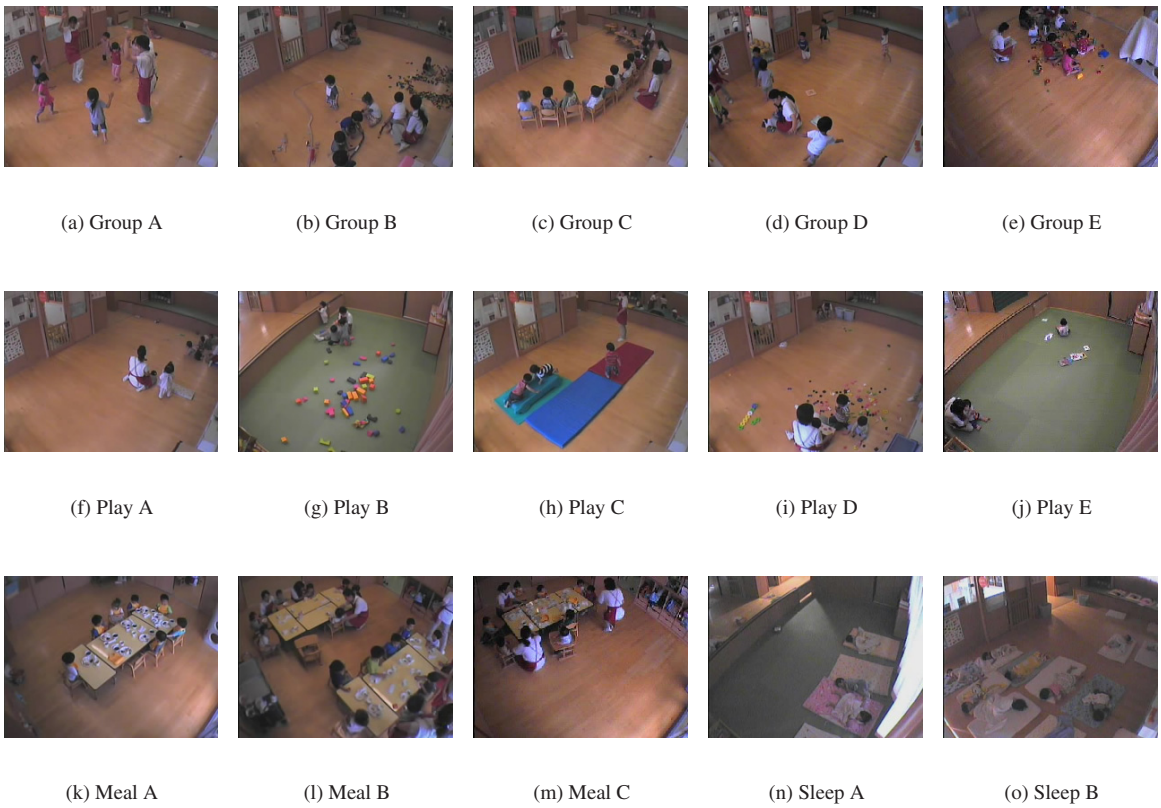
We used the learned distance metric to perform kNN classification ($k = 5$) on the testing set. The accuracy (percentage of correctly classified video segments) of the method is shown in **Table 1**. Our method has an overall accuracy of 91.7%, which is much better than the 76.6% that reported in Ref. [12].

For a comparison, we implemented two other methods on the same dataset. One is the baseline method, which also uses the same kNN classification rules. Compared to the proposed method, the baseline method used the same feature set but a different distance metric learned from the LMNN algorithm. For a fair comparison, PCA was also used to reduce the feature vector to 80-d. The other method is an improved version of the method described in Ref. [12]. In Ref. [12], the authors used a relatively simple feature set (including background subtraction, inter-fame subtraction, and black pixel extraction) and the AdaBoost to perform the classification. In order to make a fair comparison, we replaced the original feature set with the same feature set used in the proposed method and set the weak classifier number in AdaBoost to 20. The results of these two methods are shown in Table 1.

Comparing our ALMNN-based classification with the LMNN-based classification, we can see that the former shows approximately a 20% improvement in the overall classification accuracy as compared to the latter. As can be seen from the resulting confusion matrix in **Fig. 5**, LMNN performed especially poorly for the classes "play" and "group." This is because these two classes share some common visual characteristics and are similar in many feature dimensions. In LMNN, when using Euclidean distance to initialize the target neighbors, the selected target neighbors tend to be similar to the imposter data points. In such case, during the learning, when pushing the imposter data points away, the target neighbors are also being pushed away. This makes the cost minimization cannot hit a low value, and the quality of the learned metric becomes poor. Conversely, for learning, the ALMNN uses a new cost function that helps to avoid such an issue; hence, it yields a considerable improvement.

When we compare our ALMNN-based classification with AdaBoost-based classification, we also see an improvement for different kinds of events. Because the feature set is the same, the improvement is mainly beneficial due to the distance-metric-powered kNN rule. We think that this is because the categories that we deal with are of large within-class variance. For such a classification task, example-based classification methods, such as kNN, are more powerful when combined with an appropriate similarity measurement. On the other hand, the AdaBoost-based method also outperforms the LMNN-based method. We think that this is because the AdaBoost itself is robust to noise. In AdaBoost, learning is implemented by training many weak classifiers. Every loop in the weak classifier training is similar to a feature selection step. This makes it robust against the noise in feature vectors, and eventually works better than the LMNN-based classification.

We also integrated the ALMNN and LMNN algorithm with a supervised dimension reduction approach - linear discriminant analysis (LDA) [19], to see if the performance can be further improved. When utilizing LDA, we use the same 80-d subspace as we used in PCA based dimension reduction. The results are summarized in **Table 2**. We observed the performance of LMNN improves (+9.3%) but the performance of ALMNN decreases



(a) Group A (b) Group B (c) Group C (d) Group D (e) Group E

(f) Play A (g) Play B (h) Play C (i) Play D (j) Play E

(k) Meal A (l) Meal B (m) Meal C (n) Sleep A (o) Sleep B

Fig. 6 Examples of event clustering results.

Table 2 Performance of ALMNN and LMNN with LDA.

	ALMNN + LDA	LMNN + LDA
Paly	91.8%	90.0%
Group	76.4%	49.5%
Sleep	94.3%	87.8%
Meal	78.7%	78.1%
Total	87.8%	80.6%

(−3.9%). In our opinion, LDA works like a double-edged sword: it is supervised and can reduce some noises in data, but it also results in more loss of information and variation during dimension reduction, and sometimes extracts spurious features for classification. LMNN got improved because it benefited from the positive property of LDA. On the other hand, ALMNN got a decreased performance because ALMNN itself is robust against noisy features, and is mainly affected by the negative properties of LDA.

In our proposed system, another use of the learned distance metric is to divide chief events into visually different individual event clusters, and select representative videos for each cluster. To intuitively show its working, we used the video data of a single day from all the cameras and performed a clustering experiment. We used the method described in Section 4.2 to predict the chief event labels for the video segments, and then conducted agglomerative clustering within each chief category using a cut-off threshold. Please note that in the actual system, clustering is done for every child (a subset of a complete day). We use the data of a complete day because we want to see the general applicability of the clustering. (In agglomerative clustering, when the cut-off threshold is fixed, the clustering result of the subset data points is coherent with the clustering result of all data points.)

The clustering results in a number of clusters. After eliminating the small clusters, we get multiple individual events: group (5), meal (3), sleep (2), play (5). We visualize the results in **Fig. 6** by displaying the middle frames in the centroid videos of each individual event cluster. We can see that they show visually different individual events. At some level, this confirms that the clustering is capable of dividing the chief event into visually different individual event clusters.

6.2 Evaluation of Digest Generation

In order to evaluate the quality of the generated digest, we invited one child (male) to participate in the experiment. We put an RFID tag in his pocket for a day and used the proposed method to generate a digest for him. The video data we used was captured using all the seven cameras, from 9 am to 6 pm hours. The log file, which is also within the same period, comprises 15,642 records.

First, we process the entire video into the tracing video of the target child by analyzing the log file. The analysis of the log takes 15 s. We set the rejection threshold to 10 records, and 369 one-minute video segments were selected. In order to confirm the accuracy of this RFID-based pre-processing, we selected 74 video segments by uniformly taking samples in the resultant video, and manually confirmed if the target child existed in them. If the target child exists in the video for more than 80% of the one-minute duration, we treat it as a correct selection. Through verification of the 74 video segments, the selection accuracy was confirmed to be 100%.

We computed the feature vectors for these selected videos, and used the method described in Section 4.2 to divide them into chief event categories, and then into individual event clusters. Since

Table 3 Comparison between Digests A and B (manual).

	A is better	Almost the same	B is better
Q1	1	11	3
Q2	1	12	2

Table 4 Comparison between Digests A and C [12].

	A is better	Almost the same	C is better
Q1	10	3	2
Q2	11	3	1

we want to generate a ten-minute digest, we picked up the top 10 largest individual event clusters for the digest construction. The chief labels of the selected clusters are: group (3), meal (2), sleep (1), play (4). We selected the centroid videos of these clusters and linked them in a periodically linear order. This finally resulted in a ten-minute digest, which covered ten different, individual events. The whole generation process cost approximately 2 hours; most of the time was spent on feature computation (approximately 100 min) and video linking/ compression (approximately 18 min). In the following section, we refer to the resultant digest as Digest A.

For a comparison, we also produced two digests from the same data using two other methods. Digest B was produced manually following three rules: (1) the digest should cover different chief events; (2) through the digest, it should be easy to understand the daily life of the target child; (3) the digest should contain various activities. Digest C was produced using the method in Ref. [12]. In addition to the differences in the event classification accuracy, the video selection strategy in Ref. [12] is also different from that in this work. The selection rules in Ref. [12] include: (1) selection of videos with high inter-frame difference (high motion intensity), and (2) selection of videos during different time periods over a day.

We invited 15 parents to participate in our questionnaire survey. They were asked to watch Digests A and B and Digests A and C on two different days. Then, they were asked two questions: 1) which digest is better? and 2) which digest gives a better description of the child's daily life? Through question 1, we hoped to obtain an overall evaluation of the quality of the digests, and through question 2, we hoped to know how well the digests could reflect the daily life.

The evaluation result between Digests A and B is shown in **Table 3**. The table shows that the quality of the digest generated by our system is similar to the manually generated one. For the five answers that pointed out that B is better than A, we asked for reasons. Regarding the first question, three participants selected "B is better" because "B has more interesting scenes, while A appears to be plain." With respect to the second question, the participants who chose "B is better" essentially indicated that "B is smoother than A." These answers indicate that when humans manually generated digests, they tend to select interesting scenes and unconsciously link them in a comfortable manner. However, this is outside the scope of our current system.

The evaluation result between Digests A and C is shown in **Table 4**. The table shows that most people thought that A is better than C, which confirms the superiority of the proposed method. We further collected the reasons for the responses of the participants. Most participants thought that "A is better" because "A includes more types of activities," "A has less redundant scenes," and so on. This suggests that in the proposed method, the strategy of dividing the chief event into more detailed individual events was able to achieve the desired results. Specifically, the parameter "make a digest cover different activities" met the requirements of most parents.

7. Conclusion

In this paper, we proposed a novel approach for summarizing nursery school surveillance videos. The approach makes full use of a learned distance metric and generates digests that cover and reflect different activities of children. We implemented the approach as a practical system in a real nursery school environment, and confirmed its ability to generate digests that satisfy the requirements of parents. Additionally, the proposed approach only relies on general knowledge of daily lives in nursery schools, and it uses environment-independent visual features to analyze video contents. The proposed approach can easily be adapted to common nursery school environments.

Another contribution of this paper is a novel distance metric learning algorithm. Since the fundamental of the proposed summarization approach is a distance metric, its quality is crucial. To robustly learn such a distance metric from mixed, noisy feature vectors, we proposed a new learning algorithm called ALMNN. This algorithm extended the existing LMNN algorithm with a new cost function and a new feature selection-like learning process. It outperforms the existing LMNN algorithm by over 20% in the classification task, and has a strong potential to be widely used in other applications.

However, our experiments suggested that there are gaps between digests generated using our approach and those generated manually. Selecting not only visually representative but also interesting videos will be key preparing better digests. Therefore, exploring the mechanism for finding "interesting" videos is one of our future works.

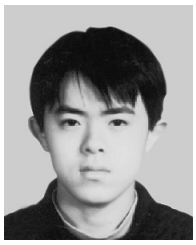
In addition, the approach discussed in this paper only dealt with visualization aspects. It makes a very limited use of information-rich nursery school surveillance videos. In order to make full use of these, we will develop high-level activity analysis methods (e.g., action recognition, interaction understanding) to extract more meaningful information from such videos for various applications.

Acknowledgments The authors would like to thank the staffs in Nagoya University Cosmos Nursery School for their support in the experiment. This work is supported by the Grant-in-Aid for JSPS Fellows Number 2410663, the Strategic Information and Communications R&D Promotion Programme Number 131306004 and the National Institute of Information and Communication Technology.

References

[1] New service "Life Camera" for Nursery Schools: available from ⟨http://www.kidscamera.net/⟩.
[2] Hauptmann, A.G., Gao, J., Yan, R., Qi, Y., Yang, J. and Wactlar, H.D.: Automated Analysis of Nursing Home Observations, *IEEE Pervasive*

Computing, Vol.3, No.2, pp.15–21 (2004).
[3] Rehg, J.M.: Behavior Imaging: Using Computer to Study Autism, *Proc. 12th IAPR Workshop on Machine Vision Applications* (2011).
[4] Rodriguez, M.: CRAM: Compact Representation of Actions in Movies, *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2010).
[5] Hashimoto, T., Shirota, Y., Mano, H. and Tanaka, H.: Prototype of Digest Viewing System for Television, *IPSJ Trans. Databases*, Vol.41, No.SIG3 (TOD6), pp.71–84 (2000).
[6] Takahashi, Y., Nitta, N. and Babaguchi, N.: Video Summarization for Large Sports Video Archives, *Proc. IEEE International Conference on Multimedia & Expo* (2005).
[7] Miura, K., Hamada, R., Ide, I., Sakai, S. and Iizawa, A.: Motion Based Automatic Abstraction of Cooking Videos, *IPSJ Trans. Computer Vision and Image Media*, Vol.44, No.SIG9 (CVIM7), pp.21–29 (2003).
[8] Bach, N.H., Shinoda, K. and Furui, S.: Robust Highlight Extraction using Multi-stream Hidden Markov Models for Baseball Video, *Proc. 2005 International Conference on Image Processing* (2005).
[9] Ren, J. and Jiang, J.: Hierarchical Modeling and Adaptive Clustering for Real-Time Summarization of Rush Videos, *IEEE Trans. Multimedia*, Vol.11, No.5, pp.906–917 (2009).
[10] Tavanapong, W. and Zhou, J.: Shot Clustering Techniques for Story Browsing, *IEEE Trans. Multimedia*, Vol.6, No.4, pp.517–527 (2004).
[11] Wang, Y., Kato, J., Zhou, W. and Yokoi, S.: Digest Generation of Kindergarten Surveillance Video with Location Information and Visual Features, *Proc. 4th International Conference on Innovative Computing, Information and Control* (2009).
[12] Ishikawa, T., Wang, Y. and Kato, J.: Daily Digest Generation of Kindergarten from Surveillance Video, *IEEJ Trans. Electronics, Information and Systems*, Vol.131, No.2, pp.385–392 (2011).
[13] Hastie, T. and Tibshirani, R.: Discriminant Adaptive Nearest Neighbor Classification, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.18, No.6, pp.607–616 (1996).
[14] Weinberger, K.Q. and Saul, L.K.: Distance Metric Learning for Large Margin Nearest Neighbour Classfication, *Journal of Machine Learning Research*, Vol.10, pp.207–224 (2009).
[15] Ramanan, D. and Baker, S.: Local Distance Functions: A Taxonomy, New Algorithms, and an Evaluation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.33, No.4, pp.794–806 (2011).
[16] Otsu, N.: A Threshold Selection Method from Gray-Level Histograms, *IEEE Trans. Sys., Man., Cyber*, Vol.9, No.1, pp.62–66 (1979).
[17] Dalal, N. and Triggs, B.: Histograms of Oriented Gradients for Human Detection, *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (2005).
[18] Gonzales, R. and Woods, R.: Digital Image Processing, *Prentice Hall* (2001).
[19] Fisher, R.A.: The Use of Multiple Measurements in Taxonomic Problems, *Annals of Eugenics*, Vol.7, No.2, pp.179–188 (1936).
[20] Mensink, T., Verbeek, J., Perronnin, F. and Csurka, G.: Metric Learning for Large Scale Image Classification: Generalizing to New Classes at Near-Zero Cost, *Proc. European Conference on Computer Vision* (2012).
[21] Guillaumin, M., Mensink, T., Verbeek, J. and Schmid, C.: TagProp: Discriminative Metric Learning in Nearest Neighbor Models for Image Auto-Annotation, *Proc. 12th International Conference on Computer Vision* (2009).
[22] Kumar, M.P., Torr, P.H.S. and Zisserman, A.: An Invariant Large Margin Nearest Neighbour Classifier, *Proc. 11th International Conference on Computer Vision* (2007).
[23] Zhang, G., Wang, Y., Kato, J. and Mase, K.: Local Distance Comparison for Multiple-shot People Re-Identification, *Proc. 11th Asian Conference on Computer Vision* (2012).
[24] Tran, D. and Sorokin, A.: Human Activity Recognition with Metric Learning, *Proc. European Conference on Computer Vision* (2008).



Yu Wang received his M.S. degree in Information Science and Ph.D. degree in Engineering from Nagoya University in 2010 and 2013, respectively. He is currently a postdoctoral researcher with the Graduate School of Information Science, Nagoya University. He has been supported by the JSPS Fellowship for Young Scientists since 2012. His research interests are object recognition and visual event categorization. He is a member of IEEE.

Jien Kato received her M.E. and Ph.D. degrees in Information Engineering from Nagoya University in 1990 and 1993, respectively. She is currently an associate professor with the Graduate School of Information Science, Nagoya University. Her research interests include computer vision, machine learning, multi-sensor perceptual computing and their applications. She is a member of IEICE, IPSJ and a senior member of IEEE.

Kenji Mase received his B.S. degree in Electrical Engineering and M.S. and Ph.D. degrees in Information Engineering from Nagoya University in 1979, 1981 and 1992 respectively. He became a professor of Nagoya University in August 2002. He is now with the Graduate School of Information Science, Nagoya University. He joined the Nippon Telegraph and Telephone Corporation (NTT) in 1981 and had been with the NTT Human Interface Laboratories. He was a visiting researcher at the Media Laboratory, MIT in 1988–1989. He has been with ATR (Advanced Telecommunications Research Institute) in 1995–2002. His research interests include gesture recognition, computer graphics, artificial intelligence and their applications for computer-aided communications. He is a member of IPSJ, JSAI, VRSJ, HISJ and ACM, a senior member of IEEE Computer Society, and a fellow of IEICE.