Incremental Learning of Novel Activity Categories from Videos

M. S. Ryoo, Jihoon Joung, Sunglok Choi, and Wonpil Yu
Robot/Cognition Research Department
Electronics and Telecommunications Research Institute
Daejeon, Korea 305-700
Email: {mryoo,jihoonj,sunglok,ywp}@etri.re.kr

***Abstract*—We present a methodology for learning novel human activities incrementally. In many real-world scenarios (e.g. YouTube), new videos of novel activities are provided additively, and the system must incrementally adjust its activity models rather than retraining the entire system after each addition. We introduce our incremental codebook learning algorithm for an efficient mining of important visual words for human activities, and propose a method that incrementally trains activity models using them. The experimental results show that our approach successfully learns human activities from increasing number of training videos, while maintaining its recognition performance comparable to previous non-incremental systems.**

I. Introduction

An automated analysis of events and activities from video data is an important problem with a large amount of public interests. Particularly, a system to learn and recognize events from online videos (e.g. YouTube) is obtaining an increasing amount of attentions from researchers. More than 20 hours of web-videos are being uploaded to these websites per minute nowadays, and the activity recognition paradigm is shifting to process and utilize these abundant videos dynamically uploaded by users. In contrast to traditional systems trained with a limited amount of videos offline, today's systems are required to incrementally update themselves and learn new activity categories from an increasing number of videos. Tons of user-created contents (UCCs) of novel categories are being uploaded, and the goal is to analyze and utilize them.

In this paper, we present an efficient *incremental concept learning* methodology for recognizing human activities from videos. The motivation is to construct an activity recognition system which efficiently adjusts itself as a new video sample is provided, instead of retraining the entire system after every video addition. The proposed methodology not only updates models for existing activity classes incrementally, but also learns an entirely new activity class as its videos are given to the system sequentially (Figure 1). Such method is especially important for interactive recognition systems whose categories increase corresponding to user demands (e.g. UCC searching), and for large-scale systems that require a significant amount of computational space and time to retrain them (e.g. systems learning from YouTube videos). Even though previous activity recognition systems [1], [2], [3], [4] have shown successful results, they were not designed for the incremental learning; they had no choice but to retrain them again as the number of available videos and their event categories (e.g. greeting, fighting, stealing, ...) are increasing.

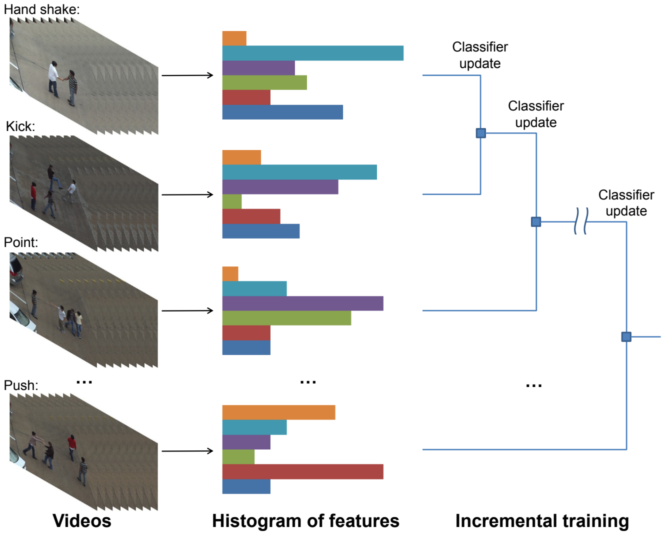

Fig. 1. An illustration of our incremental activity learning process.

Our approach is to represent an activity in terms of *visual words* (i.e. clusters of features) describing its videos, and to incrementally learn new words for existing/novel activities as training examples are provided. Each 'word' corresponds to a set of local spatio-temporal features with similar appearances, and the activity is learned by modeling the distribution of these words in its videos. We present an algorithm to learn an optimal set of words (i.e. codebook) per class incrementally. New visual words are generated as videos from an existing or novel activity class is provided, while old words are updated or merged based on the new observations. Furthermore, our algorithm is designed to share a portion of words across classes when constructing activity codebooks, enabling efficient learning of novel activity categories. A histogram of visual words required for the recognition are constructed for each activity class in an incremental fashion as well; the histogram is constructed or updated after each video addition sequentially, while maintaining characteristics of the previous distribution. As a result, our human activity classifiers using the histograms are constructed and updated incrementally.

II. Related Works

Incremental learning. Even though incremental learning of human activities from videos has not been explored in depth, several computer vision researchers have attempted to learn novel objects from images incrementally. Bart and Ullman [5] was able to learn a new object category from a single image by taking advantage of features learned for existing object categories. Torralba *et al.* [6] did not focus on incremental learning, but they have proposed an idea of sharing features for multi-class object learning that potentially benefits incremental object learning. Opelt *et al.* [7] designed an object detection system using boosting. A detector has been constructed per object, learning weak classifiers using edge features incrementally.

Activity recognition. Since early 1990s, many researchers have studied human activity recognition [8]. Hidden Markov models and other sequential models have been popularly used for the recognition of action-level activities [9], [10], [11], while hierarchical approaches have been developed for recognizing high-level activities (e.g. complex human-human interactions) [12], [13], [14], [15]. Most of these approaches take advantage of image features extracted per frame, modeling activities as a sequence of feature observations. However, such features often require good foreground segmentation method, preventing the activity recognition methodologies from being applied to videos with moving backgrounds, changing illuminations, and/or camera movements (e.g. UCC videos).

Recently, action recognition approaches using 3-D spatio-temporal local features have gained a wide range of interests, because of their reliability under noise, illumination changes, and camera movements. Schuldt *et al.* [1] presented a methodology to extract sparse local features from 3-D XYT volume constructed by concatenating images along time axis. Similarly, Dollar *et al.* [2] have introduced ‘cuboid’ feature descriptors modeling appearance changes in local spatio-temporal regions. A statistical pLSA model for an unsupervised learning of one-person actions was adopted in [3]. In order to recognize multi-person complex activities, Ryoo and Aggarwal [4] developed the spatio-temporal relationship match that models structural distributions of 3-D XYT features.

However, these systems were unable to learn activities from increasing number of videos. The previous approaches require retraining of the entire system as one video example is added to the system. In many real-world applications, a video example with important information may be added later than the others, and it was difficult for the previous approaches to update the systems corresponding to the new information. Furthermore, they must maintain all features from training examples in order to do so. It was not possible for them to learn novel activity classes incrementally.

Reddy *et al.* [16] modeled feature codebooks in a incremental fashion for human activity recognition. They have shown the idea that codebooks can be learned sequentially, but their system was unable to learn novel class. Zuniga *et al.* [17] attempted to learn new human activities by incrementally adding scene states. However, they represented an action as a transition between two states, and was difficult to learn complex activities which cannot be described with simple two-state sequential models.

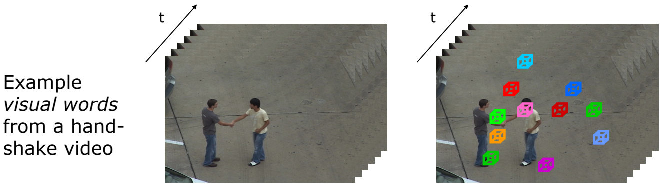

Fig. 2. Example 3-D spatio-temporal (XYT) features extracted from a video. Each feature describes an appearance of a local XYT volume corresponding to it. The features are categorized into several types (colors) based on the codebook.

III. Codebook Learning

A. Visual Words

We define a *visual word* of activity videos as follows.

Definition 1: A *visual word* is a d-dimensional sphere that satisfies the following two constraints:

1) Its feature density is greater than the given threshold.

$$\frac{|F|}{r^d} > c$$

where F is a set of d-dimensional member feature vectors included by the sphere, r is the radius of the sphere, and c is the threshold.

2) It does not overlap with any other visual words.

We say that any feature vector within the sphere is a member of the word. That is, a visual word is modeled as a particular region in a feature hyperspace which includes more than $c \cdot r^d$ features, regardless where the features have originated.

Each visual word groups a set of features with similar values, dividing entire features into several feature appearance categories as a result. If 3-D spatio-temporal local features (e.g. [1], [2]) are used, similarly shaped spatio-temporal XYT regions of videos are grouped to correspond to a single word. These visual words enable the system to represent each video as a set of words appearing in the video (e.g. Figure 2). Note that our definition of visual words are different from conventional bag-of-words definitions used in previous works [1], [2], [3]: Our sphere definition better describes discrete and dense nature of XYT appearance-based local features, since we do not force the visual words to occupy the entire feature hyperspace.

A set of all visual words forms a *codebook* (i.e. a visual word dictionary) for features, and the goal of our algorithm is to learn an optimum codebook correctly reflecting feature distributions. Diverse feature vectors are observed from training videos of various activities, which are provided incrementally. The activity recognition system is required to learn the optimum m number of visual words (i.e. a codebook with size m) incrementally so that each feature vector observed is covered by one among them.

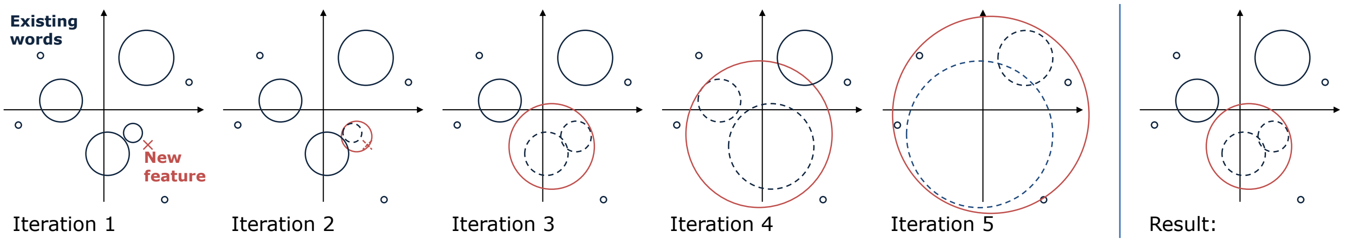

Fig. 3. The merging process for finding the largest super word continues until the density constraint is violated for the first time. Among super words constructed during the process, the largest one whose boundary does not overlap with any other existing words (e.g. iteration 3) is selected.

B. *Incremental Codebook Learning*

In this subsection, we present an efficient codebook learning algorithm which has an *incremental property*.

Definition 2: We say that a codebook learning algorithm has an *incremental property*, if and only if

$$\forall a, b : (\exists i : a \in w_i^{(t-1)}, b \in w_i^{(t-1)}) \rightarrow (\exists j : a \in w_j^t, b \in w_j^t)$$

where $w_i^{(t-1)}$ is the ith word of a codebook generated after observing $(t-1)$ features, and w_j^t is the jth word of a codebook after observing an additional feature (i.e. t features).

The incremental property of a codebook learning algorithm guarantees that member features of a learned visual word always stay together once it is formulated. That is, feature vectors assigned to an identical word do not split into two different words even if the codebook is updated by observing multiple features afterwards. In contrast to previous codebook learning using clustering algorithms such as k-means, an incremental algorithm is able to learn increasing number of visual words in a natural way without modifying (i.e. remaking) its previous codebook update decisions. This enables additive learning of visual words without storing all previously observed features; our system only maintains {centroid, radius, and feature count} of each visual word so that it represents all member features in it.

We propose an incremental codebook learning algorithm with a greedy optimization strategy. Our algorithm is an iterative algorithm that updates the codebook as a new feature is added. At every iteration, our algorithm minimizes the number of words, m, while making all observed features to be covered by the m words. When a new feature from a new training video is added to the system, our algorithm either creates a new word that only contains the new feature or finds the *largest super word* that includes the feature.

Definition 3: A word w_q is the *largest super word* of a word w_p, if and only if (i) w_q satisfies the definition of a visual word and (ii) w_q encloses the maximum number of words while including w_p.

```
// Feature f is the feature being added to the system
UPDATE_CODEBOOK(Feature f)
{
  Word $w_{max}$ = LARGEST_SUPER_WORD(makeword(f));
  update the codebook to include $w_{max}$;
}

LARGEST_SUPER_WORD(Word $w_p$)
{
  if (density($w_p$) < c) return null;
  else
  {
    Word $w_r$ = the nearest word of $w_p$;
    Word $w_q$ = LARGEST_SUPER_WORD(merge($w_p$, $w_r$));

    if ($w_q$!=null) return $w_q$;
    else if(overlaps($w_p$, $w_r$)) return null;
    else return $w_p$;
  }
}
```

Fig. 4. Our greedy algorithm to update the codebook by searching the largest super word recursively.

The largest super word can be formulated by merging all nearby words while keeping the two constraints in Definition 1. Our algorithm first creates a singleton word with zero radius based on the new feature, and then searches for the largest super word by recursively merging it with the nearest word until the density constraint is violated. Figure 3 illustrates the process of our algorithm. Finding the largest super word of a feature enables the system to minimize the number of words by merging the feature with nearby words as much as possible. Among multiple largest super words with an identical m, our algorithm selects the one with the highest density. Figure 4 shows the pseudo code of our algorithm.

Furthermore, we present an algorithm to efficiently select important visual words for the construction of class-specific codebooks. Using all visual words as a common codebook is often computationally inefficient for activity recognition, and we develop an algorithm to select the k most frequent words (i.e. representative words) for each activity class. While learning visual words of the activities, our algorithm maintains a 'rank list' of words in an incremental fashion for each activity class. This rank list is a sorted list describing the total number of occurrences of the words in training videos of each activity. Our rank list is updated after each feature observation, adding a new entry (i.e. new word) or making one of its entry have a higher rank while removing another entry (i.e. merge). If a video from a new class is provided, a new rank list is constructed. Our idea of separately maintaining visual words and rank lists enables sharing of important features among activity classes, while correctly reflecting word distributions per activity.

The time complexity of our algorithm is $O(mn)$ where n is an average number of features per video. These computations are required for each incremental video addition. On the other hand, a traditional codebook learning paradigm using k-means requires $O(lknv)$ computations per video addition, where l is the number of iteration needed for k-means to converge and v is the number of videos. In general, $O(mn) \approx O(kn)$, suggesting that our incremental codebook learning is far more efficient than the traditional clustering.

As a result of our codebook learning algorithm, we obtain k best visual words (i.e. a specialized codebook) for each activity class that has been observed incrementally.

IV. Activity Recognition

In this section, we present our incremental activity learning methodology that takes advantage of incremental codebooks generated from Section III. Our algorithm presented in the previous section updates feature codebooks of activities as videos are provided. In the traditional activity learning paradigm, activity learning must be re-done if codebook entries are modified (e.g. word merging). What we discuss in this section is a learning algorithm that overcomes such limitations; our algorithm incrementally updates the activity models as videos from existing classes are provided and codebook entires are modified, rather than retraining the entire system. Furthermore, our approach learns models for novel activity classes additively as new videos from unseen classes are provided.

We focus on the incremental property (Definition 2) of our system. We represent each activity as a mean histogram of words. That is, we maintain an array where each entry describes an average number of the corresponding word's occurrences in entire activity videos. The sizes and entry values of these histograms change as visual words are being added and merged based on new observations (i.e. training videos). Our system incrementally updates the histogram values so that the average occurrence count of a merged word (i.e. a new word) is computed based on the words being merged (i.e. existing words).

More specifically, we take advantage of the rank lists from Subsection III-B to construct the average histogram per class. A rank list maintains the total number of each word appeared in training videos of each activity class in an incremental fashion, and dividing it by the number of observed training videos provides us the average histogram. This histogram representation enables efficient addition of novel activity classes as well, since our approach automatically constructs a new rank list as a new class is added. For more efficient construction of histograms, we only consider the visual words whose ranking is better than k in at least one activity class.

We design a Bayesian classifier with a Gaussian assumption to recognize represented human activities. Each activity is modeled as a Gaussian distribution having a particular mean (i.e. a mean histogram of words) and variance. The variance is computed similarly to the mean histogram by using the rank lists. With a Gaussian assumption, the class with the maximum posterior probability of generating the given video is selected as the label of the activity:

$$P(A|h) \propto P(h|A) \cdot P(A) = N(\mu, \sigma^2) \cdot P(A) \qquad (1)$$

where h is the histogram of a testing video, A is the class of the activity, and (μ, σ^2) is a pair of mean histogram and variance histogram of the activity A. We assume a uniform prior probability.

Our activity learning algorithm only requires $O(kn)$ computations to construct/update the activity models using the rank lists. On the other hand, previous activity learning algorithm (i.e. non-incremental) with bag-of-words paradigm requires at least $O(knv)$ computations for video addition. v, the number of observed videos, can be several millions in Internet datasets.

V. Experiments

In order to evaluate our approach, we have implemented two types of activity recognition systems following our incremental class learning approach. One system uses cuboid features by Dollar *et al.* [2], and the other system uses Laptev's spatio-temporal features [18]. These two incremental systems were compared with several existing non-incremental methodologies using the same features: k-nearest neighbors (e.g. [2]), support vector machines (similar to [1]), and the Bayesian classifier similar to our algorithm presented in Section IV have been implemented.

We have used the UT-interaction dataset from the SDHA 2010 human activity recognition contest [19]. Videos in this public dataset contain complex interactions between multiple persons such as hand shaking and pushing (Figure 5). Several pedestrians are present in the scene and background/lighting conditions are changing in the videos, making the recognition problem challenging. We have used the segmented version of the UT-Interaction dataset #1 to measure the classification accuracy. This dataset is composed of 10 sets containing a total of 60 videos. The leave-one-set-out cross validation setting (i.e. 10-fold cross validation) was used, measuring the average classification performance. Our system was incrementally trained by providing videos sequentially (i.e. update one-by-one), in contrast to previous approaches trained with the entire videos at once.

Table I shows classification accuracies of our system compared to others. We are able to observed that our incrementally trained system performs comparable to other non-incremental systems. Even though the other results are based on full offline training (i.e. classifiers were learned with the entire

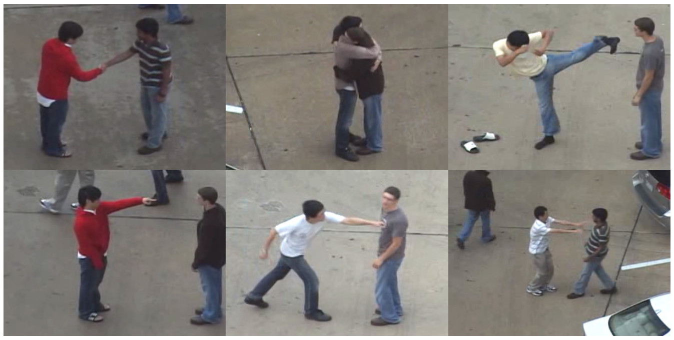

Fig. 5. Example snapshots of the dataset used.

TABLE I
ACTIVITY CLASSIFICATION ACCURACIES OF THE SYSTEMS TESTED ON THE UT-INTERACTION #1 DATASET [19].

System	**Incremental learning**	**Performances**	
		[2]'s feature	**[17]'s feature**
Random chance		16.7%	16.7%
k-NN	X	63.0%	57.0%
Bayesian	X	66.7%	58.2%
SVM	X	75.5%	64.2%
Ours	O	**65.0%**	**55.0%**

training set) which is not possible for increasing videos, the results of our incremental system were as good as the other offline systems. Particularly, our system's performances were almost identical to those of the offline version of our method (i.e. the Bayesian classifier). This result confirms that our approach efficiently learns novel activity categories from increasing number of videos. That is, our approach achieves similar activity classification accuracies without spending a large amount of computations to retrain the entire system.

VI. CONCLUSIONS

We have introduced a methodology for an incremental learning of novel human activity classes. An incremental codebook learning algorithm for efficient selection of visual words from increasing number of videos has been proposed, and an activity learning and recognition algorithm to take advantage of such codebook was presented. In contrast to previous non-incremental methodologies that need to retrain the entire system to update new training videos of existing/novel activity classes, our algorithm incrementally learns activity models as videos are provided. Our experimental results confirm that the performance of the proposed system is comparable to other non-incremental systems using same features.

VII. FUTURE WORKS

In the future, we plan to explore various visual word representations to learning codebooks. Currently, each visual word is represented as a multi-dimensional sphere in the feature hyperspace. That is, there is only one parameter describing each word: the radius. We plan to apply more flexible representations such as multi-dimensional ellipsoids for visual words, more accurately modeling non-spherical feature clusters to increase the system performances. In addition, our incremental activity learning paradigm will be extended to cope with other types of classifiers (e.g. boosting).

ACKNOWLEDGMENTS

This work was supported by the Korea Evaluation Institute of Industrial Technology (KEIT) [10035354 The Development of Autonomous Navigation Systems for a Robot Vehicle in Urban Environment].

REFERENCES

[1] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *International Conference of Pattern Recognition (ICPR)*, 2004.

[2] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *IEEE Workshop on VS-PETS*, Oct 2005, pp. 65–72.

[3] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision (IJCV)*, vol. 79, no. 3, Sep 2008.

[4] M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities," in *International Conference on Computer Vision (ICCV)*, 2009.

[5] E. Bart and S. Ullman, "Cross-generalization: learning novel classes from a single example by feature replacement," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[6] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing features: efficient boosting procedures for multiclass object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[7] A. Opelt, A. Pinz, and A. Zisserman, "Incremental learning of object detectors using a visual shape alphabet," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[8] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (to appear)*, 2010.

[9] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden markov model," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1992, pp. 379–385.

[10] D. Gavrila and L. Davis, "Towards 3-D model-based tracking and recognition of human movement," in *International Workshop on Face and Gesture Recognition*, June 1995, pp. 272–277.

[11] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T PAMI)*, vol. 22, no. 8, pp. 831–843, 2000.

[12] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T PAMI)*, vol. 22, no. 8, pp. 852–872, 2000.

[13] J. M. Siskind, "Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic." *Journal of Artificial Intelligence Research (JAIR)*, vol. 15, pp. 31–90, 2001.

[14] S. Hongeng, R. Nevatia, and F. Bremond, "Video-based event recognition: activity representation and probabilistic recognition methods," *Computer Vision and Image Understanding (CVIU)*, vol. 96, no. 2, pp. 129–162, 2004.

[15] M. S. Ryoo and J. K. Aggarwal, "Semantic representation and recognition of continued and recursive human activities," *International Journal of Computer Vision (IJCV)*, vol. 32, no. 1, pp. 1–24, 2009.

[16] K. K. Reddy, J. Liu, and M. Shah, "Incremental action recognition using feature-tree," in *International Conference on Computer Vision (ICCV)*, 2009.

[17] M. Zuniga, F. Bemond, and M. Thonnat, "Incremental video event learning," in *International Conference on Computer Vision Systems*, 2009.

[18] I. Laptev, "On space-time interest points," *International Journal of Computer Vision (IJCV)*, vol. 64, no. 2-3, pp. 107–123, 2005.

[19] M. S. Ryoo and J. K. Aggarwal, "UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA)," http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2010.