Advanced Engineering Informatics 25 (2011) 771–782

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

Learning and classifying actions of construction workers and equipment using Bag-of-Video-Feature-Words and Bayesian network models

Jie Gong [a,*], Carlos H. Caldas [b,1], Chris Gordon [a,2]

[a] Department of Construction, Southern Illinois University Edwardsville, Edwardsville, IL 62026, USA
[b] Department of Civil, Architectural, and Environmental Engineering, The University of Texas at Austin, 1 University Station C1752, Austin, TX 78712-0273, USA

ARTICLE INFO

Article history:
Received 3 February 2011
Received in revised form 21 April 2011
Accepted 3 June 2011
Available online 2 July 2011

Keywords:
Automated data collection
Computer vision
Productivity analysis
Action recognition
Bag-of-Words
Bayesian network models

ABSTRACT

Automated action classification of construction workers and equipment from videos is a challenging problem that has a wide range of potential applications in construction. These applications include, but are not limited to, enabling rapid construction operation analysis and ergonomic studies. This research explores the potential of an emerging action analysis framework, Bag-of-Video-Feature-Words, in learning and classifying worker and heavy equipment actions in challenging construction environments. We developed a test bed that integrates the Bag-of-Video-Feature-Words model with Bayesian learning methods for evaluating the performance of this action analysis approach and tuning the model parameters. Video data sets were created for experimental evaluations. For each video data set, a number of action models were learned from training video segments and applied to testing video segments. Compared to previous studies on construction worker and equipment action classification, this new approach can achieve good performance in recognizing multiple action categories while robustly coping with the issues of partial occlusion, view point, and scale changes.

1. Introduction

Advanced sensing and information technologies are increasingly used on construction jobsites for collecting and analyzing a variety of project information that traditionally relied on manual methods [1–11]. Among these technologies, video becomes an easily captured and widely spread media, serving the purposes of construction method analyses, progress tracking, and worker ergonomic studies in the construction industry [11–13]. The associated demand for reducing the burden of manual analyses in retrieving information from video motivates further research in automated construction video understanding.

Recent studies have focused on leveraging computer vision algorithms to automate the manual information extraction process in analyzing recorded videos [4,11,14–16]. However, despite considerable progress in construction object tracking, classifying the action of construction workers or construction equipment in single view video, especially in beyond simple categories like working and not working, remains a hurdle for reaping the full benefits of video-based analysis in method studies and worker ergonomic studies. Robust action analysis algorithms that are capable of differentiating subtle action categories and handling scene clutter, occlusion, and view point changes are essential to overcome such a hurdle.

In this paper, we aim to explore the potential of an emerging visual learning approach in classifying subtle action categories in a variety of construction video segments. By action, we consider the combination of rigid and non-rigid motions. This visual learning approach is composed of four major steps including feature detection, feature representation, feature modeling, and model learning. More specifically, it utilizes 3D-Harris detector as the feature detector, local histograms as the feature representation, Bag-of-Words as the feature model, and Bayesian network models as the learning mechanism for action learning and classification. For simplicity purpose, we refer this approach as the Bag-of-Video-Feature-Words in the remaining part of this paper. We developed a test bed in MATLAB to evaluate the performance of this new approach in learning and classifying action categories in construction videos. At the same time, this study also aimed to tune a set of model parameters for the model to perform well in construction scenario. Two video data sets, including backhoe actions and worker actions in a formwork activity, are constructed from a large number of construction videos as the evaluation data sets. As the main contributions of this paper, we demonstrate that the Bag-of-Words model with local action feature representations and Bayesian learning methods have a great potential in significantly

* Corresponding author. Tel.: +1 618 650 2498.
E-mail addresses: jgong@siue.edu (J. Gong), caldas@mail.utexas.edu (C.H. Caldas), cgordon@siue.edu (C. Gordon).
[1] Tel.: +1 512 471 6014; fax: +1 512 471 3191.
[2] Tel.: +1 618 650 2867.

1474-0346/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.aei.2011.06.002

advancing automated construction video understanding as it performs well in learning subtle action categories in challenging construction videos. We also characterized the impact of model parameters on the model performance; therefore, a set of good choices of model parameter values are identified.

The rest of the paper is organized as follows. Section 2 briefly reviews the relevant literature in computer vision-based construction video analysis and the background of action analysis. Section 3 explains the Bag-of-Video-Feature-Words model. Section 4 evaluates the performance of the Bag-of-Video-Feature-Words model on two video data sets. Section 5 concludes the paper.

2. Research background

Recently, extensive research studies have been devoted to developing automated data collection methods for material management [2,17], productivity monitoring [11,15,16], project status updating [8], and quality control [1]. Many of those studies have been inspired and driven by the emergence and rapid development of advanced sensing technologies such as real-time localization and/or identification technologies, and 3D imaging systems. Typical examples of real-time localization and/or identification technologies include Global Positioning System (GPS), Radio Frequency Identification (RFID), and Ultra Wide Band (UWD). Terrestrial laser scanners, Flash LADAR, and stereo vision cameras are examples of 3D imaging systems that have attracted increased attentions from the construction industry. With the rapid development of technologies, it is generally agreed that the ability of processing vast volume of data collected by new technologies is a major obstacle to gain the full benefit of these technologies [18].

2.1. Computer vision for construction activity analysis

Computer vision algorithms can be widely used in construction to improve a variety of manual processes if the problem of reliable recognition and tracking of objects on construction jobsites can be solved. In this regard, many recent studies have focused on evaluating the performance of existing vision recognition and tracking algorithms in construction environments [4,14,15]. In lieu of automated productivity measurement using videotaping, there are so far three main approaches. They include detecting the movement of construction resources [19], recognizing and tracking the trajectories of construction resources [11], and recognizing worker gestures [15].

2.2. The general approaches used in computer vision-based human action classification

Research in human action analysis quickly evolves in the computer vision domain as described in a series of comprehensive reviews [20–24]. While the automatic capture and analysis of human action has been a highly active research area for decades, there is still no silver bullet type of algorithm that can be directly applied in different applications. It is widely recognized that inferring the pose and action of humans from images or videos is a hard and often ill-posed problem.

If action analysis is the only concern, there are three types of high-level methodologies that can potentially be used (Fig. 1). In Methods I & II, it is intuitive to start with the detection of humans in the images, or more precisely, segmenting humans from the background scenes. Then, specific types of action features of detected humans will be computed. Lastly, these action features will be used to classify the actions of humans, either actions at a single moment as depicted in an image or actions in a period of time as shown in a sequence of images, into different categories. The

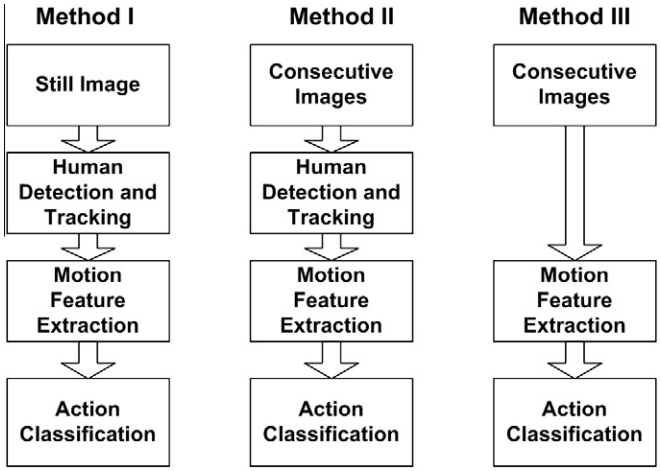

Fig. 1. Three general approaches to analyze human actions.

difference between Method I and Method II is that Method I uses a single image to infer the actions, while Method II use a sequence of images. Analyzing actions from a single image is highly dependent on the accuracy of human detection since it relies on human pose analysis. Method III reflects a recent approach for analyzing human actions in images. In principle, it uses local action features to infer global action characteristics. Computing local action features in video images often doesn't require segmentation. Several studies have argued that segmentation itself is a difficult task that often fails the rest of processing steps, and directly analyzing the actions in video images at the global level can be a viable alternative [25–26].

2.3. Local feature-based human action recognition

Local features in images have recently been extensively studied in computer vision because these features allow finding correspondences between images in spite of large changes in viewing conditions, occlusions, and image clutter as well as yield interesting descriptions of image contents [30,39]. Local features in images are often generated in two steps including detecting interest points in images and computing descriptors to describe the support region surrounding each detected interest point.

There are two main categories of interest point detectors: corner detectors and blob detectors. Commonly used corner detectors are Harris detector, SUSAN detector, and Harris-Laplace/Affine; Hessian detector, Hessian-Laplace/Affine, and Salient regions are examples of blob detectors. A comprehensive review of local features can be found in [30]. When interest points are detected in images, they typically represent significant changes in gradients along two directions (x-row and y-column) in an image. Thus, they represent changes in a spatial domain. Depending on the type of detectors used, these interest points might be invariant to scales or view point changes or both. Intuitively, these features can be extended into temporal domain by incorporating significant changes between consecutive video frames. In the context of human action analysis, a popular approach to facilitate the computation of such features in consecutive video frames is to treat human action in video sequences as silhouettes of a moving torso and protruding limbs undergoing articulated action and that such silhouettes can be described using three-dimensional shapes or volumes [25,31].

After the interest points are detected, local descriptors are used to describe the support regions surrounding interest points. Commonly used local descriptors include shape context [41], SIFT [39], steerable filters [42], Histogram of Gradient (HoG) [32], and

Histogram of Optical Flow (HoF). Among these filters, HoG and HoF have been used to represent the supporting volumes surrounding detected interest points in video sequences [26,31]. Detailed background information on calculating HoG and HoF can be found in [32] and [26,33], and is beyond the scope of this paper.

In general, a large quantity of local features can be found in images, which makes it possible to leverage highly successful classification methods used in speech recognition and text analysis. These methods often require a large amount of training data. Therefore, in essence, the use of local features for object category discovery and action classification is a reminiscent of text analysis [38]. In text analysis, the corpus of documents is summarized into a co-occurrence table, and the table records the number of occurrences of individual words in the given documents. Then, the problem of topic analysis is often formalized as a problem of finding a specific distribution of words pertaining to a topic. Such an approach is often referred as Bag-of-Words. Recently, the Bag-of-Words model has been increasingly used in image and video mining applications [25,28,29]. In these studies, local image features are quantized into clusters, and the center of each cluster becomes a visual "word". Local feature-based human action classification is a further extension of object category discovery with local image features [37]. It computes spatial–temporal local scale-invariant and viewpoint invariant features in consecutive video frames, and uses the local features as video feature words. The models used for learning topics in text vary from study to study, they can be generally categorized into generative models (Bayesian models) [38] and kernel-based models such as support vector machine [29].

2.4. Challenges of action classification in construction

This section intends to demonstrate the challenges of action classification in construction through a series of examples, and identify the possible limitations in existing studies.

The consecutive frames of two video segments, each segment recording one type of action in a construction activity, are shown in Fig. 2. The top row shows a nailing action, and the bottom row shows an action of aligning formwork. There are a number of observations that can be made from Fig. 2: (1) a action can be decomposed into a series of consecutive gestures; (2) self-occlusion of body parts due to camera view angle is prominent; and (3) different action categories can have similar gestures.

There are six video segments shown in Fig. 3. Each row of segments shows snapshots of a same action from different workers and view angles (top: nailing; bottom: aligning formwork). It can be noticed that the gestures of a same action vary significantly from each other in different view angles. Besides these challenges, it can also be noticed that low resolution and background clutter or moving objects in the background (the bottom row in Fig. 2.) are other significant challenges in analyzing actions from construction video. Considering most of construction cameras for streaming live video feeds from jobsites are cameras with a 720×480 resolution, the problem of low resolution videos can be severe. Furthermore, changing lighting conditions during a day, shaking of construction cameras caused by wind, and blur of images caused by rain, snow, and fog represent additional challenges for equipment and worker action recognition.

Analyzing equipment and worker actions using computer vision methods in challenging construction environments is subject of ongoing efforts. Of particular note is that several studies have focused on the rigid motion of equipment and workers, which only concerns the spatial positions of objects, and ignores the articulated (non-rigid) motions of objects [4,11,19]. Peddi et al. [15] proposed to use wireless cameras to develop a real-time productivity measurement system based on human poses for bridge replacement. In this study, background subtraction was used to extract human pose at each frame, and a neural network was used to train models for classifying worker performance into three classes including effective work, ineffective work, and contributory work. A potential issue with this approach is that a single worker gesture can belong to multiple categories. Gonsalves and Teizer [27] studied human actions such as walking and running using 3D ranging cameras. However, the performance of the algorithms is not reported. More importantly, both studies have relied on background segmentation and gesture recognition. It became clear from Figs. 2 and 3 that similar human gestures can belong to different action categories and one action category can have a variety of gestures. Brilakis et al. [43] developed a 3D project entity tracking method by combining 2D coordinates computed from multiple cameras. As an extension of the work reported in Gong and Caldas [11], this research focuses on a general framework of classifying actions of construction objects into intrinsic categories pertaining to the activity in which the objects are engaged. We are interested in a method that can classify actions into a level of detail that is comparable to crew balance analysis or manual ergonomic studies.

Local features provide promising approaches to address the challenges in construction worker action recognition. This is because the quantity of local features, often in the order of 10^5 for a given action category, and the quality of local features, being invariant to changes in scales and view point, render the local feature-based action recognition robust to occlusion and change of lighting conditions. However, despite the rising interest on local feature-based human action recognition in the field of computer vision, the authors could not find any instances where local

Fig. 2. Consecutive actions in two activities.

Fig. 3. Gestures in two action classes.

features have used in construction to analyze the actions of workers and equipment. Also, to date, local feature-based human action classification has been tested on a number of public human action data sets, including the KTH human action dataset, the Weizmann human action dataset, a dataset of figure skating actions, and video segments from several movies [25,26]. These video data sets recorded individuals performing certain actions, and they have limited background clutter. Therefore, they may not closely represent the challenges that can be faced in construction jobsites. This study concerns whether the local feature-based action classification method can differentiate actions of construction equipment and workers to a level of detail that can support rapid operation analysis.

3. Development of a test bed for Bag-of-Video-Feature-Words model

This section describes the development of a test bed that combines local feature representations, clustering methods, the Bag-of-Words model, and Bayesian learning methods into a powerful approach to learn action models and classify action categories in video sequences. Fig. 4 shows the overall workflow in the test bed. The learning stage of model development involves feature detection and representation, vector quantization for generating a Codebook, and learning action models. In the recognition stage, the goal is to apply the learned action model to classify new action videos. In the rest of this section, each of these steps is described.

3.1. Representing action as video feature words

We use the operator proposed by Laptev [31] to model image sequences. To make the paper self contained, a brief description of the operator is provided as follows. An image sequence is modeled by a linear scale-space representation according to the following function:

$$L(\cdot;\sigma_l^2,\tau_l^2) = g(\cdot;\sigma_l^2,\tau_l^2) \times f(\cdot) \tag{1}$$

where σ_l^2 denotes independent spatial variance, τ_l^2 denotes independent temporal variance, and the spatio-temporal separable Gaussian kernel is defined as

$$g(x,y,t;\sigma_l^2,\tau_l^2) = \frac{1}{\sqrt{(2\pi)^3\sigma_l^4\tau_l^2}} \times \exp\left(-\frac{x^2+y^2}{2\sigma_l^2} - \frac{t^2}{2\tau_l^2}\right) \tag{2}$$

Then, a spatio-temporal second-moment matrix, which is a three-by-three matrix composed of first order spatial and temporal derivatives averaged using a Gaussian weighting function $g(\cdot;\sigma_i^2,\tau_i^2)$ is used to calculate the spatial–temporal changes in this linear scale-space representation

$$\mu = g(\cdot;\sigma_i^2,\tau_i^2) \times \begin{bmatrix} L_x^2 & L_xL_y & L_xL_t \\ L_xL_y & L_y^2 & L_xL_t \\ L_xL_t & L_yL_t & L_t^2 \end{bmatrix} \tag{3}$$

where the integration scales σ_i^2 and τ_i^2 are related to the local scales according to $\sigma_i^2 = s\sigma_l^2$ and $\tau_i^2 = s\tau_l^2$. Also the first derivatives are defined as

$$L_x(\cdot;\sigma_l^2,\tau_l^2) = \partial_x(g \times f),$$
$$L_y(\cdot;\sigma_l^2,\tau_l^2) = \partial_y(g \times f),$$
$$L_t(\cdot;\sigma_l^2,\tau_l^2) = \partial_t(g \times f),$$

This matrix describes the spatio-temporal gradient distribution in a local neighborhood of a point. Now to detect the significant changes (i.e. interest points), 3D Harris corner detector, an extended version of Harris corner detector, is used. The Harris corner detector is a way of searching for regions in f with significant eigenvalues λ_1, λ_2, λ_3 of μ [40]. For the 3D Harris corner detector, this problem is formalized into finding eigenvalues λ_1, λ_2, λ_3 of μ that produce positive local maxima in the following corner function:

$$H = \det(\mu) - k\,\mathrm{trace}^3(\mu) \tag{4}$$

The eigenvalues λ_1, λ_2, λ_3 of μ constitute descriptors of variations in L along the three image directions (x, y, t). Three significantly large values of λ_1, λ_2, λ_3 of μ indicate the presence of an interest point.

After the interest points are detected, the HoG and HoF descriptors are used to describe the support regions surrounding the interest points. In the developed test bed, the HoG and HoF descriptors are computed using a similar method as described by Laptev [31]. The method involves partitioning each supporting region into

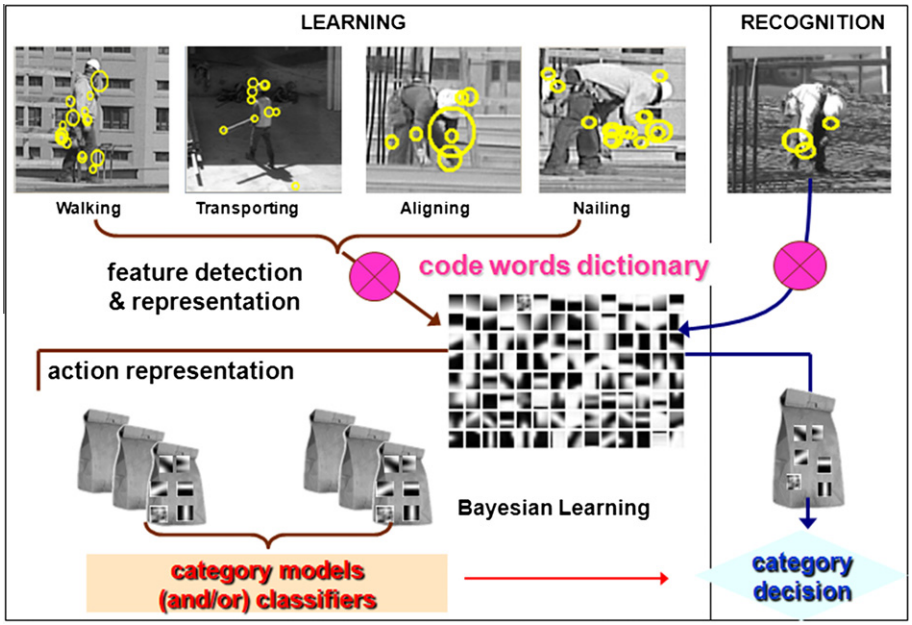

Fig. 4. An overview of the bag of video feature words method.

$3 \times 3 \times 2$ spatio-temporal blocks. The HoG descriptor uses a four-bin histogram to describe the distribution of orientation of gradients in a $3 \times 3 \times 2$ spatio-temporal block surrounding an interest point. The HoF descriptor uses a five-bin histogram to describe the orientation of the gradient calculated from optical flow. Both descriptors have shown good results on several video data sets [26,31], therefore, they are tested in this study on construction videos. The resulting HoG and HoF descriptors are 72-element and 90-element vectors, respectively. The top part of Fig. 4 shows the features (yellow circles) computed on the video frames in typical construction video sequences.

By using this method, a large set of interest points (typically in the order of 10^5) and their associated descriptors can be computed to represent the visual contents of the videos. These features and descriptors are analogous to the words in a document. The number of features produced in each video sequence depends on many factors, such as the resolution and action type. This leads to another important step, codebook formation, before these features can be effectively used for action classification.

3.2. Vector quantization and codebook formation

Considering that there are tens of thousands of features that can be computed for each category of actions, it becomes fairly difficult to discover a set of common features to one particular type of action. In practice, these large quantities of features are usually clustered into several hundred to thousands of clusters using the *K*-Means algorithm. Then the center of each of these clusters will be used to represent a set of features that belong to this cluster. In this way, a compact representation of the features can be formed. This process is often referred to as vector quantization. At this point, these centers of clusters represent the video feature words in a code book, and particular combinations of them are used to represent different categories of actions. An intuitive way to understand this process is through an analogy: the code book can be viewed as a dictionary, and the centers of the clusters are the entries in the dictionary. Video sequences for different actions have these entry words showing at different frequencies. Each of the video sequences can be represented as a bag of video feature words. A particular distribution of entry words for each action category can be learned from a training data set.

3.3. Learning action models using Bayesian approaches

In this section, we describe two Bayesian network models, namely naïve Bayesian model (also called naïve Bayesian classifier) and probabilistic Latent Semantic Analysis (pLSA), and how they are formulated in this study.

3.3.1. Naïve Bayesian model

Naïve Bayesian model is one of the simplest yet powerful Bayesian network models. It rests on strong assumptions that instances fall into one of a number of mutually exclusive and exhaustive classes and the features are conditionally independent given the instance's class [34,35]. In this research, the naïve Bayesian model is formulated as:

$$c = \arg\max p(c|w) \propto p(c)p(w|c) = p(c)\prod_{n=1}^{N} p(w_n|c) \quad (5)$$

where the w represents a set of video feature words, and the c represents action class decisions. The $p(c)$ is the prior information, the $p(w|c)$ represents vide feature word likelihood given an action class, and the $p(c|w)$ is the posterior probability of action classes given a set of video feature words. The video feature word likelihood is computed from labeled training data. More specifically, suppose there are N training video sequences containing video feature words from a vocabulary size M ($i = 1,\ldots M$), then the corpus of videos can be summarized in an M by N co-occurrence table, where each element of the table stores the number of occurrences of a particular word in a particular video sequence. Since the action type of these training video sequences is known, the probabilities of a particular video feature word generated by different action classes can be computed. The naïve Bayesian model further assumes that the probability of observing the conjunction $w_1, w_2 \ldots w_N$ given the action category is just the product of the probabilities of the individual words. Therefore, given a set of video feature words detected on a new video sequence, the $p(c|w)$ can be determined by computing the associated probability of generating these video feature words

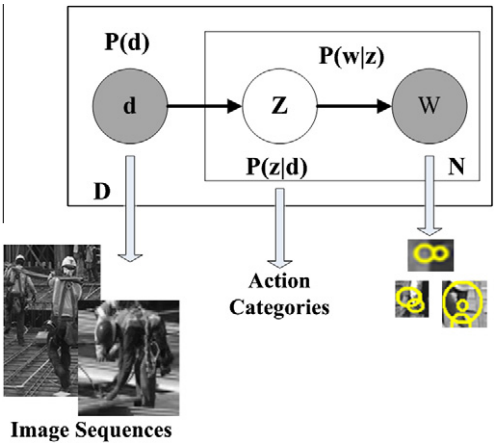

Fig. 5. The pLSA representation.

for each action category. The result can be used to determine which action category is most probable. Of particular note is that the learning process with naïve Bayesian model is a supervised approach, meaning the training samples are labeled samples.

3.3.2. Probabilistic latent semantic analysis

The pLSA model is one type of hierarchical Bayesian Text model that is often used in document classification [36]. In this study, the pLSA graphical model is formulated as shown in Fig. 5. Compared with the naïve Bayesian model-based approach, the pLSA assumes each instance (a video sequence in this case) is a mixture of several action classes rather than dictates each instance belongs to one action category. In this regard, the pLSA-based learning process is an unsupervised approach. In Fig. 5, W represents a set of video words, d represents a video sequence, and Z represents a latent topic associated with each occurrence of a video feature word in video d. Each topic corresponds to an action category. The shaded node in the figure can be observed, while the ones that are not shaded cannot. Similarly as in the naïve Bayesian classifier, the N video sequences are summarized in an M by N co-occurrence table. The essential mathematic form of the graphic model in Fig. 5 is:

$$p(d_j, w_i) = P(d_j)P(w_i|d_j) \tag{6}$$

This form can be further expanded into the following formula based on the assumption of the observation pair (d_j, w_i) being generated independently.

$$P(w_i|d_j) = \sum_{k=1}^{K} P(z_k|d_j)P(w_i|z_k) \tag{7}$$

where $P(z_k|d_j)$ is the probability of topic z_k occurring in video d_j, and $P(w_i|z_k)$ is the probability of video word w_i occurring in a particular action category z_k. K is the total number of latent topics, which correspond to the number of action categories. In essences, this method is to model each video as a mixture of action categories. To determine the model that gives a high probability to the video words in the corpus of the video, an Expectation Maximization algorithm is used to obtain a maximum likelihood estimation of the parameters.

With the action category specific video word distributions from a training dataset, the next goal is to classify a new video sequence. This new task can be achieved by projecting the unseen video on the simplex spanned by the learned $P(w|z)$ and finding the mixing coefficients $P(w|d_{test})$ such that the Kullback–Leibler divergence between the measured distribution and $P(w|d_{test})$ is minimized [28,36]. The Kullback–Leibler divergence is a non-symmetric measure of the difference between two probability distributions. Therefore, the maximum posteriors $P(z_k|w_i d_j)$ can be computed to determine which topic has the largest posterior probability.

4. Evaluation of Bag-of-Video-Feature-Words model

In this section, we describe and discuss the results of testing the Bag-of-Video-Feature-Words model on two construction video data sets with the developed test bed. The purpose of these tests is multi-folded. Broadly speaking, the study has two interrelated goals: to determine the applicability and performance of the model for construction applications. In terms of applicability, the goal is to determine whether the model is capable of: (1) classifying the actions of heavy equipment though the model is originally designed for human action classification; and (2) classifying worker actions into action categories that may be used in construction activity analysis. To quantify the performance of the model in various application scenarios as well as with different model parameters, confusion matrix and time used for learning and classifying actions are metrics used in this research. With that, this paper also aims to characterize the impact of model parameters on model performance; therefore better choices of model parameters can be made.

4.1. Video data sets

To test the Bag-Of-Video-Feature-Word model, we created two challenging video data sets from a large amount of construction videos that were recorded using either construction webcams or hand-held camcorders on several construction jobsites in different weather conditions. The video data sets consist of hundreds of labeled video segments, the label showing the type of action recorded in each video segment. The summary information of both video data sets is shown in Table 1. The first video data set includes three backhoe action categories, and the second data set includes five worker action categories in a typical formwork activity. For the backhoe action data set, we divide a backhoe's active motion into three action categories, including relocating, excavating, and swinging. The relocating action can be considered as traveling; the excavating action represents effective work; and the swinging action connects the excavating action with the action of dumping material, giving indications of operation cycles. The action of dumping material has a very short time frame. Therefore, it is not reliable to be analyzed, but it can be induced from the action of swinging. For the formwork worker action data set, we divided a worker's action into five categories, including travelling, transporting, bending down, aligning, and nailing. These categories represent the ordinary actions a worker displays in a formwork activity. In addition, these categories have practical implications on his/her working status. Time utilization studies such as work sampling often consider travelling as wasteful activity, transporting as supportive activity, and the actions such as bending down, aligning,

Table 1
Summary information of video data sets.

	Video data set I: Backhoe actions	Video data set II: Worker actions in formwork activities
Action categories	Swing, excavating, relocating	Traveling, transporting, bending down, nailing with hammer, aligning formwork
# of Video segments / category	50	60
Frame rate	7 frames/s	30 frames/s
Length of video segments	10 s	5 s
Total #of video segments	150	300

and nailing as direct work. Fig. 6 shows snapshots of some of the video segments. The challenge of view point, scale, and illumination changes, occlusion, and low resolution is evident in both video data sets.

For each of the video segments, we ran the 3D Harris corner detector on video volumes to detect the interest points. The video volumes were generated according to the method described in Section 3.1. Then, the HoG and HoF descriptors were calculated on the supporting region surrounding each interest point. As a result, there are 62,622 local features (either HoF or HoG) for the backhoe action data set, 146,800 local features for the worker action data set. The *K*-Means method was used to group these features into 400–3000 clusters, and the centers of these clusters became visual words in the codebook.

4.2. Testing configurations

During the process of evaluation, we adjusted mainly three model parameters. They are the type of model learning methods, the type of feature descriptor, and the number of code words (each code word representing the center of certain numbers of features) used for model training. Each combination of these parameters forms one testing configuration. For each test configuration (i.e. HoG descriptor, 250 code words, naïve Bayesian model), we conducted two five-folder cross validations. In each five-folder cross validation, the data set was randomly divided into five subsets. Of the five subsets, a subset is retained as the validation data for testing the model, and the remaining four subsets are used as training data. The cross-validation process is then repeated five times, with each of the five subsets used exactly once as the validation data. The ten results (five from each five-folder cross validation) was averaged to show the performance of a particular testing configuration.

For the backhoe action data set, the focus is on whether the model can distinguish different equipment actions using either naïve Bayesian model or pLSA. For the worker action data set, the focus is placed on evaluating the ability of the model in classifying worker actions into classes at different level-of-details and the impact of two model parameters, the type of descriptor and the number of code words, on the model performance. The performance of the models in terms of running time is also reported on both data sets.

4.3. Experimental evaluation results on the backhoe action data set

In this test, we set the number of code words to be 500 and the HoF to be action feature descriptor. 500 code words is a typical number used in similar studies [25]. Tables 2 and 3 shows two confusion matrices that summarize the average classification performances on training data and testing data in ten runs of evaluations using 500 code words and naïve Bayesian classifier. Three performance indices, including overall accuracy (OAC), false positive rate (FPR), and false negative rate (FNR), are extracted from a confusion matrix for classifier performance evaluation. These indices for multi-class classification are defined as follows.

Let $CM(i,j)$, $i,j = 1, \ldots C$ be the confusion matrix (i – row number, j – column number) where C is the number of classes, and assume relocating action represents normal condition.

$$\text{Overall accuracy}: \text{OAC} = \sum_{i=1}^{C} CM(i,j) \Big/ \sum_{i,j=1}^{C} CM(i,j) \tag{8}$$

$$\text{False positive rate}: \text{FPR} = \sum_{j=2}^{C} CM(1,j) \Big/ \sum_{j=1}^{C} CM(1,j) \tag{9}$$

$$\text{False negative rate}: \text{FNR} = \sum_{i=2}^{C} CM(i,1) \Big/ \sum_{i=2\,j=1}^{C} CM(i,j) \tag{10}$$

It can be noted that the overall accuracy, false positive rate, and false negative rate of classification on the testing data are 79%, 21%, and 19.3%, respectively. The standard deviation of the overall accu-

Table 2
Classification results on backhoe action training data set.

		Predicted class			Performance indices
		Relocating (%)	Excavating (%)	Swing (%)	
Actual Class	Relocating	92	5	3	OAC = 86.33%
	Excavating	4	81	15	FPR = 13.7%
	Swing	1	13	86	FNR = 13.6%

Fig. 6. Example snapshots of video segments.

Table 3
Classification results on backhoe action testing data set.

		Predicted class			Performance indices
		Relocating (%)	Excavating (%)	Swing (%)	
Actual class	Relocating	80	16	4	OAC = 79%
	Excavating	1	78	21	FPR = 21%
	Swing	1	20	79	FNR = 19.3%

racy in the ten trials for this testing configuration is 8.1%. A random guess in this case would yield 33.3% of accuracy given there are equal numbers of cases in each category. It became clear that the Bag-of-Video-Feature-Words model with the naïve Bayesian classifier performs reasonably well in distinguishing these actions.

The performance of the model with the pLSA was also tested on this data set with the same settings on the number of code words and the type of descriptor. The primary output of the pLSA is the plot of the learned $P(w|z)$ and $P(z|d)$. $P(w|z)$ describes the distribution of code words pertaining to each action category. In other words, it shows the probability for one type of action to generate each particular video word. For example, The left part of Fig. 7 shows the probabilities for the actions of relocating, swing, and excavating to generate each of the five hundred code words. The shaded bar on the right shows the scale of probability value. The brighter the bar, the higher is the probability. $P(z|d)$ shows the probabilities of a video sequence belonging to different action categories. For example, The right part of Fig. 7 plots the probabilities of each video sequence belonging to relocating, swing, and excavating. For each video sequence, the action category with the highest probability value is selected as the classification result. Since 80% of the data set was used for training in this test, the ground truth is that the first 40 of 120 training video sequences should have highest probability values in relocating, the next 40 in excavating, and the last 40 in swing. However, it can be observed in Fig. 7 that the learned model appears to misclassify many video sequences in each category. It should be noted that the label of action category for each training video is not used in the training process. The pLSA is an unsupervised learning method, seeking to define a joint distribution of video feature words, video sequences, and action classes in the maximum likelihood framework. The overall accuracy of classification is around 60% on the training data, and even worse for the testing data. It can be concluded that naïve Bayesian classifier significantly outperforms the pLSA on this data set. The performance of pLSA is further discussed in Section 4.4.4.

4.4. Experimental evaluation results on the worker action data set

The five action categories in the worker action data set, including transporting, traveling, bending down, nailing with a hammer, and aligning formwork, are typical examples of worker actions in a formwork activity. If such action categories can be accurately recognized by a computer in a video stream, automated detailed activity analysis, such as 5-min rating and crew balance study, can possibly be achieved. Also, automated recognition of stressful actions such as bending among a series of worker actions has great potentials for automated ergonomic studies. But before these applications can be realized, the purpose of this section is to assess the performance of the Bag-of-Video-Feature-Words model in classifying worker actions in a typical activity, in this case, a formwork activity. A similar training and testing process as described in Section 4.3 was conducted on the worker action data set. Each of the following sub-sections investigates an issue that is related to our global goal as stated at the beginning of Section 4.

4.4.1. Numbers of code words vs. model performance

To investigate the impact of the number of code words on the model performance, we varied the number of code words from 250 to 1500 with an increment of 250. The performances of the model under different testing configurations are shown in Fig. 8. The plots on the top and bottom rows show the performance on training data and testing data, respectively. It can be observed that the classification accuracy of the model on the training data improves as the number of code words used increases; while such accuracy on the testing data oscillates as the number of code words used increases. According to Fig. 8, the performances on the testing data for the HoF and HoG descriptor peak around 1500 code words. When the number of code words exceeds 1500, the performance on the testing data decreases. Meanwhile, the performance on the training data still showed slight improvement or remained flat, indicating the potential problem of overfitting on the training data. Overall, the testing configuration of 1500 code words, HoG descriptor, and naïve Bayesian model produces the best classification results. The confusion matrices for this case are shown in Tables 4 and 5. The overall accuracy, false positive rate, and false negative

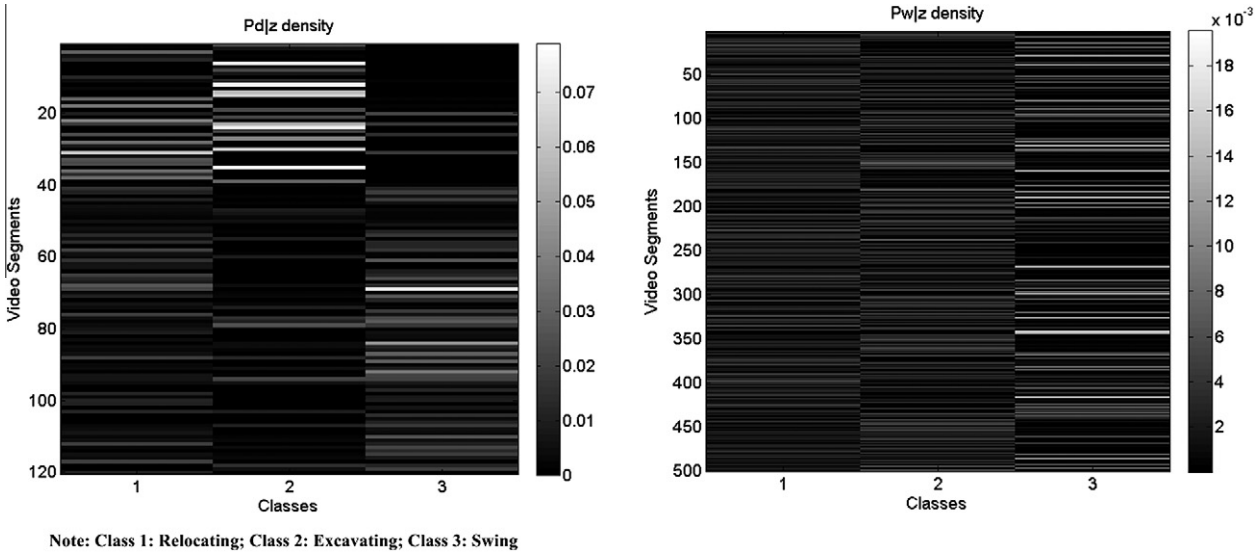

Fig. 7. The plot of Pd-Z and Pw-Z (equipment action data set).

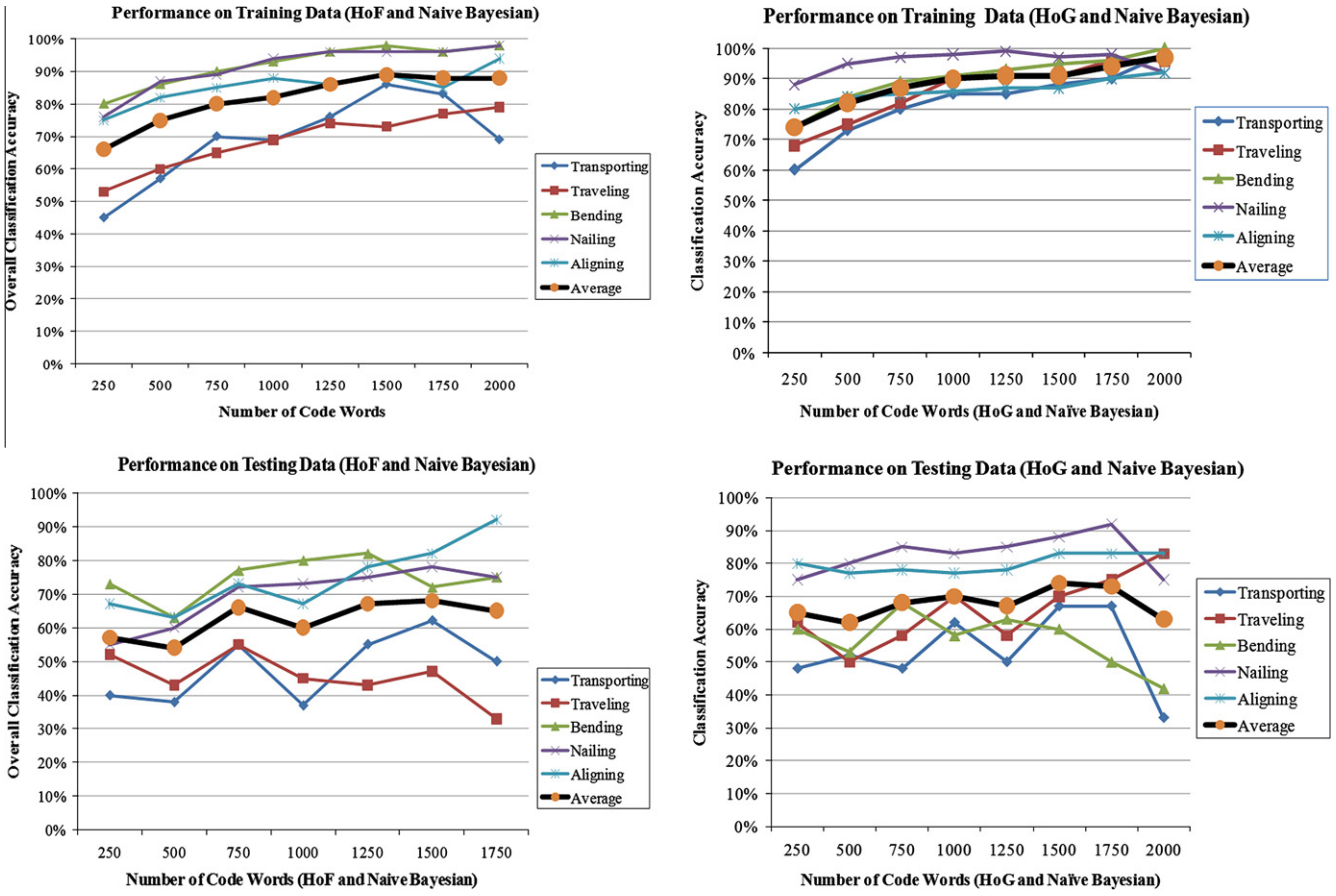

Fig. 8. Classification accuracy vs. number of code words.

rate of the classifier on the testing data set are 73.6%, 26.4%, and 25.3%, respectively. The standard deviation of the overall accuracy is 7.5% in the ten trials conducted under this testing configuration. Overall, this is a much more difficult data set than the crane action data set. Because there are more categories in this data set, the expected accuracy of random guess drops to 20%. It is clear that the learned model can do significantly better than the random guess. Notably, the learned model performs better in terms of classifying bending, nailing, and aligning actions. The most difficult action categories to classify are transporting and traveling. This is reasonable to expect since these two categories themselves have much in common in terms of action features.

4.4.2. Types of descriptors vs. model performance

It is not difficult to conclude from Fig. 8 that overall the HoG descriptor yields better classification accuracy than the HoF descriptor does. Furthermore, each HoG descriptor is a 72-element vector; while each HoF descriptor is 90-element vector. Therefore,

Table 4
Classification results on worker action training data set.

		Predicted class					Performance indices
		Transporting (%)	Traveling (%)	Bending (%)	Nailing (%)	Aligning (%)	
Actual Class	Transporting	88	6	3	3	0	OAC = 92.6%
	Travelling	6	96	1	2	0	FPR = 7.4%
	Bending	2	2	95	1	1	FNR = 8.05%
	Nailing	0	2	0	97	0	
	Aligning	5	3	1	4	87	

Table 5
Classification results on worker action testing data set.

		Predicted Class					Performance indices
		Transporting (%)	Traveling (%)	Bending (%)	Nailing (%)	Aligning (%)	
Actual Class	Transporting	67	18	7	7	2	OAC = 73.6%
	Travelling	22	70	3	2	3	FPR = 26.4%
	Bending	17	12	60	12	0	FNR = 25.3%
	Nailing	8	0	3	88	0	
	Aligning	5	0	5	7	83	

the HoF descriptor incurs significant higher computational cost than the HoG descriptor does. That means with all the other model parameters being same, the model based on the HoG descriptor outperforms the model based on the HoF descriptor in terms of speed.

4.4.3. Time used for learning and classifying action categories

The time used in the process of learning and classifying actions is consumed in the steps of computing feature representations, vector quantization, learning action models, and classifying the actions. It should be noted that the process of learning action models is an offline process. The time it consumes is not critical since it can be done beforehand. As described earlier, the learning process consists of computing feature representations, vector quantization, and learning action models. In the offline learning phase, it was found that most of the time was consumed in vector quantization. The time of vector quantization on a video sequence depends largely on the size of the code book (i.e. number of code words). The time for vector quantization in this study was benchmarked on a laptop with 2G RAM and a 1.6 GHz quad-core processor. Fig. 9 shows the relationship between the number of code words and the mean time used in vector (HoG) quantization for a given video sequence in the worker action data set. The online action classification phase involves only two steps, including computing feature representations and classifying the actions using a learned action model. The time used in online action classification phase is directly connected to how fast actions in a new video sequence can be classified. For the video sequences in the worker action data set, the calculation of HoF and HoG features runs at 6–10 video frames per second. Each of the video sequences in the worker action data set has 150 frames and runs at 30 frames per second. Therefore, if the frame rate of these video sequences can be decimated to 6–10 frames per second, the computing of the local feature descriptors can be executed at the same rate as a video is streamed. After the local features are computed, the average time used for classifying a new sequence in the construction worker action data set is around 0.3 s. A cautious note is that more time is needed in computing local feature descriptors when the resolution of video increases. This issue will be further discussed in the Section 4.5.

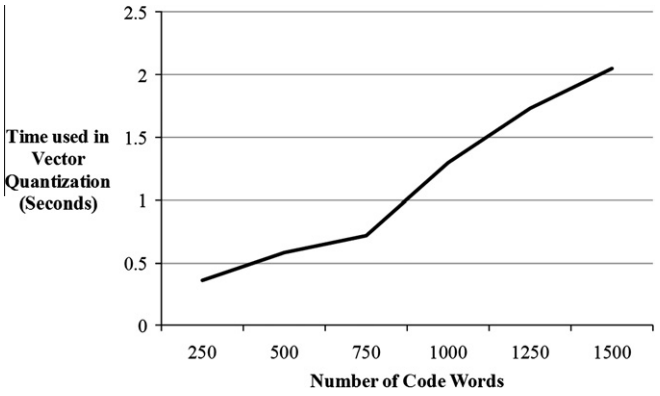

Fig. 9. The average time used in vector quantization.

4.4.4. Types of learning methods vs. model performance

We also tested the performance of the pLSA with the similar settings as above. Fig. 10 shows the distribution of code words given different action classes and the distribution of action classes in all the video sequences. Regardless of what type of descriptor being used, the learned model shows poor performance in terms of classification accuracy. The test results clearly favor use of naïve Bayesian model over pLSA. Nevertheless, studies in the computer vision domain have reported the performance of the pLSA is generally comparable or better than the naïve Bayesian classifier. The possible reason that contributed to the difference between what was discovered in our study and what was reported in the studies in computer vision is related to the size of data set. Naïve Bayesian classifier is a two-node Bayesian network in this study; while the pLSA is a three-node Bayesian network. As the number of nodes in a Bayesian network increases, an increasing amount of training data is needed to learn the joint probability distribution of the network. Therefore, naïve Bayesian classifier performs relatively better on small data set than more complex network models do, though the learned distribution from a naïve Bayesian model is not necessarily close to the true distribution [37].

4.5. Discussion

The accuracy of action classification obtained in this study is in the range of 73.6%-79%, which is comparable to what were achieved on several public human action data sets in the computer vision community [25]. Given the challenges embodied in the

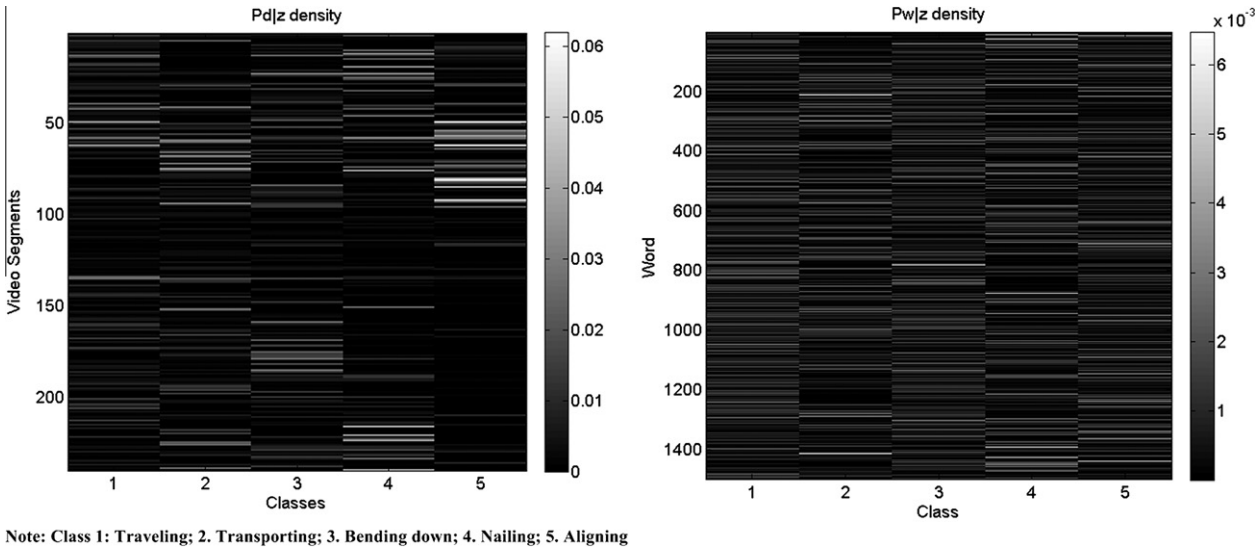

Fig. 10. The plot of Pd-z and Pw-z (worker action data set).

video sets used in this study, the Bag-of-Video-Feature-Words approach exhibited robustness to occlusions and changes in video resolution, view angles, and illumination. The researchers could not find studies in construction that studied classifying actions of workers and equipment to a level of detail that is similar to this study. Therefore, comparison of this study with others in construction cannot be drawn at the time of writing. We believe the Bag-of-Video-Feature-Words approach has a promising future in video-based operation analysis and ergonomics study, though future studies are needed to quantify the impact of misclassification.

There are several potential limitations meriting considerations in this study. First, the action analysis has been mostly constrained to a single person or equipment with some background motions. Group action analysis falls outside the scope of this study. However, it is reasonable to expect that given a video involving the actions of multiple workers, the action of an individual worker can be isolated from other workers. Then the problem of analyzing the actions of multiple workers or equipment can be decomposed into analyzing the action of individual worker or equipment. Furthermore, the process of isolating individual workers' motion can be greatly assisted by motion segmentation or moving object detection and tracking. Since a large number of local features can be computed on consecutive frames, the local feature-based action classification will be less sensitive to motion isolation errors. Therefore, we believe the approach used in this study can be extended to analyze the actions of multiple workers and equipment. Second, we did not use high resolution videos (>720 $\times$ 480) as the video data in this study. The resolution of videos used in this study is relatively low; all the video segments were trimmed and cropped from videos that have a resolution of 720 $\times$ 480 pixels (a most common resolution for construction web cameras). The approach used in this study will work with high resolution videos, but at the cost of increasing computation time. This is because the time for detecting local interest points and computing local feature descriptor will increase as the resolution of video increases. For applications where real-time processing is not needed, the increase of computing time associated with the increasing resolution is not critical. Also, there are various ways to reduce the video regions to be processed. For stationary cameras, an apparent way would be subtracting consecutive frames to isolate moving regions. For cameras in motion, appearance-based object detection (such as safety vest-based worker detection) can drastically reduce the regions to be processed for computing local features. In summary, the problem of recognizing the action of multiple workers or equipment and the increasing computational cost with high resolution videos merit future research studies.

5. Conclusion

In this study, we extended the Bag-of-Video-Feature-Words model into the construction domain. We implemented this new action learning and classification framework in MATLAB, and two construction video data sets were created for evaluating its performance. The following conclusions can be drawn from the experimental evaluations:

- The Bag-of-Video-Feature-Words model with naïve Bayesian classifier can be extended to classify the complex action of construction workers and equipment with good accuracy.
- The use of HoG descriptors as the video feature words yields better performance in classification accuracy and speed than the use of HoF descriptors.
- The accuracy of classification generally improves as more code words are used, but there is little gain in classification accuracy once the number of code words exceeds 1500.
- Naïve Bayesian classifier performs significantly better than the pLSA does on our construction video data sets. This may be caused by the size of training data.

The attractiveness of the bag of video feature words is that it does not require accurate foreground segmentation, and is robust to partial occlusion and changes in view point, illumination, and scale. Future studies in the following directions are needed. First, the performance of this method can be further improved by adding spatial information since it is well-known that the Bag-of-Words method ignores spatial information and only concerns the frequency of feature occurrence [38]. Second, there is also a need to introduce more action categories and more data into the existing video data set. Therefore, complex Bayesian network models or even Markov Random Field can be more thoroughly tested and evaluated. Last, but not the least, in addition to Bayesian learning methods, kernel-based methods, such as support vector machine, should also be tested on the video sets since these are two competing methods frequently used in similar studies in computer vision. Above all, since this is the first study on using Bag-of-Words method on construction video data set, we hope that this study can establish a baseline for further comparing the performance of other algorithms.

References

[1] B. Akinci, F. Boukamp, C. Gordon, D. Huber, C. Lyons, K. Park, A formalism for utilization of sensor systems and integrated project models for active construction quality control, Automation in Construction 15 (2) (2006) 124–138.
[2] J. Song, C.T. Haas, C.H. Caldas, A proximity-based method for locating RFID tagged objects, Advanced Engineering Informatics 21 (4) (2007) 367–376.
[3] R. Navon, Automated project performance control of construction projects, Automation in Construction 14 (2005) 467–476.
[4] J. Yang, O. Arif, P.A. Vela, J. Teizer, Z. Shi, Tracking multiple workers on construction sites using video cameras, Advanced Engineering Informatics 24 (4) (2007) 428–434.
[5] J. Gong, C.H. Caldas, Data processing for real-time construction site spatial modeling, Automation in Construction 17 (5) (2008) 526–535.
[6] I. Brilakis, L. Soibelman, Y. Shinagawa, Material-based construction site image retrieval, Journal of Computing in Civil Engineering 19 (4) (2005) 341–355.
[7] L. Soibelman, J. Wu, C. Caldas, I. Brilakis, K.Y. Lin, Management and analysis of unstructured construction data types, Advanced Engineering Informatics 22 (1) (2008) 15–27.
[8] F. Bosche, C.T. Haas, Automated retrieval of 3D CAD model objects in construction range images, Automation in Construction 17 (4) (2008) 499–512.
[9] E.J. Jaselskis, Z. Gao, R.C. Walters, Improving Transportation Projects Using Laser Scanning, Journal of Construction Engineering and Management 131 (3) (2005) 377–384.
[10] X. Luo, W.J. O'Brien, C.L. Julien, Comparative evaluation of Received Signal-Strength Index (RSSI) based indoor localization techniques for construction jobsites, Advanced Engineering Informatics 24 (2) (2011) 355–363.
[11] J. Gong, C.H. Caldas, Computer vision-based video interpretation model for automated productivity analysis of construction operations, ASCE Journal of Computing in Civil Engineering 24 (3) (2010) 252–263.
[12] J. Abeid, D. Arditi, Linking time-lapse digital photography and dynamic scheduling of construction operations, Journal Computing in Civil Engineering 16 (2002) 269–279.
[13] C.A. Quinones-Rozo, Y.M.A. Hashash, L.Y. Liu, Digital image reasoning for tracking excavation activities, Automation in Construction 17 (5) (2008) 608–622.
[14] G.M. Jog, I.K. Brilakis, D.C. Angelides, Testing in harsh conditions: Tracking resources on construction sites with machine vision, Automation in Construction 20 (4) (2011) 328–337.
[15] A. Peddi, L. Huan, Y. Bai, S. Kim, Development of human pose analyzing algorithms for the determination of construction productivity in real-time, in: Construction Research Congress 2009, vol. 1, ASCE, Seattle, WA, 2009, pp. 11–20.
[16] T.I.P. Weerasinghe, J.Y. Ruwanpura, Automated multiple objects tracking system (AMOTS), in: Construction Research Congress, vol. 1, ASCE, Banff, Canada, 2010, pp. 11–20.
[17] D. Grau, C.H. Caldas, C.T. Haas, P.M. Goodrum, J. Gong, Assessing the impact of materials tracking technologies on construction craft productivity, Automation in Construction 18 (7) (2009) 903–911.
[18] N. Lee, E.M. Rojas, Defining high-level project control data for visual information systems, in: Construction Research Congress 2010, vol. 1, ASCE, Banff, Canada, 2010, pp. 518–527.

[19] J. Zou, H. Kim, Using hue, saturation, and value color space for hydraulic excavator idle time analysis, Journal Computing in Civil Engineering 21 (2007) 238–246.
[20] C. Cedras, M. Shah, Action-based recognition a survey, Image and Vision Computing 13 (2) (1995) 129–155.
[21] D.M. Gavrila, The Visual Analysis of Human Movement: A Survey∗ 1, Computer Vision and Image Understanding 73 (1) (1999) 82–98.
[22] J.K. Aggarwal, Q. Cai, Human action analysis: A review, in: Workshop on Action of Non-Rigid and Articulated Objects, IEEE, Austin, Texas, 2002 (pp. 90–102).
[23] W. Hu, T. Tan, L. Wang, S. Maybank, A survey on visual surveillance of object action and behaviors, Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews 34 (3) (2004) 334–352.
[24] T.B. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human action capture and analysis, Computer Vision and Image Understanding 104 (2–3) (2006) 90–126.
[25] J.C. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial-temporal words, International Journal of Computer Vision 79 (3) (2008) 299–318.
[26] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, Computer Vision and Pattern Recognition CVPR, IEEE, 2008 (pp. 1–8).
[27] R. Gonsalves, J. Teizer, Human Action Analysis Using 3D Range Imaging Technology, 26th International Symposium on Automation and Robotics in Construction Austin, Texas, 2009.
[28] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, W.T. Freeman, Discovering objects and their location in images, in: International Conference on Computer Vision, vol. 1, IEEE, 2005, pp. 370–377.
[29] K. Grauman, T. Darrell, The pyramid match kernel: Discriminative classification with sets of image features, in: International Conference on Computer Vision, IEEE Computer Society, 2005.
[30] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: A survey, Foundations and Trends in Computer Graphics and Vision 3 (3) (2008) 177–280.
[31] I. Laptev, On space-time interest points, International Journal of Computer Vision 64 (2) (2005) 107–123.
[32] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, CVPR 2005, vol. 1, IEEE, pp. 886–893.
[33] B.K.P. Horn, B.G. Schunck, Determining optical flow, Artificial Intelligence 17 (1–3) (1981) 185–203.
[34] J. Pearl, Probabilistic reasoning in intelligent systems: networks of plausible inference, Morgan Kaufmann, 1988.
[35] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, The MIT Press, 2009.
[36] T. Hofmann, Probabilistic latent semantic indexing, Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval ACM, 1999, pp. 50–57.
[37] L. Fei-Fei, P. Perona, A Bayesian hierarchical model for learning natural scene categories, CVPR, IEEE 2 (2005) 524–531.
[38] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, CVPR2003, Published by the IEEE Computer Society, 2003.
[39] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.
[40] C. Harris, M. Stephens, A combined corner and edge detector, in: Alvey vision conference, vol. 15, Manchester, UK, 1988 (p. 50).
[41] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (4) (2002) 509–522.
[42] W. Freeman, E. Adelson, The design and use of steerable _lters, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (9) (1991) 891–906.
[43] I., Brilakis, M.W. Park, G. Jog, Automated vision tracking of project related entities, Journal of Advanced Engineering Informatics, Elsevier, in press.