Latent Support Vector Machine Modeling for Sign Language Recognition with Kinect

CHAO SUN, TIANZHU ZHANG, and CHANGSHENG XU, Institute of Automation, Chinese Academy of Sciences, Beijing, China

Vision-based sign language recognition has attracted more and more interest from researchers in the computer vision field. In this article, we propose a novel algorithm to model and recognize sign language performed in front of a Microsoft Kinect sensor. Under the assumption that some frames are expected to be both discriminative and representative in a sign language video, we first assign a binary latent variable to each frame in training videos for indicating its discriminative capability, then develop a latent support vector machine model to classify the signs, as well as localize the discriminative and representative frames in each video. In addition, we utilize the depth map together with the color image captured by the Kinect sensor to obtain a more effective and accurate feature to enhance the recognition accuracy. To evaluate our approach, we conducted experiments on both word-level sign language and sentence-level sign language. An American Sign Language dataset including approximately 2,000 word-level sign language phrases and 2,000 sentence-level sign language phrases was collected using the Kinect sensor, and each phrase contains color, depth, and skeleton information. Experiments on our dataset demonstrate the effectiveness of the proposed method for sign language recognition.

Categories and Subject Descriptors: I.5.4 [**Applications**]: Computer Vision

General Terms: Algorithms, Experimentation, Performance

Additional Key Words and Phrases: Sign language recognition, Kinect sensor, latent SVM

ACM Reference Format:
Chao Sun, Tianzhu Zhang, and Changsheng Xu. 2015. Latent support vector machine modeling for sign language recognition with Kinect. ACM Trans. Intell. Syst. Technol. 6, 2, Article 20 (March 2015), 20 pages.
DOI: http://dx.doi.org/10.1145/2629481

1. INTRODUCTION

Sign language is a kind of visual language that consists of a sequence of grammatically structured human gestures. It is one of the most natural means of exchanging information for deaf and hearing impaired persons. The goal of sign language recognition is to transcribe sign language into text efficiently and accurately. Currently, automatic sign language recognition is still in its infancy, roughly decades behind automatic speech

This work is supported in part by the National Basic Research Program of China (No. 2012CB316304), the National Natural Science Foundation of China (No. 61225009, 61303173), and the Microsoft Research Asia UR Project. This work is also supported by the Singapore National Research Foundation under its International Research Centre@Singapore Funding Initiative and administered by the IDM Programme Office.
Authors' address: C. Sun, T. Zhang, and C. Xu, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. No. 95, Zhongguancun East Road, Haidian District, Beijing, 100190, P.R. China; emails: {csun, tzzhang, csxu}@nlpr.ia.ac.cn.
Author's current address: T. Zhang is a visiting research scientist at Advanced Digital Sciences Center, 1 Fusionopolis Way, 138632, Singapore.

DOI: http://dx.doi.org/10.1145/2629481

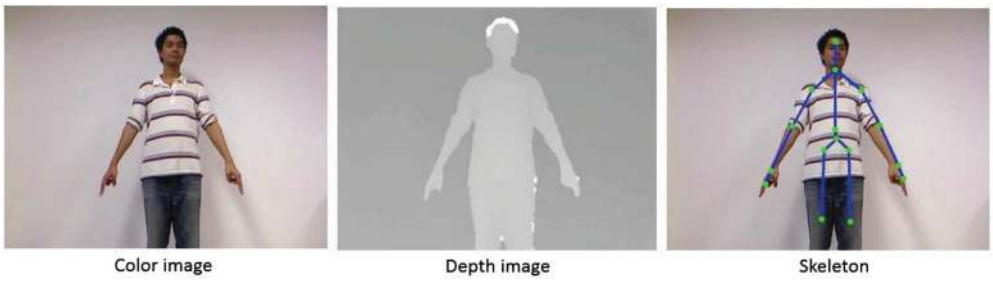

Fig. 1. Color image, depth map, and skeleton position provided by the Kinect sensor.

recognition [Von Agris et al. 2008]. It corresponds to a gradual transition from isolated to continuous recognition for a small vocabulary task. Most research has focused on identifying optimal features and classification methods to correctly recognize a given sign from a set of possible signs. Correspondingly, there exist two challenges in sign language recognition. One is how to efficiently capture the optimal features from signers, and the other is how to model different signs and classify them correctly for recognition.

In the aspect of feature collection, different kinds of sensors are explored, varying from tracking systems using data gloves [Kim et al. 1996; Liang and Ouhyoung 1998; Fels and Hinton 1993; Kadous 1996] to computer vision techniques using camera [Starner 1995; Starner et al. 1998; Vogler and Metaxas 2001] and motion capture systems [Hernandez-Rebollar et al. 2004]. Until now, commercially available depth camera systems have been expensive, and only a few researchers have used depth information to recognize hand pose. Fortunately, the release of the Microsoft Kinect sensor has provided a low-cost and off-the-shelf choice for depth sensors. The Kinect sensor involves an infrared (IR) light projector, a standard CMOS camera, a color camera, and a standard USB interface. The distortion of the IR pattern is used to calculate depth maps, which have a per-pixel depth resolution of 1cm while the camera is 2m away [Zhang 2012]. With these components, the Kinect could simultaneously provide color image, depth map, and skeleton positions developed based on color and depth information. Figure 1 illustrates the three kinds of output from the Kinect sensor at the same moment.

The extra depth map and skeleton information provided by the Kinect sensor could greatly benefit vision-based sign language recognition. First, background modeling becomes simple and robust with the depth map. We can easily and accurately extract human body parts from color images with the help of body depth information. Second, in previous 2D solutions, tracking hands is always a difficult task, and most of them need a signer to wear color gloves for help. However, with the skeleton information, hands can be tracked robustly and in real time. Third, beyond the traditional 2D features, the Kinect sensor can provide some novel 3D features, which are quite useful for improving the performance of sign language recognition. These advantages of the Kinect have been utilized into sign language recognition [Zafrulla et al. 2011]. In this article, we also utilize the Kinect sensor to perform vision-based sign language recognition.

In the aspect of sign modeling and classification, researchers have developed and applied many kinds of models. Murakami and Taguchi [1991] trained an artificial neural network (ANN) to recognize isolated signs, which was the early work on sign language recognition. Vogler and Metaxas [1997] utilized hidden Markov models (HMMs) to perform word-level sign language recognition, whose performance relies on the states selection in the HMM model. Many approaches have treated the problem as gesture recognition and have focused on statistical modeling of signs [Zhang et al. 2009, 2011,

Fig. 2. Illustrations of frames from several sign language videos. The left-most column lists the names of signs, followed by the frame sequences sampled from videos, respectively. Images with a red bounding box denote discriminative frames.

2012]. Recently, the latent support vector machine (SVM) formalism has been successfully applied to many tasks [Felzenszwalb et al. 2010; Lan et al. 2011; Yao and Fei-Fei 2010]. An advantage of latent SVM and its variants is that they allow for weak supervision of latent parts within an element to be recognized. This ability could also be applied in sign language recognition.

After observation of sign language videos, we noticed that in a video, some frames are more discriminative and representative than others. Users could recognize a sign language video correctly only according to those discriminative and representative frames, despite the other ones. As illustrated in Figure 2, each sign language video contains a sequence of frames. However, lots of frames from different videos look similar, and only a part of the frames in each video is specific. These specific frames are discriminative and representative of each video, and people could recognize a video correctly only according to these frames. Inspired by this observation, we introduce the latent SVM model into the sign language recognition. The discriminative frames are treated as latent variables in this model. After model learning and inference, we can correctly recognize the sign of a video and find out the most discriminative and representative frames within it.

The contributions of this work are summarized as follows:

(1) We collect a large dataset with ground-truth labels for research on sign language recognition. Our dataset includes both sign language of words and sign language of sentences.
(2) We adopt the latent SVM for sign language modeling, which can localize the representative frames in video and classify the sign simultaneously.
(3) Based on the latent SVM model trained on word-level signs, we develop an approach to perform recognition on sentence-level sign language.

(4) By using the information from the Kinect sensor, our proposed method can improve the performance of recognition significantly.

The rest of the article is organized as follows. Related work is reviewed in Section 2. In Section 3, we first elaborate on the latent SVM for our word-level sign language recognition, then present the approach of sentence-level sign language recognition based on words. In Section 4, our self-built dataset is introduced, followed by experimental results on this dataset. We conclude the article in Section 5.

2. RELATED WORK

In sign language recognition, data collection and feature extraction are the fundamental parts. Many early sign language recognition methods used gloves or accelerometers to track hands and measure the features [Vogler and Metaxas 1997; Hernandez-Rebollar et al. 2002]. Considering that it is more natural, vision-based sign language recognition approaches became more and more popular. Matsuo et al. [1998] implemented a system to recognize sign language with a stereo camera for recording 3D movements. Segen and Kumar [1999] developed a system that uses a camera and a point light source to track the user's hand. Tracking the hands is an important part of this kind of work and is usually implemented by requiring the signers to wear colored gloves. The gloves used to be single colored for each hand [Kadir et al. 2004]. In some works, the gloves were designed to allow hand pose to be better detected by employing colored markers [Holden and Owens 2001] or differently colored fingers [Hienz et al. 1999]. Beyond using single-colored gloves, Zhang et al. [2004] utilized multicolored gloves to detect both position and shape, in which fingers and palms of the hands were different colors. The skill color model was also used to detect hands, such as is done in Imagawa et al. [1998]. Depth can also be used to simplify the problem. Hasanuzzaman et al. [2004] used a stereo camera pair to obtain the depth image for building models of persons in the image.

The Microsoft Kinect sensor has offered an affordable depth camera that makes depth a viable option for researchers. Keskin et al. [2013] described a depth image–based real-time skeleton fitting for the hand using the Kinect and used it in an American Sign Language (ASL) recognition application. Zafrulla et al. [2011] investigated the application of the Kinect depth-mapping camera for sign language recognition and proved that the Kinect could be a viable option for sign verification.

The classification model is also important in sign language recognition. It determines how to use low-level features to describe and recognize signs. The earliest model in sign language work utilized ANNs [Murakami and Taguchi 1991]. Yang et al. [2002] presented a general method to extract motion trajectories and used them within a time-delay neural network (TDNN) to recognize sign language. Derived from automatic speech recognition, HMMs are also applied in sign language recognition. Vogler and Metaxas [1999] developed parallel hidden Markov models (PaHMMs) and demonstrated that it is a promising scheme in sign language recognition. Kadous [1996] presented a sign language recognition system that uses k-nearest neighbors (KNNs) and decision tree learning to classify isolated signs. Sun et al. [2013] introduced a discriminative exemplar coding (DEC) approach to model sign language video with a depth feature from the Kinect.

Meanwhile, there is no single, universal sign language. Regionally different sign languages have evolved, such as ASL [Brashear et al. 2003], German Sign Language (GSL) [Bauer et al. 2000], and Chinese Sign Language (CSL) [Fang et al. 2003]. For simplification, we focus on ASL in this article; the proposed method can be utilized in other sign languages as well.

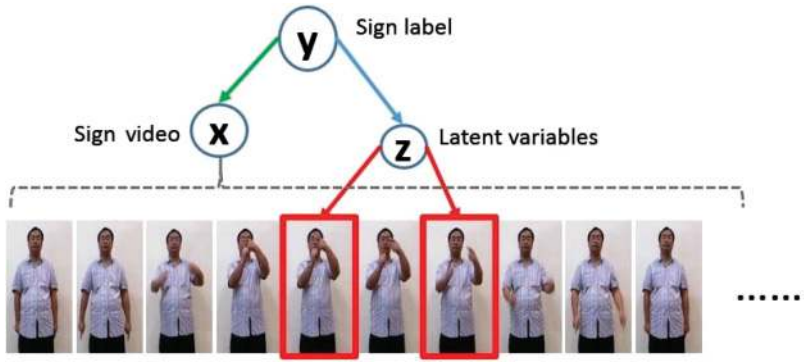

Fig. 3. A visual illustration of our model with latent variables.

3. LATENT SVM MODELING FOR SIGN LANGUAGE RECOGNITION

In this section, we first introduce how to apply the latent SVM model on signs to perform word-level sign language recognition, then explain how to perform sentence-level sign language recognition by inheritance of the latent model from word-level one.

3.1. Word-Level Sign Language Recognition

In word-level sign language recognition, each video contains only one sign corresponding to one word in the vocabulary. Our task is to develop a learning model for application to sign language recognition in sign videos. This model should generate accurate recognition of sign language videos and additionally find the frames within each video that are discriminative and representative of each sign. Latent SVM [Felzenszwalb et al. 2008; Liu et al. 2012a, 2012b] provides a framework in which we can treat the desired state value as latent variables and consider different correlations into potential functions in a discriminative manner. The desired state in word-level signs is the discriminative capability of each frame in the videos. Three types of potential functions are specially formulated to encode the latent variables representing frames into a unified learning framework. The best configurations of latent variables for all frames are searched by optimization, and the sign language videos are classified. An illustration of our latent model is shown in Figure 3.

3.1.1. Frame-Based Latent SVM. Formally, each sign language video $\mathbf{x}$ is to be classified with a semantic label y, where $y \in \{1, 2, \ldots, L\}$, and L is the quantity of all sign words in part A of our dataset (see Section 4.1). Each video consists of a set of frames, whose visual feature vectors are denoted as x_i, $i \in \{1, 2, \ldots, N\}$. For each frame, the discriminative capability is encoded in a latent variable $z_i \in \mathcal{Z} \triangleq \{0, 1\}$, where $z_i = 1$ means that the i-th frame is discriminative and should be representative of sign language recognition, and $z_i = 0$ otherwise. Therefore, $\mathbf{z} = \{z_1, z_2, \ldots, z_N\}$ specifies the discriminative frames within each training video. In the following, we will introduce how to incorporate $\mathbf{z}$ into the proposed model and how to infer it along with model parameter learning.

The goal is to learn a discriminative function f_ω over a sign language video $\mathbf{x}$ and its label y, where ω denotes all model parameters. We use $f_\omega(\mathbf{x}, y)$ to indicate the compatibility among the visual feature $\mathbf{x}$, the sign label y, and the latent variables $\mathbf{z}$. For scoring a video $\mathbf{x}$ with a class label y with the latent variable configuration $\mathbf{z}$, we take $f_\omega(\mathbf{x}, y)$ as a form of $f_\omega(\mathbf{x}, y) = \max_\omega \omega^T \mathbf{\Phi}(\mathbf{x}, \mathbf{z}, y)$, which is defined by combining

different potential functions:

$$\omega^T \mathbf{\Phi}(\mathbf{x}, \mathbf{z}, y) = \sum_{i=1}^{N} \alpha^T \cdot \phi(x_i, z_i) + \sum_{i=1}^{N} \beta^T \cdot \varphi(z_i, y) + \sum_{(i,j)} \gamma^T \cdot \psi(z_i, z_j, y). \tag{1}$$

In this form, parameter vector ω is the concatenation of the parameters in all of the factors. The model presented in the preceding equation simultaneously considers the three potential functions, whose details are described next.

Unary potential $\alpha^T \cdot \phi(x_i, z_i)$. This potential function models frame discriminative capability. Here $\phi(x_i, z_i)$ represents a certain mapping of the visual feature x_i, and the mapping result depends on the latent variable z_i. Model parameter α encodes the weight for different latent variable values. Specifically, it is parameterized as

$$\alpha^T \cdot \phi(x_i, z_i) = \sum_{b \in \mathcal{Z}} \alpha_b^T \cdot \mathbb{1}(z_i = b) \cdot x_i, \tag{2}$$

where $\mathbb{1}(\cdot)$ is the indicator function.

Unary potential $\beta^T \cdot \varphi(z_i, y)$. This potential function models the compatibility between sign label y and latent variable z_i—that is, how likely a sign language video with class label y contains a frame with latent variable z_i. It is defined as

$$\beta^T \cdot \varphi(z_i, y) = \sum_{a \in \mathcal{L}} \sum_{b \in \mathcal{Z}} \beta_{a,b} \cdot \mathbb{1}(y = a) \cdot \mathbb{1}(z_i = b). \tag{3}$$

The parameter $\beta_{a,b}$ measures the compatibility between $y = a$ and $z_i = b$. After model learning, we select the latent variable z_y^* for location y as the latent discriminative label according to $\beta_{a,b}$—that is, $z_y^* = \arg\max_{b \in \mathcal{Z}} \beta_b \cdot \mathbb{1}(z_i = b)$. Frames labeled with latent variable z_y^* are treated as discriminative and representative ones.

Pairwise potential $\gamma^T \cdot \psi(z_i, z_j, y)$. Intuitively, keyframes within the same video should have similar discriminative capability; the latent variables for those keyframes are dependent. Hence, we assume that there are certain constraints between some pairs of latent variables (h_i, h_j). This pairwise potential function models the compatibility between class label y and the dependence of latent variables z_i and z_j—that is, how likely a video with class label y contains a pair of frames with latent variables z_i and z_j. It is defined as

$$\gamma^T \cdot \psi(z_i, z_j, y) = \sum_{y \in \mathcal{L}} \sum_{b \in \mathcal{Z}} \sum_{c \in \mathcal{Z}} \gamma_{a,b,c} \cdot \mathbb{1}(y = a) \cdot \mathbb{1}(z_i = b) \cdot \mathbb{1}(z_j = c), \tag{4}$$

where model parameter $\gamma_{a,b,c}$ denotes the compatibility between class label $y = a$ and latent variable configurations $z_i = b$ and $z_j = c$.

3.1.2. Model Learning. Let $\langle \mathbf{x}^{(i)}, y^{(i)} \rangle (i = 1, 2, \ldots, K)$ be a set of K training videos; our target is to learn the model parameter ω that discriminates the correct sign label y. Here the discriminative latent variables are unobserved and automatically will be inferred along with model learning.

The latent SVM formulation [Felzenszwalb et al. 2008; Yu and Joachims 2009] is utilized to learn the model as follows:

$$\begin{aligned} &\min_{\omega,\xi\geq 0} \ \frac{1}{2}\|w\|^2 + C_1\sum_{i=1}^{K}\xi_i \\ &\text{s.t.} \max_{\mathbf{z}} \omega^T\mathbf{\Phi}\big(\mathbf{x}^{(i)},\mathbf{z},y^{(i)}\big) - \max_{\mathbf{z}}\omega^T\mathbf{\Phi}\big(\mathbf{x}^{(i)},\mathbf{z},y\big) \\ &\quad \geq \Delta\big(y,y^{(i)}\big) - \xi_i, \forall i, \forall y\in\mathcal{L}, \end{aligned} \tag{5}$$

where C_1 is the trade-off parameter similar to that in traditional SVM, and ξ_i is the slack variable for the i-th training example to handle the soft margin. Such an objective function requires that the score for ground-truth label $y^{(i)}$ is much higher than for other labels. The 0–1 loss function $\Delta(y, y^{(i)})$ is used to record the difference, which is defined as

$$\Delta_{0/1}(y,y^{(i)}) = \begin{cases} 1 & \text{if } y\neq y^{(i)} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The constrained optimization problem in Equation (5) can be equivalently rewritten as an unconstrained problem:

$$\begin{aligned} &\min_{\omega} L(\mathbf{w}) = \frac{1}{2}\|\omega\|^2 + C_1\sum_{i=1}^{K} G_i(\omega) \\ &\text{where } G_i(\omega) = \max_{y}\ \Big(\Delta_{0/1}\big(y,y^{(i)}\big) + \max_{\mathbf{z}}\omega^T\mathbf{\Phi}\big(\mathbf{x}^{(i)},\mathbf{z},y\big)\Big) \\ &\quad - \max_{\mathbf{z}}\omega^T\mathbf{\Phi}\big(\mathbf{x}^{(i)},\mathbf{z},y^{(i)}\big). \end{aligned} \tag{7}$$

Nonconvex bundle optimization in Do and Artières [2009] is adopted to solve Equation (7)—that is, the algorithm iteratively builds an increasingly accurate piecewise quadratic approximation of $L(\mathbf{w})$ based on its subgradient $\partial_{\mathbf{w}}L(\mathbf{w})$. The key issue is to compute the subgradients $\partial_{\mathbf{w}}L(\mathbf{w})$. We define the following:

$$\begin{aligned} \mathbf{z}^{(i)^*} &= \arg\max_{\mathbf{z}} \omega^T\mathbf{\Phi}\big(\mathbf{x}^{(i)},\mathbf{z},y\big), \forall i, \forall y\in\mathcal{L}, \\ \mathbf{z}^{(y)} &= \arg\max_{\mathbf{z}} \omega^T\mathbf{\Phi}\big(\mathbf{x}^{(i)},\mathbf{z},y^{(i)}\big), \forall y \\ y^{(i)^*} &= \arg\max_{y}\ \big(\Delta_{0/1}\big(y,y^{(i)}\big) + \max_{\mathbf{z}}\omega^T\mathbf{\Phi}\big(\mathbf{x}^{(i)},\mathbf{z},y\big)\big), \end{aligned} \tag{8}$$

then $\partial_{\omega}L(\omega)$ can be further computed as

$$\frac{\partial}{\omega}L(\omega) = \omega + C_1\sum_{i=1}^{M}\mathbf{\Phi}\big(\mathbf{x}^{(i)},\mathbf{z}^{(i)^*},y^{(i)^*}\big) - C_1\sum_{i=1}^{M}\mathbf{\Phi}\big(\mathbf{x}^{(i)},\mathbf{z}^{(i)},y^{(i)}\big), \tag{9}$$

By utilizing the algorithm in Do and Artières [2009], we can optimize Equation (5) by using $\partial_{\omega}L(\omega)$ and output the optimal model parameter ω.

During each iteration, we can also infer the latent variables $\mathbf{z}$ as follows:

$$\mathbf{z}^{(y)} = \arg\max_{\mathbf{z}} \omega^T\mathbf{\Phi}\big(\mathbf{x}^{(i)},\mathbf{z},y^{(i)}\big), \forall y. \tag{10}$$

This is a standard max-inference problem, and we use loopy belief propagation [Murphy et al. 1999] to approximately solve it.

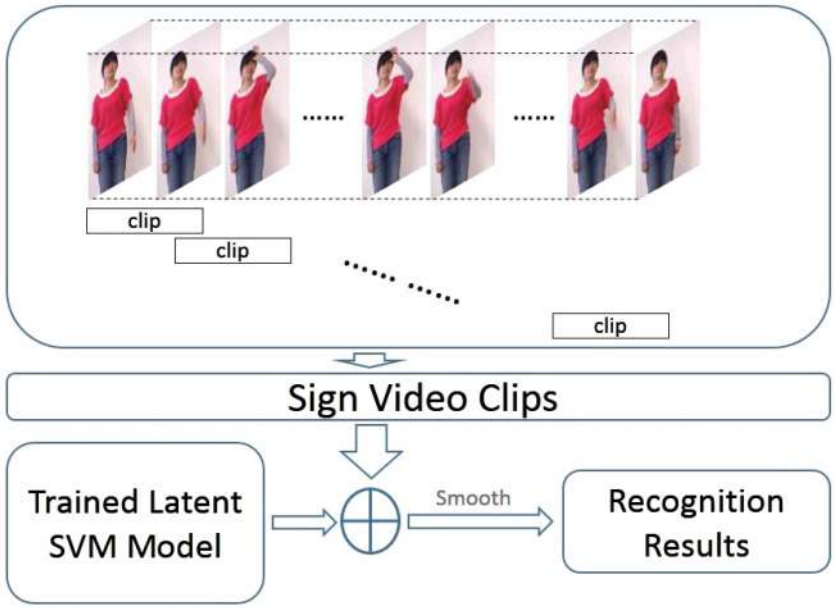

Fig. 4. The framework of our sentence-level sign language recognition approach.

3.1.3. Recognition. Given the learned model parameter ω, we can directly apply it to perform isolated sign language recognition on test video $\mathbf{x}^t$. The procedure will score a word-level sign language video and provide the discriminative frames within it. The class label y^* and latent keyframe z^* are labeled as follows:

$$(y^*, z^*) = \arg\max_{y} \left\{ \max_{\mathbf{z}} \omega^T \mathbf{\Phi}(\mathbf{x}^t, \mathbf{z}, y) \right\}. \tag{11}$$

3.2. Sentence-Level Sign Language Recognition

To explain our approach on sentence-level sign language recognition, we should first introduce the structure of our sentence-level signs. In this work, each sentence-level sign involves a sequence of word-level signs. For example, the sentence-level sign "I drink hot water" consists of four word-level signs: "I," "drink," "hot," and "water." A signer will perform this four word-level sign according to the sign language grammar to generate this sentence-level sign. Specifically, all words in sign sentences are involved in a dictionary, which is the summarization of all words in word-level sign language recognition. For the preceding example, this means that if we have the sentence "I drink hot water" in sentence-level signs, we should have the signs "I," "drink," "hot," and "water" in word level. This property enables our method for sentence-level sign language recognition.

Figure 4 indicates the framework of our sentence-level sign language recognition approach. Suppose that we already have the latent SVM model trained on word-level sign videos. This model can score a sign video and label it. For a sentence-level sign, we divide it into a sequence of video clips, where adjacent clips are temporally overlapped. The trained latent SVM model is then applied on each clip and a score is produced. Consequently, we get a sequence of scores corresponding to the sequence of video clips. ITo take account of the temporal consistency of a video, a Gaussian kernel–based temporal filtering is conducted on the score sequence for smoothing, which results in a smoothed new score sequence. After that, according to the trained latent SVM model,

each score in this smoothed new score sequence will be translated to a label, which results in a label sequence. Notice that each label in this sequence belongs to a clip, and the order of labels in sequence is the same as clips from the sentence video. Finally, we merge all adjacent same labels in label sequence and retain the distinct ones. The final retained distinct labels are treated as the label of this sentence-level sign video.

Formally, suppose that f_ω is the discriminative function trained from word-level signs. The number of classes of these word-level signs is denoted as R. Let $\mathbf{X}$ be a sentence-level sign video. $\mathbf{x}_j \in \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ denotes the overlapped video clip from video $\mathbf{X}$—that is $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$.

For each clip $\mathbf{x}_j$, discriminative function f_ω will produce an R dimension score—that is,

$$f_\omega(\mathbf{x}_j) = (y_{j1}, y_{j2}, \ldots, y_{jR}) \qquad j = 1, 2, \ldots, M.$$

Then scores of all clips for one video will form a score matrix:

$$\begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1R} \\ y_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ y_{M1} & \cdots & & y_{MR} \end{pmatrix}.$$

Here each row denotes the R dimension score over all R sign words for one clip, and each column denotes M scores over all M clips for one sign word.

To take account of the temporal consistency of a video, a Gaussian kernel–based temporal filtering is conducted on each column of score matrix. After smoothing, each column is transferred as

$$(y_{1k}, y_{2k}, \ldots, y_{Mk})^T \Rightarrow (y'_{1k}, y'_{2k}, \ldots, y'_{Mk})^T \qquad k = 1, 2, \ldots, R.$$

The score matrix is then transferred to

$$\begin{pmatrix} y'_{11} & y'_{12} & \cdots & y'_{1R} \\ y'_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ y'_{M1} & \cdots & & y'_{MR} \end{pmatrix}.$$

For elements in each row of this new matrix, we only retain the maximum one. This highest score denotes the final score of this clip—that is,

$$y_j = \max(y'_{j1}, y'_{j2}, \ldots, y'_{jR}) \qquad j = 1, 2, \ldots, M.$$

Then, the score matrix is transferred to the final score sequence $y_1, y_2, \ldots, y_M$, where each element y_j, $j = 1, 2, \ldots, M$ corresponds to a clip in this sentence-level sign video.

From the trained latent model, labels corresponding to score sequence $y_1, y_2, \ldots, y_M$ are then obtained as $l_1, l_2, \ldots, l_M$. We merge the adjacent same elements in $l_1, l_2, \ldots, l_M$ and then obtain the final labels $l^*_1, l^*_2, \ldots, l^*_D$, where $D \leq M$.

The label sequence $l^*_1, l^*_2, \ldots, l^*_D$ is the recognized result for this sentence-level sign video. This result will be treated as correct if the label sequence $l^*_1, l^*_2, \ldots, l^*_D$ is the same as the ground-truth labels of this video. In other words, D should be equal to the ground-truth number of words in this sentence, and the order from l^*_1 to l^*_D should be the same as the ground-truth order of words in this sentence.

Fig. 5. Illustration of signs in part A of our dataset.

4. EXPERIMENTS

In this section, we first introduce our self-built sign language dataset, then conduct recognition on this dataset to validate the feasibility and effectiveness of our proposed method.

4.1. Kinect Sign Language Dataset

Currently, there is no public Kinect sign language dataset. The existing public sign language datasets are totally based on the 2D camera, which lacks the depth information and thus cannot be used to evaluate the proposed method. In this situation, we built the Kinect sign language dataset by ourselves.

Our dataset includes two parts: part A and part B. Part A consists of word-level sign language, whereas part B consists of sentence-level sign language.

Part A includes 73 ASL signs, and each sign corresponds to a word. Figure 5 illustrates some signs of Part A, and Table I lists all words of signs in Part A. These signs came from 100 basic ASL signs that are frequently used by beginning signers. We discarded those that look too similar in vision and finally selected 73 of the signs. We recruited nine participants, each of whom stood in front of the Kinect sensor and performed all signs three times. A total of 1,971 phrases were collected, each of which included a set of color images, a set of depth maps, and a set of skeleton information.

Part B includes 63 sentences, and each sentence consists of 2 to 4 words. Regardless of repeated ones, 63 sentences are constructed by 28 words, which are shown in Table II. Some illustrative sentences are shown in Figure 6. These 28 words are of four types: *subject*, *verb*, *adjective,* and *object*. The permutation of these 28 words generates all

Table I. All Signs in Part A

Signs				
mom	dad	boy	girl	home
work	school	store	church	inout
day	night	week	month	year
will	today	hot	cold	pizza
milk	hotdog	egg	apple	drink
water	candy	hungry	shirt	pants
shoes	coat	wash	brushteeth	sleep
happy	angry	sad	sorry	cry
love	please	thanks	help	who
what	when	where	why	how
stop	big	blue	green	yellow
red	dollars	cat	dog	bird
horse	cow	sheep	pig	I
you	we	he	they	play
go	like	want		

Table II. Vocabulary Used in Part B

Subject	Verb	Adjective	Object
I, we, dad	drink, like go, wash, cry	hot, cold happy, angry, sad	milk, water, egg, pizza, hotdog, apple, candy, dollars, home, school, church, shirt, pants, shoes, coat

(a) I drink hot water

(b) Dad washes shirt

(c) We like candy

Fig. 6. Illustrative sentences in part B of our dataset.

63 sentences. Every sentence in part B obeys the following grammar:

$$Subject \quad Verb \quad [Adjective] \quad Object.$$

For example, we have "I drink hot water" or "Dad washes shirt." Due to the length of sentences, we will not list all sentences in part B here. Similar to part A, we recruited 10 participants, each of whom stood in front of the Kinect sensor and performed all sentences three times. Notice that these 10 participants were different from the 9 participants in part A. A total of 1,890 sentence-level sign language videos were collected, each of which also included color images, depth maps, and skeleton information.

4.2. Features

In this article, we adopt two kinds of features: Ordinary features and Kinect features. The Ordinary features include a histogram of oriented gradients (HOG) feature and an optic flow (OP) feature, which can describe the appearance and motion information. Based on output of the Kinect, we can know the position of hands and obtain their shape information and motion features. In addition, we can also estimate body pose by using the Kinect features.

Ordinary features. The Ordinary features include the HOG feature and OP feature. Based on the depth map from Kinect, it is easy to obtain the mask image and crop the foreground. Once the humans are centralized, we extract the HOG descriptor for each detected area. In human detection, the HOG has been shown to be successful [Dalal and Triggs 2005]. We follow the construction in Dalal and Triggs [2005] to define a dense representation of an image at a particular resolution. The image is first divided into 8×8 nonoverlapping pixel regions, or cells. For each cell, we accumulate a 1D histogram of gradient orientations over pixels in that cell. These histograms capture local shape properties but are also somewhat invariant to small deformations.

The gradient at each pixel is discretized into one of nine orientation bins, and each pixel "votes" for the orientation of its gradient, with a strength that depends on the gradient magnitude at that pixel. For color images, we compute the gradient of each color channel and pick the channel with the highest gradient magnitude at each pixel. Finally, the histogram of each cell is normalized with respect to the gradient energy in a neighborhood around it. We look at the four 2×2 blocks of cells that contain a particular cell and normalize the histogram of the given cell with respect to the total energy in each of these blocks. This leads to a 9×4 dimensional vector representing the local gradient information inside a cell. In our implementation, we resize each image to 256×128 and then extract HOGs in 8×8 cells. This feature vector of the human body bounding box is the $2,340$-dimensional normalized HOG cell vector. After principal component analysis (PCA) [Bao et al. 2012] at ratio 0.9, the dimension of the feature is further reduced to obtain a compact description and efficient computation, which can also be achieved by feature selection as in Liu et al. [2011b].

To generate the hand shape feature, we first crop a 48×48 pixel patch in the position of the hand point on every color frame. Then we extract the HOG feature on every patch and treat this feature as a hand shape feature. This hand shape feature has 288 dimensions.

For generating the hand motion feature, we reuse the patch mentioned earlier. The OP feature is calculated between one patch on a color frame and the patch in the same position on the previous frame. This feature is treated as a hand motion feature and has 2,304 dimensions.

To obtain a compact description and efficient computation, the combined 2,592-dimension feature of a hand patch is then reduced using PCA [Bao et al. 2012] at ratio 0.9.

Table III. Body Pose Features

3D Vectors	Angles	Distance
SR → ER (3)	∠ SC-SR-ER (1)	HR → HL (1)
ER → WR (3)	∠ SR-ER-WR (1)	
WR → HR (3)	∠ ER-WR-HR (1)	
SL → EL (3)	∠ SC-SL-EL (1)	
EL → WL (3)	∠ SL-EL-WL (1)	
WL → HL (3)	∠ EL-WL-HL (1)	
HR → HL (3)		

Note: Numbers in parentheses denote the feature dimension.

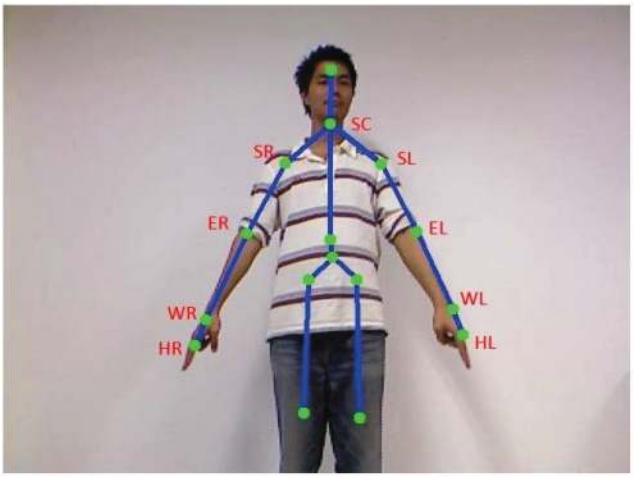

Fig. 7. The skeleton joints' names.

Kinect features. The Kinect sensor has four kinds of output: color image, depth image, mask image, and skeleton image, as shown in Figure 1. The Kinect features include body pose, hand shape, and hand motion features. The body pose features are extracted using skeleton information. By using Microsoft KinectSDK 1.5, we can obtain the positions of the shoulder, elbow, wrist, and hand, on both the right and left sides of the body.

The body pose features are a combination of three parts:

(1) The unit vectors of the elbows with respect to the shoulders, the wrists with respect to the elbows, the hands with respect to the wrists, and the left hand with respect to the right hand. See the first column of Table III.
(2) The joint angles of the shoulders, elbows, and wrists. See the middle column of Table III.
(3) The distance between the right hand and the left hand, normalized by being divided by twice the shoulder width. See the last column of Table III.

In total, the body pose feature has 28 dimensions. Figure 7 and Table III show details of the body pose feature.

The 28-dimension feature is combined to the reduced one to generate the final Kinect feature.

Table IV. Comparison of Different Methods on Part A of Our Dataset

Methods	Mean Accuracy	Features
HC [Sivic and Zisserman 2003] + SVM	67.9%	HOG + OP
LSC [Liu et al. 2011a] + SVM	74.4%	HOG + OP
Our model: Latent SVM	82.3%	HOG + OP
HC [Sivic and Zisserman 2003] + SVM	75.2%	HOG + OP + Kinect
LSC [Liu et al. 2011a] + SVM	75.0%	HOG + OP + Kinect
Our model: Latent SVM	86.0%	HOG + OP + Kinect

Note: The percentages shown are the average accuracies over all signs.

4.3. Baselines

To evaluate our proposed method for sign language recognition, we compare it to two baseline algorithms: hard-assignment coding (HC) [Sivic and Zisserman 2003] and soft-assignment coding (LSC) [Liu et al. 2011a].

Let b_i ($b_i \in R^d$) denote a visual word or an exemplar, where d is the dimensionality of a local feature or a frame representation. The total number of exemplars is n. A matrix $B = [b_1, b_2, \ldots, b_n]$ denotes a visual codebook or a set of basis vectors. Let x_i ($x_i \in R^d$) be the ith local feature in an image. Let z_i ($z_i \in R^n$) be the coding coefficient vector of x_i, with z_{ij} being the coefficient with respect to word b_j.

Hard-assignment coding [Sivic and Zisserman 2003]. For a local feature x_i, there is one and only one nonzero coding coefficient. It corresponds to the nearest visual word subject to a predefined distance. When Euclidean distance is used,

$$z_{ij} = \begin{cases} 1 & if\, j = \arg\min\limits_{j=1,\ldots,n} \left\| x_i - b_j \right\|_2^2 \\ 0 & otherwise. \end{cases}$$

Soft-assignment coding [Liu et al. 2011a]. The jth coding coefficient represents the degree of membership of a local feature x_i to the jth visual word,

$$z_{ij} = \frac{\exp\left(-\alpha \left\| x_i - b_j \right\|_2^2\right)}{\sum_{k=1}^{n} \exp\left(-\alpha \left\| x_i - b_k \right\|_2^2\right)}.$$

where α is the smoothing factor controlling the softness of the assignment. Note that all n visual words are used in computing z_{ij}.

To compensate for the missing temporal information of these two baselines, spatial pyramid matching (SPM) [Lazebnik et al. 2006] is utilized to model the temporal information for representation. Here, the SPM with two levels 1×1 and 1×3 is adopted. After coding, we trained multiclass linear SVM [Cortes and Vapnik 1995] classifiers upon the features in Sivic and Zisserman [2003], Liu et al. [2011a], and Zhang et al. [2013].

4.4. Results on Word-Level Signs

To evaluate our proposed method on isolated sign language recognition, we implemented it and the baseline algorithms on Part A of the dataset.

Meanwhile, to prove the effectiveness of the Kinect features, we also conducted comparison experiments. We designed two different cases. For the first case, all methods are conducted using only Ordinary features. For the other case, all methods are conducted using the combination of Ordinary features and Kinect features.

The recognition accuracies are shown in Table IV. From results we can infer the following:

(1) Whether using Kinect features or not, our latent SVM outperforms the other two baseline algorithms when using the same kind of feature. Specifically, the average

recognition accuracy is approximately 86% over all 73 signs when using the combination of Ordinary and Kinect features, which is nearly a 10% improvement compared to the baseline algorithm HC. This proves the effectiveness of our model on isolated sign language recognition. We believe that the latent SVM model benefits the classification and leads to the improvement. Practically, all baselines with the regular SVM only consider the mapping relations between the visual features of sign videos and the corresponding sign labels, and they have no ability to take into account the different discriminative capabilities of frames in sign videos, as well as the constraints between them. However, as discussed in Section 1, some frames in a sign video may have greater discriminative and representative abilities than others. Different from these methods, the latent SVM can model the discriminative and representative abilities of different frames, which can improve the classification performance. In the latent SVM model, each frame is assigned a latent variable that measures the discriminative power of each frame. By bringing in the latent variables, the mapping relations between frames and visual features of videos, frames, and sign labels are integrated into the discriminative function. The more discriminative and representative a frame is, the greater contribution it makes to classification. After training, the most discriminative frames could not only be sought out, but they also could benefit the classification of sign videos. The experimental results actually demonstrate the advantages of the latent SVM model in our sign language recognition task.

(2) The first three lines of the table show recognition accuracy using only Ordinary features, whereas the last three lines show recognition accuracy using a combination of Ordinary features and Kinect features. It is observed that when additionally using Kinect features, all baselines and our model outperform the ones with only Ordinary features. This result proves that features from the Kinect sensor can improve recognizing performance, and consequently, the Kinect sensor is quite suitable for sign language recognition.

We also show the confusion matrix. Due to the large number of classes, it is difficult to show all classes in one confusion matrix. For simplification, we randomly select 20 different kinds of signs to construct the confusion matrix, as shown in Figure 8. It is observed from the confusion matrix that our model could distinguish signs well. Almost every sign is distinct from others.

In addition, our model could find out the discriminative and representative frames of each sign language video, which are indicated by the latent variables. Some illustrative results are shown in Figure 9. Visually, we can see that illustrated frames in each sign are discriminative to other signs and could basically represent the sign.

Our method and all baselines are implemented on a computer with and Intel® Xeon® E7-4860 2.27GHz CPU with 32G RAM. The whole training process of the two baselines took several hours (depend on the two kinds of features). At the same time, for our method, only each complete iterative round in training, including learning and inference, would take several hours. This means that the whole training process could last more than a week. The high computational cost of training is a disadvantage of the latent SVM model. However, for a testing sign video, the testing process of our latent model takes about the same amount of time as the two baselines, and the recognition accuracy offers remarkable improvement. In this situation, we believe that the time consumption of the training is worthwhile.

4.5. Results on Sentence-Level Signs

As mentioned earlier, all words in sentences of sentence-level signs in part B of our dataset come from the 28 different words within part A. Hence, to reduce the computing time and to improve the recognition performance of sentence-level signs, we retrained a

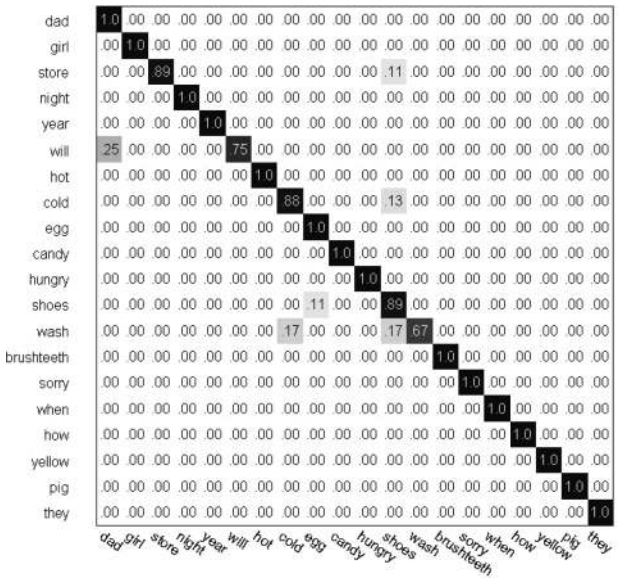

Fig. 8. Confusion matrix on 20 different signs that were randomly selected.

new latent SVM model on only these 28 word-level signs. Notice that this will not affect the classification of all 73 word-level signs in Section 4.4. Similarly, the multiclass linear SVMs for the two baseline algorithms are also retrained on the selected 28 word-level signs.

Based on the retrained latent SVM model, we conduct recognition on all 63 kinds of sentence-level signs according to the method proposed in Section 3.2. Practically, the length of a clip is set to the length of the minimum one in all 28 word-level sign videos, and the overlapped length is set to one third of the clip length. A sentence video is treated as being correctly recognized if the label sequence is the same as the ground-truth labels both in quantity and order. Recognition results of our method and both baselines are shown in Table V.

It is observed from Table V that in sentence-level sign language recognition, our method also outperforms all other baseline algorithms both when using Ordinary features and when using a combination of Ordinary and Kinect features. Specifically, the average recognition accuracy when using the combination of Ordinary and Kinect features is nearly a 13% improvement compared to the baseline algorithm HC. These results demonstrate the effectiveness of our method in sentence-level sign language recognition.

Meanwhile, for our method and the two baseline algorithms, the average recognition accuracy has nearly a 7% improvement when using a combination of Ordinary and Kinect features compared to using only Ordinary features. Once again, the Kinect feature has proved to be effective in sign language recognition.

In addition, we conduct more comparison experiments. We manually divide each sentence-level sign video into nonoverlapped clips according to the ground-truth labels, where each clip contains only one sign word. For example, a video of "I drink hot water" is divided into four clips, where adjacent clips have no overlap, and each clip corresponds to sign words "I," "drink," "hot," and "water," respectively.

Based on these manually divided clips, we conduct sentence-level sign language recognition according to the method proposed in Section 3.2. The recognition results

Fig. 9. Illustrative results of parts of selected discriminative frames.

Table V. Comparison of Different Methods on Part B of Our Dataset

Methods	Mean Accuracy	Features
HC [Sivic and Zisserman 2003] + SVM	62.7%	HOG + OP
LSC [Liu et al. 2011a] + SVM	66.1%	HOG + OP
Our model: Latent SVM	75.0%	HOG + OP
HC [Sivic and Zisserman 2003] + SVM	68.9%	HOG + OP + Kinect
LSC [Liu et al. 2011a] + SVM	73.2%	HOG + OP + Kinect
Our model: Latent SVM	82.9%	HOG + OP + Kinect

Note: The percentages shown are the average accuracies over all sentences.

are shown in Table VI. For simplification, all of these recognitions used a combination of Ordinary features and Kinect features.

From the procedures of recognition, we can infer that recognition accuracy of our proposed sentence-level sign language recognition method in Section 3.2 with any kinds

Table VI. Sentence-Level Recognition Results on Part B with Manually Divided Clips

Methods	Mean Accuracy	Features
HC [Sivic and Zisserman 2003] + SVM	72.1%	HOG + OP + Kinect
LSC [Liu et al. 2011a] + SVM	75.5%	HOG + OP + Kinect
Our model: Latent SVM	84.1%	HOG + OP + Kinect

of clip division may not be higher than the one in Table VI, as all clips in experiments of Table VI are manually cut. We notice that the average recognition accuracy of our method in Table VI is lower than the one in word-level shown in Table IV. The reason is that the meaningless transition between two words in a sentence sign video could decrease the accuracy of recognition. This issue is a challenge in all sentence-level sign language recognition and needs further research.

5. CONCLUSIONS

We have presented a latent SVM model for both word-level and sentence-level sign language recognition. This model could effectively recognize the sign language videos and find out the discriminative and representative frames within each video simultaneously. Moreover, we utilized the Kinect sensor to efficiently capture useful information from signers and hence improved the recognition accuracy. Experimental results demonstrated the effectiveness of our method both in word-level and sentence-level sign language recognition. For the challenge of eliminating the negative effect of meaningless transitions in sign sentences, we plan to continue our research of this work.

REFERENCES

Bing-Kun Bao, Guangcan Liu, Changsheng Xu, and Shuicheng Yan. 2012. Inductive robust principal component analysis. *IEEE Transactions on Image Processing* 21, 8, 3794–3800.

Britta Bauer, Hermann Hienz, and Karl-Friedrich Kraiss. 2000. Video-based continuous sign language recognition using statistical methods. In *Proceedings of the 15th International Conference on Pattern Recognition*, Vol. 2. IEEE, Los Alamitos, CA, 463–466.

Helene Brashear, Thad Starner, Paul Lukowicz, and Holger Junker. 2003. Using multiple sensors for mobile sign language recognition. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC'03)*. 45.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20, 3, 273–297.

Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conference on Vision and Pattern Recognition (CVPR'05)*, Vol. 1. IEEE, Los Alamitos, CA, 886–893.

Trinh-Minh-Tri Do and Thierry Artières. 2009. Large margin training for hidden Markov models with partially observed states. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, New York, NY, 265–272.

Gaolin Fang, Wen Gao, and Debin Zhao. 2003. Large vocabulary sign language recognition based on hierarchical decision trees. In *Proceedings of the 5th International Conference on Multimodal Interfaces*. ACM, New York, NY, 125–131.

S. Sidney Fels and Geoffrey E. Hinton. 1993. Glove-talk: A neural network interface between a data-glove and a speech synthesizer. *IEEE Transactions on Neural Networks* 4, 1, 2–8.

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. 2010. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 9, 1627–1645.

Pedro F. Felzenszwalb, David McAllester, and Deva Ramanan. 2008. A discriminatively trained, multiscale, deformable part model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*. IEEE, Los Alamitos, CA, 1–8.

Mohammad Hasanuzzaman, Vuthichai Ampornaramveth, Tao Zhang, Mohammad Al-Amin Bhuiyan, Yoshiaki Shirai, and Haruki Ueno. 2004. Real-time vision-based gesture recognition for human robot interaction. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO'04)*. IEEE, Los Alamitos, CA, 413–418.

Jose-Luis Hernandez-Rebollar, Nicholas Kyriakopoulos, and Robert W. Lindeman. 2004. A new instrumented approach for translating American Sign Language into sound and text. In *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, Los Alamitos, CA, 547–552.

Jose-Luis Hernandez-Rebollar, Robert W. Lindeman, and Nicholas Kyriakopoulos. 2002. A multi-class pattern recognition system for practical finger spelling translation. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*. IEEE, Los Alamitos, CA, 185.

Hermann Hienz, Britta Bauer, and Karl-Friedrich Kraiss. 1999. HMM-based continuous sign language recognition using stochastic grammars. In *Gesture-Based Communication inHuman-Computer Interaction*. Lecture Notes in Computer Science, Vol. 1739. Springer, 185–196.

Eun-Jung Holden and Robyn Owens. 2001. Visual sign language recognition. In *Multi-Image Analysis*. Lecture Notes in Computer Science, Vol. 2032. Springer, 270–287.

Kazuyuki Imagawa, Shan Lu, and Seiji Igi. 1998. Color-based hands tracking system for sign language recognition. In *Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, Los Alamitos, CA, 462–467.

Timor Kadir, Richard Bowden, Eng-Jon Ong, and Andrew Zisserman. 2004. Minimal training, large lexicon, unconstrained sign language recognition. In *Proceedings of the British Machine Vision Conference*. 939–948.

Mohammed Waleed Kadous. 1996. Machine recognition of Auslan signs using PowerGloves: Towards large-lexicon recognition of sign language. In *Proceedings of the Workshop on the Integration of Gesture in Language and Speech*. 165–174.

Cem Keskin, Furkan Kirac, Yunus Emre Kara, and Lale Akarun. 2013. Real time hand pose estimation using depth sensors. In *Consumer Depth Cameras for Computer Vision*. Advances in Computer Vision and Pattern Recognition 2013. Springer, 119–137.

Jong-Sung Kim, Won Jang, and Zeungnam Bien. 1996. A dynamic gesture recognition system for the Korean sign language (KSL). *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 26, 2, 354–359.

Tian Lan, Yang Wang, and Greg Mori. 2011. Discriminative figure-centric models for joint action localization and recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'11)*. IEEE, Los Alamitos, CA, 2003–2010.

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2. IEEE, Los Alamitos, CA, 2169–2178.

Runghuei Liang and Ming Ouhyoung. 1998. A real-time continuous gesture recognition system for sign language. In *Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, Los Alamitos, CA, 558–567.

Lingqiao Liu, Lei Wang, and Xinwang Liu. 2011a. In defense of soft-assignment coding. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'11)*. IEEE, Los Alamitos, CA, 2486–2493.

Si Liu, Jiashi Feng, Zheng Song, Tianzhu Zhang, Hanqing Lu, Changsheng Xu, and Shuicheng Yan. 2012a. Hi, magic closet, tell me what to wear! In *Proceedings of the 20th ACM International Conference on Multimedia*. ACM, New York, NY, 619–628.

Si Liu, Hairong Liu, Longin Jan Latecki, Shuicheng Yan, Changsheng Xu, and Hanqing Lu. 2011b. Size adaptive selection of most informative features. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*.

Si Liu, Zheng Song, Guangcan Liu, Changsheng Xu, Hanqing Lu, and Shuicheng Yan. 2012b. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*. IEEE, Los Alamitos, CA, 3330–3337.

Hideaki Matsuo, Seiji Igi, Shan Lu, Yuji Nagashima, Yuji Takata, and Terutaka Teshima. 1998. The recognition algorithm with non-contact for Japanese Sign Language using morphological analysis. In *Gesture and Sign Language in Human-Computer Interaction*. Lecture Notes in Computer Science, Vol. 1371. Springer, 273–284.

Kouichi Murakami and Hitomi Taguchi. 1991. Gesture recognition using recurrent neural networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Reaching through Technology*. ACM, New York, NY, 237–242.

Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*. 467–475.

Jakub Segen and Senthil Kumar. 1999. Shadow gestures: 3D hand pose estimation using a single camera. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1. IEEE, Los Alamitos, CA.

Josef Sivic and Andrew Zisserman. 2003. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the 9th IEEE International Conference on Computer Vision*. IEEE, Los Alamitos, CA, 1470–1477.

Thad Starner. 1995. *Visual Recognition of American Sign Language Using Hidden Markov Models.* Technical Report. Massachusetts Institute of Technology, Cambridge, MA.

Thad Starner, Joshua Weaver, and Alex Pentland. 1998. Real-time American Sign Language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 12, 1371–1375.

Chao Sun, Tianzhu Zhang, Bing-Kun Bao, Changsheng Xu, and Tao Mei. 2013. Discriminative exemplar coding for sign language recognition with Kinect. *IEEE Transactions on Cybernetics* 43, 1418–1428.

Christian Vogler and Dimitris Metaxas. 1997. Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics: Computational Cybernetics and Simulation*, Vol. 1. IEEE, Los Alamitos, CA, 156–161.

Christian Vogler and Dimitris Metaxas. 1999. Parallel hidden Markov models for American Sign Language recognition. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, Vol. 1. IEEE, Los Alamitos, CA, 116–122.

Christian Vogler and Dimitris Metaxas. 2001. A framework for recognizing the simultaneous aspects of American Sign Language. *Computer Vision and Image Understanding* 81, 3, 358–384.

Ulrich Von Agris, Jorg Zieren, Ulrich Canzler, Britta Bauer, and Karl-Friedrich Kraiss. 2008. Recent developments in visual sign language recognition. *Universal Access in the Information Society* 6, 4, 323–362.

Ming-Hsuan Yang, Narendra Ahuja, and Mark Tabb. 2002. Extraction of 2D motion trajectories and its application to hand gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 8, 1061–1074.

B. Yao and L. Fei-Fei. 2010. Modeling mutual context of object and human pose in human-object interaction activities. In *Proceeding of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*. IEEE, Los Alamitos, CA, 17–24.

Chun-Nam John Yu and Thorsten Joachims. 2009. Learning structural SVMs with latent variables. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, New York, NY, 1169–1176.

Zahoor Zafrulla, Helene Brashear, Thad Starner, Harley Hamilton, and Peter Presti. 2011. American Sign Language recognition with the Kinect. In *Proceedings of the 13th International Conference on Multimodal Interfaces*. ACM, New York, NY, 279–286.

Liang-Guo Zhang, Yiqiang Chen, Gaolin Fang, Xilin Chen, and Wen Gao. 2004. A vision-based sign language recognition system using tied-mixture density HMM. In *Proceedings of the 6th International Conference on Multimodal Interfaces*. ACM, New York, NY, 198–204.

Tianzhu Zhang, Bernard Ghanemand Si Liu, Changsheng Xu, and Narendra Ahuja. 2013. Low-rank sparse coding for image classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'13)*. IEEE, Los Alamitos, CA.

Tianzhu Zhang, Jing Liu, Si Liu, Yi Ouyang, and Hanqing Lu. 2009. Boosted exemplar learning for human action recognition. In *Proceedings of the IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops'09)*. IEEE, Los Alamitos, CA, 538–545.

Tianzhu Zhang, Jing Liu, Si Liu, Changsheng Xu, and Hanqing Lu. 2011. Boosted exemplar learning for action recognition and annotation. *IEEE Transactions on Circuits and Systems for Video Technology* 21, 7, 853–866.

Tianzhu Zhang, Changsheng Xu, Guangyu Zhu, Si Liu, and Hanqing Lu. 2012. A generic framework for video annotation via semi-supervised learning. In *IEEE Transactions onMultimedia* 14, 4, 1206–1219.

Zhengyou Zhang. 2012. Microsoft Kinect sensor and its effect. *IEEE Multimedia* 19, 2, 4–10.

Received July 2013; revised January 2014; accepted March 2014