American Sign Language Alphabet Recognition Using Microsoft Kinect

Cao Dong, Ming C. Leu and Zhaozheng Yin
Missouri University of Science and Technology
Rolla, MO 65409
{cdbm5,mleu,yinz}@mst.edu

Abstract

American Sign Language (ASL) alphabet recognition using marker-less vision sensors is a challenging task due to the complexity of ASL alphabet signs, self-occlusion of the hand, and limited resolution of the sensors. This paper describes a new method for ASL alphabet recognition using a low-cost depth camera, which is Microsoft's Kinect. A segmented hand configuration is first obtained by using a depth contrast feature based per-pixel classification algorithm. Then, a hierarchical mode-seeking method is developed and implemented to localize hand joint positions under kinematic constraints. Finally, a Random Forest (RF) classifier is built to recognize ASL signs using the joint angles. To validate the performance of this method, we used a publicly available dataset from Surrey University. The results have shown that our method can achieve above 90% accuracy in recognizing 24 static ASL alphabet signs, which is significantly higher in comparison to the previous benchmarks.

1. Introduction

American Sign Language (ASL) is a complete sign language system that is widely used by deaf individuals in the United States and the English-speaking part of Canada. ASL speakers can communicate with each other conveniently using hand gestures. However, communicating with deaf people is still a problem for non-sign-language speakers. There are some professional interpreters that can serve deaf people by real-time sign language interpreting, but the cost is usually high. Moreover, such interpreters are often not available. Therefore, an automatic ASL recognition system is highly desirable.

1.1. Related works

Researchers have been working on sign language recognition systems using different kinds of devices for decades. Sensor-based devices, such as cyber-glove [6, 7] can be used to obtain hand gesture information precisely. However, these devices are difficult to use outside laboratories because of unnatural user experience, difficulties in setting up the system, and high costs. The recent availability of low-cost, high-performance sensing devices, such as the Microsoft Kinect, has made vision-based ASL recognition potentially attractive. As a result, ASL and other hand gesture recognition using such devices have raised high interests in the past a few years [1, 15].

The most common approach to recognize hand gestures using vision-based sensors is to extract low-level features from RGB or depth images using image feature transform, and then employ statistical classifiers to classify gestures according to the features. A series of feature extraction methods have been developed and implemented, such as Scale-invariant Feature Transform (SIFT) [19, 21], Histogram of Oriented Gradients (HOG) [4, 5, 9], Wavelet Moments [16], and Gabor Filters (GF) [18, 20]. Typical classifiers include Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Decision Trees (DT). These methods are robust in recognizing a small number of simple hand gestures. For example, in [19], 96.23% accuracy was reported in recognizing six custom signs using SIFT-based bag-of-features and a SVM classifier. However, classifying ASL signs, which are complex and have a lot of inter-person variations, these methods are usually not able to achieve desirable accuracies. In [20], a Gabor Filter based method was implemented to recognize 24 static ASL alphabet signs, resulting in only 75% mean accuracy and high confusion rates between similar signs such as "r" and "u" (17% confusion rate).

In addition to ASL, many other methods have also been developed and implemented to estimate hand poses and recognize hand gestures. Oikonomidis et al. [17] developed a model-based approach that can recover a hand pose by matching a 3D hand model to the hand's image. Yeo et al. [12] proposed a contour shape analysis method that can recognize 9 simple custom hand gestures with 86.66% accuracy. Qin et al. [25] attempted to recognize 8 direction-pointing gestures using a convex shape decomposition method based on the Radius Morse function, which achieved 91.2% accuracy. Ren et al. [26] proposed a part-based hand gesture recognition method that

978-1-4673-6759-2/15/$31.00 ©2015 IEEE

parsed fingers according to the contour shape of the hand. There were 14 hand gestures containing 10 digits and 4 elementary arithmetic symbols recognized with 93.2% accuracy. Dominio et al. [11] combined multiple depth-based descriptors for hand gesture recognition. The descriptors included the hand region's edge distance and elevation, the curvature of the hand's contour, and the displacement of the samples in the palm region. An SVM classifier was employed to classify gestures and achieved 93.8% accuracy in an experiment to recognize 12 static ASL alphabet and digit signs. Still, these above methods can only recognize a small number (less than 15) of simple gestures (custom signs, ASL digits, or a small portion of ASL alphabet signs).

Shotton et al. [24] proposed a seminal approach that segmented the human body pixel-by-pixel into different parts using depth contrast features and a Random Forest (RF) classifier. This method was successfully implemented in the Kinect system to estimate human body poses. Keskin et al. [8] adapted Shotton's method [24] to segment a hand into parts, and successfully recognized 10 ASL digit signs by mapping joint coordinates to known hand gestures, resulting in 99.96% accuracy. Liang et al. [14] improved the per-pixel based hand parsing method by employing a distance-adaptive feature candidates selection scheme and super-pixel partition-based Markov Random Fields (MRF). The improved algorithm achieved 17 percentage point increase (89% vs 72%) in accuracy in per-pixel classification.

The recent achievements [8, 14, 24] based on the per-pixel classification algorithm have shown a high potential of recognizing a large number of complex hand gestures. Comparing to the low-level image features, the depth comparison features contain more informative descriptions of both the 2D shape and the depth gradients in the context of each pixel.

1.2. Research proposal

This study focused on the method of recognizing complex hand gestures using pixels' classifications information.

- We combined the advantages of the related previous works [14, 24] to segment the hand's region into parts. Where a Random Forest (RF) per-pixel classifier was used to classify pixels according to the depth comparison features [24] selected using a Distance-Adaptive Scheme (DAS) [14].
- We designed a color glove based system to help generate training dataset in order to train the per-pixel classifier.
- We developed a hierarchical mode-seeking method to localize joints under kinematic constraints.
- A hand gesture recognition method using high-level features of joint angles was developed, which achieved high recognition accuracy for 24 alphabet signs (except the dynamic signs "j" and "z" in the complete 26 alphabets).
- We have also evaluated our method using a public dataset [20] to compare the developed system with existing benchmark systems.

The paper is organized as follows. Section 2 introduces the process of hand part segmentation. Section 3 explains the methodology of joint localization and gesture recognition. Section 4 presents and discusses the experimental results. Section 5 draws the conclusions of the study.

2. Hand part segmentation

The per-pixel classification method [24] was adapted to segment the hand into parts. The input of this process was the depth image of the hand region, and the output was the classification label of each pixel. The hand was segmented into 11 parts: the palm, 5 lower finger sections, and 5 fingertips, as shown in Fig. 1.

The method of generating training data is explained in Section 2.1. The feature used for per-pixel classification is introduced in Section 2.2. The classifier's training and classifying process is described in Section 2.3.

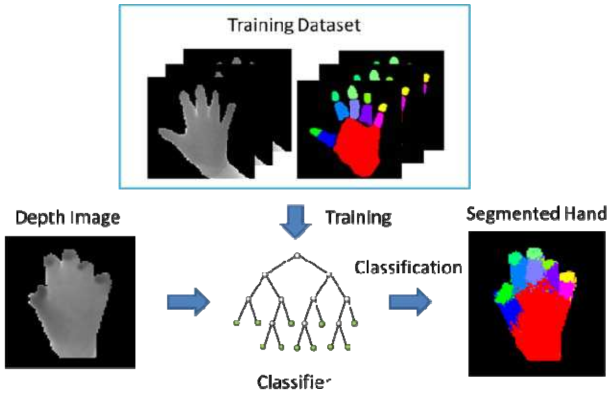

Figure 1. Hand part segmentation. The training dataset contains depth images and the ground truth configurations of the hand's parts. The classifier trained using the training dataset can segment the input depth image into hand parts pixel by pixel.

2.1. Training dataset

The depth image of the hand region can be obtained directly from the Kinect depth sensor. Obtaining the ground truth classification for each pixel, however, is not trivial. Segmenting each depth image manually would be a massive job; Generating synthetic data [8, 24] requires building a high-quality 3D hand model, and simulating the distortion and noise for synthetic data is necessary and challenging. Therefore, a color glove was designed in order to generate realistic training data conveniently; as shown in Fig. 2.

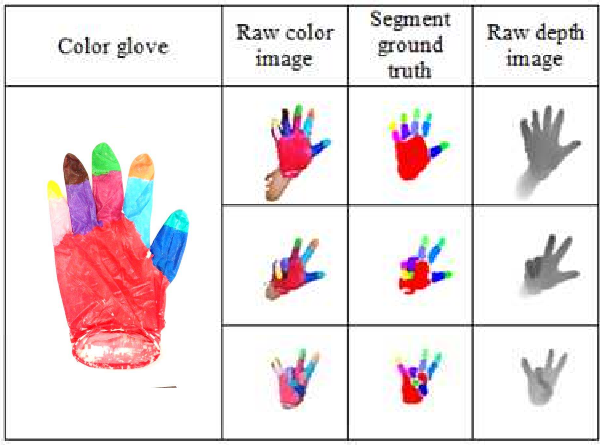

Figure 2. Color glove, color images with glove, segmentation ground truth and corresponding depth images

The glove was painted using 11 different colors according to the configuration of hand parts. The glove can fit the human hand's surface perfectly because it is made from an elastic material. In this way, not only RGB images with colored hand parts but also precise human hand depth images can be obtained using a Kinect sensor. The RGB images were then processed in a hue-saturation-value color space to segment the hand parts according to colors. Therefore, the dataset for hand parsing (depth images and their ground truth) can be generated efficiently by performing various hand gestures wearing the glove.

2.2. Feature extraction

The depth comparison features [24] were employed to describe the context information of each pixel in the hand depth image. For each pixel $\boldsymbol{x}$ in the depth image I, a feature value is described as:

$$f_n(I, \boldsymbol{x}) = I(\boldsymbol{x} + \boldsymbol{v_n}) - I(\boldsymbol{x}) \qquad (1)$$

where the feature $\{\boldsymbol{f_n}\}$ is calculated using the depth value contrast between the pixel $\boldsymbol{x}$ and the offset pixel $\boldsymbol{x} + \boldsymbol{v_n}$. A set of features are extracted for each pixel according to a certain feature selection scheme that contains a set of offset vectors $\{\boldsymbol{v_n}\}$. A large number of features insure a comprehensive description of the pixel's context, but it also may result in considerable computational costs.

In order to improve the efficiency of feature usage, the Distance-Adaptive Scheme (DAS) was employed [14]. The hand region pixels are usually clustered in a relatively small area of the whole depth image. Thus, depth value contrasts between hand pixels and background pixels which are far away will typically provide very little useful information. The contrasts between closer pixels can, however, provide important information. Therefore, a feature selection scheme was generated randomly using a Gaussian distribution kernel to focus on context pixels in the central region of a hand.

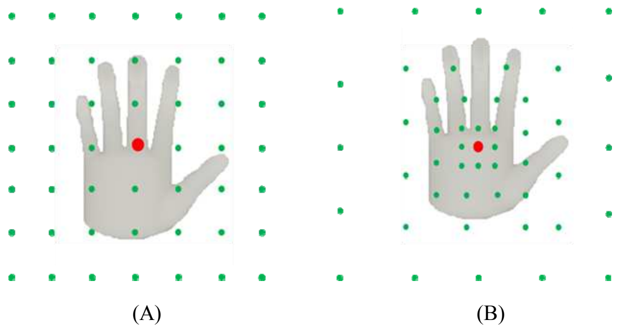

Figure 3. Illustration of feature-selection schemes: (A) an Evenly Distribute Scheme (EDS) and (B) a Distance Adaptive Scheme (DAS).

Fig. 3 illustrates two feature selection schemes which are generated using an EDS and DAS, respectively. The distance adaptive context points are more focused in the the hand region. As a result, DAS features are more likely to contain detailed information in a hand region than EDS features.

2.3. Per-pixel classifier

Labeling pixels according to their corresponding hand part is a typical multi-class classification task. A number of statistical machine learning models can be used, including the Artificial Neural Networks (ANN), Support Vector Machine (SVM), Decision Tree (DT) and Random Forest (RF) [3]. The RF has been proven to be effective for human body segmentation using depth contrast features in [24]. It is robust to outliers, can avoid over-fitting situations in multi-class tasks, and is highly efficient in large database processing. Therefore, RF was selected as the machine learning model in this study.

The RF classifier consists of a set of independent decision trees. At each split node of a decision tree, a feature subset is used to determine the split by comparing the feature values to corresponding thresholds. At each leaf node, the prediction is given as a set of classification probabilities $P(c|\boldsymbol{f}(I, \boldsymbol{x}))$ for each class c. The final prediction of the forest is obtained by a voting process of all trees.

In the process of per-pixel classification, each pixel of the hand's depth image is assigned a set of probabilities $P(c|\boldsymbol{f}(I, \boldsymbol{x}))$ of all classes using the RF classifier. The probability distribution maps of several different classes are illustrated in Fig. 4. A sample of hand part segmentation result is also illustrated in this figure, where each pixel is colored according to the class that has the highest probability. Each hand is segmented into 11 parts (classes).

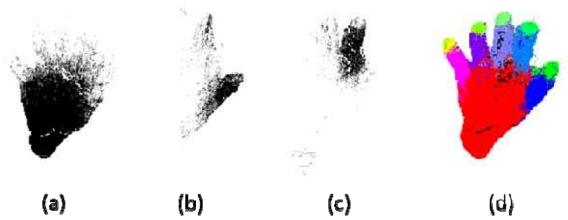

Figure 4. Per-pixel classification results. (a), (b) and (c) Probability distribution maps of “palm,” “thumb finger,” and “middle finger” respectively (Darker pixel values represents higher probabilities). (d) Per-pixel classification result on a hand depth image (hand parts are represented using different colors).

3. Gesture recognition

The RF-based per-pixel classification process classifies each pixel by assigning classification probabilities $P(c|\boldsymbol{f}(I,\boldsymbol{x}))$ for classes representing different hand parts. In [8], the joint positions are obtained by the mean-shift local mode-seeking algorithm [10] performed on the probability distribution maps of the classes {c}. The hand gestures are then recognized by mapping the estimated joint coordinates to known hand gestures. However, both noise and misclassifications in the probability distribution maps make it difficult to localize joint positions accurately. Moreover, the joint coordinates not only can be determined by different gestures but also can be significantly affected by the hand’s size and rotational direction. Thus, joint coordinates are not suitable descriptions of the hand gestures. In addition, lacking constraints can result in unjustified joint positions that make the joint position information unreliable.

In this section, the approach to recognize hand gestures that can overcome the above problems is introduced. In Section 3.1, the mean-shift mode-seeking algorithm is improved by adapting the searching window size with the target hand part size. A confidence function is also employed to evaluate the reliability of the hand part localization. In Section 3.2, the method to constrain joint locations based on the hierarchical kinematic structure of the hand is proposed. Thus, the joint localization algorithm is more robust to outlier clusters in the probability distribution maps. In Section 3.3, the joint angle features are used to describe the hand gestures, thus the feature is invariant to the hand’s size and rotational directions.

3.1. Joint Localization

The hand part segmentation process assigns the classification probabilities $P(c|\boldsymbol{f}(I,\boldsymbol{x}))$of each pixel x for each class (hand part) c . Typically, a multi-modal probability distribution map would be obtained for each hand part from the per-pixel classification algorithm. Thus, the global mass center of the probability distribution map is not suitable to represent the joint position. Therefore, the mean-shift local mode-seeking algorithm [10] was adapted to estimate the joint positions. The mean function can be written as:

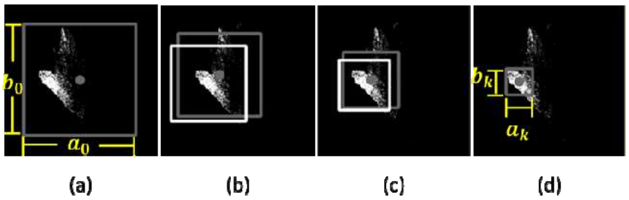

Figure 5. Mean-shift based joint localization process. (a) Initial searching window $a_0 \times b_0$. (b), (c) Dimension-adaptive mean-shift process. (d) Final window $a_k \times b_k$ that localized the global mode.

$$\boldsymbol{m}(\boldsymbol{x}) = \frac{\sum_{i=1}^{N} K(\boldsymbol{x}_i - \boldsymbol{x})\boldsymbol{x}_i}{\sum_{i=1}^{N} K(\boldsymbol{x}_i - \boldsymbol{x})} \tag{2}$$

where $\{\boldsymbol{x_i}\}_{i\epsilon[1,N]}$ is the set of neighborhood pixels, and N is the number of pixels in the searching window. The algorithm starts with an initial estimate $\boldsymbol{x}$, and sets $\boldsymbol{x} \leftarrow \boldsymbol{m}(\boldsymbol{x})$ iteratively until $\boldsymbol{m}(\boldsymbol{x})$ converges. A weighted Gaussian kernel K is used as follows:

$$K(\boldsymbol{x} - \boldsymbol{x}_i) = I(\boldsymbol{x}_i)^2 w_{ic} e^{-\sigma\|\boldsymbol{x}-\boldsymbol{x}_i\|} \tag{3}$$

where

$$w_{ic} = P\big(c\big|\boldsymbol{f}(I,\boldsymbol{x}_i)\big) \tag{4}$$

and σ is a constant parameter to determine the bandwidth of the Gaussian function, w_{ic} is the weight of the pixel $\boldsymbol{x}_i$in the image I. $I(\boldsymbol{x}_i)^2$ is used to estimate the pixel area in the world coordinate system, which is related to the distance of the object to the camera.

In order to find the global mode, the dimension-adaptive method is used. The searching window is initialized at the center of the probability distribution map with a large size $N_0 = a_0 \times b_0$ (Fig. 5a). Then, the window shrinks in each iteration (Fig. 5 b,c) until the size is approximately similar to the size of the hand part (Fig. 5d). The final window size $N_k = a_k \times b_k$ and the shrinking rates a_k/a_{k-1} and b_k/b_{k-1} are constant parameters determined by the size of each hand part.

In some cases, some hand joints may be invisible or unreliably classified. Therefore, a confidence score S_c of the hand part c is given by averaging all the pixel weights w_{ic} in the final searching window. Joints that have poor scores will be considered as “missing” joints. The location of a “missing” joint is assigned by the location of its parent joint. Specifically, the locations of missing fingertips are assigned to the locations of their corresponding fingers, and the locations of missing fingers are assigned to the location of the palm center.

The X and Y coordinates of the joint $\boldsymbol{J}_c$ in the world coordinate system can be obtained by transforming the center position of the final searching window from the image coordinate system to the world coordinate system. The Z coordinate z_c is defined using an average value in the final searching window W_c as:

$$z_c = \frac{\sum_{x \epsilon W_c} I(\boldsymbol{x}) u\big(I(\boldsymbol{x})\big)}{\sum_{x \epsilon W_c} u\big(I(\boldsymbol{x})\big)} \tag{5}$$

$$u(\boldsymbol{x}) = \begin{cases} 1, & I(\boldsymbol{x}) \in [M - \varepsilon, M + \varepsilon] \\ 0, & \quad\quad\quad\quad\quad otherwise \end{cases} \tag{6}$$

$$M = Median(\{I(\boldsymbol{x}) | \boldsymbol{x} \in W_c\}) \tag{7}$$

where I is the depth image, $\boldsymbol{x}$ is the pixel's position vector, and ε is a constant threshold value. The function $u(\boldsymbol{x})$ is used to determine if the depth of the pixel $\boldsymbol{x}$ is valid, where the depth values larger than $m + \epsilon$ or smaller than $m - \epsilon$ are considered as noise. Note, M represents the median distance of the hand pixels regarding to the camera.

3.2. Kinematic constraints

As discussed in Section 3.1, the joint positions can be obtained using the mode-seeking algorithm. However, sometimes the mode-seeking process cannot localize the correct joint position because the pixels of neighborhood hand parts are likely misclassified. For example in Fig.4 (b), besides the global mode locating at the "thumb" position, there is another significant cluster locating at the "index finger" position that is possible to be recognized as the "thumb".

A kinematic constraining method is developed to solve this problem. The concept is employing kinematic probability $P(c|\boldsymbol{x}_i)$ to penalize the weights w_{ic} of pixels that cannot fit the kinematic structure of the hand, i.e., Equation 4 becomes:

$$w_{ic} = P(c|\boldsymbol{x}_i) \cdot P\big(c\big|\boldsymbol{f}(I, \boldsymbol{x}_i)\big) \tag{8}$$

The kinematic probability distribution $\{P(c|x_i)\}$ was obtained from the training dataset generated by the color glove, which contains a large number of segmented hand images (e.g., Fig. 2) for different gestures. The probability distribution map was obtained by

$$P(c \mid \boldsymbol{x}_i) \propto \frac{\sum_{j=1}^{M} \delta\big(L(\boldsymbol{x}_{i,j})=c\big)}{M} \tag{9}$$

where $L(\boldsymbol{x})$ is the class label of pixel $\boldsymbol{x}$, and M is the number of training images. The statistical distribution is obtained by counting the number of pixels from all M images that belong to class c.

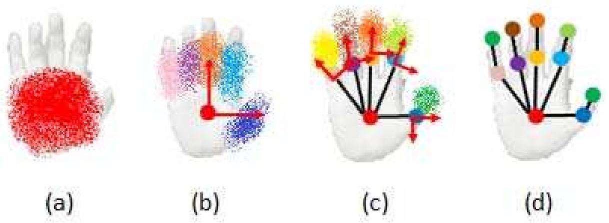

Figure 6. Hierarchical kinematic constraints. (a) Kinematic probability of the palm region. (b) Localize the palm and obtain the kinematic probabilities of the lower fingers. (c) Localize the lower fingers and obtain the kinematic probabilities of the fingertips. (d) Localize the fingertips.

The kinematic probability distribution maps are generated hierarchically (Fig. 6). Firstly, the "palm" joint is localized under the constraints by the kinematic probabilities (Fig. 6a). Secondly, the lower fingers' kinematic probabilities are obtained on the reference coordinate system of the palm, where the origin is the center of the palm, and the x-axis and y-axis are taken to be horizontal and vertical respectively. Then the lower fingers can be localized (Fig. 6b). Thirdly, the kinematic probabilities of five fingertips are obtained on the reference coordinate system of the five fingers respectively (Fig. 7), where the origins are the lower finger joints, the y-axis is along the directions from the palm to the lower finger joints, the x-axis is perpendicular to the y-axis (Fig. 6c). Thus, the fingertips can be localized (Fig. 6d).

Using the method above, the hand joints can be constrained in a smaller region that is kinematically

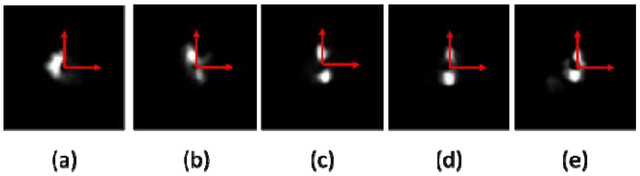

Figure 7. Kinematic probability distribution maps of the fingertips, where the reference coordinates are shown in red. (a) Thumb fingertip. (b) Index fingertip. (c) Middle fingertip. (d) Ring fingertip. (e) Pinky fingertip.

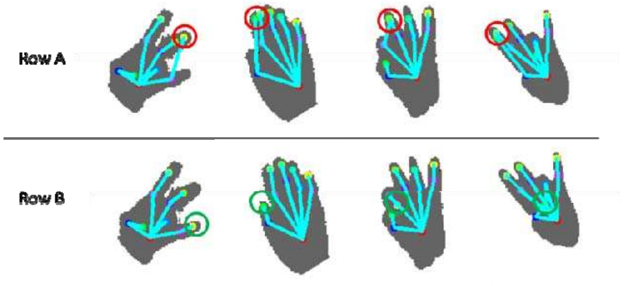

Figure 8. Joint localization results. (Row A) Localized joints without constraints (Row B) Localized joints with constraints

possible. Especially for the fingertip joints, which are highly constrained by the positions and directions of their parent lower finger joints, the hierarchical constraints can effectively improve the joint localization accuracy (Fig. 8).

3.3. Gesture recognition

The joint positions $\{\boldsymbol{J}_{c1}, \boldsymbol{J}_{c2} \dots \boldsymbol{J}_{c11}\}$ in the 3D world coordinate system can be obtained by using the joint localization method discussed in Section 3.1. Thus, the hand gesture can be described using a joint angle feature vector (see Fig. 9).

The feature vector contains the angles between neighborhood lower fingers $\{\theta_1, \theta_2, \theta_3, \theta_4\}$, the angles between each pair of lower and upper fingers $\{\theta_5, \theta_6, \theta_7, \theta_8, \theta_9\}$, and the angles between neighborhood upper fingers $\{\theta_{10}, \theta_{11}, \theta_{12}, \theta_{13}\}$. Using the feature vector $\{\theta_1, \theta_2 \dots \theta_{13}\}$ as the input, the hand gesture as the ground truth, a RF gesture classifier can be trained to recognize ASL alphabets.

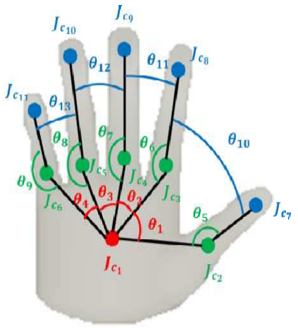

Figure 9. Joint angle features

4. Results

In order to evaluate the developed method, we have done three experiments. First, the RF-based per-pixel classification results are shown in Section 4.1. We tested the per-pixel classifier using the dataset generated using the color glove. Second, in Section 4.2, we compared our method with the method developed by Keskin et al. [8] on our dataset. Third, we used the public dataset from Surrey University [20] to compare our method with other benchmark methods in Section 4.3.

4.1. Per-pixel classification

Our training dataset contains 3,000 images generated using the color glove, of which 2,000 images were picked randomly for training and the rest were used for validation. The resolution of the training image was normalized to 256×256. For each pixel, 100 depth comparison features were extracted. We tested the Evenly Distribute Scheme (EDS) and Distance Adaptive Scheme (DAS) feature selection methods. The accuracy corresponding to the training sample amount is shown in Fig. 10.

The accuracy on the dataset containing 10 million pixels using DAS was 88.96%, comparing to 81.34% using the EDS. These results show that the adaption of DAS we developed has significantly improved the accuracy of per-pixel classification. In addition, according to the trend of the accuracy curve, the classification accuracy could still increase if the size of training database expands.

4.2. ASL alphabet recognition

In order to evaluate the performance of the developed system for ASL alphabet recognition, 72,000 depth images of a hand were generated using the Kinect, of which 48,000 of the data were used for training and the rest 24,000 were used for testing. The gestures included 24 alphabet signs (excluding the dynamic signs “j” and “z”). The signed alphabets followed the standard from the ASL University website (http://www.lifeprint.com) with varying distances and view angles from the Kinect sensor.

Firstly, the Random Forest and Joint Angle (RF-JA) method classifies gestures using a RF classifier and the feature vector containing 13 key angles of the hand skeleton. Secondly, the method of Random Forest and Joint Angles with Constraints (RF-JA+C) was developed based on the RF-JA method, where the hierarchical kinematic constrains were added to improve the joint localization accuracy. Thirdly, the Random Forest and Joint Positions (RF-JP) method introduced in [8] was also implemented to compare with our method. In [8], the joints were localized using the basic mean-shift algorithm. The hand gestures were recognized by mapping the joint position coordinates to the known gestures. The results obtained using the above three methods above are shown in Fig. 11.

The highest accuracy achieved using the RF-JP method was 66.32%, our RF-JA method achieved 83.65%, and our RF-JA+C method achieved 91.85%. It is clear that the joint angle features can provide informative description of the hand gesture. Furthermore, the developed kinematic constraining method can further improve the recognition accuracy of the hand gestures.

4.3. Experiments on a public dataset

To evaluate and compare our method with previous methods by other researchers, a public dataset [20] was used. The dataset contained 24 ASL fingerspelling signs (not including “J” and “X”) performed by 5 subjects, where 500 samples of each sign were recorded for every subject. The subjects were asked to sign facing the sensor and to move their hand around while keeping the hand shape fixed. This dataset was generated using Kinect V1.

In the experiment, we firstly used the same per-pixel

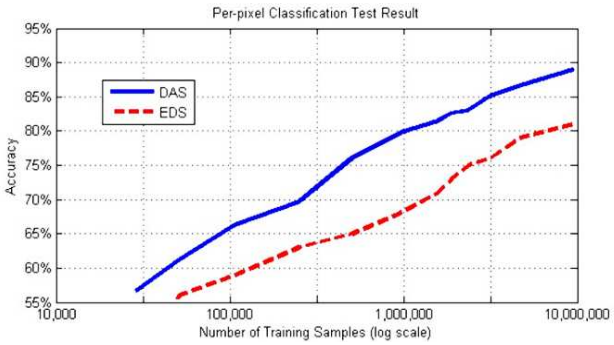

Figure 10. Per-pixel classification accuracy using different features selected using Distance-Adaptive Scheme (DAS) and Evenly Distributed Scheme (EDS).

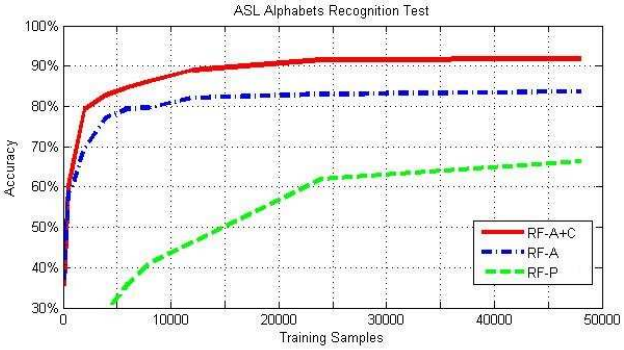

Figure 11. ASL Alphabets recognition accuracy using methods of our Random Forest and Angle under Constraints (RF-JA+C), our Random Forest and Joint Angles (RF-JA), and Random Forest and Joint Positions (RF-JP) [8]

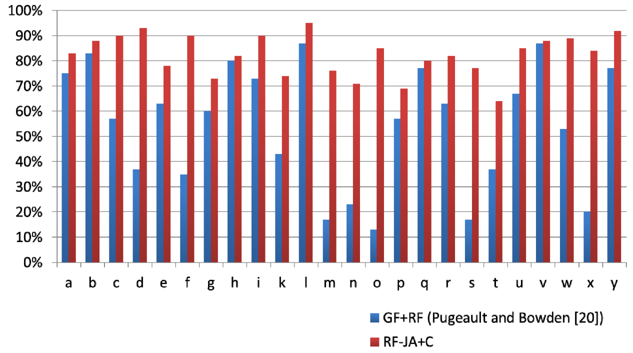

Figure 12. Comparison of the accuracy for each alphabet sign using the RF-JA+C (red) and the RF+GF method [20] (blue).

Table 1
Comparison of ASL alphabet recognition accuracies on the public dataset [20] using "half-half" (h-h) and "leave one out" (l-o-o) experimental tests.

Method	h-h	1-o-o
GF-RF [20]	75%	49%
ESF-MLRF [2]	87%	57%
RF-JP [8]	59%	43%
RF-JA+C (our method)	90%	70%

GF-RF: Gabor Filter-based features and Random Forest [20].
ESF-MLRF: Ensemble of Shape Function and Multi-Layer Random Forest [2].
RF-JP: Random Forest and Joint Positions [8].
RF-JA+C: Random Forest and Joint Angles with Constraints.

classifier trained with our color glove dataset to segment the hand into parts. Then two kinds of validation methods were used for comparison. In "half-half" (h-h) experiment, one half of the dataset were used for training, while the other half dataset were used for testing. The "leave-one-out" (l-o-o) accuracy, however, was obtained by testing the dataset of one subject using the classifier trained with the dataset of the other four subjects,

Fig. 12 illustrates the comparison of the h-h recognition accuracy for each alphabet between the results obtained using the Gabor filter-based hand shape feature and random forest classifier (GF-RF) [20] and using the RF-JA+C method we developed. The recognition accuracy has been significantly improved using our method especially for complex and confusing gestures including "m", "e", "n" ,"o" and "s", "t". The mean accuracy of RF-JA+C method showed 15 percentage point improvement in the h-h experiment and 21% percentage point in the 1-o-o experiment in comparison to the GF+RF method reported in [20].

As illustrated in Table 1, we also compared our results obtained using RF-JA+C with the results obtained using the ensemble of Shape Function Descriptor (ESF) and Multi-Layer Random Forest (MLRF) (ESF-MLRF) [2] and using the RF-JP [8] method. Although in the h-h experiment, the ESF-MLRF method also achieved high mean accuracy (87%), which is close to our RF-JA+C method (90%), in the 1-o-o experiment our RF-JA+C method still had significantly higher mean recognition accuracy. Several samples of recognized signs are shown in Fig. 13.

5. Conclusions

This paper describes a new method developed for American Sign Language (ASL) alphabet recognition. By using depth data obtained from the Kinect sensor, the per-pixel classification algorithm was used to segment a human hand into parts. We employed a latex color glove instead of a commonly used synthetic 3D hand model in order to generate realistic per-pixel training data. The joint positions were obtained using a dimension-adaptive mean-shift mode-seeking algorithm. To improve the joint localization accuracy, we employed kinematic probabilities in the mode-seeking algorithm to constrain the joints within possible motion ranges. The assemblies of the 13 key angles of the hand skeleton were used as the features to describe hand gestures. An Random Forest (RF) gesture classifier was implemented in the end to recognize ASL

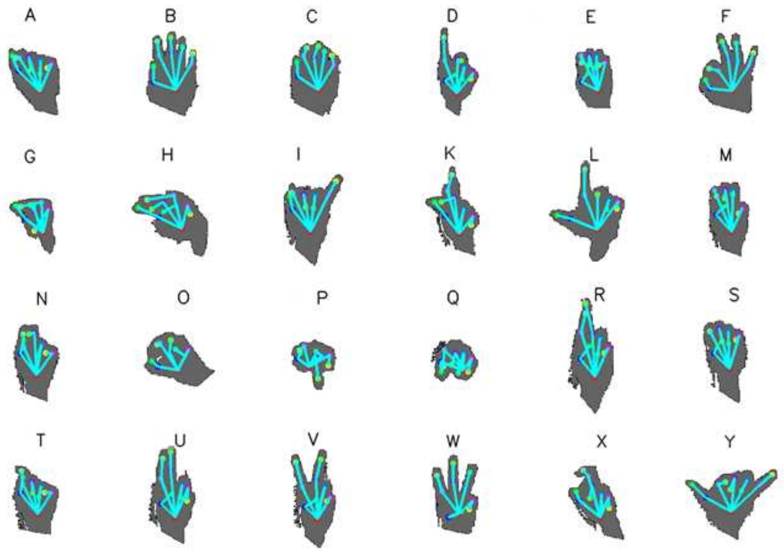

Figure 13. Samples of recognized ASL alphabets, joints and skeletons are shown using colors.

signs. The system achieved a mean accuracy of 92% on a dataset containing 24 static alphabet signs. In comparison with previous methods on Surrey University's dataset, our method achieved the highest accuracy in recognizing ASL signs. Since ASL signs represent complex hand gestures, the capability of recognizing ASL alphabets implies that our method has a great potential of being applicable to other applications that involve the use of hand gestures, for example controlling industrial robots on a factory floor by bald hands or remotely communicating with healthcare assistants from a hospital room when oral communication is disfunctional.

Acknowledgements This work was partially supported by the NSF grant CNS-1205695 and Intelligent Systems Center of Missouri S&T.

References

[1] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. Computer Vision and Image Understanding, Vol. 108, pp. 52-73, 2007.

[2] A. Kuznetsova, L. L. Taixe, and B. Rosenhahn. Real-time sign language recognition using a consumer depth camera. Computer Vision Workshops, pp. 83-90, 2013.

[3] B. Leo. Random Forests. Machine Learning, Vol. 45, pp. 5-32, 2011.

[4] C. Nölker, and H. Ritter. Detection of Fingertips in Human Hand Movement Sequences. Gesture and Sign Language in Human-Computer Interaction, Vol.1371, pp. 209-218, 1998.

[5] C. Nölker, and H. Ritter. GREFIT: Visual Recognition of Hand Postures. Gesture and Sign Language in Human-Computer Interaction, Vol.1739, pp. 61-72, 1999.

[6] C. Oz, and M. C. Leu. Recognition of finger spelling of American sign language with artificial neural network using position/orientation sensors and data glove. Advances in Neural Networks, pp. 157-164,2005.

[7] C. Oz, and M. C. Leu. Linguistic Properties Based on American Sign Language Recognition with Artificial Neural Networks Using a Sensory Glove and Motion Tracker. Computational Intelligence and Bioinspired Systems, pp.1197-1205, 2005.

[8] C. Keskin, F. Kirac, Y. E. Kara, and L. Akarun. Real Time Hand Pose Estimation using Depth Sensors. 2011 Computer Vision Workshops, pp. 1228-1234, 2011.

[9] C. R. Mihalache, B. Apstol. Hand pose estimation using HOG features from RGB-D data. System Theory, Control and Computing (ICSTCC), pp. 356-361, 2013.

[10] D. Comaniciu, and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE Trans. PAMI, pp. 603-619, 2002.

[11] F. Dominio, M. Donadeo, and P. Zanuttigh. Combining multiple depth-based descriptors for hand gesture recognition, Pattern Recognition Letters, pp. 101-111, 2014.

[12] H. S. Yeo, B. G. Lee, and H. Lim. Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware, Multimedia Tools and Applications, pp. 1-29, 2013.

[13] H. Liang, J. Yuan, D. Thalmann, Z. Zhang. Model-based hand pose estimation via spatial-temporal hand parsing and 3D fingertip localization. The Visual Computer, Vol. 29, pp. 837-848, 2013.

[14] H. Liang, J. Yuan, and D. Thalmann. Parsing the Hand in Depth Images. Multimedia, Vol. 16, pp. 1241-1253, 2014.

[15] J. Suarez, and R. Robin. Hand Gesture Recognition with Depth Images: A Review. RO-MAN, pp. 411-417, 2012.

[16] K. Chen, X. Guo, and J. Wu, Gesture recognition system based on wavelet moment. Applied Mechanics and Materials, Vol. 401-403, pp. 1377-1380, 2013.

[17] L. Oikonomidis, N. Kyriazis, and A. A. Argyros. Markerless and Efficient 26-DOF Hand Pose Recovery, Computer Vision, pp. 744-757, 2011.

[18] M. A. Amin, and H. Yan. Sign Language Finger Alphabet Recognition from Gabor-PCA Representation of Hand Gestures. Machine Learning and Cybernetics, Vol. 4, pp. 2218-2223, 2007.

[19] N. H. Dardas, and N. D. Georganas. Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques. Instrumentation and Measurement, Vol. 60, pp. 3592-3607, 2011.

[20] N. Pugeault, and R. Bowden. Spelling It Out: Real-Time ASL Fingerspelling Recognition. 2011 IEEE Workshop on Consumer Depth Cameras for Computer Vision, pp. 1114-1119, 2011.

[21] P. Gurjal, and K. Kunnur. Real Time Hand Gesture Recognition Using SIFT. International Journal of Electronics and Electrical Engineering, Vol. 2, Issue 3, 2012.

[22] R. Y. Wang, and J. Popovic. Real-Time Hand-Tracking with a Color Glove. ACM Transactions on Graphics (TOG), Vol. 28, 2009.

[23] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient Regression of General-Activity Human Pose from Depth Images. ICCV '11 Proceedings of the 2011 International Conference on Computer Vision, pp. 415-422, 2011.

[24] S. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-Time Human Pose Recognition in Parts from Single Depth Image. Communications of the ACM (CACM), pp. 116- 124, 2011.

[25] S. Qin, X. Zhu, H. Yu, S. Ge, Y. Yang, and Y. Jiang. Real-Time Markerless Hand Gesture Recognition with Depth Camera. Advances in Multimedia Information Processing, pp. 186-197, 2012.

[26] Z. Ren, J. Yuan, J. Meng, and Z. Zhang. Robust Part-Based Hand Gesture Recognition Using Kinect Sensor. IEEE Transactions on Multimedia, pp. 1110-1120, 2013.