Visual Synthetic Data Generation for Sign Language Recognition

Ahmed Ibrahim
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
a24ibrah@uwaterloo.ca

Rasha Kashef
Department of Computer Science
College of Computing and Information Technology
Arab Academy for Science and Technology
Cairo, Egypt
rkashef@pami.uwaterloo.ca

Abstract One of the essential steps in pattern recognition is having a training set that contains diversity of exemplars. This step is a major obstacle in sign language recognition. The availability of large public corpus with diversity of examples is limited due to the large number of degrees of freedom inherent in sign language hand gestures and the major variances between different signers. This causes difficulties in collecting a large number of exemplars for each hand gesture and makes it either expensive or impractical. This paper attempts to address this challenge by using hand motion carried out by synthetic 3D animated models. The experiments examine how the recognition accuracy will be changed when the training set is enlarged. The results demonstrate how the recognition accuracy has been improved even when the expressing ability of the classifier is limited. Moreover, it enhances the quality of training set by including a signer independent manner.

Keywords: signer-independent recognition, sign language recognition, synthetic data generation

1. Introduction

The need for a dataset with large diversity is an important step in many research fields. In sign language hand gesture recognition, the recognition accuracy not only depends on the inherent properties of the detected object to be recognized and the design of classifiers, but also on the how much diversity this training data contains (Jiang F. et al., 2009). Most of the benchmarked dataset are limited in the number of signers and/or the number of each gesture tires which often resulting in over-fitting and poor generalization as in Mitra S. and Acharya T., (2007) and Poppe R. (2012). Overcome this challenge has been supported in many ways one of them is using synthetic data to enrich the real dataset as shown in Jiang F. et al. (2009).

In Nonnemaker J., and Baird H. (2009), it has been indicated that synthesis can be carried out in at least three types of spaces: sample space (the set of all real samples as they are originally described), parameter (or, generator) space (vectors of values determining how samples are generated), and finally, feature space (vectors by numerical feature values).

It’s difficult to use parameter space due to the large number of degrees of freedom inherent in sign language hand gestures and the major variances between different signers. In feature space, many statistical-based approaches in data enlargement have been used by Mitra S. and Acharya T., (2007), Poppe R. (2012), Nonnemaker J., and Baird H. (2009), Wu, C.F.J. (1986) and Pavlovic V. (1997). The drawbacks of this class of data enlargement approaches include: (1) the generated data inherits the structure of real data which yields the same defects and amplifies the embedded noise; (2) the new dataset
does not have enough flexibility to be generalized because the original dataset is signer dependent as shown in Jiang F. et al. (2009).

Currently, research trends are going toward using animated videos for human movements to overcome the challenge of limited training data as in Jiang F. et al. (2009), Maik V. et al. (2010), and Tian Y. et al. (2010). Maik V. et al. (2010) had used this approach for validating a new method for classifying human body poses. Their experiments were carried out on their own in-house animated poses to evaluate the performance of their proposed classification scheme. Also, Tian Y. et al. (2010) presented a discriminative framework for 3D pose inference. This framework has been tested using an animated synthetic sequences produced by Poser software (Web-1). Poser is a 3D animation software package for the human figure design, posing, and animation. It allows the user to load actors, props, illumination and cameras for still and animated human characters.

In sign language recognition, Péporté M. (2009) have used visual synthesis approach in generating sign language hand gestures to train an Irish Sign Language (ISL) recognition system. She used synthetic data in order to simplify the generation of motion variations of the animated signer which allows her to generate different forms of similar gesture with minor variations in the motion parameters. She converted the motion patterns into symbols that were used in training a Hidden Markov Model (HMM) classifier after projecting data into subspace using Principal Component Analysis (PCA).

In this paper, we are extending Péporté M. (2009) approach from a different prospective. First, we use synthetic data to enrich the real data by adding more diversity. Second, we combine real and synthetic data together into the same subspace and propose a homogeneity (coherency) measure which not only permits the enlargement of the training data to be more signer-independent data but it provides the flexibility in analyzing and testing the data under different parameters settings. Moreover, the proposed approach will not only reduce the chance of over-fitting but also it will increase the generalization in the available training datasets as well.

The rest of this paper is organized as follows: In section 2, the proposed data enlargement approach is presented. Synthetic data generation tool is described in section 3. Sections 4 discuss the experimental setup and results, respectively. Finally, we draw some conclusions and outline future work in section 5.

2. Visual synthetic data set enlargement

In this section, we briefly describe and discuss the proposed approach for synthesizing sign language gestures visually. The proposed approach is composed of three phases as shown in Fig 1.

2.1. Synthetic data generation

Artificially generated datasets are a common approach in computer vision systems like in Mike V. et al. (2010) and Tian Y. et al. (2010). Once a simulated data is developed, it is fast and cheap to produce large quantities of high quality data (Poppe R. 2012). However it is important for the data to preserve the characteristics of real data. Hence, we use a purely image-based approach to combine real image sequences with realistic 3D animated models in order to generate visually convincing animations of sign language gestures. This is motivated by the intention to preserve the practicality of the original data, without losing quality due to intermediate conversion steps into a 3D representation. Several articles like in Lee J. and Kunii T. (1993), Barreit W.A. and Cheney A.S. (2002), Igarashi T. et al (2005) and Chuang Y. et al (2005) have shown the promising potential of such solely image-based animation approaches.
Generating 3D animated human gesture model is composed of several stages as shown in Fig. 2. In the first stage, we build a 3D basic gesture model from the real gestures manually. This is an important and fundamental stage because we need to map the real gestures video into a 3D animated model. There must be a closest match between the generated model and the real one. The second stage is implementing PythonPoser script (Web-1) that contains all possible variations of sign gestures. In our experiment, we only cover the variations in motion. There are many other variations that can be handled by PythonPoser but our research concern here is on modelling the gesture movements. The final stage in this phase is generating the videos for the 3D human model we built. This stage is held by Poser where it can realize automatically a video from created animations.

2. Processing sign language gestures videos

In sign languages, hand gestures represent the word meaning intended by the signer. Thus, by observing a sequence of hand gestures and extracting suitable features, the required sign could be recognized. Appearance-based features are used to recognize the underlying sign. Those features acquired from the image sequence without any pre-processing or segmentation. This can be done by reshaping every image into a high dimensional intensity vector. Such a high dimensional feature space is too large to permit robust classification. Applying dimension reduction technique such as Principal Component Analysis (PCA) as shown in Mitra S. and Acharya T., (2007) is a common way to resolve this problem. Such dimension reduction technique is able to projecting the original data into a new space which combine the original features and mapping them into a different space. In PCA space, every image will be represented as a point and the entire image sequence will be characterized as a unique trajectory. This method can provide a measure for hand configuration and orientation as mentioned in Mitra S. and Acharya T., (2007) and Poppe R. (2010). Furthermore, it can be used to combine both real and synthetic generated data in the same feature space as well. A detailed description about PCA can be found in Gonzalez R.C. et al. (2008).

Fig. 3 shows the stages of combining both types of sign gestures. First, the signer is detected in the gestures’ videos. This is done by getting the region-of-interest (ROI) or the boundary box around the signer for all videos frames using Blub Analysis approach in Gonzalez R.C. et al. (2008). Second, the ROI is cropped and reshaped to realize the features matrix. The feature matrix is shaped by ‘n’ rows of observations (frames) and ‘p’ columns of variables (pixel intensities) which tackled the ROI area. Last, the principal components analysis (PCA) performed on the n-by-p data matrix. Finally, both gestures’ data are needed to be projected into the principal component (PC) space. Now, we have both real and synthetic sign language gestures in the same space.
2.3. Measuring the accuracy

In order to illustrate the feasibility of using the proposed approach against “real” data in terms of the overall recognition accuracy, we have used one of the most traditional machine learning algorithms as k-nearest neighbour (KNN) as shown in Cunningham P. and Delany S. (2007). KNN have been selected because of its inherit properties that are suitable for the nature of the experiment. In KNN, no residual classifier is needed to be built ahead of time. It does not build a classification model in advance. In this classification paradigm, k nearest neighbours of a training data is calculated first. Then the correspondences of one sample from testing data to the k nearest neighbours are combined according to the class of the neighbours, and the testing sample is assigned to the best related class.

Furthermore, KNN uses the pixel intensity of both real and synthetic sign language gestures with the same weight. Measuring the accuracy is established in three stages as shown in Fig.4. The output of phase 2 is first filtered where we select the synthetic dataset that attains the same homogeneity (coherency) of data or better compared to the original real dataset. The coherency of a dataset is expressed as the pairwise similarity between dataset points (Intra similarity between points). The higher the coherency is, the higher the homogeneity of points in the dataset. The coherency of a dataset $D$ can be calculated by:

$$D_{coh} = \frac{1}{n(n-1)} \sum_{i=1}^{n} Similarity(x_i, y_i)$$

Where $(x, y)$ are the real and synthetic ‘n’ data points with respect to the correspondence between different types. Similarity can be evaluated using any similarity measure (e.g. Cosine Coefficient (Eq.2)) or distance measures (e.g. Euclidian Distance (Eq.3)).

$$\text{cosSim} = \frac{x \cdot y}{\|x\| \|y\|}$$

The cosine coefficient as in Steinbach M. et al. (2000) is widely used in text mining and document clustering applications. Where $(.)$ indicates the vector dot product and $\|\|$ indicates the length of the vector. Euclidian Distance is defined as:

$$\|x - y\|_2 = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

Euclidian distance can be defined as the straight line between any two points in the feature space. In the second stage the filtered data is randomly divided into two sets: training set and testing set. The KNN classifier uses those sets to measure the overall enhancement in recognition accuracy due to combining synthetic gestures into the real ones in stage 3. The following section discusses the adopted synthetic data generation approach.
3. Synthetic data generation tool

Poser software (Web-1) is 3D character animation package used in building a 3D human figure model design. Using Poser is expected to give more insight in understanding the contents of sign language gestures. The human figure’s weights and heights can be easily modified, even for single parts of the body. Moreover, one can add different clothes and head hair to the character. Environment objects can be imported from other commercial 3D software. The body can be set freely in the three dimensional environment, in every pose. All body parts can be moved separately, either in physiologic or in pathologic motions. Even every single finger can be modified. Poser allows the construction of a set of different poses for still image reproduction. Poser knows the centre of gravity of a body and it is possible to move a character only round this centre of gravity. Even the possibility to set movements against anatomical order is important for accident scene reconstruction. Otherwise it is possible to move characters only in anatomic boundaries.

4. Experimental Setup

The used synthetic gesture set was developed by Péporté M. (2009). It contained the following gestures: “Alive”, “Call”, “Calm” and “Dance”. The first two gestures were one hand gestures and the remaining were two hands gestures. In future work we plan to extend this choice to a more complex gesture set in order to find out how different types of gestures will satisfy the same enlargement average accuracy for its similar class.

![Fig.5. The used animated hand gestures which enlarge the correspondent real hand gestures by (Péporté M. 2009): (a) Alive (b) Call (C) Calm (d) Dance.](image)

Different sets of tests have been constructed with various configurations of real data (RD) and synthetic data (Poser). The main factor in the comparison is the classification accuracy of the KNN classifier for each run. In the following, we report the experimental results for synthetic and real data, comparing the results obtained using datasets described in below.

4. 1. Datasets descriptions

The dataset was collected from the Irish Sign Language data as shown in Péporté M. (2009) which contains 3 signers in total. The dataset was divided into two subsets based on the signers. We assumed that one of them is a registered signer (RS) and the other is as unregistered signers (US). The KNN classifier is trained using the samples of the RS. The samples of the US were used as a testing set which consists of 10 different tries for 4 types of gestures. Synthetic data generated by the Poser had four gestures: ‘Alive’, ‘Call’, ‘Calm’ and ‘Dance’ as shown in Fig.5. Those synthetic gestures had same motion identical to the real gesture set. The Poser gesture set was captured from the same angle (main camera view) but with different parameters’ variations. In Figure 6, although the synthetic gestures had same motion identical to the real gesture set but it is expected to see that the generated data are far outside the area of the original one due to the difference in environment conditions. Thus, the generated data still considered as valid because both type of data still have the same motion pattern structure.
4. 2. Experimental Results

The experiments are designed to examine the effect of combining synthetic data on the recognition accuracy in an offline mode. Fig. 6 illustrates the projections of real dataset in part (a), and enlarged data (twice its size) in part (b). The vertical and horizontal axis represents the principal components one and two respectively. Although we have ‘p’ number of principal components which are normally bigger than the number of observations (frames). We selected the first two principal components for visualization purposes.

Table 1. Mean classification accuracy for each class

<table>
<thead>
<tr>
<th>Mean classification accuracy (%)</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
<th>Mean Average Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small size dataset</td>
<td>34.5</td>
<td>50.0</td>
<td>43.0</td>
<td>38.5</td>
<td>41.5</td>
</tr>
<tr>
<td>Enlarged dataset</td>
<td>46.8</td>
<td>67.2</td>
<td>71.8</td>
<td>59.9</td>
<td>61.4</td>
</tr>
<tr>
<td>Mean Accuracy Improvement</td>
<td>12.4</td>
<td>17.2</td>
<td>28.8</td>
<td>21.4</td>
<td>19.9</td>
</tr>
</tbody>
</table>

The recognition results using KNN classifiers are shown in Table 1. The four gestures (classes) ‘Alive’, ‘Call’, ‘Calm’ and ‘Dance’ were assigned labels from 1 to 4 respectively. We measured the average performance of the KNN on both datasets for each class. The classification accuracy is calculated over 7 runs and the overall average performance is obtained (Eq.4). Where ‘n’ is the number of classes and Ai is the accuracy per class at each run.

\[
\text{Mean Classification Accuracy} = \frac{1}{n} \sum_{i=1}^{n} A_i
\]  

(4)
As shown in Table 1, the mean average classification accuracy is improved with more than 19% after the incremental addition of data (i.e. enlarged dataset) for different gestures. For example the mean classification accuracy was increased with ~28% and ~21% for class3 and class4, respectively.

4. 3. Registered Vs. Un-registered signer

In this experiment, the classifier was first trained on the RS training set and then tested on the RS test samples. This test is performed to show that the classifier performs regularly well on previously known samples as shown in Table 2.

Table. 2. Overall accuracy for registered and unregistered signer

<table>
<thead>
<tr>
<th>Run</th>
<th>Training</th>
<th>Dataset size (gestures)</th>
<th>Testing</th>
<th>Recognition Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RD</td>
<td>4</td>
<td>RS</td>
<td>100%</td>
</tr>
<tr>
<td>2</td>
<td>RD</td>
<td>4</td>
<td>US</td>
<td>~ 23%</td>
</tr>
<tr>
<td>3</td>
<td>RD + Poser</td>
<td>12</td>
<td>US</td>
<td>~ 36%</td>
</tr>
<tr>
<td>4</td>
<td>RD + Poser</td>
<td>24</td>
<td>US</td>
<td>~ 56%</td>
</tr>
<tr>
<td>5</td>
<td>RD + Poser</td>
<td>24</td>
<td>RS “new tries”</td>
<td>from 76%: 89%</td>
</tr>
<tr>
<td>6</td>
<td>RD + Poser + New RD</td>
<td>24+5</td>
<td>US</td>
<td>~ 56%</td>
</tr>
</tbody>
</table>

In addition, the classifier was trained on the RS training set and then tested on the US test sample. This experiment shows how the recognition accuracy for the US is penalized to only 23% accuracy due to the involvement of the unsigned signers.

4. 4. Synthetic vocabulary enlargement

As shown in Table 2, the classifier was trained on both real and synthetic training sets and then tested by the RS and US test samples. This set of experiments indicates that the trained classifier on both types of data performed well on previously known gestures for the RS and US samples where the classification accuracy was increased to 76%. Furthermore, the classifier was trained on the additional synthetic gesture samples to examine how well the classifier performed when trained on extra synthetic gesture samples. The accuracy was enhanced to 89%.

Finally, it is shown that combining synthetic data enhances the accuracy in most cases. It never produced more errors, when tested with the real samples. Furthermore, a classifier trained with synthetic data regularly leads to enlarge the real training set and makes it more signer-independent.

5. Conclusions and Future Work

Experimental results have demonstrated that using synthetic data to enlarge training data has never worsened the mean recognition accuracy but in most cases improves the overall system performance. Experiments show some modest results of the mean recognition accuracy for the first two classes (table 1). A primary investigation showed that the synthetic data generated is not consistent with the real data and thus reduces the density of the class. This is the inverse effect of generating synthetic data. Second unpredictable result was found in experiment run number 6 (table 2). It’s hard to find out the reason why there was a reduction in the recognition accuracy while there was an increase in the previous runs. A rapid analysis has been done on the added real data. It seems that the difference in the signer shape and scene illumination have a great effecting on the system performance.

Although there were some drawbacks using proposed approach, but there are a lot of benefits too. The improvement happened in the dataset diversity was increased when generating and embedding more synthetic data. The way of generating synthetic data was more practical, affordable and consumes less time. In general, the experimental results backup the claim that using visual synthetic data can effectively improve the performance of the gesture recognition system.
In future work, further studies should be made to automate the generation of 3D animated models to resist its unnaturalness. Furthermore, additional investigation is needed to show that there might be an additional bias factor in our set of experiments to see if the trained classifier on synthetic data outperforms its current performance. More comprehensive experiments of a wider range of sign gestures might be used in the generation of our additional samples.

**Acknowledgements**
I thank Prof. M. Kamel for his guidance and support and the University of Waterloo Centre for Pattern Analysis and Machine Intelligence (UW PAMI) for financial support. Also, I would like to thank Péporté, Michèlle and her research group for the gesture files which have been used in our experiment.

**References**