An invariant-based approach to static Hand-Gesture Recognition

Pujan Ziaie
Technical University of Munich
ziaie@cs.tum.edu

Alois Knoll
Technical University of Munich
knoll@cs.tum.edu

Abstract

In this paper, a fast and robust approach for static hand gesture recognition is presented. The method is an invariant-based approach and avails Hu moments for this purpose. The invariants are classified by means of a modified K-Nearest Neighbors classifier called Locally Weighted Naive Bayes classifier. The Hu moments are obtained from the outer-contour of the observed hand and then classified by the LWNB classifier. Three types of gestures (pointing, grasping and holding-out) were used in practice under different lighting conditions with different users. Given the experimental results in the domain of the Joint-Action Science and Technology (JAST) project which is a Human-Robot interaction system, this method has a considerable performance of about 93% correct classification results on average.

1. Introduction

One of the major issues in the 21st century is, by far, the interaction between human and robot. Robots are getting more and more prevalent in every stratum of most developed societies. This is due to the fact that with new technologies, owing a robot for private purposes like house-cleaning is cheaper and more viable that ever before. Moreover, in many cases in which robots either need the supervision and direction of a human-being, the collaboration of humans with machines is inevitable. Machines, or specifically robots, require communication with people to receive and process the corresponding data to start a transaction or finish an assignment.

In some areas the interaction with humans is a must. In entertainment industry, for example, a good understanding of what people want is important. A robot for selling tickets, for instance, needs to communicate with people to see what they want and then carry out the corresponding task. Another example would be in bomb detection where the supervision of an expert is required to reduce the risk. Trying to address these needs, new methods nave been sought to ease the process of communication. Not every customer or specialist needs to know how to program a robot and insert the right instructions! Therefore, it is crucial to build up a natural way of interaction, so that robots can obtain the relevant data from the surrounding people. This process should be performed in a way that people would not need any special knowledge or protocol for this type of communication.

Considering this need, gestures are an indispensable part of every communication between humans. They are used for everything from pointing at a person to conveying specific information or implying a message. Researches indicate that gesturing does not only embellish spoken language, but is an essential part of the language generation process [5]. It happens very often that one cannot simply express his or her feelings or opinions without using additional gestures. Hand gestures, among other necessary domains in HRI, play an important role, both as an accompaniment to speech and as a means of input in their own right. This paper focuses on the task of static *hand-gesture recognition*, which is recognizing and classifying different hand shapes of a human user. For this purpose, hu-moments of the outer-contour of the observed hand are extracted and classified by using a modified K-nearest neighbors algorithm. The whole process is in the context of a cooperative human-robot assembly task.

2. Related Works

Numerous methods exist that have been developed recently to perform a successful gesture recognition. Most of these systems use model-based approaches, whereas some of them exploit invariant-classification methods. The invariant-based approaches consist of two main steps: Extraction of invariants and classification of gestures based on those invariants.

For hand-gesture recognition, some researchers have tried to perform the early segmentation process using skin-color histograms [10, 2, 9]. The problem with this approach is that they do not operate well in cases when there are some other objects in the scene with the same color as skin color, or where the hand has other colors than the per-defined one. In the target JAST application, the background is static and

ICAT 2008
Dec. 1-3, Yokohama, Japan
ISSN: 1345-1278

can easily be eliminated and therefore, the concentration can be mainly on the geometric characteristics of the objects.

Kuno and Shirai [4] defined seven invariants to do hand gesture recognition, including the position of the fingertip. This is not practical when we have not only pointing gestures, but also several other gestures, like grasping. However, the invariants they considered inspired us for our defined invariants.

Classification is a method to assign a class to a point (or vector in spaces of more than one dimensions) in an N-dimensional space. The classes may be predefined and learned beforehand (supervised learning), or may be extracted automatically based on a similarity metric (unsupervised learning).

K-nearest neighbors (KNN) classifiers has a good performance when the attributes of a system are linearly separable. It finds the K nearest (already classified) vectors to the input vector. The class which most vectors in those K neighbors belong to is chosen to be the right class of the input vector. K-nearest neighbors with distance weighting (KNNDW) is an improvement to the regular KNN algorithm which has been proved to perform better than KNN in numerous aplications [6]. In KNNDW, the contribution of each neighbor to the overall classification is weighted by its distance from the point being classified. The classes are then assigned with a likelihood value based on a simple naïve Bayes approach.

The most relevant work to our classification approach issued in this paper has been performed by Frank *et al.* [1] which introduces a locally weighted naïve Bayes (LWNB) classifier. Their evaluation shows that LWNB outperforms KNN and KNNDW when K is big enough.

To increase the performance of the classification, sometimes a combination of two invariant classes or two classification methods [8] is used along with Bayes probability theory. By combining the results of two classifiers a better performance is achieved without manipulating the training data or any complicated modification. We will try to enhance the quality of the recognition by using such an approach in the future works.

3. Gesture Recognition Approach

3.1. Invariants

The invariants used for classifying gestures are the first six Hu-moments which are extracted from the outer-contour of the hand performing a gesture. Once the regions of interest (ROI) have been identified as described in the preceding section, the next step is to extract some meaningful geometric invariants from the binary image to be used for the classification.

Hu moments [3], are scale, translation and rotation invariant. Hu derived these expressions from algebraic invariants applied to the moment generating function under a rotation transformation. In this work, only the first six moments are being used, because the seventh moment, which is the skew-invariant one, apparently adds no values to the recognition results.

3.2. Training the Classifier(s)

Before performing classification a training pool should be created for each of the invariant-classes. It is of great importance that the data would be produced by different users under various lighting conditions to increase the robustness. Each training instance is labeled with its corresponding gesture type. The gesture types are defined as:

$$\vec{C} = \{c_1, c_2, \ldots, c_Z\} \tag{1}$$

where Z is the total number of gesture types and is *three* in this application. Extending the gestures can be simply done by adding the corresponding training data to the pools.

Assuming there are N vectors in the training pool (N samples), each vector will be defined as:

$$\begin{aligned} &Inv_n^h(l) = \{hu_0, hu_1, \ldots, hu_L\} \\ &hu_0 \subseteq C \\ &l \in 1..L \& n \in 1..N \end{aligned} \tag{2}$$

where Inv^h represents the vector of hu-moments and the first element(hu_0) correponds to their class label.

After constructing the pool of labeled vectors, classification can proceed.

3.3. Classification: LWNB

In our application, the well-known K-nearest neighbors algorithm is used as our classifier, with two modifications. First, before performing the classification, the elements of the given vector (the invariants) are being weighted based on their influence on the process. The proper weights have been extracted off-line based on empirical empirical findings.

The second modification is after finding the K nearest neighbors. Instead of simply calculating the distance of each vector in the space, a weight is assigned to each node based on its distance to the input vector. The probability of a class is then extracted based on the weights of that class in the first K neighbors.

We can define the distance-weighting vector as:

$$\vec{whu} = \{whu_1, whu_2, \ldots, whu_L\} \tag{3}$$

The distance from the input invariant-vector in^h to the nth training-node in the training pool can then be computed in

Euclidean space as:

$$dist(Inv_n^h, in^h) = \sqrt{\sum_{l=1}^{L} \frac{(Inv_n^h(l) - in^h(l))^2}{whu_l}} \quad (4)$$

the distance is indeed normalized, so that all the values be between 0 and 1.

In the next step, K^f defined invariant vectors and K^h Hu-invariant vectors with the shortest distance from their respective input vector are selected from the training pools ($\vec{Inv}^f{}_n$ and $\vec{Inv}^h{}_n$) for both invariant-classes. These selected vector is $s\vec{Inv}^h{}_n$.

$$s\vec{Inv}^h{}_{kh} = \{s\vec{Inv}^h{}_1, \ldots, s\vec{Inv}^h{}_{K^h}\} \quad (5)$$

$$s\vec{Inv}^h{}_{kh} \in \{\vec{Inv}^h{}_1 \ldots \vec{Inv}^h{}_N\} \quad (6)$$

$$kf \in \{1, \ldots, K^f\}$$

The first elements of each of the vector,$sInv_{kh}^h(1)$ is, in fact, the label of the class it belongs to (according to 3). Hence having c^h as the variables corresponding to classes as:

$$c^h(kh) = \{sInv_{kh}^h(1)\} \quad (7)$$

To take advantage of the effect of distances and improve the result, we add weights to the selected nodes (neighbors). Considering each node as t, this weight $wB(t) = f(ds_t)$ is a function of the already computed Euclidean distance ds_x of each node and can be any monotonically decreasing function. According to the experimental results, Hu invariant-vectors result in a maximum of 93 percent of correct classification.

3.4. Localization of Gestures

Once the class of the gesture has been selected as described above, the next step is to identify the attributes corresponding the gestures. For each gesture, there is a location that is most important for determining the meaning of the gesture in the context of the system.

For the holding-out gesture, all that is needed is the center of the hand, which is very easy to obtain. However, for pointing and grasping gestures, it is more complicated: for pointing gestures, what the system needs is the position of the finger-tip plus the angle of the pointing-finger.

For grasping gestures, it is the position between two grasping fingers. In the following sections the approach to locate these positions in these types of gestures is described.

The gesture localization-attributes can also be sought in the ROI by doing template-matching as well [7]. However it is computationally of more advantage to extract them based on the gradients points (edges).The results (see figures 1,2,3) are precise enough and absolutely satisfactory for the JAST application.

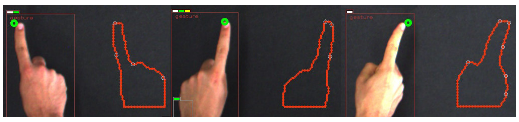

Figure 1. Finding the Position of the Fingertip in Pointing Gesture

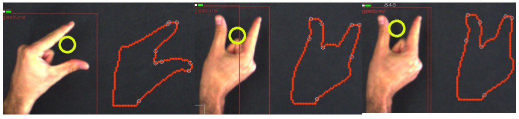

Figure 2. Finding the Position of Grasping in Grasping Gesture

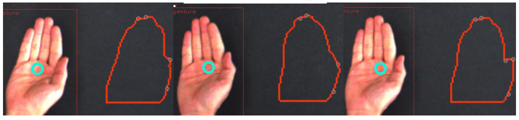

Figure 3. Finding the Center of the Palm in Holding Gesture

4. Experimental Results

After trying several weighting combinations for Hu-invariants as input vector, a performance of 93.46% correct classification was achieved, with a weighting vector of $whu = [443411]$ which is again intuitively reasonable, putting a higher weight on the first four moments.

The results, with and without weighting the moments, can be viewed in figures 5 and 4.

The table 1 shows the performance of the method for some random weighting vectors.

Hu	K^h
92.94%	6
92.94%	5
92.94%	5
92.98%	5
92.98%	5
93.46%	5
92.98%	5
92.98%	5

Table 1. Performance results of Hu invariants for different K

By following this approach, almost 93% correct recognition occures on average.

Running the full gesture-recognition process on a frame takes less than 15 msec on average (usually between 14-16msec). Of this time, segmentation takes about 11 msec, while the (gesture) recognition process along with its autonomous segmentation module takes approximately 4-6 msec.

5. Conclusion and Future Work

Using LWNB as classifier together with *Hu* invariants along with apt weighting values results in a performance of more than 93.0% correct recognition for our three defined gestures.

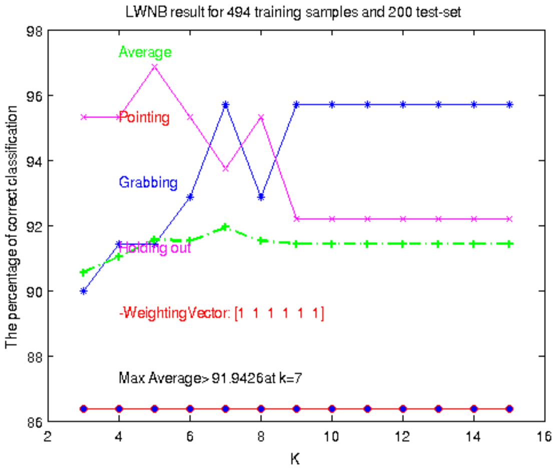

Figure 4. Recognition Performance Without Weighting(92.00%)

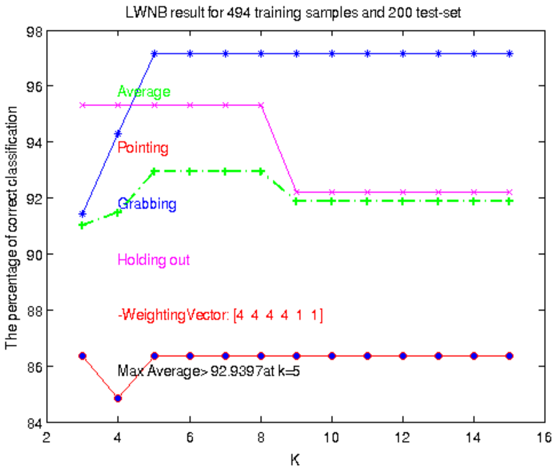

Figure 5. Recognition Performance With Weighting (93.00%)

To achieve these results, a testing pool with about 200 samples was constructed for all of the gestures (roughly 70 each), for a total of 500 samples in the training pool. The training data were made by five persons (three boys and two girls) in different lighting conditions.

The results of the classifier given different weighting vectors (with and without weighting) was demonstrated for different values of Ks. The X axis of these graphs represents the value of the corresponding K for the K-nearest neighbor selection, while the Y axis shows the percentage of gesture instances of each type that were correctly recognized.

We are considering adding up a new class of invariants to the system and then combine the results of Hu-moments with the other invariant-set to enhance the quality of the recognition. We are contemplating combining the results of the given two invariant-classes based on a Bayesian system. For this purpose, a likelihood will be assigned to the result of each invariant-class and then the proper gesture will be chosen based on the highest likelihood of a gesture. This approach will hopefully increase the correct recognition rate and also generates results with less fluctuation.

Another point for future works is to add up more gesture types. This is important to see how this system works in an environment with 5 or more types of gestures.

6. Acknowledgement

This research was supported by the EU project JAST (FP6-003747-IP), http://www.euprojects-jast.net/.

References

[1] E. Frank, M. Hall, and B. Pfahringer. Locally weighted naive bayes. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, pages 249–25, San Francisco, CA, 2003. Morgan Kaufmann. 2

[2] H. Hongo, M. Ohya, M. Yasumoto, and K. Yamamoto. Face and hand gesture recognition for human-computer interaction. In *Proc. IEEE 15th Int. Conf. Pattern Recognition*, volume 2, pages 921–924, 2000. 1

[3] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Information Theory*, 8:179–187, 1962. 2

[4] K. Kuno and Y. Shirai. Manipulative hand gesture recognition using task knowledge for human computer interaction. In *FG '98: Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, page 468, Washington, DC, USA, 1998. IEEE Computer Society. 2

[5] D. McNeill and E. Levy. *Conceptual Representations in Language Activity and Gesture*. John Wiley and Sons Ltd, thirteenth edition, 1982. 1

[6] R. L. Morin and B. E. Raeside. A reappraisal of distance-weighted k-nearest neighbor classification for pattern recognition with missing data. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(3):241–243, 1981. 2

[7] R. O'Hagan and A. Zelinsky. Finger track - a robust and real-time gesture interface. In *AI '97: Proceedings of the 10th Australian Joint Conference on Artificial Intelligence*, pages 475–484, London, UK, 1997. Springer-Verlag. 3

[8] Y. Weiss and E. H. Adelson. Slow and smooth: A bayesian theory for the combination of local motion signals in human vision. Technical Report AIM-1624, Massachusetts Institute of Technology, 1998. 2

[9] H. Wu, T. Shioyama, and H. Kobayashi. Spotting recognition of head gestures from color image series. In *ICPR '98: Proceedings of the 14th International Conference on Pattern Recognition*, volume 1, page 83, Washington, DC, USA, 1998. IEEE Computer Society. 1

[10] H. Zhou, D. J. Lin, and T. S. Huang. Static hand gesture recognition based on local orientation histogram feature distribution model. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop*, page 161, 2004. 1