Sparse Representation for Gesture Recognition

Arthur Pajot
ENS De Rennes

Adrien Luxey
ENS De Rennes

***Abstract*—This document presents our group's research work on the subject of sparse representation applied to gesture recognition. This representation primary aims at reducing the dimensionality of gestural inputs. The obtained results are promising: they show that sparse representation offers valid answers to the challenges brought by gestural recognition, among which the segmentation of the movement into meaningful gestures. Finally, it provides solid solutions to the input variability issue.**

I. Introduction

The perpetual improvements of computers' processing power and of machine learning techniques have allowed the development of news areas of Human Computer Interaction (HCI). Gestural recognition is one of these. It already has applications in the fields of video gaming and computer assisted surgery, to cite a few. But a lot of challenges are still to be overcome before computers can successfully interpret complex gestures.

A. Gesture recognition

Gesture recognition is a process that aims at extracting meaningful information from a 3D human motion stream, which can be acquired with a lot of different sensors, such as accelerometers and gyroscopes (for example a Nintendo Wii-mote), or motion-tracking cameras (like a Microsoft Kinect device). The meaningful parts in a motion stream are called *gestures*, and must be interpreted by the computer as orders. Different processes are involved here: *segmentation*, that is recognizing a gesture inside a motion stream, and *classification*, that is differentiating types of gestures (a collection of gestures is shown in Figure 5), resulting in different kinds of orders.

Several issues make gestural recognition very complex to study. Firstly, the *high dimensionality* of the input: a point in space is represented by a 3D vector (x, y, z) and motion sensors provide a minimal acquisition frequency of 60 Hz, leading to a 120 by 3 vector to process for a gesture that would only last 2 seconds. Secondly, gestures are prone to a very high degree of *variability*: inter-users, meaning that two different persons will not perform the same gesture the same way, and intra-user, that is the same person will never reproduce the same exact gesture twice. Thirdly, *segmentation* is a challenge on its own, because the computer has to determine whether the user is trying to accomplish a gesture, or if he is just making irrelevant moves.

Some algorithms already bring satisfying solutions for both the segmentation and classification of 3D gestures [1] [2], but they remain very specific to their application, and do not provide a compliant basis that could be extended as the application needs evolve.

B. Sparse representation

The approach that we will now present tries to answer these issues in a novel way, by representing every gesture as a composition of elementary gestures, called *atoms*, precomputed and stored in a dictionary. Thus, a gesture will be represented as a set of atoms associated with various meta-data: a *sparse representation* of the original movement.

The generation of such a dictionary of atoms is a supervised machine learning process. It is composed of two main steps: the creation of the dictionary from a *training set* (a set of gestures of which we know the types), and the representation of any new gesture with our dictionary. To be efficient, a dictionary should be composed of the least possible number of atoms, and still provide a good sparse representation (a measure for a representation's fitness is presented below).

Because we want a minimal number of atoms, they should be usable for a gesture representation regardless of their position, rotation, and scale. Quentin Barthelemy, in his thesis on the sparse representation of multivariate signals [3], addresses this issue by introducing several techniques to provide atoms rotation and scale *invariants*. To handle the atom's shift in a gesture, we create, in the dictionary, one atom per possible position of it. This way, a single line atom, saved at different shifts, could be efficient in the representation of any kind of Swipe gesture (see Figure 5), for example.

So, given D a dictionary of atoms ϕ and a gesture G:

$$D = [\phi_m \in \mathbb{R}^{N\times 3}]_{m=1}^{M}, G \in \mathbb{R}^{N\times 3}$$

The so called *Matching Pursuit* (or MP) algorithm will produce g, the sparse representation of G, composed of a set of *patterns* ψ, containing, for each atom used in the sparse representation, its rotation matrix R and scale x:

$$g = \text{MP}(D, G) = [\psi_p]_{p=1}^{P} \text{ where } \psi = [x, \phi, R]$$

From g, we can build the reconstruction G' of G, leading to a residual (also called reconstruction) error ϵ:

$$G' = \sum_{p=1}^{P} x_p \phi_p R_p \in \mathbb{R}^{N\times 3} \rightarrow G = G' + \epsilon$$

Such a representation, empowered by a good dictionary, should be able to represent any kind of gesture sparsely and efficiently. The classification and segmentation using this representation still need to be studied before sparse coding can be used as a production solution, which was the topic of our research. In Section II, we will present the classification of pre-segmented gestures using our approach, the topic on which we focused the most; Section III will then present our attempts to segment a movement stream into meaningful gestures; finally, we will conclude our article in Section IV.

II. Gesture classification

Classification is the problem of identifying to which of a set of categories a new observation belongs. Mathematically, it comes back to finding the closest group of elements to which a data vector might belong in a space with many dimensions. In the field of gesture recognition, different approaches already exist to tackle the problem, providing satisfying results, but they remain very specific to their application. The sparse representation offers promising results to extend the capacity of the other approaches.

A. General classification process

Classification is a very common task in computer sciences, notably in data mining. A simple and yet efficient algorithm is *k*-NN (for k Nearest Neighbours).

In a N-dimensions space, provided with existing labelled points (the training set), the algorithm locates the k nearest neighbours of the new point that we want to classify, and sets its category to which is the most frequent among its neighbours. In Figure 1, the test point is the green circle. If $k = 3$, it will be set to the red triangles category; if $k = 5$, it will be set to the blue squares one.

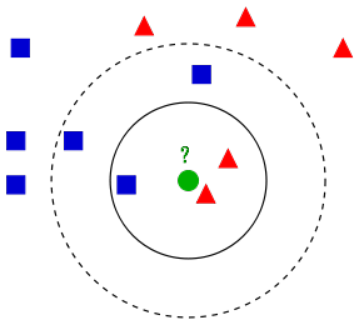

Fig. 1. An example of a KNN search with k=3 (solid line) and k=5 (dashed line) in a 2D space

To be efficient, such an algorithm must be provided with a training set where the points of the same category are as close to each other as possible. Besides, classical classifiers like *k*-NN can take as input any kind of data vector, whatever their size.

B. Existing approaches

The most straight-forward approach we could think of is called Pattern Matching: by associating a typical gesture (the pattern) to each category, one could try to match a new gesture with each pattern, and conclude that it belongs to the category to which it fits the most. The crucial point, here, is to find an efficient fitness operator. Because of the input's variability, Pattern Matching is very error prone, and is only acceptable for single-user applications with few very distinguishable categories.

Another solution, with far better results, is to use features. Features are numeric values, extracted from a gesture, that provide meaningful information on it. We could think of the movement amplitude, its duration, extrema of speed, acceleration, curvature, or any other value that could give relevant information on the gesture. In the end, the movement is only represented as a vector of features, which can be used as-is in a statistical classifier (like *k*-NN), or passed to other classification algorithms. This approach has been covered by Chen et al. in [2], with both a statistical classifier and a Hidden Markov Model (HMM) classifier [4]. While this kind of solution can provide excellent recognition rates, it depends entirely on the choice of the features. For this reason, it is still very application-specific.

C. Classification using sparse representation

Gesture classification using sparse representation can be solved by two main approaches [5]:

A first approach, called the *multi dictionaries* approach, consists in using several dictionaries to recognize a gesture, as shown in Figure 2. At first, we build a different dictionary for each category using the training set. To classify a new gesture, we compute its sparse representation with each dictionary and compare each reconstruction error. In the end, the selected category is the one for which this error is minimal.

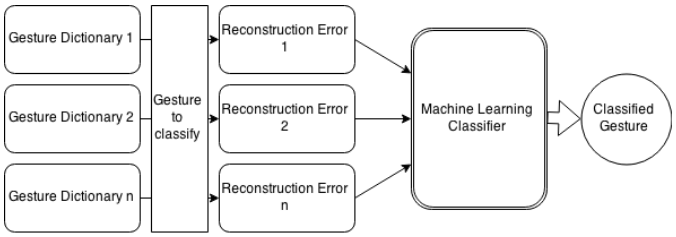

Fig. 2. The multi dictionaries approach workflow

Another approach, called the *single dictionary* approach (see Figure 3), consists in using a single dictionary for all the gestures: first, we build a dictionary for each gesture set, then, we combine the atoms of each dictionary into a single one. We can now classify gestures by looking at the patterns of their reconstruction, instead of using only the reconstruction error. In addition, the Matching Pursuit algorithm will only be called once, and not for each gesture type.

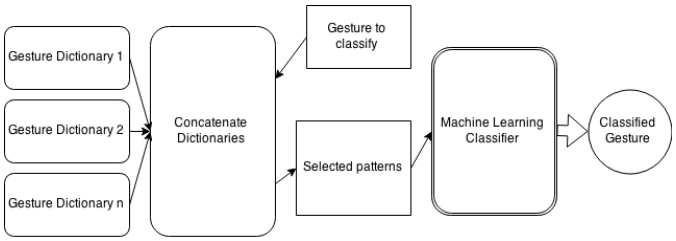

Fig. 3. The single dictionary approach workflow

The first multi dictionaries approach used to select the category which had the dictionary with the smallest reconstruction error, but it occurred that the decision rules were often more complicated.

As we can see on Figure 4, the error rates probabilities tend to overlap, even though the first error curve should correspond to the gesture we try to recognize, which makes the classification error-prone.

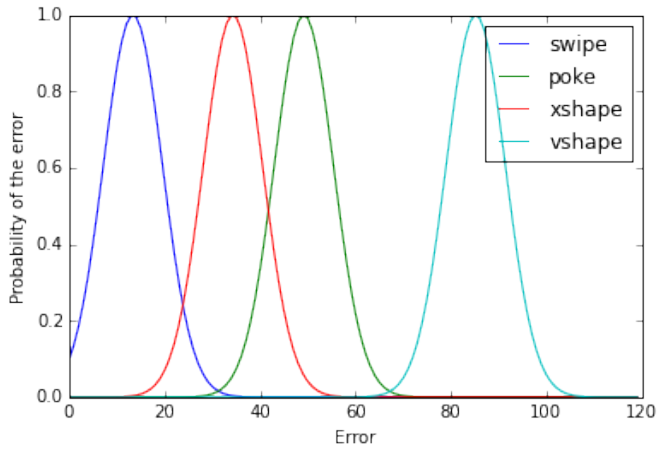

Fig. 4. Repartition of the reconstruction errors of a Swipe gesture set against all the dictionaries

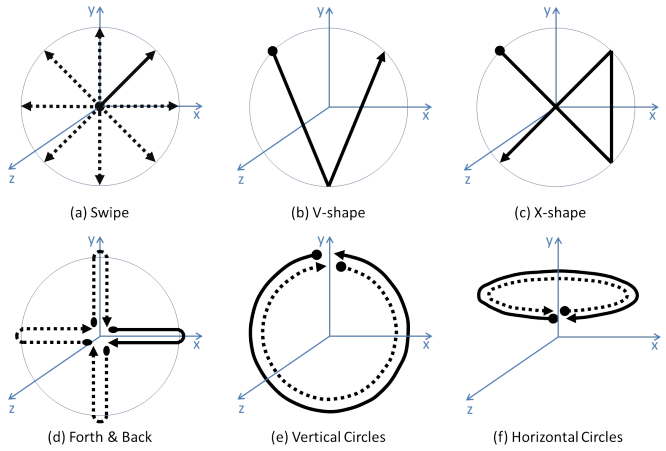

Fig. 5. The different types of gestures in the 6DMG database [2]

Our idea was to help the algorithm make a decision, by giving him other features than just the smallest error. Using a supervised learning algorithm like *k*-NN, and the vector of the recognition error of our gesture against each dictionary as the classifier input, the decision rule takes into account the relationship between the errors against each dictionary.

However, the multi dictionaries approach was not quite satisfactory, as it requires the building of a dictionary per gesture type and as many Matching Pursuit runs to compute the recognition errors. On the other hand, the single dictionary approach is far more interesting, as it takes as a classifier input all the data from the patterns (atom's label, rotation matrix (or quaternion), and scale), which makes it more robust and flexible. A new gesture type could be incorporated to the classifier without the need to rebuild the dictionary: even though the reconstruction might not be optimal, the patterns signature might still be unique. This makes the single dictionary approach extendible, and thus, far more independent from the application context, and close to the sparse representation goal.

D. Results

In this section, we present our comparison of the two approaches, that we implemented and tested with the Matlab environment.

To test gesture classifiers without the need to bother about the segmentation issue, some researchers have compiled some pre-segmented and labelled gesture data sets, such as the 6DMG (for 6D Motion Gesture) database, provided by Chen et al. [2], that we used extensively. Furthermore, it has been widely used in research papers since it came out in 2013, which makes it a very good comparison database.

In 6DMG, the positions of the users' hands are captured with an optical tracking system at 60 Hz, coupled with a Nintendo Wiimote to gather additionally the acceleration and orientation. As we can see it in Figure 5, the database defines a total of 16 gestures, including swiping motions in eight directions, poke gestures that swipe rapidly forth and back in four directions, a V shape, an X shape, and clockwise and counter clockwise circles in both vertical and horizontal planes. Because sparse representation is rotation invariant, we started by merging all the rotated gesture datasets, resulting in 5 gesture type: swipe, V-shape, X-shape, poke (forth & back), and circle. Each gesture being performed several times by different users, we can efficiently test the inter-users and intra-user adaptability of our framework.

In a classification task, the quality measure for a class is the number of true positives (i.e. the number of items correctly labelled as belonging to the class) divided by the total number of elements labelled as belonging to the class, in percent.

Our experimental protocol is the following: we randomly select gestures to split them into two groups: a train dataset, and a test dataset. We use the train dataset to build the dictionary (or the dictionaries) and use the test dataset to test the classification.

To ensure that our results were truthful, we applied a k-fold cross validation protocol on our data. Cross-validation is a model validation technique for assessing how the results of a classification task will generalize to any dataset. In k-fold cross-validation, the original sample (the gestures database) is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining $k-1$ subsamples are used as training data.

Gesture	Minimum	Multiple	Unique	Rotation
Poke	98.47%	100%	100%	99.38%
Swipe	99.45%	99.4%	99%	98.62%
X-shape	95.56%	96.1%	93.3%	97.24%
V-shape	98.33%	98.9%	94.4%	95.58%
Circle	99.72%	100%	99.4%	98.47%

TABLE I. CLASSIFICATION RESULTS WITH OUR FOUR CONFIGURATIONS

We have tested our data sets in 4 configurations, with the results shown in Table I. The first one (labelled **Minimum**) is the one described by Duccoffe in [5]: we built several test samples sing the k-fold cross validation protocol, and chose the gesture type whose dictionary had the smallest representation error. In the second configuration (labelled **Multiple**), we did the same as in Minimum, but added a machine learning

classifier with the error vector as input. The third configuration (labelled **Unique**) implemented the single dictionary approach without the rotation matrices. Finally, the last configuration (labelled **Rotation**) used the unique dictionary approach with the 5 gestures and the rotation matrices.

Those results are not perfect: the 6DMG database has only 5 different gestures, which might be too small to test the potential of the unique dictionary approach. Furthermore, such a database comes with noiseless and pre-segmented gestures, which does not correspond to a production reality.

Despite that fact, the results still allowed us with interesting observations. The first one is that for 5 gestures, the addition of a machine learning classifier to the multiple dictionary approach improves the results, which seems perfectly logical, as it is the purpose of machine learning algorithms to solve that kind of problems. We even reach 100% for the poke and the swipe, which is quite impressive. The second observation is that the unique dictionary approach (without the orientation) does not perform better than the multiple dictionaries. We can emit the hypothesis that it is due to the fact that the reconstruction errors adds more significant information on the data than the parts of the patterns we gave to the algorithm.

However, when we add the rotation of the gestures to the unique dictionary approach, the score is again very high. For the X-shape and the V-shape, which were only performed in one orientation, the rotation matrix adds information, as the classification rate is better. Thus, the unique dictionary approach seems more suited for complex gesture alphabet.

To conclude on this part, we can state that sparse representation seems to be a very promising approach for gesture recognition. Indeed, it drastically reduces the dimensionality of segmented gestures, while providing very decent features for classification (as reads the Rotation column of Table I), empowering classification with a very good execution speed, once a good dictionary has been crafted. It also shows interesting results in the variability issue, wiping out the distinction between rotations of the same gestures (while providing a rotation matrix on which we could also work to differentiate a left swipe from a right one, for example). Still, this approach is very new, and still needs a lot of work before it can achieve the ambitions it carries. Optimizing the differentiation between the patterns of different gestures (to improve classification), allows the extension of a learning algorithm with new gestures by crafting a stronger dictionary and can be a very interesting extension of our algorithm.

III. Gesture Segmentation

Our two approaches, as they efficiently solve the dimensionality issue, can be used to solve the other big problematic of gesture recognition: segmentation. It consists in cutting a complex motion stream into the only relevant parts: the gestures. Compared to a pre-segmented database as 6DMG, gesture intentions might be noisy, poorly done, or even a combination of the expected gestures.

This part is certainly the hardest to overcome in the vast topic of gesture recognition. Indeed, an algorithm that could recognize effortlessly complex successions of gestures, or even a combination of different ones (a swipe and a circle, for example) would allow a really seamless interface between the human and the computer. Alas, at the time of writing, the most efficient way to segment a gesture is still to add a button to the user controller, so that he can tell when his gesture starts, and when it ends. Not quite impressing.

A. Approach

The idea of our approach is to use a supervised learning classifier to recognize whether a segment of the motion stream is a gesture or not. In order to achieve this task, we add a new prediction class to our machine learning system: the zero class, that is non-recognized gestures. To recognize a gesture as a non-significant move, we look at the probability that a new gesture belongs to each category. If this probability is below 50% for every gesture type, then, we consider the studied gesture irrelevant.

In order to segment gestures, we have to define how to extract potential gestures for the stream. A naive approach is to create a moving window, moving along the input data and calculating the probability for each segment to be a gesture. If this probability is high enough, then we have a segmented gesture. The problem with this approach is that we have a fixed window size for different gestures length. It does not work.

Thus, we have chosen another approach: determining key points in the motion stream, such as a brutal change in direction or velocity. We then compute the gesture probability for every segment between two key points that has a credible length (a gesture should not last less than 1/2 second nor more than 2 seconds). Finally, we only keep the segments that have been recognized as proper gestures. The advantage of this approach is that we can see if a gesture is composed of several sub-gestures (a poke is the combination of two swipes in opposite direction, for example), and thus segment more complex gestures.

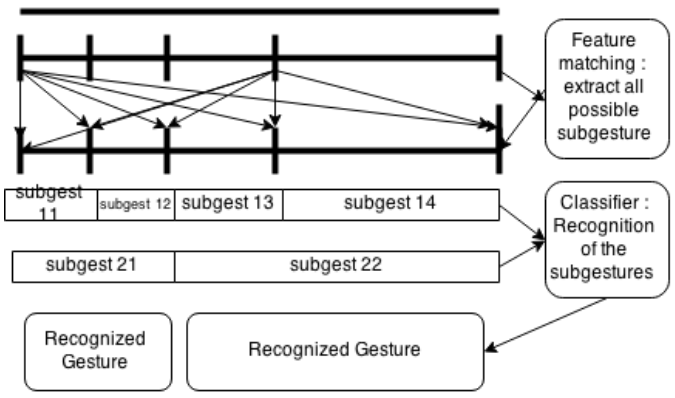

Fig. 6. Segmentation workflow

B. Results

Testing our segmentation framework has been quite challenging. The first challenge was the lack of a pre-definite dataset. Because segmentation is still a prominent issue, there is no dataset or metric to compare our results with a different method. We had to come up with our own test dataset. Unfortunately, creating an artificial motion stream to segment is complicated. We came up with two methods: a concatenation

of 6DMG gestures, separated by credible noise (a white noise would have been unrealistic), and real-life experiments generated by us with a Razer hydra (a magnetic device, which can record 3D positions while being held like a Nintendo Wiimote).

We tested our framework on those gestures. Because we lack proper experiment metrics, we cannot provide quantitative results. However, the 6DMG concatenated gestures were successfully recognized by a single dictionary trained with the same types of gestures. Still, we cannot really believe our results, as the separation between gestures in the stream was crafted by ourselves. A white noise separation would not be realistic, while drawing a line between two gestures could be misinterpreted as a swipe.

Our experiments with the Razer Hydra were not conclusive either, because the key points rising was hard to fine-tune. If we put too much sensitivity, key points could be detected due to measure noise. If we put less sensitivity, our gestures beginning and ending were then hard to catch for the algorithm, leading in a poor recognition rate. Still, we managed to interpret some of our moves correctly, so we believe that such an approach could give promising results, if studied thoroughly. In particular, we believe that the detection of gestures and sub-gestures could provide a very good starting point to recognize complex gestures (a X gesture can be interpreted as 3 consecutive swipes).

IV. Conclusion

During this research project, we have experimented different methods to apply sparse representation to gesture recognition, working on the two main aspects of this topic: classification, and segmentation. Our results were not always positive, but, it appears that this approach is really innovative, and that it could empower gesture recognition with brilliant solutions to the difficult challenges it embodies.

Classification, on which we focused the most, has already given very promising results, with the recognition rate of certain gestures (swipe and poke) rising up to 100%, despite the multi-users component of our test database. This rare result proves that sparse coding is a right way to follow to get rid of the issue of the variability of the input, besides the fact that it already solves the dimensionality one.

Despite these results, it is clear that sparse representation of the movement is still a work in progress, as a lot of aspects are still to be covered: the generation of dictionaries could be critically enhanced, for example. At this time, the atoms contained in the dictionary are more than half the length of a gesture. Their atomicity could be greatly enhanced by reducing this size, but it brings out a lot of other issues.

Where the rubber hits the road is obviously on the segmentation part. This problematic is really the hardest to overcome in gesture spotting. We believe that our key points approach could offer promising results, but the lack of a common research ground on this point can only be a brake. Most industrial gesture recognition frameworks today use the application's context (in a video game, an algorithm can guess when the player is supposed to accomplish a particular gesture) or a push-button to segment pertinent gestures. A lot of research would be needed to perfect the estimation of key points in a motion stream. The choice of features, the adaptivity of the gesture window, there would be a lot to enhance.

Acknowledgement

The present work benefited from the precious help of F. Arguelaguet, post-doctoral fellow at the Hybrid Team at INRIA Rennes, who presented us with this very interesting subject, and provided us with a valuable assistance during our research summarised here. It also benefited of quality work of M. Ducoffe who provided the background of our work.

The authors would also like to thanks the IRISA and the ENS of Rennes for making our research possible.

References

[1] F. G. Hofmann, P. Heyer, and G. Hommel, "Velocity profile based recognition of dynamic gestures with discrete hidden markov models," in *Gesture and Sign Language in Human-Computer Interaction*, ser. Lecture Notes in Computer Science, I. Wachsmuth and M. Fröhlich, Eds. Springer Berlin Heidelberg, 1998, vol. 1371, pp. 81–95. [Online]. Available: http://dx.doi.org/10.1007/BFb0052991

[2] M. Chen, G. AlRegib, and B.-H. Juang, "Feature processing and modeling for 6d motion gesture recognition," *Multimedia, IEEE Transactions on*, vol. 15, no. 3, pp. 561–571, April 2013.

[3] Q. Barthelemy (Barthélemy), "Représentations parcimonieuses pour les signaux multivariés," Ph.D. dissertation, 2013, thèse de doctorat dirigée par Mars, Jérôme Sciences de l'univers Grenoble 2013. [Online]. Available: http://www.theses.fr/2013GRENU008

[4] L. R. Rabiner, "Readings in speech recognition," A. Waibel and K.-F. Lee, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, ch. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pp. 267–296. [Online]. Available: http://dl.acm.org/citation.cfm?id=108235.108253

[5] M. Ducoffe, "Dictionary learning for sparse gesture representation," Master's thesis, ENS Rennes, 2014.