Automatic human action recognition in videos by graph embedding

Ehsan Zare Borzeshi, Richard Xu, and Massimo Piccardi

School of Computing and Communications
Faculty of Engineering and IT
University of Technology, Sydney (UTS)
Sydney, Australia
{ezarebor,ydxu,massimo}@it.uts.edu.au
www.inext.uts.edu.au

Abstract. The problem of human action recognition has received increasing attention in recent years for its importance in many applications. Yet, the main limitation of current approaches is that they do not capture well the spatial relationships in the subject performing the action. This paper presents an initial study which uses graphs to represent the actor's shape and *graph embedding* to then convert the graph into a suitable feature vector. In this way, we can benefit from the wide range of statistical classifiers while retaining the strong representational power of graphs. The paper shows that, although the proposed method does not yet achieve accuracy comparable to that of the best existing approaches, the embedded graphs are capable of describing the deformable human shape and its evolution along the time. This confirms the interesting rationale of the approach and its potential for future performance.

Keywords: Graph edit distance, Graph embedding, Object classification

1 Introduction and related work

Human action recognition has been the focus of much recent research for its increasing importance in applications such as video surveillance, human-computer interaction, multimedia and others. Recognising human actions is challenging since actions are complex patterns which take place along the time. Due to the nature of human physiology and the varying environmental constraints, each person performs the same action differently for every instance. More so, different people may perform the same action in a pronouncedly different way in both spatial extent and temporal progression. When translated into feature vectors, actions give place to a probing feature space with very high intra-class variance and low inter-class distance. To mollify this issue, this paper presents an initial study into the possibility of using *graph embedding* for obtaining a more suitable feature set.

Many approaches have been proposed for human action recognition to date, including bag of features [1] [2], dynamic time warping [3], hidden Markov models [4] and conditional random fields [5]. A recent survey has offered a systematic review of these

approaches [6]. However, the problem of a suitable feature set which can well encapsulate the deformable shape of the actor is still partially unresolved. Graphs offer a powerful tool to represent structured objects and as such are promising for human action recognition. Ta *et al*. in [7] have recently used graphs for activity recognition. However, to assess the similarity of two instances, they directly compare their graphs which is prone to significant noise. An alternative to the direct comparison of action graphs is offered by graph embedding: in each frame, the graph representing the actor's shape can be converted to a finite set of distances from prototype graphs, and the distance vector then used with conventional statistical classifiers. Graph embedding has been successfully used in the past for fingerprint and optical character recognition [8]. To the best of our knowledge, this is the first work proposing to employ graph embedding for human action recognition. Such an extension is not trivial since feature vectors need to prove action-discriminative along the additional dimension of time. In this paper, we propose to extract spatial feature points from each frame and use them as nodes of a graph describing the actor's shape. With an adequate prototype set, we convert the graph to a set of distances based on the probabilistic graph edit distance (GED) of Neuhaus and Bunke [9]. Probabilistic GED is a sophisticated edit distance capable of learning edit costs directly from a training set and weigh each edit operation individually. The feature vectors of each frame are then composed into a sequence and analysed by means of a conventional sequential classifier. The recognition accuracy that we obtain is not yet comparable with that of the best methods from the literature; however, results show unequivocally that the embedded vectors are capable of representing the human posture as it evolves along the time and set the basis for potential future improvements.

The rest of this paper is organised as follows: Section 2 provides a brief recall of graph embedding. Section 3 describes the methodology proposed by this paper to incorporate graph embedding into an action recognition approach. Section 4 presents and discusses an experimental evaluation of the proposed approach on the challenging KTH action dataset. Finally, we give concluding remarks and a discussion of future work in section 5.

2 A brief recall of graph embedding

Based on various research studies, different definitions for graphs can be found in the literature. In this work we use an *attributed graph* g represented by $g = (V, E, \alpha, \beta)$ where

- $V = \{1, 2, ..., M\}$ is the vertices (nodes) set, where $M \in \mathbb{N} \cup \{0\}$,
- $E \subseteq (V \times V)$ is the set of edges,
- $\alpha : V \rightarrow L_V$ is a vertex labeling function, and
- $\beta : E \rightarrow L_E$ is a edge labeling function.

Vertex and edge labels are restricted to fixed-size tuples, ($L_V = \mathbb{R}^p$, $L_E = \mathbb{R}^q$, p, $q \in \mathbb{N} \cup \{0\}$).

With a graph-based object representation, the problem of pattern recognition changes to that of graph matching. One of the most widely used methods for error-tolerant graph matching is the graph edit distance (GED). It measures the (dis)similarity of arbitrarily

structured and arbitrarily labeled graphs and it is flexible thanks to its ability to cope with any kind of structural errors [10], [11]. The main idea of the graph edit distance is: find the dissimilarity of two graphs by the minimum amount of distortion required to transform one graph into the other [10]. In the first step, the underlying distortion models (or edit operations) are defined as insertion, deletion and substitution for both nodes and edges, $(e_1, e_2, e_3, e_4, e_5, e_6)$. Based on this definition, every graph can be transformed into another by applying a sequence of edit operations (i.e. an edit path). Then, given a set of edit operations and an edit cost function, the dissimilarity of a pair of graphs is defined as the minimum cost edit path that transforms one graph into the other. Let $g_1 = (V_1, E_1, \alpha_1, \beta_1)$ and $g_2 = (V_2, E_2, \alpha_2, \beta_2)$ be two graphs. The graph edit distance of such graphs is defined as:

$$d(g_1, g_2) = \min (e_1, ..., e_k) \in E(g_1, g_2) \sum_{i=1}^{k} C(e_i) \tag{1}$$

where $E(g_1, g_2)$ denotes the set of edit paths between two graphs, C denotes the edit cost function and e_i denotes the individual edit operation. Based on (1), the problem of evaluating the structural similarity of graphs is changed into the problem of finding a minimum-cost edit path between two graphs.
Among different methods, the *probabilistic graph edit distance* (P-GED) proposed by Neuhaus and Bunke [12, 9] was chosen to automatically find the cost function from a labeled sample set of graphs. To this aim, the authors represented the structural similarity of two graphs by a learned probability $p(g_1, g_2)$ and defined the dissimilarity measure as:

$$d(g_1, g_2) = -\log p(g_1, g_2) \tag{2}$$

The main advantage of this model is that it learns the costs of edit operations automatically and is able to cope with large sets of graphs with huge distortion between samples of the same class [12, 9].

2.1 Graph embedding

Graph embedding converts a graph into an n-dimensional real vector. Its motivation is that of trying to take advantage of the rich space of statistical pattern recognition techniques yet retaining the spatial representational power of graphs. Let $G = \{g_1, g_2, ..., g_m\}$ be a set of graphs , $P = \{p_1, p_2, ..., p_n\}$ be a set of prototypes with $m > n$ (detail in subsection 2.2), and d be a (dis)similarity measure (detail in section 2). For graph embedding, dissimilarity d_{ji} of graph $g_j \in G$ to prototype $p_i \in P$ is computed. Then, an n-dimensional vector $(d_{j1}, ..., d_{jn})$ can be achieved through the computation of the n dissimilarities, $d_{j1} = d(g_j, p_1), ..., d_{jn} = d(g_j, p_n)$. As a result of this, any graph $g_j \in G$) can be transformed into a vector of real numbers.
Formally, the mapping $t^P : G \to \mathbb{N}^n$ is defined as the following function:

$$t^P(g) \to (d(g, p_1), ..., d(g, p_n)) \tag{3}$$

where $d(g, p_i)$ is a dissimilarity measure between graph g and prototype p_i [8].

2.2 Prototype selector

Based on the definition in section 2.1, selecting informative *prototypes* from the underlying graph domain plays a vital role in graph embedding. In other words, in order to have meaningful vectors in the embedding space, a set of selected prototypes $P = \{p_1, p_2, ..., p_n\}$ should be uniformly distributed over the whole graph domain while avoiding redundancies in terms of selection of similar graphs [8], [13], [14].

3 Methodology

Our approach to action recognition is based on i) using graph embedding to create a feature vector of the actor in each frame, ii) concatenating all the feature vectors from the first to the last frame of the action video into a vector sequence, and iii) using a sequential classifier for action classification. As sequential classifier, we have used the well-known hidden Markov model [15]. Moreover, prior to extracting graphs of the actor's shape, we have used a tracker to extract a bounding box of the actor in each frame [16]. Due to limitation in space, we do not describe the tracker and classifier further and focus the next paragraphs on graph construction and embedding.

3.1 Graph building

A number of SIFT keypoints are extracted within the actor's bounding box in each video frame using the software of Vedaldi and Fulkerson [17]. Based on the chosen threshold, this number typically typically varies between 5 and 8. Moreover, a Gaussian outlier elimination method is applied on the selected SIFT keypoints to eliminate points which are estimated to be far away from the actor. Example results after these steps are illustrated in figure 1. Then, the location of the remaining SIFT keypoints (x, y) is expressed relatively to the actor's centroid and employed as a node label for an attributed graph describing the actor's shape. In a preliminary study not reported in this paper, we found that graphs with only labelled nodes performed as well as graphs with both labelled nodes and edges and were faster to process. We therefore decide to employ graphs consisting only of labelled nodes.

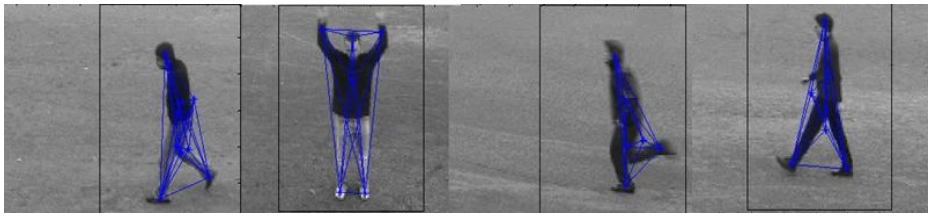

Fig. 1. Bounding box generated by the proposed tracker in the KTH action dataset and final selected SIFT keypoints which are used to build a graph.

3.2 Posture selection

In order to have a semantic prototype set which could lead to meaningful feature vectors in the embedded space, a number of different reference postures was chosen to describe all of the human shapes in the action dataset. For the dataset at hand, (KTH [18]; details provided in section 4.1), we chose a set of 16 different reference postures across all of the human actions (running, walking, boxing, jogging, hand-waving, hand-clapping). For training purposes, we manually selected a number of different frames varying in scenario (e.g. outdoor, outdoor with different clothes, indoor), action (e.g. hand waving, hand clapping, jogging) and actor (e.g. person01, person25, person12) (see figure 2).

Fig. 2. Examples of selected postures which are used to describe all of the human actions in the KTH action dataset.

3.3 Prototype selection

Among various prototype selection algorithms [8], [13], [14], the *class-wise center prototype selection* (c-cps) method [8] was chosen in this study. With this method, a prototype set $P = \{p_1, ..., p_n, ..., p_N\}$ is found from a class set $C = \{c_1, ..., c_n, ..., c_N\}$, $N = 16$ with each p_n prototype located in, or near, the centre of class c_n. For selecting the center graph from the sample set of class $c_n = \{g_{n1}, ..., g_{nj}, ..., g_{nN_n}\}$, we choose g_{nj} such that the sum of distances between g_{nj} and all other graphs in c_n is minimal (eq.4) [8].

$$p_n = g_{nj} = \arg \min_{g_{nj} \in c_n} \sum_{g_{ni} \in c_n} d(g_{nj}, g_{ni}) \tag{4}$$

3.4 Feature vector

The embedding of a graph leads to a 16-dimensional feature vector describing the shape of a single actor in a frame. In order to exploit other available information, we add the displacement between the bounding boxes of two successive frames (which is proportional to the horizontal speed component) and the location of the actor's centroid relative to the bounding box. This leads to an overall 19-dimensional feature vector to describe the shape, motion and location of the actor in each frame (see figure 3).

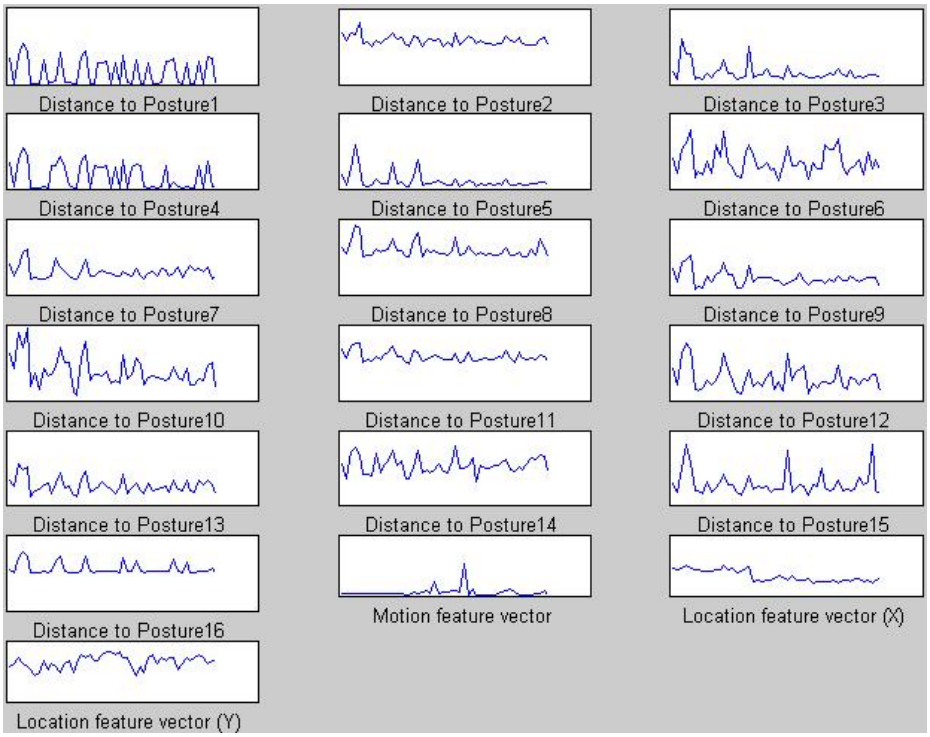

Fig. 3. The time-sequential values of a 19-dimensional feature vector obtained from graph embedding for one action (boxing) performed by one subject in the KTH action dataset.

4 Experiments

For the experimental evaluation of our approach, we have chosen a popular dataset, KTH [18], which allows comparison of our results with other, state-of-the-art action recognition methods.

4.1 KTH action dataset

The KTH human action dataset contains six different human actions: walking, jogging, running, boxing, hand-waving and hand-clapping, all performed various times over homogeneous backgrounds by 25 different actors in four different scenarios: outdoors, outdoors with zooming, outdoors with different clothing and indoors. This dataset contains 2391 sequences, with each sequence down-sampled to the spatial resolution of 160×120 pixels and a length of four seconds on average. While this dataset consists of simplified actions, it is challenging in terms of illumination, camera movements and variable contrasts between the subjects and the background. KTH has been a de-facto benchmark in the last few years and many results are available for comparison.

4.2 Experimental set-up and results

In this section, we evaluate the recognition accuracy of the proposed method. We first evaluate various choices of feature vectors and then compare our approach based on the best feature vector with the state of the art. All of these experiments were performed on a computer with an Intel(R) Core(TM)2 Duo CPU (E8500, 3.16GHz) and 4GB RAM using Matlab R2009b.

Evaluation of the feature vectors The 19-dimensional feature vector described in section 3.4 contains shape, motion and location features jointly. In order to assess the individual contribution of these different types of features, we have conducted experiments with feature vectors containing only shape, motion or location features in isolation. To this aim, we have used *leave one (actor) out cross validation* reporting a correct classification rate (CCR) for each feature vector. It is possible to see that none of the individual type of features was capable of achieving high accuracy in isolation; in all cases, recognition accuracy was below 50% (table 1). However, these features show interesting complementarity: for instance, the motion features report good accuracy in recognising the Jogging class, but a rather low performance on the Boxing class (which is mainly a stationary class). Conversely, the graph-embedded shape features report good accuracy on the Boxing class, but cannot discriminate well between classes such as Jogging and Running where the articulated shape is similar, yet speed of execution varies remarkably. This complementarity is at the basis of the higher performance achieved by the joint vector which jumps to 70.00%, as shown by table 2.

Table 1. The average CCRs on the KTH action dataset based on separate feature vectors for motion, location and shape.

validation technique	motion	location	shape
LOOCV-CCR	49.34%	45.67%	47.63%

Comparison to the state of the art Accuracy measurements on the KTH database have been performed with different methods by different papers in the literature. For easier comparison, in this section we have used the test approach presented by Schuldt *et al*. in [18]. With this test approach, all sequences are divided into 3 different sets with respect to actors: training (8 actors), validation (8 actors) and test (9 actors). The classifier is then tuned using the first two sets (training and validation sets), and the accuracy on the test set is measured by using the parameters selected on the validation set, without any further tuning. The confusion matrix obtained with the proposed approach is presented in table 3. The overall accuracy is 70.17%, slightly higher than the 70.00% obtained with the leave one out cross validation. This result is not yet comparable with the best accuracies reported in the literature: it is not far from the accuracy reported by *Schuldt et al.* [18], but much lower than that reported by Guo *et al.* in [19] (table 4).

Table 2. Action confusion matrix (%) for the proposed method based on the LOOCV test approach on the KTH action dataset. The average CCR is 70.00%.

	Boxing	Clapping	Waving	Jogging	Running	Walking
Boxing	80	9	8	1	1	1
Clapping	11	59	25	1	2	2
Waving	8	22	66	1	0	3
Jogging	0	0	0	56	21	23
Running	0	0	0	17	74	9
Walking	0	0	0	8	7	85

Table 3. Action confusion matrix (%) for the proposed method based on the Schuldt test approach on the KTH action dataset. The average CCR is 70.17%.

	Boxing	Clapping	Waving	Jogging	Running	Walking
Boxing	92	1	5	0	1	1
Clapping	18	62	17	1	1	1
Waving	9	35	55	0	0	1
Jogging	1	0	0	56	18	25
Running	1	0	0	14	66	19
Walking	0	0	0	5	5	90

Table 4. Average class accuracy on the KTH action dataset.

Method	**Ours**	Schuldt *et al.*[18]	Dollar *et al.*[1]	Laptev *et al.*[20]	Guo *et al.*[19]
LOOCV-CCR	**70.00%**	-	80%	-	98.47%
Schuldt-CCR	**70.19%**	71.70%	-	91.80%	97.40%

4.3 Discussion

In section 4.2 we have showed our initial experimental results from the application of graph building and embedding to human action recognition. Despite the good posture discrimination provided by P-GED (not reported quantitatively here for reasons of space), the overall action recognition accuracy on the KTH dataset is not yet very high. Based on our judgment, the main difficulty faced by the proposed approach is the extraction of a reliable set of keypoints in each frame. Due to noise and variable appearance, the extracted set changes significantly over the frames. Another possible limitation is the limited accuracy of the employed classifier (HMM). However, we believe that the

work conducted to date already provides evidence that the features obtained by graph embedding are capable of encoding the actor's shape to a significant extent.

5 Conclusions and future work

In this paper, we have presented a novel approach for human action recognition based on graph embedding. To this aim, an attributed graph is used to represent the actor's shape in each frame and then graph embedding is used to convert the graph into a feature vector so as to have access to the wide range of current classification methods. Although our method does not yet match the accuracy of existing approaches, it generates a novel methodology for human action recognition based on graph embedding and may outperform existing methods in the future. With reference to limitations discussed in section 4.3, we plan to further investigate other keypoint sets to improve the stability of the graph-based representation along the frame sequence and employ different classification methods for the classification stage.

Acknowledgments. The authors wish to thank the Australian Research Council and its industry partners that have partially supported this work under the Linkage Project funding scheme - grant LP 0990135 "Airport of Future".

References

1. P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65–72. IEEE, 2006.
2. I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005.
3. Jaron Blackburn and Eraldo Ribeiro. Human motion recognition using isomap and dynamic time warping. In *Proceedings of the 2nd conference on Human motion: understanding, modeling, capture and animation*, pages 285–298, Berlin, Heidelberg, 2007. Springer-Verlag.
4. J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden Markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 379–385, 1992.
5. A. Quattoni, S. Wang, L. p Morency, M. Collins, T. Darrell, and Mit Csail. Hidden-state conditional random fields. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.
6. R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.
7. Anh-Phuong Ta, Christian Wolf, Guillaume Lavoue, and Atilla Baskurt. Recognizing and localizing individual activities through graph matching. volume 0, pages 196–203, Los Alamitos, CA, USA, 2010. IEEE Computer Society.
8. K. Riesen, M. Neuhaus, and H. Bunke. Graph embedding in vector spaces by means of prototype selection. In *Proceedings of the 6th IAPR-TC-15 international conference on Graph-based representations in pattern recognition*, pages 383–393. Springer-Verlag, 2007.
9. M. Neuhaus and H. Bunke. Automatic learning of cost functions for graph edit distance. *Information Sciences*, 177(1):239–247, 2007.

10. X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Analysis & Applications*, 13(1):113–129, 2010.
11. D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(3):265–298, 2004.
12. M. Neuhaus and H. Bunke. A probabilistic approach to learning costs for graph edit distance. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 389–393. IEEE, 2004.
13. G.R. Hjaltason and H. Samet. Properties of embedding methods for similarity searching in metric spaces. *IEEE Transactions on Pattern Analysis and machine intelligence*, pages 530–549, 2003.
14. K. Riesen and H. Bunke. Graph classification by means of Lipschitz embedding. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(6):1472–1483, 2009.
15. L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
16. T.P. Chen, H. Haussecker, A. Bovyrin, R. Belenov, K. Rodyushkin, A. Kuranov, and V. Eruhimov. Computer vision workload analysis: case study of video surveillance systems. *Intel Technology Journal*, 9(2):109–118, 2005.
17. A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.
18. C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, 2004.
19. K. Guo, P. Ishwar, and J. Konrad. Action Recognition Using Sparse Representation on Covariance Manifolds of Optical Flow.
20. I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.