A discriminative prototype selection approach for graph embedding in human action recognition

Ehsan Zare Borzeshi, Massimo Piccardi, Richard Yi Da Xu
Faculty of Engineering and Information Technology
University of Technology, Sydney (UTS), Australia
{Ehsan.ZareBorzeshi, Massimo.Piccardi, YiDa.Xu}@uts.edu.au

Abstract

This paper proposes a novel graph-based method for representing a human's shape during the performance of an action. Despite their strong representational power, graphs are computationally cumbersome for pattern analysis. One way of circumventing this problem is that of transforming the graphs into a vector space by means of graph embedding. Such an embedding can be conveniently obtained by way of a set of "prototype" graphs and a dissimilarity measure: yet, the critical step in this approach is the selection of a suitable set of prototypes which can capture both the salient structure within each action class as well as the intra-class variation. This paper proposes a new discriminative approach for the selection of prototypes which maximizes a function of the inter- and intra-class distances. Experiments on an action recognition dataset reported in the paper show that such a discriminative approach outperforms well-established prototype selection methods such as center, border and random prototype selection.

1. Introduction and related literature

Many approaches have been proposed to date for human action recognition, including bag-of-features [6] [11], dynamic time warping [2], hidden Markov models [32] and conditional random fields [19]. However, the problem of finding a suitable feature set to more effectively represent the deformable shape of a human performing an action is still partially unresolved. Graphs provide a very powerful and flexible way to describe relations between parts, and could therefore be used to encapsulate the object's structure and support human action recognition. However, a major drawback of graph-based representations is that even basic operations such as sums and products cannot be performed on graphs, making them unsuitable for conventional pattern recognition approaches based on feature vectors. One way of resolving this problem is to apply *graph embedding*, converting a graph into a point in a vector space. Amongst the various graph embedding approaches [17] [31] [22] [7], we employ prototype-based graph embedding for its theoretical simplicity. With this approach, a graph is converted to an n-dimensional feature vector by way of a set of "prototype" graphs and a dissimilarity measure (often, a graph edit distance): the feature vector consists of the distances between the graph and each prototype. In order for such vectors to prove class-discriminative, the prototype set should be able to cover the graph domain in a uniform way. However, this is difficult to ensure in principle since uniformity over a graph domain is a vague concept.

In this paper, we study the use of class-based prototype selection and propose a novel discriminative prototype selection method maximizing a function of the inter- and intra-class distances. The method is compared with well-established class-based prototype selection methods such as *center*, *border* and *random* prototype selection [17] [22]. The application addressed in this paper is action recognition in videos: in our approach, we first apply a tracker to obtain a bounding box of each actor in each frame. Within the bounding box, spatial feature points (SIFT [12]) are then detected and used as nodes of a graph representing the human shape. Eventually, the graph is converted to a set of distances based on a prototype set and the probabilistic graph edit distance (P-GED), a sophisticated edit distance capable of learning edit costs from a training set [15]. The feature vectors from the individual frames of each video are then composed into a time series to describe the evolution of the actor's shape along the time dimension and permit action recognition. As time-series classifier we have used the hidden Markov model; yet, conditional random fields or structural SVM could be equally applied - the classification algorithm is not the focus of this paper. As action dataset, we have used the KTH dataset [26] for its widespread past utilisation, while we plan to extend the study to other datasets such as the UCF sports action dataset [23], the Olympics Sports dataset [16] and MuHAVi [28] in the near future.

The rest of the paper is organized as follows: a brief

overview of prototype-based graph embedding is offered by section 2. In section 3, we describe the use of graph embedding for action recognition. In section 4, we present an experimental evaluation of the proposed approach on the KTH action dataset. Finally, we give concluding remarks and a discussion of future work in section 5.

2. Overview of prototype-based graph embedding

An *attributed graph*, $g = (V, E, \alpha, \beta)$, is a tuple defined by

- $V = \{1, 2, ..., M\}$, a set of vertices (nodes),
- $E \subseteq (V \times V)$, a set of edges,
- $\alpha : V \rightarrow L_V$, a vertex labeling function, and
- $\beta : E \rightarrow L_E$, an edge labeling function.

Vertex and edge labels are restricted to fixed-size tuples, ($L_V = \mathbb{R}^p$, $L_E = \mathbb{R}^q$, $p, q \in \mathbb{N} \cup \{0\}$). When attribute graphs are used to represent objects, 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). The graph edit distance between any two graphs is defined as the cost of transforming the first graph into the second [8]. It measures the (dis)similarity of arbitrarily structured and arbitrarily labeled graphs and is flexible thanks to its ability to cope with any kind of structural errors [8], [5]. The edit transformation is usualy broken up into atomic edit operations which can be of six basic types: insertion, deletion and substitution, for either nodes or edges, and noted as $(e^{i,n}, e^{d,n}, e^{s,n}, e^{i,e}, e^{d,e}, e^{s,e})$. It can be proven that every arbitrary graph can be transformed into another, equally arbitrary graph by applying a finite sequence of edit operations (also called an *edit path*). The distance between the two graphs is defined as the minimum cost amongst all edit paths transforming the first graph into the other. Let $g_i = (V_i, E_i, \alpha_i, \beta_i)$ and $g_j = (V_j, E_j, \alpha_j, \beta_j)$ be a pair of graphs in a set. The graph edit distance of such graphs is formally defined as:

$$d(g_i, g_j) = \min_{(e_1,...,e_k) \in E(g_i,g_j)} \sum_{l=1}^{k} C(e_l) \qquad (1)$$

where $E(g_i, g_j)$ denotes the set of edit paths between the two graphs, C denotes the edit cost function and e_l denotes the individual edit operation. Based on (1), the problem of evaluating the structural similarity of two graphs is changed into that of finding a minimum-cost edit path between them.

Among the various methods, the *probabilistic graph edit distance* (P-GED) proposed by Neuhaus and Bunke [14], [15] is capable of automatically inferring the cost function from a training set of manually-paired graphs. P-GED measures the similarity of two graphs by a learned probability, $p(g_i, g_j)$, and defines the dissimilarity measure as:

$$d(g_i, g_j) = -\log p(g_i, g_j) \qquad (2)$$

A further advantage of P-GED is its claimed ability to learn from large sets of graphs with huge distortion between samples of the same class, which makes it suitable for application to vision problems [14], [15].

2.1. Graph embedding

In the literature, "graph embedding" refers interchangeably to the embedding of a graph as a whole into a point in vector space, or the embedding of its set of nodes into a set of corresponding points in vector space. In this work, we assume the former meaning, although similar embedding techniques can be applied in the two cases and for other types of non-vectorial objects such as strings or trees [21]. The embedding assumes that a set of objects is given alongside distance values between any two objects in the set. The goal is that of converting the set of objects into a set of points in a vector space of given dimensionality while ensuring certain properties or constraints. Well-known embedding techniques include Laplacian eigenmaps, commute times, symmetric polynomials, and kernel principal component analysis, amongst others [1], [18], [31], [25]. After the embedding of the initial set of objects, it is also possible to embed new, out-of-sample objects, albeit not always straightforward. An alternative embedding approach is to make use of a given set of "prototype" objects (or prototypes, for short) which can equally embed in-sample and out-of-sample data, in a way not unlike that of eigenvectors in principal component analysis. Let $G = \{g_1, g_2, ..., g_m\}$ be a set of graphs, $P = \{p_1, p_2, ..., p_n\}$ be a set of prototype graphs with $m > n$, and d be a dissimilarity measure. For embedding any graph $g_j \in G$ by way of P, the dissimilarity measure $d_{ji} = d(g_j, p_i)$ of graph g_j to prototype $p_i \in P$ is computed $\forall i$. Then, an n-dimensional vector $(d_{j1}, ..., d_{jn})$ is assembled from all the n dissimilarities. With this procedure, any graph can be individually transformed into a vector of real numbers [22], [17]. Prototype-based embedding is certainly the simplest and fastest embedding approach and for these reasons is adopted hereafter.

2.2. Prototype selection

Based on the definition given in section 2.1, selecting informative *prototypes* from the underlying graph domain plays a critical role in graph embedding. In other words, in order to obtain useful graphs' representations in vector space, the set of prototypes, $P = \{p_1, ..., p_n, ..., p_N\}$, should be uniformly distributed over the whole graph domain, at the same time avoiding redundancies in terms of

selection of similar graphs [17], [22], [10]. Prototype selection methods mainly sub-divide into class-independent (or unlabeled) and class-based (or labeled) approaches. Class-independent approaches select N prototypes globally for the entire training set, C, and can be applied to unlabeled training data. Class-based approaches are instead only possible if the training data are labeled into classes, and their aim is that of selecting one prototype for each of N classes, $C_1, ..., C_N$. Given that in our application graphs can be labeled into classes, in this work we have decided to adopt class-based approaches. Various existing methods will be reviewed in subsection 3.3.

3. Methodology

The approach used for human action recognition consists of the following main steps:

- use of a modified tracker [4] to extract a bounding box of each actor in each frame, and extracting the SIFT keypoints within such a bounding box;
- for each bounding box, building a graph using the locations of the SIFT keypoints as nodes;
- embedding the graph into a feature vector by means of P-GED with the prototype set of choice;
- concatenating the feature vectors for a single actor from a whole video into a time series;
- applying a sequential classifier to the time series. As sequential classifier, we have used the well-known hidden Markov model [20] and a variant to be detailed in section 4.1.

The following subsections provide further details of the approach.

3.1. Graph building

As a preliminary step, a modified tracker is used to extract a bounding box of each actor in each frame [4]. Over the dataset at hand, (KTH [26]; details provided in section 4), the tracker performs really well, providing bounding boxes which almost invariably contain the actor in full size. As the next step, a number of *scale invariant feature transform* (SIFT) keypoints [12] are extracted within the actor's bounding box in each video frame using the software of Vedaldi and Fulkerson [30]. Based on the chosen threshold, their number typically varies between 5 and 8. Example results of this step are illustrated in figure 1. After extraction, the location of each SIFT keypoint, (x, y), is expressed relatively to the actor's centroid and employed as a node label for an attributed graph describing the human's shape. In a preliminary study, we found that graphs with only labeled nodes granted comparable accuracy to graphs with both labeled nodes and labeled edges, yet resulted in faster processing. We therefore decided to employ graphs consisting only of labeled nodes.

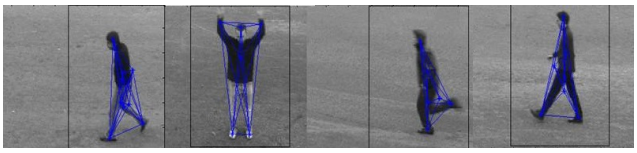

Figure 1. Bounding box generated from a modified tracker [4] using the KTH action dataset and the extracted SIFT keypoints composed into a graph.

3.2. Posture set

In order to identify a prototype set which could lead to meaningful feature vectors in the embedded space, a number of different reference postures was chosen to describe all human shapes in the action dataset. For KTH, we arbitrarily chose a set of 16 different reference postures across all human actions (running, walking, boxing, jogging, hand-waving, hand-clapping). Such selected postures should prove adequate for recognising human actions also in any other dataset where the actors are approximately in full view such as UCF Sports [23] and MuHAVi [28]. 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).

Figure 2. Examples of selected postures from the KTH action dataset.

3.3. Prototype selection

As stated in subsection 2.1, an appropriate choice of the prototype set, P, plays a critical role in this approach as it impacts the classification accuracy. Given that we avail ourselves of a labelled training set, we have decided to employ class-based approaches for prototype selection. In the following, we describe three popular, existing approaches and the approach proposed in this work.

In *class-based center prototype selection* (c-cps), a prototype set, $P = \{p_1, ..., p_n, ..., p_N\}$, is generated from a labeled training set, $C = \{C_1, ..., C_n, ..., C_N\}$, with each p_n prototype located in, or near, the "center" of the graphs from the n-th class, C_n. To implement the notion of center, we select the median graph from sample set C_n =

$\{g_{n1}, ..., g_{nj}, ..., g_{nN_n}\}$, defined as the g_{nj} graph such that the sum of distances between g_{nj} and all other graphs in C_n is minimal [22]:

$$p_n = \arg \min_{g_{nj} \in C_n} \sum_{g_{ni} \in C_n, g_{ni} \neq g_{nj}} d(g_{nj}, g_{ni}) \quad (3)$$

As an alternative, *class-based border prototype selection* (c-bps) chooses the prototype set, P, with each p_n prototype situated at the "farthest border" of its class, C_n. Again, the notion of border is vague in class domain. The rationale for this selection is that of having prototypes which are at maximum distance from the training graphs and generate feature vectors with the largest values. To implement it, we select the marginal graph from the sample set of class $C_n = \{g_{n1}, ..., g_{nj}, ..., g_{nN_n}\}$, defined as the g_{nj} graph such that the sum of distances between g_{nj} and all other graphs in C_n is maximal [22]:

$$p_n = \arg \max_{g_{nj} \in C_n} \sum_{g_{ni} \in C_n, g_{ni} \neq g_{nj}} d(g_{nj}, g_{ni}) \quad (4)$$

Given the relative arbitrariness of the above selections, a random choice of the class prototype is a plausible alternative. In *class-wise random prototype selection* (c-rps), each p_n prototype is randomly selected from class C_n with uniform probability [22]:

$$p_n = g_{nj} \in C_n, j \sim p(k = 1...N_n) = \frac{1}{N_n} \quad (5)$$

All of the above selection approaches choose the class' prototype based solely on the graphs in the class. This is in a way reminiscent of generative classifiers, where a class' parameters are estimated based on only the samples from that class. Discriminative classifiers, instead, choose parameters based on the information from multiple classes at once, maximizing objective functions such as the class margin, Fisher discriminants and others, and often proving more accurate than their generative counterparts. Inspired by discriminative approaches, we propose herewith a *class-based discriminative prototype selection* approach (c-dps), where each p_n prototype is chosen as the graph g_{nj} that minimizes the ratio between the sum of distances between g_{nj} and all other graphs in C_n and the sum of distances between g_{nj} and all graphs in the other classes, $\overline{C_n}$:

$$p_n = \arg \min_{g_{nj} \in C_n} \frac{\sum_{g_{ni} \in C_n, g_{ni} \neq g_{nj}} d(g_{nj}, g_{ni})}{\sum_{g_{ni} \in \overline{C_n}} d(g_{nj}, g_{ni})} \quad (6)$$

This selection approach is analogous to minimizing the ratio between the within-class and between-class scatter matrices in vector spaces.

3.4. Feature vector

The embedding of a graph by any of the above prototype selection methods leads to a 16-dimensional feature vector describing the shape of a single actor in a frame. Time series of such vectors may prove action-discriminative. Yet, we decided to augment the feature vector by some basic information about the actors' global motion and location relative to the bounding box. We thus added the horizontal displacement between the bounding boxes of two successive frames (which is proportional to the horizontal velocity) and the location of the actor's centroid relative to the bounding box. This leads to an overall 19-dimensional feature vector with information about the shape, motion and location of the actor in a frame. Figure 3 shows time series of the feature vector for a boxing action in KTH (the embedding is obtained by $c - dps$). An analysis of the individual contributions of the shape, motion and location information is presented in [3].

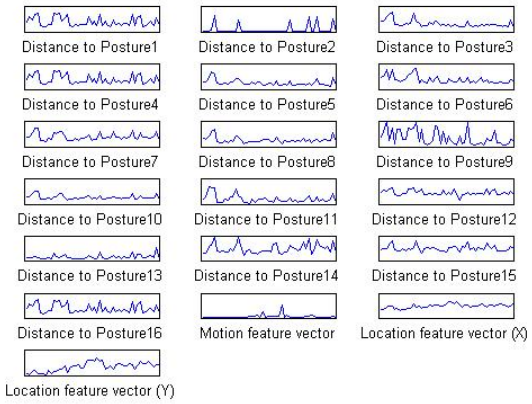

Figure 3. The time-sequential values of a 19-dimensional feature vector obtained from graph embedding based on the $c - dps$ for one action (boxing) performed by one subject in the KTH action dataset.

4. Experiments

A popular action dataset, KTH [26], has been chosen to compare the recognition accuracy from feature vectors obtained with different prototype selection approaches. 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 sec-

onds 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. In some sense, KTH is a stepping stone towards more recent datasets which add multiple views, non-staged actions and other challenges.

4.1. Experimental set-up and results

In this section, the recognition accuracies for the feature vectors extracted by the four different prototype selectors are given. For accuracy evaluation, we have used the evaluation procedure proposed by Schuldt *et al.* in [26]. In this procedure, all sequences are divided into three sets with respect to actors: training (8 actors), validation (8 actors) and test (9 actors). Each classifier is then tuned using the first two sets (training and validation sets), and the accuracy on the test set is measured "blindly" by using the parameters selected on the validation set, without any further tuning. All our experiments were performed on a personal computer with an Intel(R) Core(TM)2 Duo CPU (E8500, 3.16GHz) and 4GB RAM using Matlab R2009b. As software, we have used Murphy's HMM toolbox for Matlab, modified as needed [13].

4.2. Evaluation of feature vectors with maximum likelihood training

The hidden Markov model (HMM) is a generative approach which can be used to recognise human actions in time series. The assumption used in the following is that each action class is in correspondence with one HMM. The learning of the HMM parameters for each class is achieved by the Baum-Welch re-estimation algorithm [20] and classification of an unseen observation sequence, O_{new}, is obtained by maximum-likelihood (ML) classification. In other words, let us denote as $A = \{a_1, ..., a_k, ..., a_K\}$ the set of K different action classes; $\lambda = \{\lambda_1, ..., \lambda_k, ..., \lambda_K\}$, the set of HMM parameters associated with each action class in A; $O = \{O_1, ..., O_k, ..., O_K\}$, the set of K different groups of observation sequences, one per class; and, eventually, each $O_k = \{O_k^1, ..., O_k^{N_k}\}$ as the group of N_k observation sequences for action class k. Then, parameters $\lambda_k^*, k = 1...K$, are estimated with maximum likelihood as:

$$\lambda_k^* = \arg\max_{\lambda_k}(\prod_{e=1}^{N_k} p(O_k^e|\lambda_k)) \tag{7}$$

After training of the λ parameters, the action class, a_k^*, for an unseen sequence, O_{new}, can be chosen by maximum likelihood as:

$$a_{k^*} : k^* = \arg\max_k(p(O_{new}|\lambda_k)), \quad k = 1..K. \tag{8}$$

where the likelihood of O_{new} in action class k, $p(O_{new}|\lambda_k)$, can be efficiently evaluated by the forward or backward algorithm [20]. The Correct Classification Rate (CCR) obtained with this method is reported in table 1. Table 1 shows that the discriminative prototype selector achieves greater accuracy than the compared methods.

Table 1. Classification accuracy of a maximum-likelihood HMM applied to feature vectors from different prototype selectors (c-dps, c-cps, c-bps and c-rps).

Schuldt's validation	
Prototype selector	CCR(%)
c-dps	67.80
c-cps	66.75
c-bps	64.05
c-rps	65.50

4.3. Evaluation of feature vectors with maximum conditional likelihood training

In addition to the positions above, let $Y = \{Y_1, ..., Y_k, ..., Y_K\}$ be the set of K different groups of ground-truth labels for the observation sequences in each class; and each $Y_k = \{y_k^1, ..., y_k^{N_k}\}$ be the group of ground-truth labels for the N_k observation sequences of action class k. Each such a label takes value in A, the set of action classes. Here, the availabilty of the ground-truth labels allows defining a different objective function, known as *conditional likelihood*, for the setting of the λ parameters [29]:

$$\mathcal{L}(\lambda; Y, O) = \prod_{k=1}^{K}\prod_{e=1}^{N_k} p(y_k^e|O_k^e, \lambda_k) \tag{9}$$

Parameters $\lambda = \{\lambda_1, .., \lambda_k, .., \lambda_K\}$ are then selected to maximize the conditional likelihood as in:

$$\lambda^* = \arg\max_{\lambda}(\mathcal{L}(\lambda; Y, O)) \tag{10}$$

The parameters estimated by maximizing (9) are more promising for classification than those estimated with the conventional likelihood since conditional likelihood $p(y'|O', \lambda')$ for a given class, y', and meauserement, O', is, with different wording, the posterior probability of class y' given measurement O'. In essence, training the parameters with the conditional likelihood target maximizes the posterior probability of the correct class labels over the entire training set. As such, it is an example of *maximum score* training [27].

However, maximizing the conditional likelihood for the HMM is not trivial. Therefore, in this work we resort to

an approximation: at each iteration of the Baum-Welch algorithm (which is guaranteed to increase the conventional likelihood), we evaluate (9) and store the parameters. At convergence of Baum-Welch, the value of the parameters corresponding to the largest conditional likelihood encountered during the iterations is selected.

Table 2 shows the recognition accuracy with the maximum conditional likelihood criterion. Again, the proposed discriminative selector, $c - dps$, achieves the highest accuracy. In addition, the proposed conditional likelihood training permits higher accuracy than conventional likelihood training in most cases.

Table 2. Classification accuracy of a maximum-conditional-likelihood HMM applied to feature vectors from different prototype selectors (c-dps, c-cps, c-bps and c-rps).

Schuldt's validation	
Prototype selector	CCR(%)
c-dps	70.35
c-cps	68.85
c-bps	64.15
c-rps	65.50

4.4. Discussion

To position our work properly, it is very important to state that current results on KTH are well in excess of 90% accuracy [9]. The goal of our paper is not that of proposing a more accurate action recognition method; rather, assessing the comparative accuracy of discriminative prototype selection in a significant classification exercise. As for what action recognition is concerned, we have gathered empirical evidence that the graphs built by using SIFT keypoints as their nodes are rather unstable and noisy, and we are working on the use of graph-cut techniques to substantially improve nodes' extraction [24]. In addition, the usefulness of discriminative prototype selection extends well beyond action recognition: typical problems approached by graph embedding include, for instance, fingerprint recognition, character recognition and general object classification [5].

5. Conclusions and future work

In this paper, we have proposed a discriminative prototype selector for graph embedding and evaluated its use in an application of human action recognition. In our action recognition approach, an attributed graph is built in each frame to represent the actor's shape. Thence, a set of prototypes is used to embed this graph into a point in vector space. The sequence of vectors for a whole video depicting an action is then collected as a time series, and the hidden Markov model is used for action classification. The experiments reported in the paper show that the proposed prototype selector allows accuracies that are 1.05÷1.50 percentage points higher than that of the best competing selector. In addition, we have shown that an approximate maximum conditional likelihood training of the HMM allows accuracies that are up to 2.55 percentage points higher than those with conventional likelihood training. In the near future, we plan to provide improvements to the graphs' extraction, extend our study to other action datasets, as well as experiment with the discriminative selector over other domains such as object and character recognition.

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 LP0990135 "Airports of the Future".

References

[1] M. Belkin and P. Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396, June 2003. 2

[2] J. Blackburn and E. 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. 1

[3] E. Z. Borzeshi, R. Y. D. Xu, and M. Piccardi. Automatic human action recognition in videos by graph embedding. In *16th International Conference on Image Analysis and Processing, ICIAP 2011*, 2011. 4

[4] T. 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. 3

[5] 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. 2, 6

[6] 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. 1

[7] D. Emms, R. Wilson, and E. Hancock. Graph embedding using quantum commute times. *Graph-Based*

Representations in Pattern Recognition, pages 371–382, 2007. 1

[8] X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Analysis & Applications*, 13(1):113–129, 2010. 2

[9] K. Guo, P. Ishwar, and J. Konrad. Action Recognition Using Sparse Representation on Covariance Manifolds of Optical Flow. 6

[10] G. 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. 3

[11] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005. 1

[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 1, 3

[13] K. Murphy. Hidden markov model (hmm) toolbox for matlab. *online at http://www. ai. mit. edu/˜ murphyk/Software/HMM/hmm. html*. 5

[14] 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. 2

[15] M. Neuhaus and H. Bunke. Automatic learning of cost functions for graph edit distance. *Information Sciences*, 177(1):239–247, 2007. 1, 2

[16] J. Niebles, C. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. *Computer Vision–ECCV 2010*, pages 392–405, 2010. 1

[17] E. Pekalska and R. Duin. *The dissimilarity representation for pattern recognition: foundations and applications*. World Scientific Pub Co Inc, 2005. 1, 2, 3

[18] H. Qiu and E. Hancock. Clustering and embedding using commute times. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1873–1890, 2007. 2

[19] A. Quattoni, S. Wang, L. p Morency, M. Collins, T. Darrell, and M. Csail. Hidden-state conditional random fields. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007. 1

[20] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. 3, 5

[21] K. Rieck and P. Laskov. Linear-Time Computation of Similarity Measures for Sequential Data. *Journal of Machine Learning Research*, 9:23–48, Jan. 2007. 2

[22] 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. 1, 2, 3, 4

[23] M. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 1, 3

[24] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23:309–314, 2004. 6

[25] B. Scholkopf, A. Smola, and K.-R. Muller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Comp.*, 10(5):1299–1319, July 1998. 2

[26] 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. 1, 3, 4, 5

[27] Q. Shi, L. Wang, L. Cheng, and A. Smola. Discriminative human action segmentation and recognition using semi-markov model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008. 5

[28] S. Singh, S. Velastin, and H. Ragheb. Muhavi: A multicamera human action video dataset for the evaluation of action recognition methods. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 48–55. IEEE, 2010. 1, 3

[29] C. Sutton and A. McCallum. 1 an introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, page 93, 2007. 5

[30] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008. 3

[31] R. Wilson, E. Hancock, and B. Luo. Pattern vectors from algebraic graph theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1112–1124, 2005. 1, 2

[32] 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. 1