Real-Time Exact Graph Matching with Application in Human Action Recognition

Oya Çeliktutan[1,2], Christian Wolf[2], Bülent Sankur[1], and Eric Lombardi[2]

[1]Electrical and Electronics Engineering,
Boğaziçi University, Istanbul, Turkey
[2]Université de Lyon, CNRS,
INSA-Lyon, LIRIS, UMR CNRS 5205, F-69621, France

Abstract. Graph matching is one of the principal methods to formulate the correspondence between two set of points in computer vision and pattern recognition. However, most formulations are based on the minimization of a difficult energy function which is known to be NP-hard. Traditional methods solve the minimization problem approximately. In this paper, we show that an efficient solution can be obtained by exactly solving an approximated problem instead of approximately solving the original problem. We derive an exact minimization algorithm and successfully applied to action recognition in videos. In this context, we take advantage of special properties of the time domain, in particular causality and the linear order of time, and propose a novel spatio-temporal graphical structure.

Keywords: Space-time graph, Hyper-graph matching, Action recognition

1 Introduction

In many applications involving the recognition of complex visual patterns, as for instance recognition of object classes or actions in video scenes, salient local features collected on sparse set of points provide a compact yet rich representation, for classification or matching. This approach can be robust, e.g. against occlusion and bypasses the tedious segmentation task. The resulting representation is inherently structural and is therefore difficult to use in a statistical learning framework without sacrificing all or a part of the spatial or spatio-temporal relationships. In fact, the ensemble of local features is often converted into a numerical representation, discarding all or most of the structural information in the process. A typical example is the bag-of-words (BoW) formalism, originally developed for image classification [1]. However, graphs (and hypergraphs) form a natural description of this type of data.

In the context of human action recognition, a graph can effectively represent the relationship between low-level features such as spatio-temporal interest points, descriptors, body parts etc. In [2], a number of interest points per frame is structured into a graph. Features that are computed based on graph edit distance to a set of predetermined prototypes are fed to Hidden Markov Models (HMM) for classification. Several entities, i.e. histogram of spatio-temporal descriptors, histogram of spin-image descriptors and action classes, are modeled by a graphical structure using Fiedler embedding

in [3]. In [4], the nodes correspond to the five body parts and the energy function is penalized with the prior information on the human body configuration.

In this work we concentrate on hyper-graph matching and point set matching, where the nodes of the graph(s) encode both position and description of spatio-temporal interest points, and the neighborhood relationship is derived from proximity information. Matching corresponds to finding an action model point set in a (usually larger) scene point set. Up to our knowledge, prior work on space-time graph matching can be summed up by a few recent papers. In [5] matching is done via temporally ordered local feature-graphs where each graph models spatial configuration of the features in a small temporal segment. In [6] graphs are built from adjacency relationships of space-time tubes produced from oversegmenting the test video, and in [7] graphs are built from proximity by thresholding distances in space time. These methods resort to off-the-shelf spectral methods or slightly modified versions of them. In contrast, we propose to take advantage of some properties of the 3D space in which the data is embedded to devise an exact algorithm.

There are alternative approaches taking into account space-time 3D geometry: In [8], the spatial position of the features is combined within a probabilistic framework where they divided the features into clusters and modeled each cluster by its relative spatial position as well as the distribution of the appearance and position of interest points. In [9], the correlation of spatio-temporal (ST) patterns is measured and ST correlograms are constructed. Pairwise spatio-temporal relations are introduced in [10], based on a set of rules, and this information is transformed into 3D histograms. In [11], interest points, optical flow and image segmentation are mixed, and classification is done with multiple search trees. In [12], a parts-based model integrates spatio-temporal configuration, appearance, and human-object interactions. Finally, in [13], a maximum-weight connected subgraph is searched over temporal or spatio-temporal subvolumes where the weights are assigned by binary Support Vector Machines (SVMs).

The linear nature of the time dimension is frequently used to devise methods based on sequence alignment. In [14], a chain graph model exploits a priori knowledge of the nature and semantics of relationships between different variables. More examples are trajectory matching with Gabor filters [15], learning salient state transitions by HMMs [16], and modeling the evaluation of silhouettes over time [17].

Our proposed algorithm is related to sequence alignment in that it exploits temporal information and its linear nature in a similar way. However, we do not perform simple sequence alignment. The novelty of our approach is that we use a full-fledged hyper-graph model with all its rich structural information stored in its nodes, embedded in space-time, and in its hyper-edges built from proximity information. The derived minimization algorithm is capable of dealing with classical energy functions including unary, binary and ternary terms, which makes it possible to include scale invariant potentials, as the formulations in [18–20] and others. Once the graph representation of a given video sequence is obtained, action recognition problem boils down to searching for the closest prototype graph in the graph-space. Overview of our approach is illustrated in Figure 1.

Techniques for graph matching and for point set matching with graphs have been studied intensively in pattern recognition. While the graph isomorphism problem can

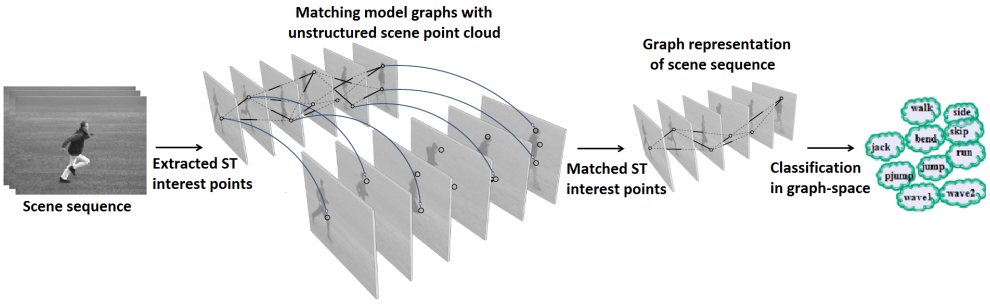


Fig. 1. Overview of the proposed algorithm for action recognition.

be calculated in polynomial time, it is widely known that exact subgraph matching is NP-complete [20], as is subgraph isomorphism [21]. Formulations like the one in (1) are known to be NP-hard [20]. In the context of object recognition, a method which approximates the graph, which in turn enables computation of the exact solution in polynomial time has been proposed in [22]: a k-tree is built randomly from the spatial interest points on an object, which allows for the application of the classical junction tree algorithm [23]. Spectral methods like the one in [24] relax the binary assignment problem into a continuous one and show that the solution for the continuous problem is the principal eigenvector of the constraints matrix. The solution of the original problem is calculated by thresholding the solution of the continuous problem, which is an approximation — the discrete optimum is not necessarily related to a continuous solution. In [19] this is extended to hyper-graphs and the Eigenproblem is solved efficiently with an iterative algorithm. In [25], a convex-concave programming approach is employed on a least-squares problem of the permutation matrices. Several methods decompose the original problem into sub problems which are solved with different optimization tools like graph cuts [20, 26]. In [27], a multi-label graph cuts minimizer is extended to 2D problems by alternating between labels and nodes. In [28], a candidate graph structure is created and the problem is formulated as a multiple coloring problem on a layered structure. A solution for the resulting integer quadratic programming problem is advanced in [29], the problem is extended to relationships of general order (> 3) and solved with random walks. Finally, in a related paper dynamic programming and graph algorithms [30] are described.

The contributions in this paper are two-fold:

- A theoretical result stating that for the data embedded in space-time, the exact solution to the point set matching problem with hyper-graphs can be calculated in complexity exponential on a small number, which becomes bounded when the hyper-graph is structured using proximity relationships.
- A practical solution to the action recognition problem in videos applying the proposed algorithm to graphs designed with a special structure. This allows calculating matches with computational complexity, which grows linearly in the number of model nodes and linearly in the number of scene nodes.

The paper is organized as follows: Section 2 formulates the graph matching problem and discusses related work on the problem. Section 3 discusses the special properties of the space in which our data are embedded and proposes an exact space-time matching algorithm taking advantage of these properties. In Section 4, we propose a special structure of our model graphs and derive an algorithm which further reduces the computational complexity of the matching algorithm. Section 5 describes the experiments and Section 6 finally concludes.

2 Problem Formulation

In this paper, we formulate the problem as a particular case of the general correspondence problem between two point sets. The objective is to assign points from the model set to points in the scene set, such that some geometrical invariance is satisfied. We solve the problem through a global energy minimization which takes into account a hyper-graph[1] constructed from the model point set. The M points of the model are organized as a hyper-graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of nodes (corresponding to the points) and $\mathcal{E}$ is the set of edges. From now on we will abusively call hyper-graphs "graphs" and hyper-edges "edges". The edges $\mathcal{E}$ in our graph connect sets of three nodes, thus triangles.

While our method requires the data in the model video to be structured into a graph, this is not necessarily so for the data in the scene video. While structural information on the scene data *can* be integrated easily into our formulation, which allows to add structural terms into the minimization framework, and thus results in a classical graph-matching problem. Our formulation is thus more general but can also deal with graph matching.

Each point i of the two sets (model and scene) is also assigned a position $p_i = [\, p_i^{<x>}\ p_i^{<y>}\ p_i^{\langle t \rangle}\ ]^T$ and a feature vector f_i describing the appearance of a local space-time region around this point. When necessary, we will distinguish between model and scene values by the superscripts $^{\langle m \rangle}$ and $^{\langle s \rangle}$: $p_i^{\langle m \rangle}, f_i^{\langle m \rangle}, p_i^{\langle s \rangle}, f_i^{\langle s \rangle}$ etc. Note that symbols in superscripts enclosed in angle brackets $\langle . \rangle$ are not numerical indices, they are mere symbols indicating a category.

Each node i of the model graph is assigned a discrete variable $x_i,\ i = 1..M$, which represents the mapping from the ith model node to some scene node, and can take values from $\{1 \dots S\}$, where S is the number of scene nodes. The whole set of variables x_i is also abbreviated as x. A solution of the problem is given through the values of the x_i, where a value of $x_i = j$ is interpreted as model node i being assigned to scene node j. To handle occlusions, an additional dummy value ϵ is admitted, which semantically means that no assignment has been found for the given variable.

Each combination of assignments x evaluates to an energy value using an energy function $E(x)$. In principle, the energy should be lower for assignments that correspond to a realistic transformation from the model image to the scene image, and it should be high otherwise. We search for the assignments that minimize this energy.

[1] A hyper-graph is a generalization of a graph, where a hyper-edge can connect any number of vertices, typically more than two [31].

Using pairwise edges mostly restricts geometrical coherence constraints to distance similarities, which are not invariant to scale changes. Higher order matching through hyper-graphs has been proposed in the context of object recognition [24]. Typically, hyper-edges connect 3 nodes, which allows to formulate geometrical constraints between pairs of triangles. In particular, geometrical similarity can be measured through angles, which are scale invariant. Our proposed energy function is of the following form:

$$E(x) = \lambda_1 \sum_i U(x_i) + \lambda_2 \sum_{(i,j,k)\in\mathcal{E}} D(x_i, x_j, x_k) \tag{1}$$

where U is a data attached term taking into account feature distances, D is the space-time geometric distortion between two triangles and λ_1 and λ_2 are weighting parameters. For convenience, all dependencies on all values over which we do not optimize have been omitted. U is defined as the Euclidean distance between the appearance features of assigned points, taking into account a penalty W^P for the dummy assignment:

$$U(x_i) = \begin{cases} W^p & \text{if } x_i = \epsilon, \\ ||f_i^{\langle m \rangle} - f_{x_i}^{\langle s \rangle}|| & \text{else.} \end{cases} \tag{2}$$

Since our data is embedded in space-time, angles are projections from 3D+t to 2D, thus include a temporal component not related to scale changes induced by zooming. We therefore split the geometry term D into a temporal distortion term D^t and a spatial geometric distortion term D^g:

$$D(x_i, x_j, x_k) = D^t(x_i, x_j, x_k) + \lambda_3 D^g(x_i, x_j, x_k) \tag{3}$$

where the temporal distortion D^t is defined as truncated time differences over two pairs of nodes of the triangle and geometric distortion D^g is defined over differences of angles.

3 Space-time matching

In our work, the geometric data are embedded in space-time. We assume the following commonly accepted properties of space-time to derive an efficient algorithm:

Hypothesis 1: Causality — Each point in the two sets (i.e., model and scene) lies in a 3-dimensional space :

$(p_i^{\langle x \rangle}, p_i^{\langle y \rangle}, p_i^{\langle t \rangle})$. The spatial and temporal dimensions should not be treated in the same way. While objects (and humans) can undergo arbitrary geometrical transformations like translation and rotation, which is subsumed by geometrical matching invariance in our problem, human actions can normally *not* be reversed.

In a correct match, the temporal order of the points should be retained, which can be formalized as follows

$$\forall\, i, j : p_i^{\langle m \rangle \langle t \rangle} \leq p_j^{\langle m \rangle \langle t \rangle} \Leftrightarrow p_{x_i}^{\langle s \rangle \langle t \rangle} \leq p_{x_j}^{\langle s \rangle \langle t \rangle} \tag{4}$$

Let us recall that the superscript $\langle t \rangle$ stands for the time dimension, and it is not an index.

Hypothesis 2: Temporal closeness — Another reasonable assumption is that the extent of time warping between model and scene time axes must be limited. In other words, two points which are close in time must be close in both the model set and the scene set. This property can be used to further decrease the search space during inference. Since our graph is created from proximity information (we threshold space-time distances between nodes to extract the hyper-edges), it can be formalized as

$$\forall\, i,j,k \in \mathcal{E} : |p_{x_i}^{\langle s\rangle\langle t\rangle} - p_{x_j}^{\langle s\rangle\langle t\rangle}| < T^t \;\vee\; |p_{x_j}^{\langle s\rangle\langle t\rangle} - p_{x_k}^{\langle s\rangle\langle t\rangle}| < T^t \tag{5}$$

Hypothesis 3: Unicity of time instants — We assume that time instants cannot be split or merged. In other words, all points of the same model frame should be matched to points of the same scene frame.

$$\begin{aligned} \forall i,j : (p_i^{\langle m\rangle\langle t\rangle} = p_j^{\langle m\rangle\langle t\rangle}) \Leftrightarrow (p_{x_i}^{\langle s\rangle\langle t\rangle} = p_{x_j}^{\langle s\rangle\langle t\rangle}) \wedge \\ (p_i^{\langle m\rangle\langle t\rangle} \neq p_j^{\langle m\rangle\langle t\rangle}) \Leftrightarrow \;\; (p_{x_i}^{\langle s\rangle\langle t\rangle} \neq p_{x_j}^{\langle s\rangle\langle t\rangle}) \end{aligned} \tag{6}$$

In [32], we showed that (under these hypotheses) the complexity of exactly minimizing in Eq. (1) is exponential only on the maximum number of points per frame, which is typically a small number, e.g., $1-4$. However, in practice and for general graphs it is still too high for practical usage. The next section will introduce a special structure which further decreases complexity.

4 A special graphical structure

Since most formulations of point set matching or graph matching problems in computer vision are NP-complete, one classically resorts to approximate solutions. In this work we advocate an alternative and perhaps better idea, which is to approximate the problem — the graphical structure in this case — and to solve the new problem exactly. This is particular appealing in point matching problems where the structure of the graph is less related to the description of the object, but rather to the constraints of the matching process. We recall that the graphical structure is obtained from adjacency or proximity information, so changing it will not significantly harm the description of the space-time object.

We propose to structure the model points as follows:

- We keep a single point in each model frame by choosing the most salient one, i.e. the ones with the highest confidence of the interest point detector. However, no restrictions are applied to the scene frames, which may contain an arbitrary number of points.
- Each model point i is connected to its two immediate predecessors $i-1$ and $i-2$ as well as to its two immediate successors $i+1$ and $i+2$.

This creates a planar graph with triangular structure, as illustrated in Figure 2. The general case of the energy function (1) can be simplified in this case. The neighborhood system can be described in a very simple way using the index of the nodes of the graph,

similar to the dependency graph of a second order Markov chain:

$$E(x) = \sum_{i=1}^{M} U(x_i) + \sum_{i=3}^{M} D(x_i, x_{i-1}, x_{i-2}). \tag{7}$$

The general recursive formula of the inference algorithm can be derived as

$$\alpha_i(x_{i-1}, x_{i-2}) = \min_{x_i} \Big[U(x_i) + D(x_i, x_{i-1}, x_{i-2}) + \alpha_{i+1}(x_i, x_{i-1}) \Big] \tag{8}$$

with the initialization

$$\alpha_M(x_{M-1}, x_{M-2}) = \min_{x_M} \left[U(x_M) + D(x_M, x_{M-1}, x_{M-2}) \right]. \tag{9}$$

During the calculation of the trellis, the arguments of the minima in equation (8) are stored in a table $\beta_i(x_{i-1}, x_{i-2})$. Once the trellis completed, the optimal assignment can be calculated through classical backtracking:

$$\widehat{x_i} = \beta_i(x(i-1), x(i-2)), \tag{10}$$

starting from an initial search for x_1 and x_2:

$$(\widehat{x_1}, \widehat{x_2}) = \arg\min_{x_1, x_2} [U(x_1) + U(x_2) + \alpha_3(x_1, x_2)]. \tag{11}$$

The algorithm as given above is of complexity $O(M{\cdot}S^3)$: a trellis is calculated in a $M \times S \times S$ matrix, where each cell requires to iterate over S possible combinations.

Exploiting the different hypotheses on the spatio-temporal data introduced in section 2, the complexity can be decreased further:

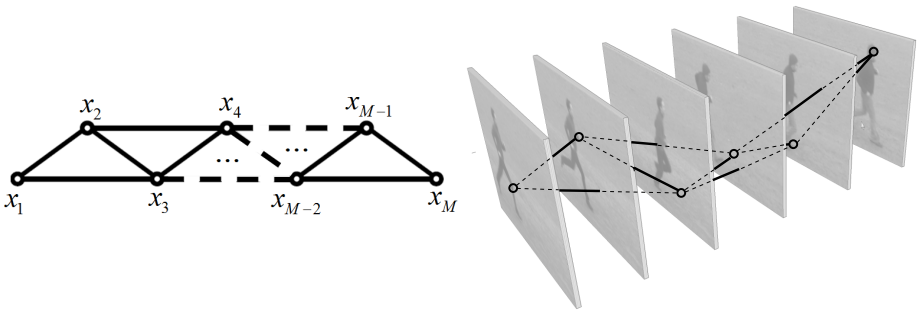
Ad) Hypothesis 1 — taking causality constraints into account we can prune many combinations from the trellis of the optimization algorithm. In particular, if we calculate possibilities in the trellis given a certain assignment for a given variable x_i, all values for the predecessors x_{i-1} and x_{i-2} must be necessarily *before* x_i, i.e. lower.

Ad) Hypothesis 2 — similar as above, given a certain assignment for a given variable x_i, we will allow a maximum number of T^t possibilities for the values of the successors x_{i-1}, x_{i-2}, which are required to be *close*.

Thus, the expression in equation (8) is only calculated for combinations satisfying the following constraints:

$$\begin{array}{c} |x_i - x_{i-1}| < T^t \wedge |x_{i-1} - x_{i-2}| < T^t \wedge \\ x_i > x_{i-1} \quad \wedge \quad x_{i-1} > x_{i-2}. \end{array} \tag{12}$$

These pruning measures decreases the complexity to $O(M{\cdot}S{\cdot}{T^t}^2)$, where T^t is a small constant measured in the number of frame, so the complexity is linear on the number of points in the scene: $O(M{\cdot}S)$. For example, letting the number of model nodes and scene nodes be $M = 30$ and $S = 500$, respectively, we achieve 2500 fold complexity reduction when $T^t = 10$.



(a) (b)

Fig. 2. (a) A special graphical structure for the model point set designed for very low computational complexity: a second order chain. No requirements whatsoever are imposed on the scene point set, however. (b) An example model graph.

	B	HC	HW	J	R	W
B	**100**	0	0	0	0	0
HC	0	**100**	0	0	0	0
HW	3	26	**71**	0	0	0
J	0	0	0	**69**	31	0
R	0	0	0	25	**75**	0
W	0	0	0	3	3	**94**

(a)

	B	HC	HW	J	R	W
B	**100**	0	0	0	0	0
HC	3	**97**	0	0	0	0
HW	6	15	**79**	0	0	0
J	0	0	0	**72**	28	0
R	0	0	0	8	**89**	3
W	0	0	0	0	0	**100**

(b)

Table 1. Confusion matrix without (a) and with (b) prototype selection. Respective accuracies: 84.8%, 89.3%. (B: Box, HC: Handclap, HW: Handwave, J: Jog, R: Run, W: Walk).

Method	B	HC	HW	J	R	W	Tot.
Laptev et al. [33]	97	95	91	89	80	99	91.8
Schuldt et al. [34]	98	60	74	60	55	84	71.8
Li et al. [35]	97	94	86	**100**	83	97	**92.8**
Niebles et al. [36]	99	**97**	**100**	78	80	94	91.3
Our method	**100**	**97**	79	72	**88**	**100**	89.3

Table 2. Comparison with existing methods using the same KTH dataset protocol. (B: Box, HC: Handclap, HW: Handwave, J: Jog, R: Run, W: Walk).

5 Experimental Results

We tested the proposed method on the widely used public KTH dataset [34]. It includes 25 subjects performing 6 actions (*walking*, *jogging*, *running*, *handwaving*, *handclapping* and *boxing*) recorded in four different scenarios including indoor/outdoor scenes and different camera viewpoints. Spatio-temporal interest points extracted with the 3D Harris detector [33] constitute the nodes of the proposed graphical structure. Appearance features f_i are the well known HoG/HoF extracted with the publicly available code

from [33]. As mentioned in section 4, we choose a single point per model frame based on the confidence score of the detector. All points are kept for testing videos.

The parameters are fixed as follows. The penalty parameter W_P should theoretically be higher than the average local energy of correctly assigned triangles and lower than the average local energy of incorrectly assigned triangles. We estimate it by sampling energies (without penalty) of pairs of training sequences in two settings: intra-class and inter-class, resulting in two histograms of local energies. We set $W^P = 8.4$ as the point of minimal Bayes error. The weighting parameters are optimized over the validation set: $\lambda_1 = 0.6$, $\lambda_2 = 0.1$, $\lambda_3 = 10$, $T^t = 30$, and $W^t = 60$.

First, we build up a model dictionary using 16 training subjects, 383 sequences. We generate several model graphs by partitioning the sequences into subsequences each containing between 20 to 30 number of frames with salient interest points. This results in 1429 model graphs in total. Action classes on the unseen subjects are recognized with a nearest neighbor (NN) classifier where the distance is defined as the matching energy (1). The average recognition performance of the proposed scheme is found to be 84.8%. The main cause of this modest performance is the poor discrimination between the *jogging* and *running* classes (see Table 1a). The algorithm also suffers from *handwaving*, while significantly successful in *boxing*, *handclapping* and *walking*. We conjecture that this issue can be handled by a prototype selection algorithm.

Prototype selection — In prototype-based approaches, prototype selection plays a key role in recognition performance. Intra-variation can be large among action categories; some categories need different numbers of views or different categories can be similar, thus misleading, in the graph-space constructed. We balanced and optimized the dictionary with Sequential Floating Backward Search (SFBS), which removes irrelevant model graphs from the training set. SFBS has been successfully used as a supervised feature selection method in many previous studies [37]. Briefly, we start with a full dictionary and proceed to remove conditionally the least significant models from the set, one at a time, while checking the performance variations. Deletions which improve the performance are made permanent in this greedy search. After a number of removal steps, we reintroduce one or more of the removed ones provided they improve the performance. At each step, performance is evaluated on a validation set. We use the same data partition protocol (8/8/9) as proposed in [34]. We select 44 models out of 705 as our best subset of model graphs, which increased test performance to 89.3%. As expected, the *handwaving* and *running* sequences benefit the most from dictionary learning (see Table 1b).

Sample matched model and scene sequences are illustrated in Figure 3, where the first three actions (*handwaving*, *boxing*, *walking*) are successfully recognized while the last one (*running*) gives an example of misclassification. Table 2 proves that our method has a comparable performance with state-of-the-art methods in the literature while using much less information. We want to point out that many results have been published on the KTH database, but the protocols are not comparable for most of them, see the excellent review in [38]. In the figure, we chose results obtained with the same protocol.

The algorithm has been implemented in Matlab. Matching each model graph is done simultaneously with 0.02 seconds per frame, i.e. for an average scene of 30 seconds ($S = 750$) recognition takes 13.8 seconds on a CPU with 3.33GHz and 4GB RAM.

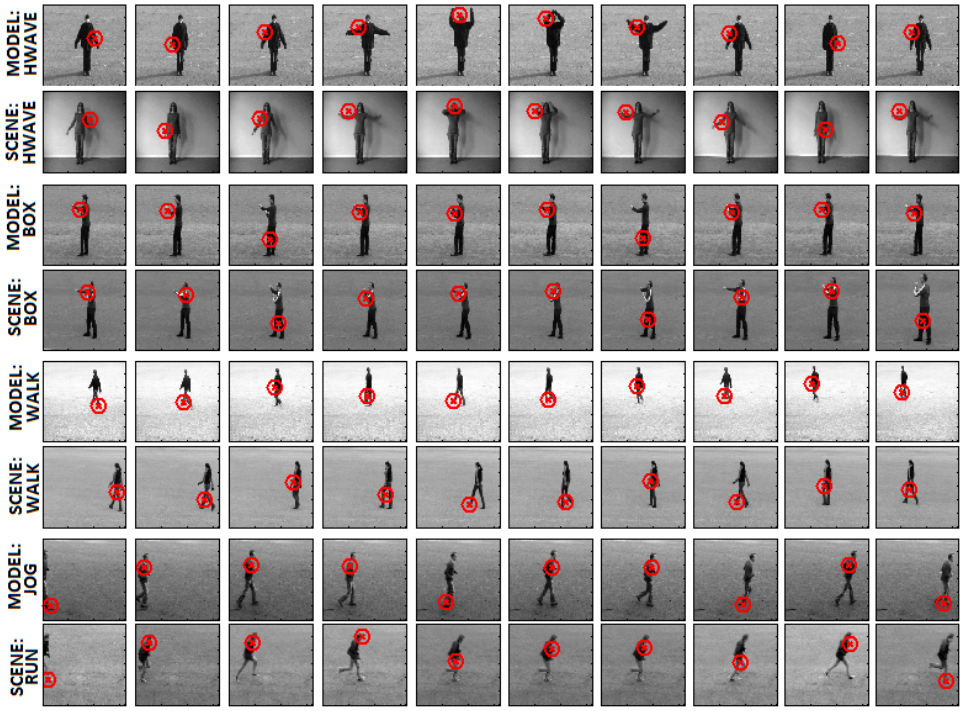


Fig. 3. Examples for matched sequences: while the top three matches result in correct recognition, the bottom match is misclassified.

A real-time GPU implementation– A first preliminary GPU implementation allows real-time performance on standard medium end GPUs, e.g. a Nvidia GeForce GTS450. Table 3 compares run times of the CPU implementation in Matlab/C and the GPU implementation running on different GPUs with different characteristics, especially the number of calculation units. The run times are given for matching a single model graph with 30 nodes against scene blocks of different lengths. If the scene video is cut into smaller blocks of 60 frames, which is necessary for continuous video processing anyway, than real time performance can be achieved even on the low end GPU model. With these smaller chunks of scene data, matching all 44 graph models to a block of 60 frames (roughly 2 seconds of video) takes roughly 3ms regardless of the GPU model.

The processing time of 3ms/fr is very much lower than the limit for real time processing, which is 40ms for video acquired at 25fps. Additional processing will be required in order to treat overlapping blocks, which increases running time to 6ms/fr. The times given above also do not include interest point detection and feature extraction, but these are negligible compared to the matching requirements and can also be calculated on a GPU. This system is currently being integrated in our mobile robotics platform.

Implementation	Nodes	Frames	Time (ms)	Time/fr (ms)	Time/fr (ms)
			— — A single model — —		— All 44 models —
CPU: Intel Core 2 Duo, E8600 @ 3.33Ghz, Matlab/C(mex)	754	723	13800	19.09	840
Nvidia GeForce GTS450, 192 cuda cores, 128 bit memory interface	754	723	748	1.03	45
	60	55	4	0.07	3 (real time)
Nvidia GeForce GTX560, 336 cuda cores, 256 bit memory interface	754	723	405	0.56	25 (real time)
	60	55	4	0.07	3 (real time)

Table 3. Running times in milliseconds for two different GPUs and for 4 different scene block sizes. The last column on the right gives times **per frame** for matching the whole set of 44 model graphs. Times < 40ms mean real time processing.

6 Conclusions and Future Work

In this paper we showed that — when the data is embedded in space-time — the exact solution to the point set matching problem with hyper-graphs can be calculated in complexity exponential on a small number, which is bounded when the hyper-graph is structured with proximity information. As a second contribution we presented a special graphical structure which allows to perform exact matching with very low complexity, linear in the number of the model nodes and the number of scene nodes. The method has been tested on the KTH dataset where it shows competing performance with very low runtime.

Our current work concentrates on extension of graphical structure to more than one interest point per frame. This idea is formulated through ”meta” graph or ”frame” graph matching in which each node in the graph corresponds to a frame of the video and each frame is characterized by ST interest points and triangles. Following this, we will use more realistic videos, e.g., UCF sports action database [39].

References

1. Sivic, J., Zisserman, A.: Video google: a text retrieval approach to object matching in videos. In: ICCV. Volume 2. (2003) 1470–1477
2. Borzeshi, E.Z., Piccardi, M., Xu, R.Y.D.: A discriminative prototype selection approach for graph embedding in human action recognition. In: ICCVW. (2011)
3. Liu, J., Ali, S., Shah, M.: Recognizing human actions using multiple features. In: CVPR. (2008)
4. Raja, K., Laptev, I., Prez, P., Oisel, L.: Joint pose estimation and action recognition in image graphs. In: ICIP. (2011)
5. Gaur, U., Zhu, Y., Song, B., Roy-Chowdhury, A.: A string of feature graphs model for recognition of complex activities in natural videos. In: ICCV. (2011)
6. Brendel, W., Todorovic, S.: Learning spatiotemporal graphs of human activities. In: ICPR. (2011)

7. Ta, A.P., Wolf, C., Lavoue, G., Başkurt, A.: Recognizing and localizing individual activities through graph matching. In: AVSS. (2010)
8. Niebles, J.C., Fei-Fei, L.: A hierarchical model of shape and appearance for human action classification. In: CVPR. (2007) 1–8
9. Savarese, S., Delpozo, A., Niebles, J., Fei-Fei, L.: Spatial-temporal correlatons for unsupervised action classification. In: WMVC, Los Alamitos, CA (2008)
10. Ryoo, M.S., Aggarwal, J.K.: Spatio-temporal relationship match: video structure comparison for recognition of complex human activities. In: ICCV. (2009)
11. Mikolajczyk, K., Uemura, H.: Action recognition with appearance motion features and fast search trees. CVIU **115**(3) (2011) 426–438
12. Filipovych, R., Ribeiro, E.: Robust sequence alignment for actor-object interaction recognition: Discovering actor-object states. CVIU **115** (2011) 177–193
13. Chen, C., Grauman, K.: Efficient activity detection with max-subgraph search. In: CVPR. (2012)
14. Zhang, L., Zeng, Z., Ji, Q.: Probabilistic image modeling with an extended chain graph for human activity recognition and image segmentation. IEEE Tr. on IP (2011)
15. Dyana, A., Das, S.: Trajectory representation using gabor features for motion-based video retrieval. Pattern Recognition Letters **30**(10) (2009) 877–892
16. Cuntoor, N., Yegnanarayana, B., Chellappa, R.: Activity modeling using event probability sequences. IEEE Tr. on IP **17**(4) (2008) 594–607
17. Abdelkader, M.F., Abd-Almageed, W., Srivastava, A., Chellappa, R.: Silhouette-based Gesture and Action Recognition via Modeling Trajectories on Riemannian shape manifolds. CVIU **115**(3) (2010) 439–455
18. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. IJPRAI **18**(3) (2004) 265–298
19. Duchenne, O., Bach, F.R., Kweon, I.S., Ponce, J.: A tensor-based algorithm for high-order graph matching. In: CVPR. (2009) 1980–1987
20. Torresani, L., Kolmogorov, V., Rother, C.: Feature correspondence via graph matching: Models and global optimization. In: ECCV. (2008) 596–609
21. Zampelli1, S., Deville, Y., Solnon, C.: Solving subgraph isomorphism problems with constraint programming. Constraints (2009)
22. T.S.Caetano, Caelli, T., Schuurmans, D., Barone, D.: Graphical models and point pattern matching. IEEE Tr. on PAMI **28**(10) (2006) 1646–1663
23. Lauritzen, S., Spiegelhalter, D.: Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society B **50** (1988) 157–224
24. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: ICCV, Washington, DC, USA (2005) 1482–1489
25. Zaslavskiy, M., Bach, F., Vert, J.: A path following algorithm for the graph matching problem. IEEE Tr. on PAMI **31**(12) (2009) 2227–2242
26. Zeng, Y., Wang, C., Wang, Y., Gu, X., Samaras, D., Paragios, N.: Dense non-rigid surface registration using high-order graph matching. In: CVPR. (2010)
27. Duchenne, O., Joulin, A., Ponce, J.: A graph-matching kernel for object categorization. In: ICCV. (2011)
28. Lin, L., Zeng, K., Liu, X., Zhu, S.C.: Layered graph matching by composite cluster sampling with collaborative and competitive interactions. CVPR **0** (2009) 1351–1358
29. Leordeanu, M., Zanfir, A., Sminchisescu, C.: Semi-supervised learning and optimization for hypergraph matching. In: ICCV 2011. (2011)
30. Felzenszwalb, P., Zabih, R.: Dynamic programming and graph algorithms in computer vision. IEEE Tr. on PAMI **33**(4) (2011) 721–740

31. Zass, R., Shashua, A.: Probabilistic graph and hypergraph matching. In: CVPR. (2008)
32. Çeliktutan, O., Wolf, C., Sankur, B.: Fast exact matching and correspondence with hypergraphs on spatio-temporal data. LIRIS UMR 5205 CNRS/INSA de Lyon/Universit'e Claude Bernard Lyon 1/Universit'e Lumi'ere Lyon 2/Ecole Centrale de Lyon **Report No. RR-LIRIS-2012-002** (2012)
33. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR. (2008) 1–8
34. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local svm approach. In: ICPR. (2004) 32–36
35. Li, B., Ayazoğlu, M., Mao, T., Camps, O.I., Sznaier, M.: Activity recognition using dynamic subspace angles. In: CVPR. (2011)
36. Niebles, J.C., Chen, C.W., Fei-Fei, L.: Modelling temporal sturcture of decomposable motion segments for activity classification. In: ECCV. (2010) 1–14
37. Pudil, P., Ferri, F.J., Novovicov, J., Kittler, J.: Floating search methods for feature selection with non-monotonic criterion functions. In: ICPR. (1994) 279–283
38. Gao, Z., Chen, M., Hauptmann, A., A.Cai: Comparing evaluation protocols on the kth dataset. In: Human Behavior Understanding. Volume LNCS 6219. (2010) 88–100
39. Rodriguez, M.D., Ahmed, J., Shah, M.: Action mach: A spatio-temporal maximum average correlation height filter for action recognition. In: CVPR. (2008)