Fast Exact Matching and Correspondence with Hyper-graphs on Spatio-temporal Data

Technical Report LIRIS-RR-2012-002

LIRIS UMR CNRS 5205
Université de Lyon, France

Oya Celiktutan [a] Christian Wolf [b] Bülent Sankur [a]

[a]Electrical and Electronics Engineering
Bogazii University, Istanbul, Turkey
{oya.celiktutan,bulent.sankur}@boun.edu.tr

[b]Universit de Lyon, CNRS
INSA-Lyon, LIRIS, UMR CNRS 5205, F-69621, France
christian.wolf@liris.cnrs.fr

Abstract

Graphs and hyper-graphs are frequently used to recognize complex and often non-rigid patterns in computer vision, either through graph matching or point-set matching with graphs. Most formulations resort to the minimization of a difficult energy function containing geometric or structural terms, frequently coupled with data attached terms involving appearance information. Traditional methods solve the minimization problem approximately, for instance with spectral techniques. In this paper we deal with data embedded in a 3D "space-time", for instance in action recognition applications. We show that, in this context, we can take advantage of special properties of the time domain, in particular causality and the linear order of time. We show that the complexity of the exact matching problem is far inferior to the complexity of the general problem and we derive an algorithm calculating the exact solution. As a second contribution, we propose a new graphical structure which is elongated in time. We argue that, instead of approximately solving the original problem, a better solution can be obtained by exactly solving an approximated problem. An exact minimization algorithm is derived for this structure and successfully applied to action recognition in videos.

1. Introduction

In many applications involving the recognition of complex visual patterns, as for instance recognition of object classes or of 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 [35]. On the other hand, graphs (and hyper-graphs) form a natural description of this type of data and graph matching algorithms yield distances between graphs, so that classification schemes such as k-nearest neighbor can be used.

In this work we concentrate on hyper-graph matching and point set matching, where the nodes of the graph(s) correspond to space-time points, and the neighborhood relationship is derived from proximity information. The goal application in this work is human action recognition, where 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 two recent papers only: in [4] graphs are built from adjacency relationships of space-time tubes produced from oversegmenting the test video, and in [37] graphs are built from proximity by thresholding distances in space time. Both 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.

Since the literature on human action detection has become vast, we opt to focus our review only on methods making explicit use of 3D space-time geometry, and otherwise we refer the interested reader to some recently published surveys [2, 29]

Several methods for the detection of space-time interest points have been proposed: periodic points [7], the 3D Harris corner detector [16], extension of SIFT to space-time [34], the 3D Hessian [40] and Gabor filters [28]. They are usually combined with appearance and motion features like cuboids (concatenation of gradient values) [7], spatial-temporal jets [33], histogram of gradient and optical flow values (HoG and HoF) [16] or 3D SIFT [34].

Among methods that use space-time 3D geometry, one can cite [17], which divides the space-time volume into several grids and construct spatio-temporal histograms, each referenced with its grid position. In [27], 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 [32], the correlation of spatio-temporal (ST) patterns is measured and ST correlograms are constructed. Similarly, in [24], ST correlograms are employed and mutual information is maximized in clustering and determining the number of clusters. Pairwise spatio-temporal relations are introduced in [31], based on a set of rules, and this information is transformed into 3D histograms.

Alternative methods consider an action as a three-dimensional object in space-time. Examples are motion history images [3] and solutions of the Poisson equation [14]. Although these methods are relatively simple, they can properly work in controlled settings only. In real life situations, several confounding factors such as background clutter, illumination variations, shadows, clothing and camera resolution can affect the performance severely. Optical flow features introduce dense motion information but suffer from high computational complexity, e.g. [15]. In [25], 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, states, and appearance.

The linear nature of the time dimension is frequently used to devise methods based on sequence alignment. Examples are dynamic time warping, trajectory matching with Gabor filters [10], accumulated co-occurrence data of trajectories [36] etc. In [6], salient state transitions in HMMs are learned. In [45], a chain graph model for action recognition exploits a priori knowledge of the nature and semantics of relationships between different variables. In [1], the evolution of silhouettes over time is modelled.

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 minimization algorithm we derived is capable of dealing with classical energy functions including unary, binary and ternary terms, which makes it possible to include scale invariant potentials, as e.g. the formulations in [5, 8, 38] and others.

Techniques for graph matching and for point set matching with graphs have been studied intensively in the field of pattern recognition. While the graph isomorphism problem can be calculated in polynomial time, it is widely known that exact subgraph matching is NP-complete [38], as is subgraph isomorphism [41]. Formulations like the one in (1) are known to be NP-hard [38]. In fact, for two graphs $\mathcal{G}_M$ and $\mathcal{G}_S$ with respective numbers of nodes M and S, the brute force method needs to take into account S^M possible assignments over the whole set of nodes in $\mathcal{G}_M$. 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 [39]: 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 [18]. In [43] a hyper-graph matching method is presented which proceeds through convex optimization of the relaxed problem in a probabilistic setting. Recently, [8] generalized the spectral matching method for pairwise graphs in [20] to hyper-graphs by using a tensor-based algorithm which solves an eigen-problem on the assignment matrix. In [42], 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 [38, 44]. In [9], a multi-label graph cuts minimizer is extended to 2D problems by alternating between labels and nodes. In [23], 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 [21] and in [19], 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 [11] 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

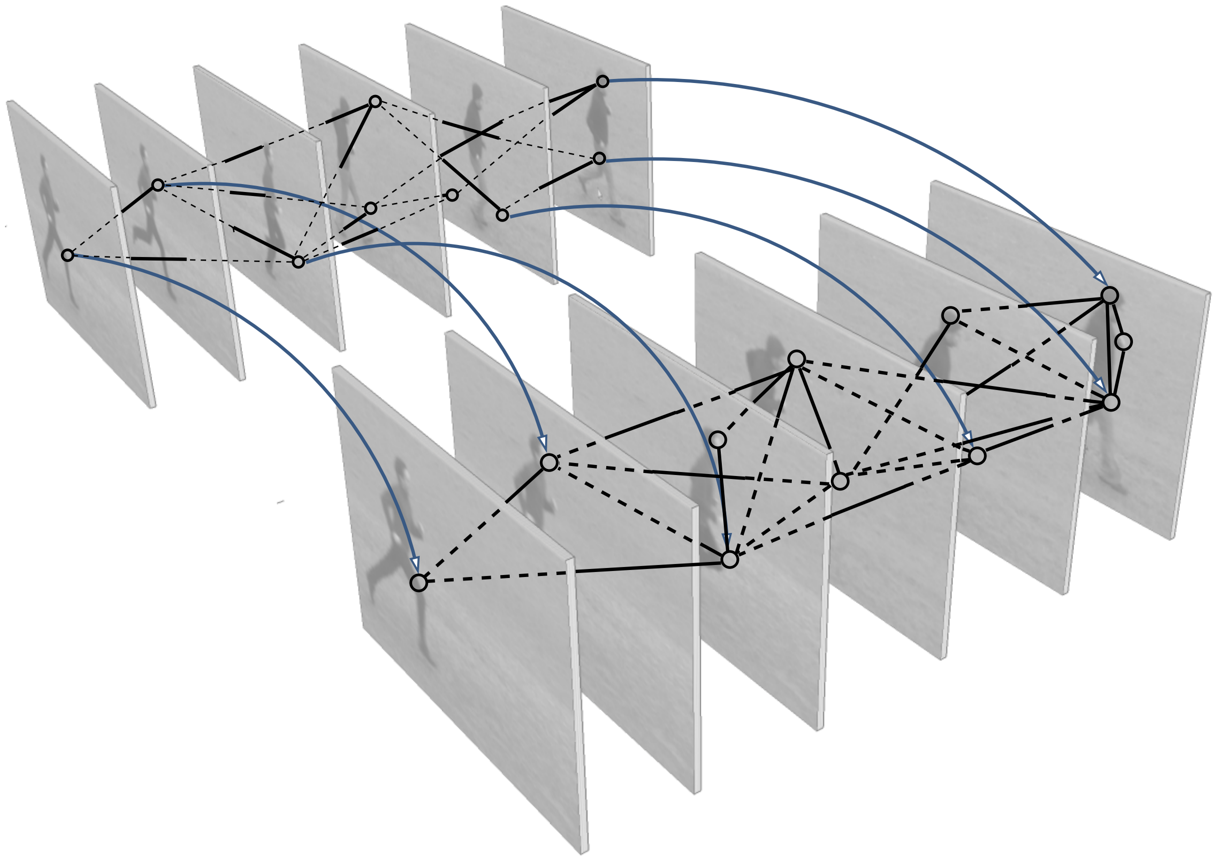

Figure 1. A model point cloud structured into a hyper-graph is matched with a scene point cloud, eventually but not necessarily structured into a hyper-graph.

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. Figure 1 illustrates the problem.

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, giving a classical graph-matching problem, this is not necessary for the method to work. 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 distin-

[1] A hyper-graph is a generalization of a graph, where a hyper-edge can connect any number of vertices, typically more than two [43].

guish 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 $\{1 \ldots S\}$, 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.

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 [20]. 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:

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

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:

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

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:

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

where the temporal distortion D^t is defined as truncated time differences over two pairs of nodes of the triangle:

$$\begin{aligned} &D^t(x_i, x_j, x_k) = \\ &= \begin{cases} W^t & \text{if } \Delta(i,j) > T^t \ \vee \Delta(j,k) > T^t \\ \Delta(i,j) + \Delta(j,k) & \text{else} \end{cases} \end{aligned} \qquad (4)$$

Here, $\Delta(a, b)$ is the time distortion due to the assignment of node pair (a, b):

$$\Delta(a,b) = |(p_a^{\langle m\rangle\langle t\rangle} - p_b^{\langle m\rangle\langle t\rangle}) - (p_{x_a}^{\langle s\rangle\langle t\rangle} - p_{x_b}^{\langle s\rangle\langle t\rangle})| \qquad (5)$$

and D^g is defined over differences of angles:

$$D^g(x_i, x_j, x_k) = \left\| \begin{array}{l} \phi^{\langle m\rangle}(i,j,k) - \phi^{\langle s\rangle}(x_i, x_j, x_k) \\ \phi^{\langle m\rangle}(j,i,k) - \phi^{\langle s\rangle}(x_j, x_i, x_k) \end{array} \right\| \qquad (6)$$

Here, $\phi^{\langle m\rangle}(a,b,c)$ and $\phi^{\langle s\rangle}(a,b,c)$ denote the angles subtended at point b for, respectively, model and scene triangles indexed by (a, b, c).

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

$$\forall\, i,j : p_i^{\langle m\rangle\langle t\rangle} \le p_j^{\langle m\rangle\langle t\rangle} \Leftrightarrow p_{x_i}^{\langle s\rangle\langle t\rangle} \le p_{x_j}^{\langle s\rangle\langle t\rangle} \qquad (7)$$

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

$$\forall\, i,j,k \in \mathcal{E} : |p_{x_i}^{\langle s\rangle\langle t\rangle} - p_{x_j}^{\langle s\rangle\langle t\rangle}| < T^t \vee |p_{x_j}^{\langle s\rangle\langle t\rangle} - p_{x_k}^{\langle s\rangle\langle t\rangle}| < T^t \qquad (8)$$

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.

$$\begin{array}{lcl} \forall i,j: \quad (p_i^{\langle m\rangle\langle t\rangle} = p_j^{\langle m\rangle\langle t\rangle}) & \Leftrightarrow & (p_{x_i}^{\langle s\rangle\langle t\rangle} = p_{x_j}^{\langle s\rangle\langle t\rangle}) \wedge \\ \quad (p_i^{\langle m\rangle\langle t\rangle} \neq p_j^{\langle m\rangle\langle t\rangle}) & \Leftrightarrow & (p_{x_i}^{\langle s\rangle\langle t\rangle} \neq p_{x_j}^{\langle s\rangle\langle t\rangle}) \end{array} \tag{9}$$

Matching

Hypothesis nr. 3 implies that a correct sequence match consists of a collection of single model frame to scene frame matches. We therefore first reformulate the energy function in equation (1) by splitting each variable x_i into two subsumed variables z_i and $x_{i,l}$, which are interpreted as follows: z_i denotes the index of the scene frame the model frame is matched to. The number of model frames is denoted $\overline{M}$. Each frame i also possesses a number $\overline{M}_i$ of node variables $x_{i,1}, \ldots, x_{i,M_i}$, where $x_{i,1}$ denotes the node number in the scene graph the model node $x_{i,1}$ is matched to. Note that the number of possible values for variable $x_{i,l}$ depends on the value of z_i, since different frames may contain different amounts of nodes. For convenience we will also simplify the notation by representing a hyper-edge (and the corresponding frame indices and node indices) as c and the corresponding variables as (z_c, x_c); we also drop the parameters λ_1 and λ_2 which can be absorbed into the potentials U and D. The reformulated energy function is now given as:

$$E(z,x) = \sum_{(i,l)\in\overline{M}\times\overline{M}_i} U(z_i, x_{i,l}) + \sum_{c\in\mathcal{E}} D(z_c, x_c) \tag{10}$$

We now introduce a decomposition of the set of hyper-edges $\mathcal{E}$ into disjoint subsets $\mathcal{E}^i$, where $\mathcal{E}^i$ is the set of all hyper-edges which contain at least one node with temporal coordinate equal to i and no node has a higher (later) temporal coordinate. It is clear that the set of all possible sets $\mathcal{E}^i$ forms a complete partition of $\mathcal{E}$, i.e. $\mathcal{E} = \bigcup_i \mathcal{E}^i$. We can now exchange sums and minima according to this partitioning:

$$\begin{array}{ll} \min\limits_{z,x} E(z,x) = & \\ \min\limits_{z_1;x_{1,1},\ldots,x_{1,\overline{M}_1}} & \left[\sum\limits_{l=1}^{\overline{M}_1} U(z_1, x_{1,l}) + \sum\limits_{c\in\mathcal{E}^1} D(z_c, x_c) + \right. \\ \min\limits_{z_2;x_{2,1},\ldots,x_{2,\overline{M}_2}} & \left[\sum\limits_{l=1}^{\overline{M}_2} U(z_2, x_{2,l}) + \sum\limits_{c\in\mathcal{E}^2} D(z_c, x_c) + \right. \\ & \vdots \\ \min\limits_{z_{\overline{M}};x_{\overline{M},1},\ldots,x_{\overline{M},\overline{M}_{\overline{M}}}} & \left[\sum\limits_{l=1}^{\overline{M}_{\overline{M}}} U(z_{\overline{M}}, x_{\overline{M},l}) + \sum\limits_{c\in\mathcal{E}^{\overline{M}}} D(z_c, x_c)\right]\vdots\Bigg] \end{array} \tag{11}$$

In general the hyper-edges have variable temporal spans, which makes it impossible to define a recursion scheme with regular structure. We therefore define the concept of the *reach* $\mathcal{R}^i$ of frame i, which consists of the set of edges which reach into the past of frame i and which are part of i or its future:

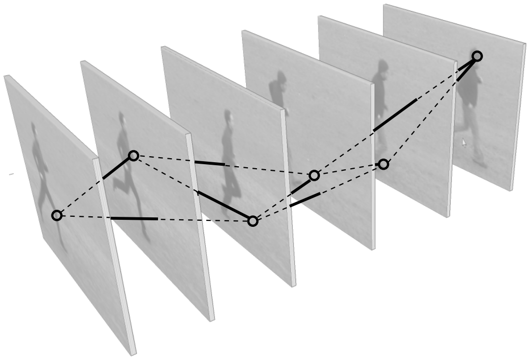

Figure 2. A special graphical structure for the model point set designed for very low computational complexity: a second order chain. No requirements whatsoever are imposed on the scene point set, however.

$$\mathcal{R}^i = \left\{c \in \mathcal{E} : [\min{}^{\langle t\rangle}(c) < i] \wedge [\max{}^{\langle t\rangle}(c) \geq i]\right\} \tag{12}$$

where $\min^{\langle t\rangle}(c)$ and $\max^{\langle t\rangle}(c)$ are, respectively, the minimum and the maximum temporal coordinate of the nodes of edge c. Note that, per definition $\mathcal{E}^i \subseteq \mathcal{R}^i$.

We also introduce the expression $\mathcal{X}^i$ for the set of all variables z_i or $x_{i,j}$ involved in the edges of the reach $\mathcal{R}^i$:

$$\begin{array}{rl} \mathcal{X}^i = & \left\{z_j : \exists k : (j,k) \in c \wedge c \in \mathcal{R}^i\right\} \cup \\ & \left\{x_{j,k} : (j,k) \in c \wedge c \in \mathcal{R}^i\right\} \end{array} \tag{13}$$

Finally, the set of reach variables $\mathcal{R}^i$ without the variables of frame i itself is denoted as $\mathcal{X}^{i-}$

$$\mathcal{X}^{i-} = \mathcal{X}^i \setminus \{z_i\} \setminus \{x_{i,k} : k \in \{1\ldots\overline{M}_i\} \tag{14}$$

Now a recursive calculation scheme for (11) can be devised by defining a recursive variable α_i which minimizes the variables of a given frame as a function of the variables of its reach:

$$\begin{array}{rcl} \alpha_i(\mathcal{X}^{i-}) & = & \min\limits_{z_i;x_{i,1},\ldots,x_{i,\overline{M}_i}} \left[\sum\limits_{l=1}^{N_i} U(z_i, x_{i,l}) + \right. \\ & + & \left. \sum\limits_{c\in\mathcal{E}^i} D(z_c, x_c) + \alpha_{i+1}(\mathcal{X}^{(i+1)^-})\right] \end{array} \tag{15}$$

For this recursion to work the following relation must hold, which it does per definition: $\mathcal{X}^{(i+1)^-} \subseteq (\mathcal{X}^{i-} \cup z_i; x_{i,1}, \ldots, x_{i,\overline{M}_i})$.

Calculation starts at the last frame $i = \overline{M}$ and iterates by calculating α_i from α_{i+1}. At each step, a minimum is calculated over all variables of frame i for all possible values of the variables in $\mathcal{X}^{i^-}$. The computational complexity thus depends on the number of variables in the reach $\mathcal{R}^i$ and on the size of their domain:

$$O\left(\max_i \left[\prod_{v\in\mathcal{V}^i} |\text{domain}(v)|\right]\right) \approx O\left(\max_i \left[\overline{S}^{|\mathcal{X}_z^i|}\langle\langle s\rangle\rangle^{|\mathcal{X}_x^i|}\right]\right) \tag{16}$$

where $\overline{S}$ is the number of scene frames, $\langle\langle s\rangle\rangle$ is the average number of nodes per frame, and $|\mathcal{X}_z^i|$ is the number of variables of set z in $\mathcal{X}^i$. The complexity is thus very much lower than the complexity of the brute force approach, which is given by $O(S^M M|\mathcal{E}|)$. Let us recall that S is the total number of nodes in the scene and M is the total number of nodes in the model, i.e. $S \gg \overline{S}$ and $S \gg \langle\langle s\rangle\rangle$. Furthermore, both $|\mathcal{X}_z^i|$ and $|\mathcal{X}_x^i|$ are bounded and low when the graph is constructed from proximity constraints. However, in practise 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

Most formulations of point set matching or graph matching problems in computer vision are NP-complete. Since exact minimization is infeasible 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. As mentioned in section 3, a similar philosophy has been put forward by [39] in the context of object recognition, where the graph is structured into a k-tree.

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 variable split into pairs (z_i, x_i) we introduced in section 3 is not necessary anymore. Furthermore, 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:

$$E(x) = \sum_{i=1}^{M} U(x_i) + \sum_{i=3}^{M} D(x_i, x_{i-1}, x_{i-2}) \tag{17}$$

The reach of this structure is constant and consists of two edges only, as $\mathcal{R}^i = \{(x_{i-2}, x_{i-1}, x_i), (x_{i-1}, x_i, x_{i+1})\}$; the set of reach variables is also constant $\mathcal{X}^i = \{x_{i-2}, x_{i-1}, x_i\}$. The general recursive formula of the inference algorithm can be derived as

$$\begin{aligned}\alpha_i(x_{i-1}, x_{i-2}) = \min_{x_i} \Big[\quad & U(x_i) + D(x_{i-2}, x_{i-1}, x_i) \\ & + \alpha_{i+1}(x_i, x_{i-1}) \Big]\end{aligned} \tag{18}$$

with the initialization

$$\alpha_M(x_{M-1}, x_{M-2}) = \min_{x_M} \left[U(x_M) + D(x_M, x_{M-1}, x_{M-2})\right] \tag{19}$$

During the calculation of the trellis, the arguments of the minima in equation (18) 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:

$$\widehat{x_i} = \beta_i(x(i-1), x(i-2)), \tag{20}$$

starting from an initial search for x_1 and x_2:

$$(\widehat{x_1}, \widehat{x_2}) = \arg\min_{x_1,x_2} \left[U(x_1) + U(x_2) + \alpha_3(x_1, x_2)\right] \tag{21}$$

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 corresponds to a possible value of a given variable. The calculation of each cell requires to iterate over all S^2 possible combinations of its two successors.

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 (18) is only calculated for values (x_{i-1}, x_{i-2}) satisfying the following constraints:

$$\begin{array}{ccccc} |x_i - x_{i-1}| < T^t & \wedge & |x_{i-1} - x_{i-2}| < T^t & \wedge \\ x_i > x_{i-1} & \wedge & x_{i-1} > x_{i-2}. & \end{array} \tag{22}$$

	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
H	3	**97**	0	0	0	0
H	6	15	**79**	0	0	0
J	0	0	0	**72**	28	0
R	0	0	0	8	**89**	3
W	0	0	0	6	0	**100**

(b)

Table 1. Confusion matrix with (a) and without (b) model pruning. 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.* [17]	97	95	91	89	80	99	91.8
Schuldt *et al.* [33]	98	60	74	60	55	84	71.8
Li *et al.* [22]	97	94	86	**100**	83	97	**92.8**
Niebles *et al.* [26]	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).

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: $O(M{\cdot}S)$.

5. Experimental Results

We tested the proposed method on the widely used public KTH dataset [33]. 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. We use the same protocol as proposed in the original paper [33]. First, we build up a model dictionary which consists of subsequences extracted from sequences of the training and validation set. We use 383 sequences of the KTH set for training out of total number 599 sequences and use the remaining portion for testing. We generate several model graphs by partitioning the sequences into subsequences each containing somewhere between 20 to 30 number of frames with salient interest points. This results in 1429 model graphs in total.

Spatio-temporal interest points extracted with the 3D Harris detector [17] 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 [17]. 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 the 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 $W^t = 60$.

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 the disparate number of model sets could be a factor.

Dictionary Learning — 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 [30]. 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. The half of the training sequences is used as validation set during dictionary learning. 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). However, the algorithm still mixes *jogging* up with *running*.

Sample matched model and scene sequences are illustrated in Figure 3, where the first three actions (*handwaving*, *boxing*, *walking*) are successfully matched while the last one (*running*) gives an example of mismatch. 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 [13]. In the table we chose results obtained with the same protocol.

The algorithm has been implemented in Matlab. It matches a model graph in approximately 13.8 seconds for an average scene of 30 seconds ($S = 755$) on a CPU with 3.33GHz and 4GB RAM.

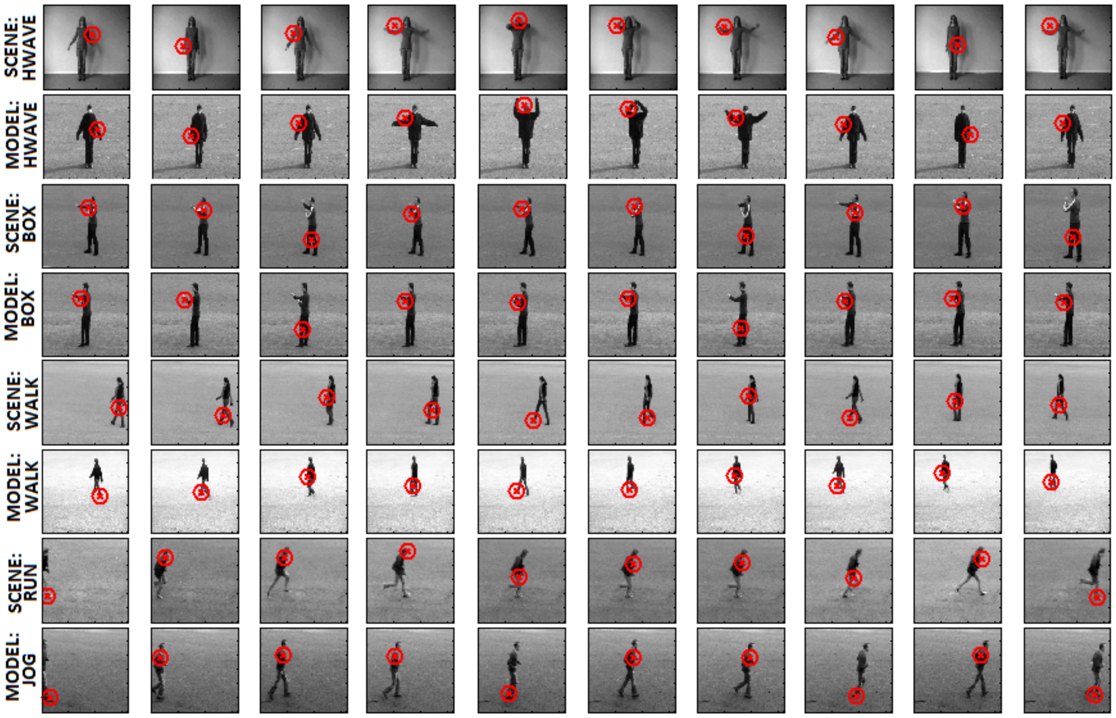

Figure 3. Examples for matched videos: the top three matches are correct, whereas the bottom match is incorrect.

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. Future work will concentrate on a GPGPU implementation and on modelling more complex activities with hierarchical models.

References

[1] M. F. Abdelkader, W. Abd-Almageed, A. Srivastava, and R. Chellappa. Silhouette-based Gesture and Action Recognition via Modeling Trajectories on Riemannian shape manifolds. *CVIU*, 115(3):439–455, 2010. 2

[2] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: a review. *ACM Computing Surveys*, 2011. 2

[3] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Tr. on PAMI*, 23(3):257–267, 2001. 2

[4] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *ICPR*, 2011. 1

[5] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *IJPRAI*, 18(3):265–298, 2004. 2

[6] N. Cuntoor, B. Yegnanarayana, and R. Chellappa. Activity modeling using event probability sequences. *IEEE Tr. on IP*, 17(4):594–607, 2008. 2

[7] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *ICCV VS-PETS*, Beijing, China, 2005. 2

[8] O. Duchenne, F. R. Bach, I.-S. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. In *CVPR*, pages 1980–1987, 2009. 2

[9] O. Duchenne, A. Joulin, and J. Ponce. A graph-matching kernel for object categorization. In *ICCV*, 2011. 2

[10] A. Dyana and S. Das. Trajectory representation using gabor features for motion-based video retrieval. *Pattern Recognition Letters*, 30(10):877–892, 2009. 2

[11] P. Felzenszwalb and R. Zabih. Dynamic programming and graph algorithms in computer vision. *IEEE Tr. on PAMI*, 33(4):721–740, 2011. 2

[12] R. Filipovych and E. Ribeiro. Robust sequence alignment for actor-object interaction recognition: Discovering actor-object states. *CVIU*, 115:177–193, 2011. 2

[13] Z. Gao, M. Chen, A. Hauptmann, and A.Cai. Comparing evaluation protocols on the kth dataset. In *Human Behavior Understanding*, volume LNCS 6219, pages 88–100, 2010. 7

[14] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Tr. on PAMI*, 29(12):2247–2253, 2007. 2

[15] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *ICCV*, pages 166–173, 2005. 2

[16] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, pages 432–439, 2003. 2

[17] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008. 2, 7

[18] S. Lauritzen and D. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society B*, 50:157–224, 1988. 2

[19] J. Lee, M. Cho, and K. Lee. Hyper-graph matching via reweighted random walks. In *CVPR X*, 2011. 2

[20] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, pages 1482–1489, Washington, DC, USA, 2005. 2, 4

[21] M. Leordeanu, A. Zanfir, and C. Sminchisescu. Semi-supervised learning and optimization for hypergraph matching. In *ICCV 2011*, 2011. 2

[22] B. Li, M. Ayazoğlu, T. Mao, O. I. Camps, and M. Sznaier. Activity recognition using dynamic subspace angles. In *CVPR*, 2011. 7

[23] L. Lin, K. Zeng, X. Liu, and S.-C. Zhu. Layered graph matching by composite cluster sampling with collaborative and competitive interactions. *CVPR*, 0:1351–1358, 2009. 2

[24] J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR*, Los Alamitos, CA, 2008. 2

[25] K. Mikolajczyk and H. Uemura. Action recognition with appearance motion features and fast search trees. *CVIU*, 115(3):426–438, 2011. 2

[26] J. C. Niebles, C. W. Chen, and L. Fei-Fei. Modelling temporal sturcture of decomposable motion segments for activity classification. In *ECCV*, pages 1–14, 2010. 7

[27] J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *CVPR*, pages 1–8, 2007. 2

[28] H. Ning, Y. Hu, and T. S. Huang. Searching human behaviours using spatial-temporal words. In *ICIP*, volume 6, pages 337–340, 2007. 2

[29] R. Poppe. A survey on vision-based human action recognition. *Image Vision and Computing*, 28, 2010. 2

[30] P. Pudil, F. J. Ferri, J. Novovicov, and J. Kittler. Floating search methods for feature selection with non-monotonic criterion functions. In *ICPR*, pages 279–283, 1994. 7

[31] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: video structure comparison for recognition of complex human activities. In *ICCV*, 2009. 2

[32] S. Savarese, A. Delpozo, J. Niebles, and L. Fei-Fei. Spatial-temporal correlatons for unsupervised action classification. In *WMVC*, Los Alamitos, CA, 2008. 2

[33] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, pages 32–36, 2004. 2, 7

[34] P. Scovanner, S. Ali, and M. Shah. A 3d sift descriptor and its application to action recognition. In *ACM Multimedia*, pages 357–360, New York,USA, 2007. 2

[35] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *ICCV*, volume 2, pages 1470–1477, 2003. 1

[36] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Tr. on PAMI*, 22(8):747–757, 2000. 2

[37] A. P. Ta, C. Wolf, G. Lavoue, and A. Başkurt. Recognizing and localizing individual activities through graph matching. In *AVSS*, 2010. 1

[38] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV*, pages 596–609, 2008. 2

[39] T.S.Caetano, T. Caelli, D. Schuurmans, and D. Barone. Graphical models and point pattern matching. *IEEE Tr. on PAMI*, 28(10):1646–1663, 2006. 2, 6

[40] G. Willems, T. Tuytelaars, and L. V. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008. 2

[41] S. Zampelli1, Y. Deville, and C. Solnon. Solving subgraph isomorphism problems with constraint programming. *Constraints*, 2009. 2

[42] M. Zaslavskiy, F. Bach, and J. Vert. A path following algorithm for the graph matching problem. *IEEE Tr. on PAMI*, 31(12):2227–2242, 2009. 2

[43] R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. In *CVPR*, 2008. 2, 3

[44] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios. Dense non-rigid surface registration using high-order graph matching. In *CVPR*, 2010. 2

[45] L. Zhang, Z. Zeng, and Q. Ji. Probabilistic image modeling with an extended chain graph for human activity recognition and image segmentation. *IEEE Tr. on IP*, 2011. 2