A graph-kernel method for re-identification

Luc Brun, Donatello Conte, Pasquale Foggia, Mario Vento

To cite this version:

Luc Brun, Donatello Conte, Pasquale Foggia, Mario Vento. A graph-kernel method for re-identification. Mohamed Kamel University ofWaterloo, Department of Electrical and Computer Engineering, Waterloo, ON, Canada Aurélio Campilho University of Porto, Faculty of Engineering, Institute of Biomedical Engineering, Porto, Portugal. International Conference on Image Analysis and Recognition, Jun 2011, Burnaby, Canada. Springer Verlag Berlin, 6753, pp.173-182, 2011. <hal-00770260>

HAL Id: hal-00770260

https://hal.archives-ouvertes.fr/hal-00770260

Submitted on 4 Jan 2013

A graph-kernel method for re-identification

Luc Brun[1], Donatello Conte[2], Pasquale Foggia[2], and Mario Vento[2]

[1] GREYC UMR CNRS 6072
ENSICAEN-Université de Caen Basse-Normandie,
14050 Caen, France
luc.brun@greyc.ensicaen.fr
[2] Dipartimento di Ingegneria dell'Informazione e di Ingegneria Elettrica,
Università di Salerno, Via Ponte Don Melillo, 1 I-84084 Fisciano (SA), Italy
{dconte, pfoggia, mvento}@unisa.it

Abstract. Re-identification, that is recognizing that an object appearing in a scene is a reoccurrence of an object seen previously by the system (by the same camera or possibly by a different one) is a challenging problem in video surveillance. In this paper, the problem is addressed using a structural, graph-based representation of the objects of interest. A recently proposed graph kernel is adopted for extending to this representation the Principal Component Analyisis (PCA) technique. An experimental evaluation of the method has been performed on two video sequences from the publicly available PETS2009 database.

1 Introduction

In the last years, research in the field of intelligent video surveillance has progressively shifted from low-level analysis tasks (such as object detection, shadow removal, short term tracking etc.) to high-level event detection (including long term tracking, multicamera tracking, behavior analysis etc.).

An important task required by many event detection methods is to establish a suitable correspondence between observations of people who might appear and reappear at different times and across different cameras. This kind of problematic is commonly known as "people re-identification".

Several applications using single camera setup may benefit from information induced by people re-identification. One of the main appllications is loitering detection. Loitering refers to prolonged presence of people in an area. This behaviour is interesting in order to detect, for example, beggars in street corners, or drug dealers at bus stations, and so on. Beside this, information on these re-occurrences is very important in multi-camera setups, such as the ones used for wide area surveillance. Such surveillance systems create a novel problem of discontinuous tracking of individuals across large sites, which aims to reacquire a person of interest in different non-overlapping locations over different camera views.

Re-identification problem has been studied for last five years approximately. A first group [9, 17, 2, 3] deals with this problem by defining a unique signature which condenses a set of frames of a same individual; re-identification is then performed using a similarity measure between signatures and a threshold to assign old or new labels to successive scene entrances. In [9] a panoramic map is used to encode the appearance of a person extracted from all cameras viewing it. Such a method is hence restricted to multicamera systems. The signature of a person in [17] is made by a combination of SIFT descriptors and color features. The main drawback of this approach is that people to be added into the database are manually provided by a human operator. In [2] two human signatures, which use haar-like features and dominant color descriptor (DCD) respectively, are proposed while in [3] the signature is based on three features, one capturing global chromatic information and two analyzing the presence of recurrent local patterns.

A second group ([25, 4]) deals with re-identification of people by means of a representation of a person in a single frame. Each representation corresponds to a point in a feature space. Then a classification is performed by clustering these points using a SVM ([25]) or a correlation module ([4]). Both [25, 4] use the so-called "color-position" histogram: the silhouette of a person is first vertically divided into n equal parts and then some color features (RGB mean, or HSV mean, etc.) are computed to characterize each part.

This paper can be ascribed to the second group but with some significant novelty: first, we have a structural (graph-based) representation of a person; second, our classification scheme is based on *graph kernels*. A graph kernel is a function in graph space that shares the properties of the dot-product operator in vector space, and so can be used to apply many vector-based algorithms to graphs.

Many graph kernels proposed in the literature have been built on the notion of *bag of patterns*. Graphlets kernels [21] are based on the number of common sub-graphs of two graphs. Vert [14] and Borgwardt [22] proposed to compare the set of sub-trees of two graphs. Furthermore, many graph kernels are based on simpler patterns such as walks [13], trails [8] or paths.

A different approach is to define a kernel on the basis of a graph edit distance, that is the set of operations with a minimal cost transforming one graph into another. Kernels based on this approach do not rely on the (often simplistic) assumption that a bag of patterns preserves most of the information of its associated graph. The main difficulty in the design of such graph kernels is that the edit distance does not usually corresponds to a metric. Trivial kernels based on edit distances are thus usually non definite positive. Neuhaus and Bunke [15] proposed several kernels based on edit distances. These kernels are either based on a combination of graph edit distances (trivial kernel, zeros graph kernel), use the convolution framework introduced by Haussler [11] (convolution kernel, local matching kernel), or incorporate within the kernel construction schemes several features deduced from the computation of the edit distance (maximum similarity edit path kernel, random walk edit kernel). Note that a noticeable exception to this classification is the diffusion kernel introduced by the same authors [15] which defines the gram matrix associated to the kernel as the exponential of a similarity matrix deduced from the edit distance.

We propose in this paper to apply a recent graph kernel [5, 10] based on edit distance, together with statistical machine learning methods, to people reidentification. The remaining of this paper is structured as follows: we first describe in Section 2 our graph encoding of objects within a video. Moving objects are acquired from different view points and are consequently encoded by a set of graphs. Given such a representation we describe in Section 3 an algorithm which allows to determine if a given input graph corresponds to a new object. If this is not the case, the graph is associated to one of the objects already seen. The different hypotheses used to design our algorithm are finally validated through several experiments in Section 4.

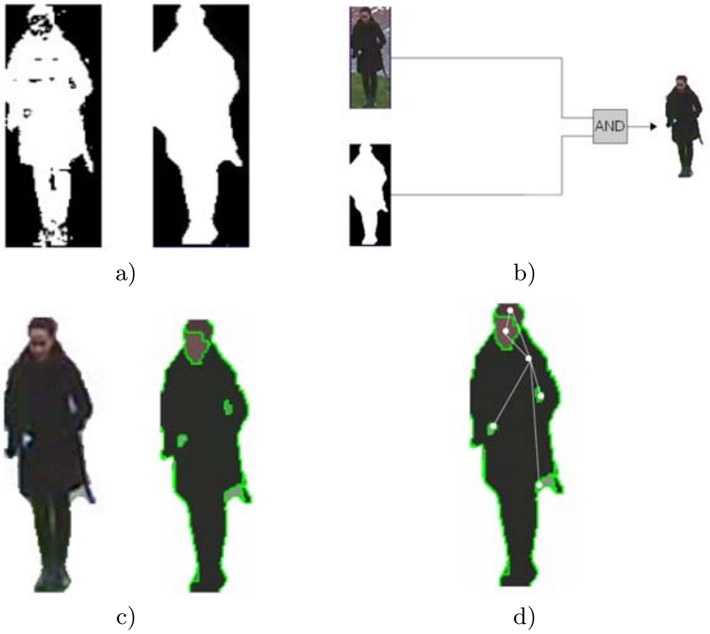
2 Graph-based Object Representation

The first step of our method aims to separate pixels depicting people on the scene (foreground) from the background. We thus perform a detection of moving areas, by background subtraction, combined with a shadow elimination algorithm [6]. This first step provides a set of masks which is further processed using mathematical morphology operations (closing and opening) (Fig. 1a). Detected foreground regions are then segmented using Statistical Region Merging (SRM) algorithm [16] (Fig. 1c). Finally, the segmentation of the mask within each rectangle is encoded by a Region adjacency Graph (RAG). Two nodes of this graph are connected by an edge if the corresponding regions are adjacent. Labels of a node are: the RGB average color, the area, and the size η normalized with respect to the overall image (Fig. 1d).

3 Comparisons between objects by means of Graph Kernels

Objects acquired by multiple cameras, or across a large time interval, may be subject to large variations. Common kernels [13] based on walks, trails or paths are quite sensitive to such variations. On the other hand, graph edit distances correspond to the minimal overall cost of a sequence of operations transforming two graphs. Within our framework, such distances are parametrized by two sets of functions $c(u \to v), c(u \to \epsilon)$ and $c(e \to e'), c(e \to \epsilon)$ encoding respectively the substitution, and deletion costs for nodes and edges. Using such distances, small graph distortions may be encoded by small edit costs, hence allowing to capture graph similarities over sets having important within-class distance. Unfortunately, the computational complexity of the exact edit distance is exponential in the number of involved nodes, which drastically limits its applicability to databases composed of small graphs.

This paper is based on a sub optimal estimation of the edit distance proposed by Nehauss and Bunke [18, 19]. Let us consider two labeled graphs $g_1 =$



a) b)

c) d)

Fig. 1. a) Application of a suited morphological operator; b) Extraction of person appearance; c) Image segmentation; d) RAG construction.

(V_1, E_1, μ_1) and $g_2 = (V_2, E_2, \mu_2)$ where μ_1 and μ_2 denote respectively the vertice's labels of g_1 and g_2. For any vertex w of V_1 or V_2, let us further denote by $\angle(w)$ the set of edges incident to w. The distance between g_1 and g_2 is estimated by first computing for each couple of vertices $(u, v) \in V_1 \times V_2$ the best mapping between $\angle(u)$ and $\angle(v)$. Such a mapping is defined by a permutation σ from a set $S_{u,\sigma} \subset \angle(u)$ to a set $S_{v,\sigma} \subset \angle(v)$, the remaining edges $\angle(u) - S_{u,\sigma}$ and $\angle(v) - S_{v,\sigma}$ being respectively denoted by $N_{u,\sigma}$ and $N_{v,\sigma}$. The cost of a mapping is defined as the overall cost of edges substitutions from $S_{u,\sigma}$ to $S_{v,\sigma}$ and edge deletions from $N_{u,\sigma}$ and $N_{v,\sigma}$. The optimal mapping, denoted $\Delta^e(u, v)$ being defined as the mapping of minimal cost:

$$\Delta^e(u, v) = \min_{\sigma \in M_{u,v}} \sum_{e \in S_{u,\sigma}} c(e \to \sigma(e)) + \sum_{e \in N_{u,\sigma} \cup N_{v,\sigma}} c(e \to \epsilon)$$

where $M_{u,v}$ denotes the set of mappings from $\angle(u)$ to $\angle(v)$.

This optimal mapping is determined using the Hugarian Algorithm [19] applied on the sets $\angle(u)$ and $\angle(v)$. The total cost of mapping vertex u to vertex v together with the sets of incident edges of both vertices is denoted $\Delta(u, v) = c(u \to v) + \Delta^e(u, v)$. The Hugarian algorithm between V_1 and V_2 based on the cost functions $\Delta(u, v)$ and $c(u \to \epsilon)$ provides an optimal mapping σ^* between the nodes of both sets denoted $Editcost(g_1, g_2)$:

$$Editcost(g_1, g_2) = \sum_{u \in S_{1,\sigma^*}} \mathit{\Delta}(u, \sigma^*(u)) + \sum_{u \in N_{1,\sigma^*} \cup N_{2,\sigma^*}} c(u \to \epsilon) \qquad (1)$$

where S_{1,σ^*} (resp. S_{2,σ^*}) corresponds to the set of vertices of V_1 (resp. V_2) mapped to some vertices of V_2 (resp. V_1) by the optimal mapping σ^* while N_{1,σ^*} (resp. N_{2,σ^*}) corresponds to the set of deleted vertices in g_1 (resp. g_2).

Now we will discuss the four cost functions used for defining the edit distance. Within our framework, each node u encodes a region and is associated to the mean color (R_u, G_u, B_u) and to the normalized size η_u of the region (Section 2). We experimentally observed that small regions have larger chances to be deleted between two segmentations. Hence, the normalized size of a region can be used as a measure of its relevance within the whole graph.

The cost of a node substitution is defined as the distance between the mean colors of the corresponding regions. We additionally weigh this cost by the maximum normalized size of both nodes. Such a weight avoids to penalize the matching of small regions, which should have a small contribution to the global similarity of both graphs. Also, a term is added to account for the size difference between the regions:

$$c(u \to v) = \max(\eta_u, \eta_v) \cdot d_c(u, v) + \gamma_{NodeSize} \cdot |\eta_u - \eta_v|$$

where $d_c(u, v)$ is the distance in the color space, and $\gamma_{NodeSize}$ is a weight parameter selected by cross validation. The distance $d_c(u, v)$ is not computed as the Euclidean distance between RGB vectors, but uses the following definition that is based on the human perception of colors:

$$d_c(u, v) = \sqrt{(2 + \frac{\overline{r}}{2^k})\delta_R^2 + 4\delta_G^2 + (2 - \frac{(2^k - 1) - \overline{r}}{2^k})\delta_B^2}$$

where k is the channel depth of the image, $\overline{r} = \frac{R_u + R_v}{2}$ and δ_R, δ_G and δ_B encode respectively the differences of coordinates along the red, green and blue axis.

The cost of a node deletion should be proportional to its relevance encoded by the normalized size, and is thus defined as:

$$c(u \to \epsilon) = \gamma_{NodeSize} \cdot \eta_u$$

Using the same basic idea, the cost of an edge removal should be proportional to the minimal normalized size of its two incident nodes.

$$c((u, u') \to \epsilon) = \gamma_{Edge} \cdot \gamma_{EdgeSize} \cdot \min(\eta_u, \eta_{u'})$$

where $\gamma_{EdgeSize}$ encodes the specific weight of the edge removal operation while γ_{Edge} corresponds to a global edge's weight.

Within a region adjacency graph, edges only encode the existence of some common boundary between two regions. Moreover, these boundaries may be drastically modified between two segmentations. Therefore, we choose to base the cost of an edge substitution solely on the substitution's cost of its two incident nodes.

$$c((u,u') \rightarrow (v,v')) = \gamma_{Edge} \cdot (c(u \rightarrow v) + c(u' \rightarrow v'))$$

Note that all edge costs are proportional to the weight γ_{Edge}. This last parameter allows thus to balance the importance of node and edge costs.

3.1 From graph edit distance to graph kernels

Let us consider a set of input graphs $\{G_1,\ldots,G_n\}$ defining our graph test database. Our person re-identification is based on a distance of an input graph G from the space spanned by $\{G_1,\ldots,G_n\}$. Such a measure of novelty detection requires to embed the graphs into a metric space. Given our edit distance (Section 3), one may build a $n\times n$ similarity matrix $W_{i,j} = exp(-EditCost(G_i,G_j)/\sigma)$ where σ is a tuning variable. Unfortunately, the edit distance does not fulfill all the requirements of a metric; consequently, the matrix W may be not semi-definite and hence does not define a kernel.

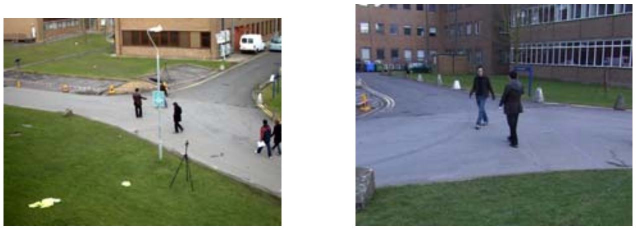
As mentioned in Section 1, several kernels based on the edit distance have been recently proposed. However, these kernels are rather designed to obtain a definite positive matrix of similarity than to explicitly solve the problem of kernel-based classification or regression methods. We thus use a recent kernel construction scheme [5, 10] based on an original remark by Steinke [23]. This scheme [5, 10] exploits the fact that the inverse of any regularised Laplacian matrix deduced from W defines a definite positive matrix and hence a kernel on $\{G_1,\ldots,G_n\}$. Thus, our kernel construction scheme first builds a regularised Laplacian operator $\tilde{L} = I + \lambda L$, where λ is a regularisation coefficient and L denotes the normalized Laplacian defined by: $L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ and D is a diagonal matrix defined by $D_{i,i} = \sum_{j=1}^{n} W_{i,j}$. Our kernel is then defined as: $K = \tilde{L}^{-1}$. Using a classification or regression scheme, such a kernel leads to map graphs having a small edit distance [5, 10] (and thus a strong similarity) to close values.

3.2 Novelty detection and person re-identification

Within our framework, each reappeared person is represented by a set of graphs encoding the different acquisitions of this person. Before assigning a new input graph to an already created class, we must determine if this graph corresponds to a person already encountered. This is a problem of novelty detection, with the specific constraint that each class of graphs encoding an already encountered person has a large within-class variation. Several methods, such as one class SVM [20] or support vector domain description [24] have been used for novelty detection. However, these methods are mainly designed to compare an incoming data with an homogeneous data set. The method of Desobry [7] has the same drawback and is additionally mainly designed to compare two sets rather than one set with an incoming datum.

The method introduced by Hoffman [12] is based on kernel Principal Component Analysis (PCA). An input datum is considered as non belonging to a class if its squared distance from the space spanned by the first principal components of the class is above a given threshold. Note that this method is particularly efficient using high dimensional spaces such as the one usually associated to kernels. This method has the additional advantage of not assuming a strong homogeneity of the class.

Given an input graph G and a set of k classes, our algorithm first computes the set $\{d_1(G), \dots, d_k(G)\}$ where $d_i(G)$ is the squared distance of the input graph G from the space spanned by the first q principal component of class i. Our novelty decision criterion is then based on a comparison of $d(G) = \min_{k=1,n} d_k(G)$ against a threshold.



a) View 001

b) View 005

Fig. 2. Sample frames from the PETS2009 dataset.

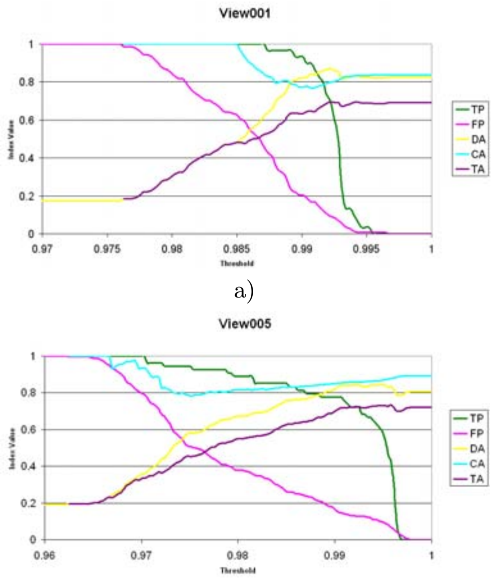
If $d(G)$ is greater than the specified threshold, G is considered as a new person entering the scene. Otherwise, G describes an already encountered person, which is assigned to the class i that minimizes the value of $d_i(G)$.

4 Experimental Results

We implemented the proposed method in C++ and tested its performance on two video sequences taken from the PETS2009 [1] database (Fig. 2). Each video sequence is divided in two parts so as to build the training and test sets. In this experiment we have used one frame every 2 seconds from each video, in order to have different segmentations of each person. The training set of the first sequence (View001) is composed of 180 graphs divided into 8 classes, while the test set contains 172 graphs (30 new and 142 existing). The second sequence (View005) is composed of 270 graphs divided into 9 classes for the training set, and 281 graphs (54 new and 227 existing) for the test set.

In order to evaluate the performances of the algorithm, we have used the following measures:

- The **true positives rate (TP)**, i.e the rate of test patterns correctly classified as novel (positive): $TP =$ true positive/total positive



a)

b)

Fig. 3. Performances result on the view001 (a) and view005 (b) of the PETS2009 dataset.

- The **false positives rate (FP)**, i.e the rate of test patterns incorrectly classified as novel (positive): $TP =$ false positive/total negative
- The **detection accuracy (DA)**:

$$DA = (\text{true positive} + \text{true negative})/(\text{total positive} + \text{total negative})$$

- The **classification accuracy (CA)**, i.e the rate of samples classified as negatives which are then correctly classified with multi-class SVM
- The **Total Accuracy**: $TA = DA \times CA$.

As shown on Fig. 3 we obtained around 85% of novelty detection accuracy, and 70% of total accuracy for both View001 and View005 sequences. These results were obtained with the Graph Laplacian Kernel using $\sigma = 4.7$ and $\lambda = 10.0$.

These results appear very promising. For a wide interval of threshold values the classification accuracy rate remains close to 100%. Furthermore, the True Positive Rate curve has a high slope in correspondence of a high value of the threshold, while the False Positive Rate has a smoother behavior; this means that the algorithm can reliably find a threshold value that is able to discard most of the false positives while keeping most of the true positives. Finally, the ROC curves (Fig. 4) are close to the upper and left edges of the True Positive/False Positive space, confirming the discriminant power of the proposed method.

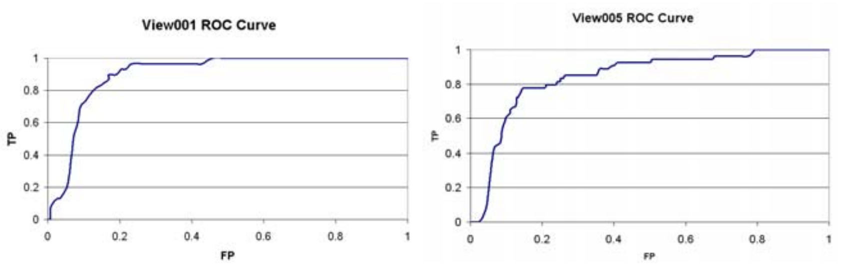


Fig. 4. ROC curves for the two sequences from the PETS2009 dataset.

5 Conclusions

This paper presents a novel method for people re-identification based on a graph-based representation and a graph kernel. It combines our graph kernel with a novelty detection method based on Principal Component Analysis in order to detect if an incoming graph corresponds to a new person and, if not, to correctly assign the identity of a previously seen person. Our future works will also extend the present method to people re-identification within groups. In such cases, a whole group is encoded by a single graph. Thus, the used kernel should be able to match subgraphs within larger graphs. We plan to study the ability of graphlet kernels to perform this task.

References

1. Database: Pets2009. http://www.cvg.rdg.ac.uk/PETS2009/
2. Bak, S., Corvee, E., Brmond, F., Thonnat, M.: Person re-identification using haar-based and dcd-based signature. In: 2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (2010)
3. Bazzani, L., Cristani, M., Perina, A., Farenzena, M., Murino, V.: Multiple-shot person re-identification by hpe signature. In: Proceedings of 20th International Conference on Pattern Recognition, ICPR 2010 (2010)
4. Bird, N., Masoud, O., Papanikolopoulos, N., Isaacs, A.: Detection of loitering individuals in public transportation areas. IEEE Transactions on Intelligent Transportation Systems 6–2, 167–177 (2005)
5. Brun, L., Conte, D., Foggia, P., Vento, M., Villemin, D.: Symbolic learning vs. graph kernels: An experimental comparison in a chemical application. In: 14th Conf. on Advances in Databases and Information Systems (ADBIS) (2010)
6. Conte, D., Foggia, P., Percannella, G., Vento, M.: Performance evaluation of a people tracking system on pets2009 database. In: Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (2010)
7. Desobry, F., Davy, M., Doncarli, C.: An online kernel change detection algorithm. IEEE Transaction on Signal Processing 53–8, 2961–2974 (2005)
8. Dupé, F.X., Brun, L.: Tree covering within a graph kernel framework for shape classification. In: XV ICIAP (2009)

9. Gandhi, T., Trivedi, M.M.: Panoramic appearance map (pam) for multi-camera based person re-identification. In: IEEE International Conference on Video and Signal Based Surveillance, 2006. AVSS '06. (2006)
10. Gauzere, B., Brun, L., Villemin, D.: Graph edit distance and treelet kernels for chemoinformatic. In: Graph Based Representation 2011. IAPR-TC15, Munster, Germany (May 2011), submitted
11. Haussler, D.: Convolution kernels on discrete structures. Tech. rep., Department of Computer Science, University of California at Santa Cruz (1999)
12. Hoffmann, H.: Kernel pca for novelty detection. Pattern Recognition 40(3), 863 – 874 (2007)
13. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized kernel between labeled graphs. In: In Proc. of the Twentieth International conference on Machine Learning (2003)
14. Mah, P., Vert, J.P.: Graph kernels based on tree patterns for molecules. Machine Learning 75(1), 3–35 (2008)
15. Neuhaus, M., Bunke, H.: Bridging the Gap Between Graph Edit Distance and Kernel Machines. World Scientific Publishing Co., Inc., River Edge, NJ, USA (2007)
16. Nock, R., Nielsen, F.: Statistical region merging. IEEE Transaction on Pattern Analysis and Machine Intelligence 26–11, 1452–1458 (2004)
17. de Oliveira, I.O., de Souza Pio, J.L.: People reidentification in a camera network. In: IEEE Int. Conf. on Dependable, Autonomic and Secure Computing (2009)
18. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of bipartite graph matching. Image Vision Computing 27(7), 950–959 (2009)
19. Riesen, K., Neuhaus, M., Bunke, H.: Bipartite graph matching for computing the edit distance of graphs. In: Escolano, F., Vento, M. (eds.) Graph-Based Representations in Pattern Recognition. No. 4538 in LNCS (2007)
20. Scholkopf, B., Platt, J., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Computation 13, 1443–1471 (2001)
21. Shervashidze, N., Vishwanathan, S.V., Petri, T.H., Mehlhorn, K., Borgwardt, K.M.: Efficient graphlet kernels for large graph comparison. In: Twelfth International Conference on Artificial Intelligence and Statistics (2009)
22. Shervashidze, N., Borgwardt, K.: Fast subtree kernels on graphs. In: Advances in Neural Information Processing Systems 22. Curran Associates Inc (2009)
23. Steinke, F., Schlkopf, B.: Kernels, regularization and differential equations. Pattern Recognition 41(11), 3271 – 3286 (2008)
24. Tax, D., Duin, R.: Support vector domain description. Pattern Recognition Letters 20, 1191–1199 (1999)
25. TruongCong, D.N., Khoudour, L., C.Achard, C.Meurie, O.Lezoray: People reidentification by spectral classification of silhouettes. Signal Processing 90, 2362–2374 (2010)