People re-identification by Graph Kernels Methods

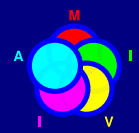
Luc Brun[1], Donatello Conte[2], Pasquale Foggia[2] and Mario Vento[2]

[1] GREYC UMR CNRS 6072
ENSICAEN – Université de Caen Basse-Normandie
FRANCE
E-mail: luc.brun@greyc.ensicaen.fr

[2] Dip. di Ing. Elettronica e Ing. Informatica
Università degli Studi di Salerno
ITALY
E-mail: {dconte,pfoggia,mvento}@unisa.it

8th IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition – Münster, May 2011

Outline



The re-identification problem

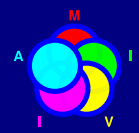
In recent years, intelligent video-surveillance gained more and more interest due to its important role in security.

Early research focused on low-level processing:

- object detection
- tracking
- shadow removal
- ...

More recently, the interest is shifting towards high-level event detection:

- behaviour analysis
- abandoned object detection
- ...

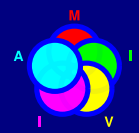


The re-identification problem

An important task for high-level event detection is to establish a correspondence between observations of people who appear and disappear at different times, possibly on different cameras (people re-identification).

Applications:

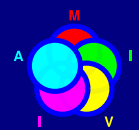
- loitering detection
- multi-camera tracking
- ...



The re-identification problem

Usually re-identification is performed by defining a "signature" (a vector of measures) for a detected person, and thresholding a similarity measure between signatures.

In this paper we propose a re-identification method based on a structural, graph-based representation of a person, and using graph kernels for comparing such representations.



Outline

Graph kernels

Graph Kernel: A function in graph space:

$$K : G \times G \longrightarrow R$$

satisfying the properties of the vector dot-product:

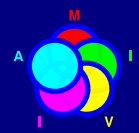
- symmetric:

$$K(g_1, g_2) = K(g_2, g_1)$$

- positive-definite:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i \cdot c_j \cdot K(g_i, g_j) \geq 0$$

for any n graphs $g_1, \ldots, g_n$ and any n real constants $c_1, \ldots, c_n$

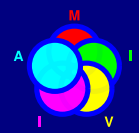


Graph kernels

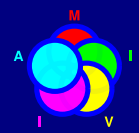
Given a graph kernel, several vector-based algorithms can be applied to to graphs (e.g. Support Vector Machines).

Example of graph kernels:

- bag of patterns (walks, trails, paths, graphlets, subtrees)
- kernels based on graph edit distance
- convolution kernels
- ...

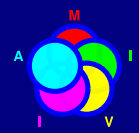


Outline



Graph-based representation

- object detection through background subtraction (with shadow removal)
- detected objects are segmented using Statistical Region Merging (SRM)
- each object is represented using a Region Adjacency Graph (RAG)



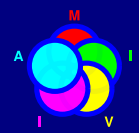
Graph-based representation

Graph-based representation

Node attributes:

- Average color (RGB)
- Area
- Size (normalized with respect to the detected object size)

Edges do not have attributes.



The adopted graph kernel

Our kernel is based on a recently proposed kernel construction scheme to derive a positive definite kernel from a graph-edit distance.

For the graph-edit distance, we have adopted a sub-optimal estimation proposed by Riesen and Bunke in 2007:

- this method requires the definiton of a cost function for node substitution, node deletion, edge substitution and edge deletion
- the cost of mapping two nodes is approximated by taking into account only the edges directly adjacent to the nodes
- the best mapping is found using the Hungarian algorithm (polynomial)

Node substitution cost

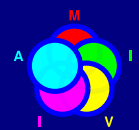
The node substitution cost is defined as:

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

where u and v are the nodes, η is the relative size, $\gamma_{NodeSize}$ is a weight parameter and $d_c(u, v)$ is the distance in color space, computed according to human perception:

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

$\overline{r}$ is the average red value of u and v; δ_R, δ_G and δ_B are the difference between the two nodes in the average red, green and blue components.



Edge substitution cost

In our representation, edges do not have attributes. So we chose to base the cost of edge substitution on the corresponding cost of the incident nodes.

Hence, edge substitution cost is defined as:

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

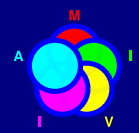
Node and edge deletion cost

The node deletion cost is defined as:

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

while the edge deletion cost is defined as:

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



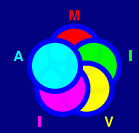
From the Graph Edit Distance to the Laplacian Kernel

Given our distance function, and a set of graphs $\{g_1, \ldots, g_n\}$, we first construct a similarity matrix W defined as:

$$w_{ij} = exp\left(-\text{dist}(g_i, g_j)/\sigma\right)$$

where σ is a tuning variable.

W is not guaranteed to be positive-definite, so it does not define a proper kernel.



From the Graph Edit Distance to the Laplacian Kernel

We use a remark by Steinke to costruct a positive-definite kernel from W: the inverse of any regularized Laplacian matrix deduced from W is a positive definite matrix.

So, we first build a regularized Laplacian operator:

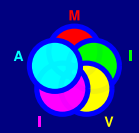
$$\tilde{L} = I + \lambda L$$

where λ is a regularization coefficient and

$$L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

is the normalized Laplacian. The matrix D is a diagonal matrix computed as: $D_{i,i} = \sum_{j=1}^{n} W_{i,j}$

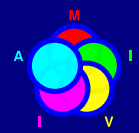
Finally, our kernel is defined by the matrix $K = \tilde{L}^{-1}$



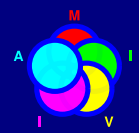
People re-identification

Our method considers each person as a different class.

- Kernel PCA is applied to the graphs of a same class to compute the first q principal components
- Given an input graph g, we compute the distance $d_i(g)$ of g from the space spanned by the first q principal components of class i
- if $d_i(g) > \theta$ for all i, the object is considered as a new person, and a new class is created
- otherwise, the person is classified as belonging to class $i = \arg\min d_i(g)$



Outline



The dataset

We have used two video sequences from PETS2009 (View 001 and 005):

View 001: 8 classes; 63+172 graphs (training+test)

View 005: 6 classes; 33+89 graphs

Performance measures

TP: true positive (novel) graphs

FP: false positive graphs

NDA: Novelty Detection Accuracy

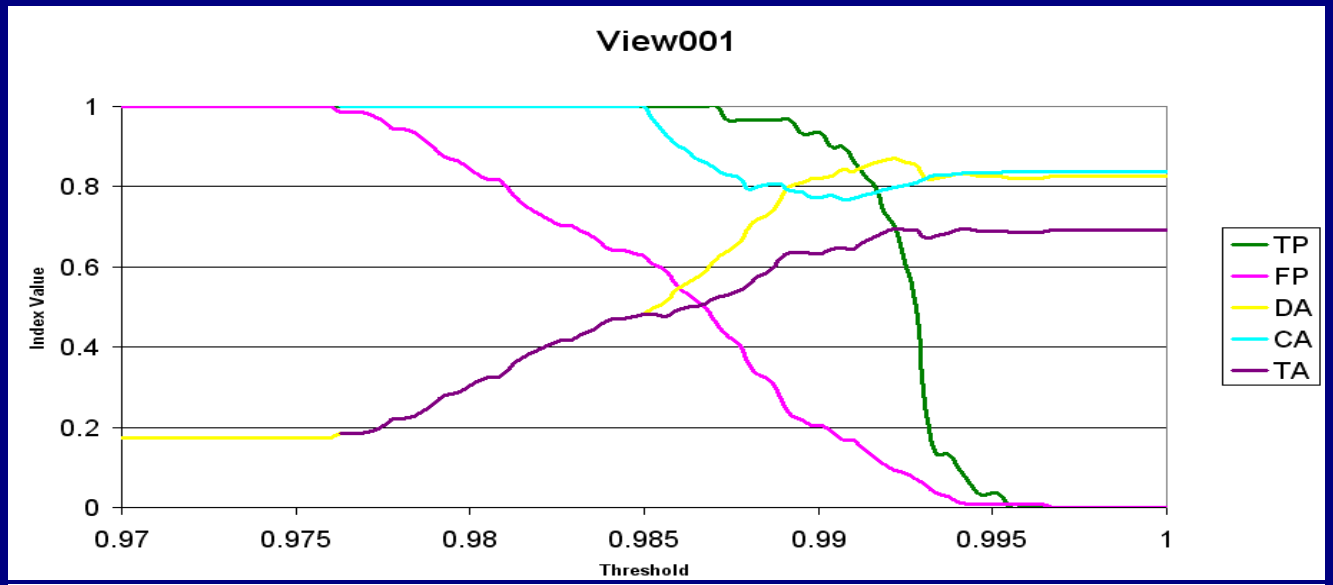
$$NDA = (TP + TN)/(TP + FP + TN + FN)$$

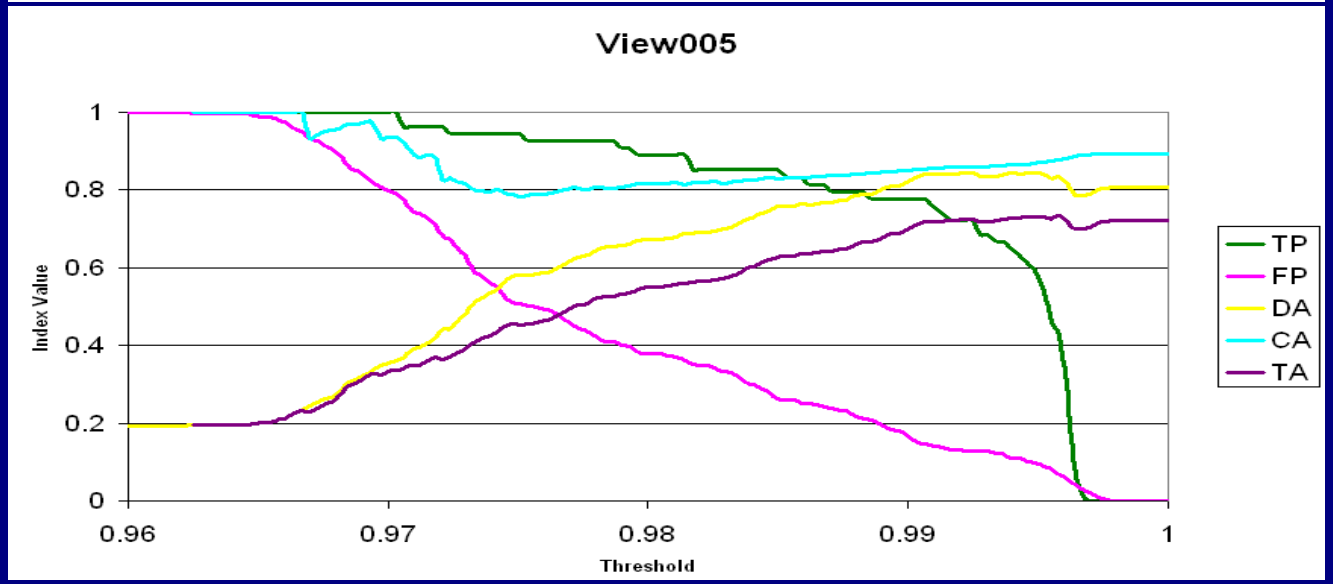
CA: Classification Accuracy (negative samples assigned to the right class)

TA: Total Accuracy

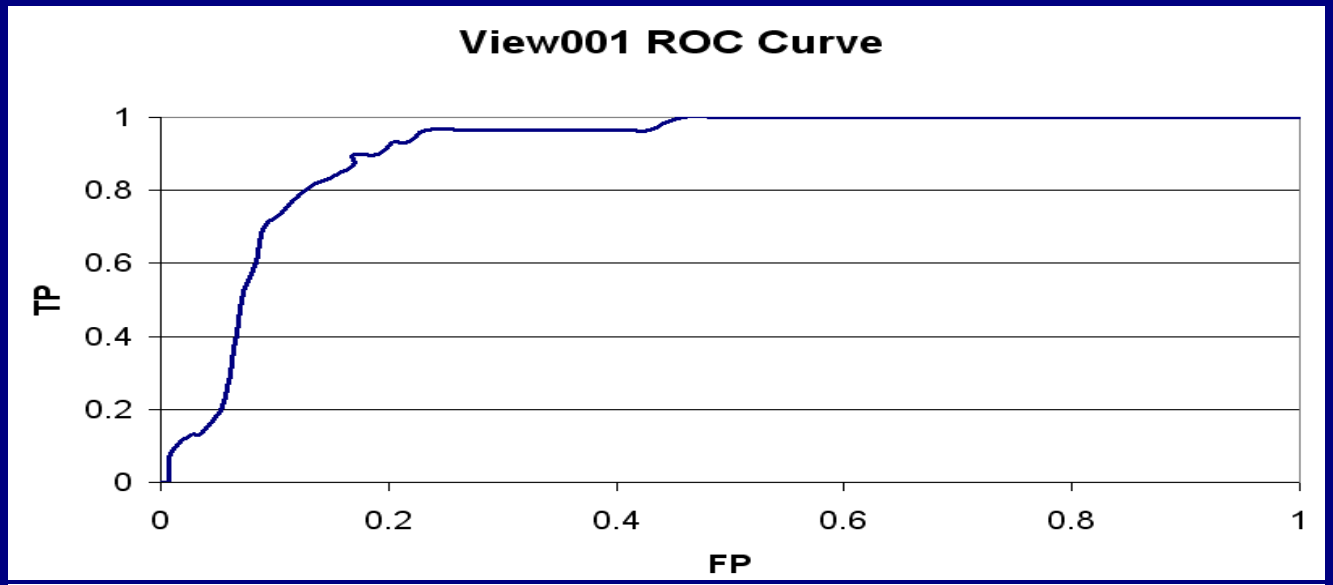
$$TA = NDA \times CA$$

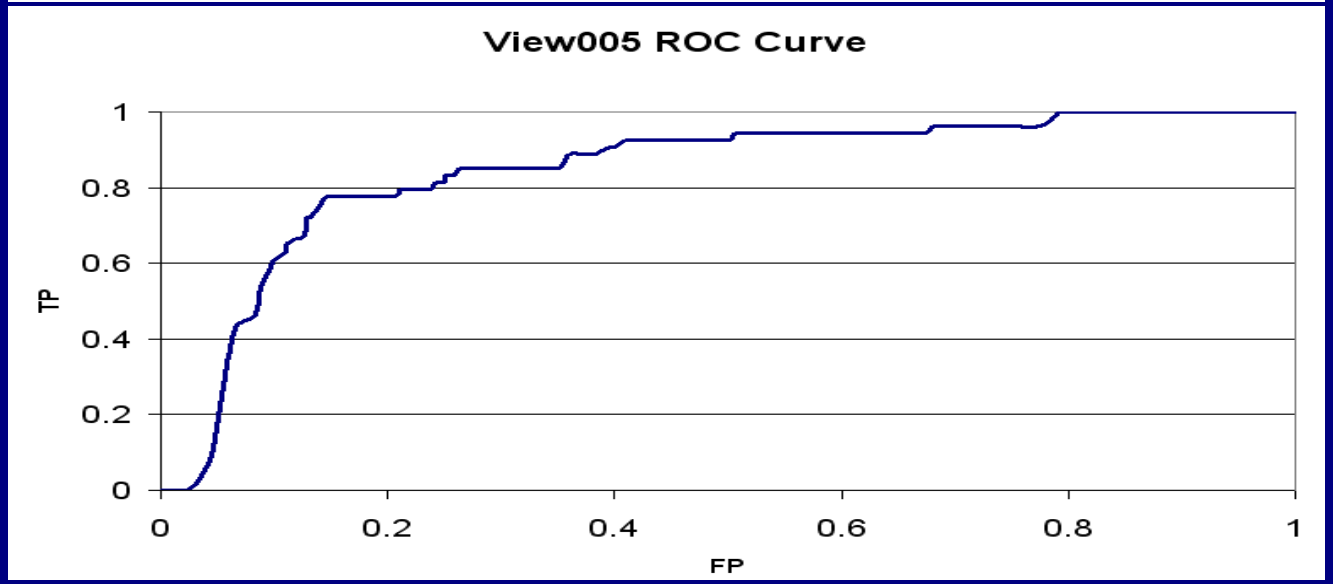
Obtained performance





ROC curves


View001 ROC Curve
TP
FP
0
0.2
0.4
0.6
0.8
1

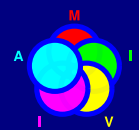

View005 ROC Curve
TP
FP
0
0.2
0.4
0.6
0.8
1

Conclusions

- we have defined a novel people re-identification method, that combines our Laplacian graph kernel with a novelty detection method based on kPCA
- the method shows good performance on the PETS2009 dataset
- future work: extending to persons within groups, using kernels able to match subgraphs within a graph (e.g. graphlet kernels)

Thanks for your attention...

...are there any questions?



Conclusions

- we have defined a novel people re-identification method, that combines our Laplacian graph kernel with a novelty detection method based on kPCA
- the method shows good performance on the PETS2009 dataset
- future work: extending to persons within groups, using kernels able to match subgraphs within a graph (e.g. graphlet kernels)

Thanks for your attention...

...are there any questions?