



Label Diffusion on Graph for Face Identification

Pierre Buysens, Marinette Revenu

► **To cite this version:**

Pierre Buysens, Marinette Revenu. Label Diffusion on Graph for Face Identification. ICB, Mar 2012, New Delhi, India. pp.46 - 51, 2012, <10.1109/ICB.2012.6199757>. <hal-00805790>

HAL Id: hal-00805790

<https://hal.archives-ouvertes.fr/hal-00805790>

Submitted on 28 Mar 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Label Diffusion on Graph for Face Identification

Pierre Buysens, Marinette Revenu

GREYC - CNRS, ENSICAEN, University of Caen, 14050 Caen, France

Pierre.Buysens@unicaen.fr

Abstract

We present a generic classification method based on a label diffusion on graph. Inspired by the watershed transform widely used in the image segmentation field, we propose a simple algorithm working on distances graphs of any topology. Applied to the faces identification problem, this method can deal with any type of distances. We propose also two penalty functions that clearly improve the identification results when used within the diffusion process.

1. Introduction

Face recognition is a topic which has been of increasing interest during the last two decades due to a vast number of possible applications: biometrics, video-surveillance, advanced HMI of image/video indexation.

Biometric *identification* consists in finding an unknown identity (the *probe* image) among a set of known identities (the *gallery*). Most of the approaches proposed in the literature [6] are built on the same three-steps scheme: 1) preprocessing of the images, 2) extraction of features from faces, 3) classification of these features. This paper deals with the last part, the classification step.

Image preprocessing The first step tends to localize the faces in an image, to resize it, to apply a geometric normalization and to use some algorithms to improve the image quality.

Features extraction The second step computes facial features from the preprocessed image. These features have to be simple to compute, robust to facial appearance (facial expression), and discriminative in order to differentiate persons. For a recent survey on facial features extraction, see [9]. This step can globally be divided into two main parts:

- the local approaches, which act locally on the face by extracting salient interest points (like eyes or mouths), and combine them into a global model [3] [15];

- the global approaches which often rely on a projection of the whole image onto a new low-dimensional space (these methods are then named *Subspace* methods). Numerous dimension reduction techniques have been used like PCA [14], LDA [8], or their non linear version Kernel-PCA [11], and Kernel-LDA [10].

Classification The last step aims to compare a features vector extracted from a *probe* face to a set of features vector computed from known subjects. Data classification is a general task in computer science that is not restricted to the biometric area. Numerous algorithms can then be used. However, learning-based methods, like SVM or neural networks, cannot be used in practice. These methods indeed learn a discriminative function that try to separate identities of the gallery. As the number of classes (the known identities) may change (a subject is removed/added to the gallery), these classification models have to be reconsidered and the learning phase to be re-processed, which may be impracticable in a real scenario.

Classification algorithms generally used in biometric systems compare directly features vectors in a one-to-one scheme.

The well-known *Nearest Neighbor* algorithm simply selects the features vector v_g from the gallery \mathbf{G} that is the closest to the probe vector v_p according to a given distance measure:

$$v_g = \min_{i \in \mathbf{G}} \|v_{g_i} - v_p\|$$

The identity corresponding to v_p is then the one of v_g .

Other classification algorithms make use of the features vectors of the entire gallery set to obtain a model of the probe. That is the case with the mixture of gaussians. Used when there are more than one image per person in the gallery, a probe vector is decomposed as a weighted sum of gaussian models:

$$v_p = \sum_i \pi_i * \mathcal{G}(id)$$

where π_i are the mixture coefficients and $\mathcal{G}(id)$ are the created models (gaussians) for each subject id . The identity

of v_p often corresponds to the identity with the maximum mixture coefficient.

Another classification algorithm makes use of the entire gallery to decompose in a sparse manner a probe vector onto the gallery set [16]. Here the sparse decomposition is processed according to the energy:

$$E = \min_{\mathbf{x} \in \mathbb{R}^m} (\|\mathbf{v}_p - A\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1)$$

where $A \in \mathbb{R}^{n \times m}$ is the matrix containing the features vectors of the gallery (in column) and $\mathbf{x} \in \mathbb{R}^m$ is the vector containing the coefficients of the sparse decomposition. The identity of v_p is then deduce with:

$$identity(v_p) = identity(\min_i \|v_p - A_i x_i\|_2)$$

In this paper, we propose another scheme for the identification: using both the entire gallery and probe sets for the classification. Relations between vectors are modeled with a graph where each node represents a features vector. A label (the identity) is given to each node corresponding to features vectors of the gallery. A label diffusion onto the graph then assigns a label to each node. As relation between two unlabeled nodes can be less relevant that a relation between an initially-labeled and an unlabeled node, we propose two penalty functions that can be used during the process to favor the last case (or to penalize the first case).

The paper is organized as follow: Section 2 details the proposed algorithm for the label diffusion and two penalty functions that may be used during the diffusion process. The conducted experiments and results obtained are described in section 3. In section 4, we discuss about the algorithm and its links with other diffusion on graph technics. Finally we present our conclusions and further work in section 5.

2. Label diffusion on graphs

In this section, we recall definitions of graphs, detail the proposed algorithm, and explain the penalization functions used within the algorithm.

2.1. Notations and definitions

Preliminaries on Graphs We consider the general situation where any discrete domain can be modeled as a weighted graph. A weighted Graph $G = (V, E, w)$ is composed of a finite set $V = \{u_1, \dots, u_N\}$ of N vertices (or nodes), a set of edges $E \subset V \times V$, and a weighting function $w: E \rightarrow \mathbb{R}^+$. An edge $(u, v) \in E$ connects two adjacent or neighbor vertices u and v of V . In the rest of the paper, such an edge will be noted $u \sim v$. We assume that the graph G is simple, connected and undirected, which implies that the weighting function w is symmetric i.e. $w(u, v) = w(v, u)$ if $(u, v) \in E$.

In the rest of the paper, we only deal with distances graphs. The weight associated to an edge in such a graph represents the distance between two vertices according to a given distance.

The label diffusion The *Watershed* transform is a well-known tool principally used in image segmentation. Intuitively, the watershed of a function (seen as a topographical surface) is composed of locations from which a drop of water could flow towards different minima. The formalisation and proof of this statement is related to the *optimal spanning forests relative to the minima* [5]. Widely used in image segmentation, classical algorithms use automatically-computed or user-defined seeds as starting pixels. The gradient image is also used as the topological map, seen as a relief map. As a result, each pixel belongs to a basin derived from an initial seed, and all pixels are clustered in a local manner.

The watershed transform can be seen as a clustering tool which assigns to pixels (in the case of image segmentation) the label of the seed to which they are attached. Inspired from the watershed transform, we propose a similar transformation applied to a weighted graph. In the rest of the paper, the data attached to each vertice v correspond to a feature vector.

2.2. Proposed algorithm

Given a weighted graph G of distances, we propose a simple algorithm (Algorithm 1) that computes a label diffusion on graph.

The algorithm starts with a set S of labeled vertices. A label function f is then defined on each vertex such that:

$$\begin{cases} f(u) > 0 & \forall u \in S \\ f(u) = 0 & \forall u \notin S \end{cases}$$

Algorithm 1: Label diffusion on distances graph

Data: a weighted graph G ,

a set S of vertices u with label $f(u) > 0$

Result: S

initialize $f(u) = 0, \forall u \notin S$;

initialize the list L of edges $e_{u \sim v}, \forall u \in S$;

while $L \neq \emptyset$ **do**

$e_{u \sim v} = \min_w L$;

if $f(v) = 0$ **then**

$f(v) = f(u)$;

$S = S \cup \{v\}$;

$L = L \cup e_{k \sim v} \quad \forall k \sim v \text{ such that } f(k) = 0$;

$L = L - \{e_{u \sim v}\}$;

All edges that contains an element of S are added to a list L of edges. At each step, the algorithm finds the edge $e_{u \sim v}$ in L of minimal weight that has one and only one

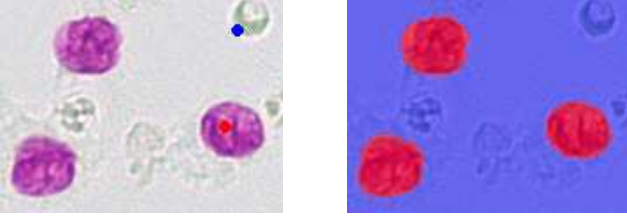


Figure 2. Left: The original image, the two seeds are the black and white dots. Right: the clustering result. All the three nuclei are well detected.

labeled vertex. The unlabeled vertex v connected to the labeled vertex u is then labeled $f(v) = f(u)$. The edges $e_{v \sim k}$ connecting any unlabeled neighbor k of v are then added to L .

The complexity of the algorithm essentially relies on the function that aims to find the minimum weight among a list of weighted edges. This has been conveniently processed with a heap structure which complexity is $\mathcal{O}(n \log(n))$ where n is the number of edges.

A schematic view of the algorithm is shown at the figure 1. In this example, seeds (blue and red dots) are associated to the nodes a and d . Assuming $w_1 < w_2 < w_3$, nodes b and c take each the label of a (red label).

A clustering example of nuclei in a cytological image obtained with this algorithm is shown at figure 2. In this example, each pixel (in the RGB colorspace) corresponds to a vertex $u \in \mathbb{R}^3$, and the weighting function is given by $w(u, v) = \|u - v\|_2$. Note that the constructed graph is a k -nearest neighbor graph, in which each vertex is connected to its k nearest neighbors according to the \mathcal{L}_2 norm, independently of their location on the grid. To ensure that there are no disjoint parts in the graph, each vertex u corresponding to a pixel (i, j) is also connected (4-connexity) to its neighbors on the grid. The algorithm allows the detection of the three nuclei although only one was initially marked.

2.3. Penalty functions

The proposed algorithm acts locally on nodes. Since a propagated label could be less relevant than an initial one, we introduce two penalization functions to reflect the saliency of a label. These functions act on the edges of the graph, and introduce the notion of globality in the process. Intuitively, these functions penalize edges between two vertices that are not labeled: the farthest an edge is from a seed, the higher its penalty will be.

We first introduce the notion of *step* (noted $p_{u \rightarrow v} \in \mathbb{N}^*$) that represents for a given node u the number of edges to reach a node v . This step can easily be updated during the diffusion process when edges are added to L . Nodes $u \in S$ have a fixed step $p_u = 0$. In the example (Figure 1), $p_a = p_d = 0$, $p_{b \rightarrow a} = p_{c \rightarrow d} = 1$, and $p_{c \rightarrow a} = p_{b \rightarrow d} = 2$.

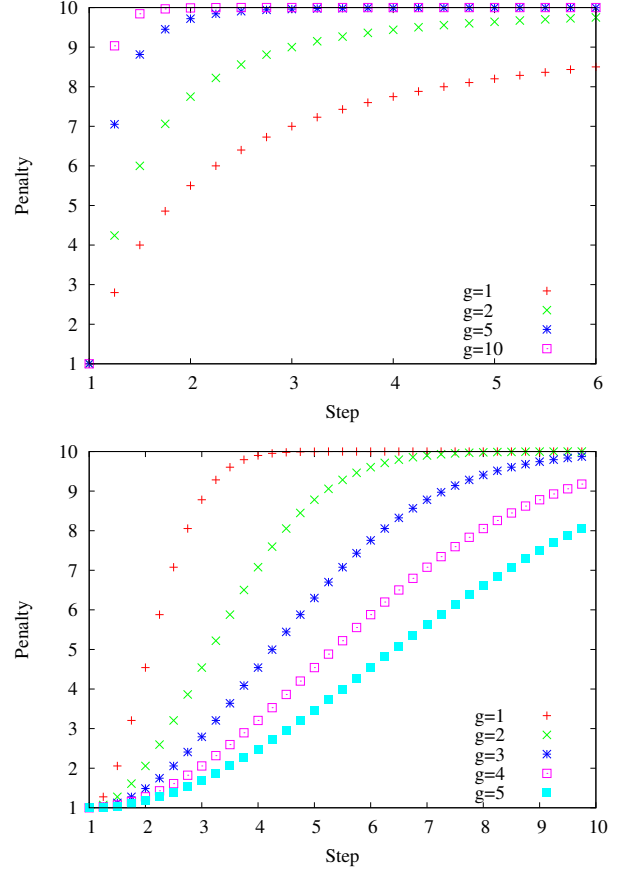


Figure 3. Penalty functions Ψ^1 (top) and Ψ^2 (bottom) for different values of γ (noted as g) and with $C = 9$.

The two penalty functions proposed Ψ^1 and Ψ^2 are:

$$\begin{aligned} \Psi_{C,\gamma}^1(p) &= 1 + C\left(1 - \frac{1}{p^\gamma}\right) \\ \Psi_{C,\gamma}^2(p) &= 1 + C\left(1 - \exp\left(-\frac{1}{2} \frac{(p-1)^2}{\gamma^2}\right)\right) \end{aligned}$$

These functions are strictly increasing according to p (with $C > 0$), and are defined on $[1, +\infty[$ taking values in $[1, C + 1]$ (see figure 3). Since these functions are only used for unlabeled nodes u , we always have $p_u > 0$. Note that for $C = 0$, we have $\Psi^1 = \Psi^2 = 1$, which means that no penalization is applied.

The penalization of an edge $e_{u \sim v}$ added to L is then computed as:

$$w(u, v) \leftarrow \Psi_{C,\gamma}^i(p) \cdot w(u, v) \text{ with } i \in \{1, 2\}$$

These functions allow to control the diffusion speed of the label. When $C = 0$, the process is exactly the one described by the algorithm 1. If $\gamma \rightarrow +\infty$ for Ψ^1 (or $\gamma \rightarrow 0$ for Ψ^2), weights become equal to $(C + 1) \cdot w(u, v)$.

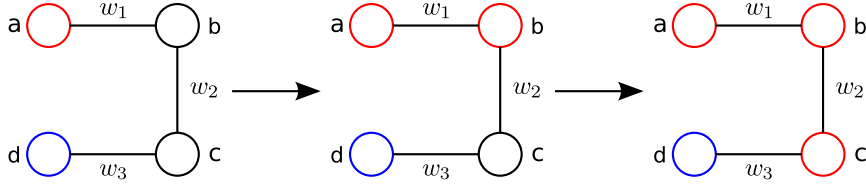


Figure 1. Algorithm process for a graph where $w_1 < w_2 < w_3$.

3. Experiments and results

In this section, we detail the database used as benchmark, the preprocessing applied on the images, the features extraction step, and the conducted experiments.

3.1. Details of the database

The Notre-Dame database [4] (see figure 4 for some samples) is a reference publicly available database. A test protocol is given with images lists to be use for learning/enrolment/test. Two main experiments have been designed, named *Same-session* and *Time-lapse*. In this paper we focus on the *Time-lapse* experiment which includes images taken within weeks/months. This experiment seems more relevant than the *Same-session* one which is less complex.

The *Time-lapse* experiment is composed of 16 separate sub-experiments, varying in illumination and facial expression of the gallery/probe sets (see [4] for details).

Each sub-experiment consists in finding the identity of 431 probe images among a gallery set composed of 63 images of 63 different subjects.

In the rest of the paper, and for clarity purposes, although the 16 sub-experiments have been conducted separately, only the mean rank-1 recognition rates are presented.

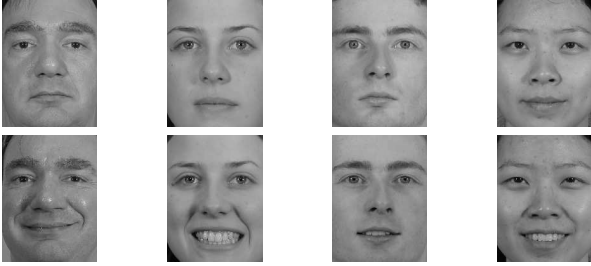


Figure 4. Samples of the Notre-Dame database. Top: some gallery images. Bottom: some probe images.

3.2. Preprocessing step

Each image of the database is normalized before the features extraction step. Images are geometrically normalized, eyes are roughly at the same position for each image (see figure 5). Images are resized to a fixed size 150×200 . Their pixel values are scaled to ensure they are centered ($\mu = 0$) and normalized ($\sigma = 1$).

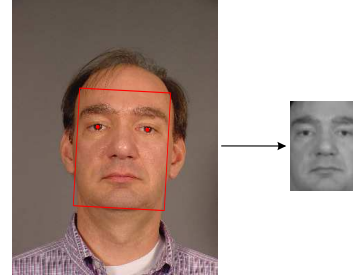


Figure 5. Preprocessing of the images.

3.3. Features extraction step

Once all the images have been preprocessed, the features extraction step consists in a simple Principal Component Analysis (*Eigenfaces* method [14]). Each image is characterized by a vector $\Phi \in \mathbb{R}^n$. In our tests, eigenvectors of the PCA representing 95% of the total energy induced by the eigenvalues are retained, which corresponds to vectors of size 97.

3.4. Identification results

The identification tests involve a comparison between the label diffusion method with a nearest neighbor classifier (noted *NN*) and the *Sparse Representation-based Classification* (noted *SRC*) framework [16].

Distances used Some distances have been tested for the graph construction and the nearest neighbor classifier. For the distances related to the Mahalanobis space (Equations 5, 6 and 7), features vectors u and v have to be transformed such that variance along each dimension is equal to 1. Let m and n be the vectors in Mahalanobis space corresponding to u and v . The vectors are related through the following equations [2]:

$$m_i = \frac{u_i}{\sigma_i}, n_i = \frac{v_i}{\sigma_i}$$

Interested reader can find more details on equations 1 through 7 in [2]. Equations 8 and 9 are explained in [13] and [1] respectively.

$$D_{CityBlock}(u, v) = \sum_i |u_i - v_i| \quad (1)$$

$$D_{Euclidean}(u, v) = \sqrt{\sum_i (u_i - v_i)^2} \quad (2)$$

$$D_{Correlation}(u, v) = \frac{\sum_i (u_i - \bar{u})(v_i - \bar{v})}{(N-1)\sqrt{\frac{\sum_i (u_i - \bar{u})^2}{N-1}}\sqrt{\frac{\sum_i (v_i - \bar{v})^2}{N-1}}} \quad (3)$$

$$D_{Covariance}(u, v) = \frac{\sum_i u_i v_i}{\sqrt{\sum_i u_i^2}\sqrt{\sum_i v_i^2}} \quad (4)$$

$$D_{MahL1}(u, v) = \sum_i |m_i - n_i| \quad (5)$$

$$D_{MahL2}(u, v) = \sqrt{\sum_i (m_i - n_i)^2} \quad (6)$$

$$D_{MahCosine}(u, v) = \frac{m \cdot n}{|m||n|} \quad (7)$$

$$D_{Hellinger}(u, v) = \sqrt{\sum_i (\sqrt{|u_i|} - \sqrt{|v_i|})^2} \quad (8)$$

$$D_{Canberra}(u, v) = \sum_i \frac{|u_i - v_i|}{|u_i + v_i|} \quad (9)$$

Results The table 1 presents the main results obtained from our experiments. For each distance, the rank-1 mean recognition rates (and their standard deviation) are shown. The results for the label diffusion (*Lab. Diff.* in the table) are composed of two parts according to the penalty function applied (see section 2.3).

Since the SRC method cannot be derived for the considered distances, its results have not been included in the table. It gives a rank-1 mean identification rate of 60.70% (with a standard deviation of 9.81).

For the label diffusion method, a grid search has been performed on the two parameters γ and C . The best parameters (those giving the best results) are grouped into the variable ϕ .

One can see that the label diffusion process gives always better results than the nearest neighbor classifier, whatever the distance used.

When no penalization is used during the label diffusion process ($C = 0$), identification rates are the lower. This clearly shows that faces from different persons are close in the projection space defined by the main eigenvectors of the PCA, which indicates that the eigenfaces method is not optimal for the face clustering task.

4. Discussion

The constructed graphs in all our experiments are fully connected graphs: each node is connected to all other. We preferred this type of graph, since we made no assumption about the distribution of the features vectors in the eigenspace. Nevertheless, other types of graph can be considered, like a k -nearest neighbor graph (similar to the one used in the example at figure 2). For such a graph, one has to be careful that there exist no disjoint sub-graphs.

The label diffusion method has several links with the diffusion methods on graphs like the Dijkstra [7] or the more

	<i>NN</i>	<i>Lab. Diff.</i> Ψ^1	<i>Lab. Diff.</i> Ψ^2
Eq.1	69.76% (10.44)	74.59% (9.24) $\phi = (1, 1)$	75.20% (9.53) $\phi = (4, 3)$
		$C = 0 : 52.43\%$ (17.71)	
Eq.2	59.09% (8.17)	65.74% (8.24) $\phi = (1, 1)$	65.61% (8.72) $\phi = (1, 1)$
		$C = 0 : 42.24\%$ (15.16)	
Eq.3	58.06% (8.27)	65.84% (8.83) $\phi = (2, 1)$	65.83% (9.31) $\phi = (2, 1)$
		$C = 0 : 41.79\%$ (14.99)	
Eq.4	58.55% (8.22)	65.71% (8.58) $\phi = (2, 1)$	65.82% (8.84) $\phi = (2, 1)$
		$C = 0 : 42.32\%$ (15.04)	
Eq.5	68.14% (11.89)	73.53% (11.05) $\phi = (1, 1)$	75.50% (11.82) $\phi = (5, 4)$
		$C = 0 : 59.61\%$ (20.24)	
Eq.6	71.70% (11.65)	76.42% (10.38) $\phi = (1, 1)$	78.19% (10.80) $\phi = (10, 5)$
		$C = 0 : 58.91\%$ (21.33)	
Eq.7	74.23% (10.57)	80.48% (8.90) $\phi = (1, 2)$	80.32% (9.62) $\phi = (3, 2)$
		$C = 0 : 60.78\%$ (21.91)	
Eq.8	54.81% (11.77)	58.40% (11.90) $\phi = (1, 2)$	62.83% (11.46) $\phi = (3, 3)$
		$C = 0 : 43.31\%$ (18.81)	
Eq.9	41.83% (12.14)	42.57% (12.33) $\phi = (1, 1)$	49.20% (16.06) $\phi = (1, 3)$
		$C = 0 : 34.71\%$ (16.37)	

Table 1. Rank-1 mean identification rates, standard deviation in parenthesis. ϕ : parameters (C, γ) used within the penalty functions Ψ^1 and Ψ^2 . $C = 0$: Rank-1 mean identification rates obtained without penalization. Best rates per distance in bold.

general eikonal-based methods [12]. A difference is that it does not explicitly compute any distance between two nodes.

Dijkstra algorithm computes a sum of weights of edges which represents the distance of a node from a seed. In the particular case of a complete connected graph, it is equivalent to a nearest neighbor classification (we have indeed for the graph example at figure 1: $w_{a \rightarrow c} \leq w_{a \rightarrow b} + w_{b \rightarrow c}$). The classification method differs from the eikonal-based resolution methods since it only deals with edges of the graph when eikonal-based methods compute gradients on nodes for example. Moreover, algorithms needs often numerous iterations to converge [12], which can be slow in practice.

Parameters C and γ of the penalty functions play a significant role for the final identification rates. Although the best combination was found by a grid search for the results presented in table 1, some other combinations work almost as well. Table 4 shows the rank-1 mean identification rates obtained with the penalty function Ψ^2 for different values of C and γ . For a fixed couple (C, γ), they are computed as the mean identification rates of all the distances considered above. This table clearly shows that there exists some couples which offers good results whatever the considered distance. A similar result has been observed for the Ψ^1 penalty

$\gamma \backslash C$	1	2	3	4	5	10
1	66.82%	65.21%	64.28%	63.56%	63.04%	62.19%
2	65.58%	66.68%	66.72%	66.07%	65.68%	64.06%
3	62.79%	65.52%	66.38%	66.75%	66.64%	65.52%
4	60.64%	63.39%	65.08%	65.70%	66.28%	66.59%
5	59.11%	61.66%	63.33%	64.53%	65.26%	66.65%
6	57.95%	60.20%	61.97%	63.08%	63.85%	66.02%
7	57.24%	59.27%	60.60%	61.91%	62.75%	65.44%
8	56.23%	58.46%	59.55%	60.75%	61.69%	64.53%
9	55.60%	57.68%	58.99%	59.83%	60.72%	63.38%
10	54.86%	57.22%	58.44%	59.29%	59.92%	62.70%

Table 2. Rank-1 mean identification rates of the tested distances for different values of C and γ for the penalty function Ψ^2 . Rates greater than 66% in bold.

function, but not shown in this paper to avoid overburdening.

The proposed classification scheme is quite new in the biometric field. It is efficient when one has to classify many unknown faces, especially when some of these faces belong to the same subject. A typical operational scenario involves face recognition through time via a camera for example. In such a case, faces spread out on the feature space, and form a graph, on which a label diffusion can be processed.

Note that the proposed classification scheme can easily be adapted to similarity graph. In such graphs, weights measure the degree of similarity between two nodes, and are generally in $[0, 1]$.

5. Conclusion and future work

We presented a generic classification method applied to a face identification problem. Inspired from the well-known watershed transform, it is based on a label diffusion on graph. Samples of the gallery are considered as initial seeds, the diffusion process then assigns to each probe a label corresponding to an identity. Tested on the Notre-Dame database, the proposed algorithm together with the penalty functions provides better identification results than the other classification methods considered. For comparison purposes and simplicity, we have tested this method only with a PCA approach for the features computation, which has been showed to be not optimal. Further experiments will be conducted on other types of features, other types of graphs, as well as other biometrics.

References

- [1] D. Androustos, K. Plataniotis, and A. Venetsanopoulos. Distance measures for color image retrieval. *International Conference on Image Processing*, pages 770–774, 1998.
- [2] J. R. Beveridge, D. S. Bolme, B. A. Draper, and M. Teixeira. The CSU face identification evaluation system: Its purpose, features, and structure. *Machine Vision and Applications*, 16(2):128–138, Feb. 2005.
- [3] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.
- [4] X. Chen, P. J. Flynn, and K. W. Bowyer. IR and visible light face recognition. *Computer Vision and Image Understanding*, 99(3):332–358, Sept. 2005.
- [5] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1362–1374, Aug. 2009.
- [6] D. P. Delacretaz, G. Chollet, and B. Dorizzi. *Guide to Biometric Reference Systems and Performance Evaluation*. Springer, 2009.
- [7] D. B. Johnson. A note on dijkstra’s shortest path algorithm. *Jrnl A.C.M.*, 20(3):385–388, July 1973.
- [8] D. J. Kriegman, J. P. Hespanha, and P. N. Belhumeur. Eigenfaces vs. fisherfaces: Recognition using class-specific linear projection. In *European Conference on Computer Vision*, pages I:43–58, 1996.
- [9] S. Z. Li and A. K. Jain. *Handbook of Face Recognition*, 2nd ed. Springer, 2011.
- [10] S. Mika and J. Weston. Fisher discriminant analysis with kernels. *Neural Networks for Signal Processing*, May 06 1999.
- [11] B. Scholkopf, A. Smola, and K. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, pages 1299–1319, 1998.
- [12] V.-T. Ta, A. Elmoataz, and O. L  zoray. Adaptation of eikonal equation over weighted graphs. In *International Conference on Scale Space and Variational Methods in Computer Vision*, LNCS 5567, pages 187–199. Springer, 2009.
- [13] R. Taylor. A user’s guide to measure-theoretic probability. *Journal of the American Statistical Association*, 98(462), June 2003.
- [14] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuro Science*, 3(1):71–86, 1991.
- [15] L. Wiskott, J. M. Fellous, N. Kruger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, July 1997.
- [16] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.