Constellations and the Unsupervised Learning of Graphs

B. Bonev, F. Escolano, M.A. Lozano, P. Suau, M.A. Cazorla[1] and W. Aguilar[2]

[1] Robot Vision Group, Departamento de Ciencia de la Computación e IA
Universidad de Alicante, Spain

[2] IIMAS: Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas
Univesidad Nacional Autónoma de México UNAM, México

Abstract. In this paper, we propose a novel method for the unsupervised clustering of graphs in the context of the constellation approach to object recognition. Such method is an EM central clustering algorithm which builds prototypical graphs on the basis of fast matching with graph transformations. Our experiments, both with random graphs and in realistic situations (visual localization), show that our prototypes improve the set median graphs and also the prototypes derived from our previous incremental method. We also discuss how the method scales with a growing number of images.

1 Introduction

Structural criteria, graph matching, and even graph learning, have been considered as fundamental elements in the set up of the constellation (part/features-based) approach to object recognition [12]. Most of research in such direction has been concentrated in exploiting feature (local) statistics, whereas structural (global) statistics have been typically confined to the joint Gaussian of feature locations [5]. However, there has been a recent interest in modelling and learning structural relationships. This is the case of the *tree-structured models* [6][10] and the *k-fans* graph model [3]. However, models with higher relational power are often needed for solving realistic situations. In this regard, a key question is to find an adequate trade-off between the complexity of the model and the computational cost of learning and using it.

In this paper, we present a novel method for the unsupervised learning of general graph models under the constellation approach. Here, we follow central graph clustering [2][14][7], and the core element is *prototype building* or *graphs fusion*. In [16] we proposed an incremental method which depends on the order in which the graphs are fused. In this paper, we present an alternative method which overcomes such problem. It is based on the information provided by the diffusion kernels [4][11] in order to decide which matches are preferable to be considered in order to fuse the nodes of the graphs in the set. Our algorithm works both with continuous graph-matching methods like Softassign, or our kernelized version [15], and with faster alternative discrete matching methods. From this point of view, as in the constellation approach node attributes coming from describing salient features play a key role, here we also propose *graph-transformation matching* [1] a novel fast and reliable method, emerging from putative matches between feature sets, which yields a *consensus graph*, provided that such subgraph exists.

Our graph-learning method for the constellation approach is tested in a *visual localization* (scene recognition) context. The early approach is coarse-to-fine: (i) Given an input image, an appearance-based classifier, trained with the optimal (minimal) number of features finds the most probable submap; (ii) The statistics of the sub-map are exploited to speed-up the extraction of invariant salient features [9]; (iii) Given proper feature descriptors [13] graph-transformation matching finds common subgraphs with images in the same submap; (iv) The image with the highest number of nodes in the subgraph is chosen as output and the viewing coordinates are reported. Here, we compare this early design with the one resulting from replacing (iii) by finding the closest structural prototype in the submap and then match the input image only to the images in such cluster.

The rest of the paper is organized as follows. The core of our proposal, the graph-fusion method, is presented in Section 2. In Section 3 we describe the graph-transformation matching and its implications in the EM clustering algorithm, together with graph-fusion. Experimental results are presented in Section 4, and, finally, in Section 5 we present our conclusions and future works.

2 Mapping Graphs to Prototypes via Diffusion Kernels

2.1 Building the Super-Graph

Given a set of graphs S, with $N = |S|$, to be clustered, each graph $\mathbf{G}_i \in S$ is a 4-tuple $\mathbf{G}_i = (V_i, E_i, \beta_i)$ where: V_i is the set of nodes, $E_i \subseteq V \times V$ is the set of edges, $\beta_i : V_i \longrightarrow \mathbb{R}^n$ are the node attributes (descriptors of salient points). In order to obtain the prototype, firstly it is necessary to obtain $O(N^2)$ pairwise matching matrices M^{ij} between al pairs $< \mathbf{G}_i, \mathbf{G}_j > \in S \times S$ with $i \neq j$. With respect to the incremental method [14][16], pairwise matchings will be computed only once, which is critical for the efficiency of the EM-clustering (more precisely to the E-steps).

Super-Graph . The latter matching matrices will be used in order to build a super-graph $\mathbf{G}_M$ which encodes the possible matchings among the graphs in S. This super-graph is a 5-tuple $\mathbf{G}_M = (V_M, E_M, \theta, \nu, \xi)$, where

- $V_M = \cup_{i=1}^{|S|} V_i$ is the union of the nodes from all the graphs in S,
- $\theta : V_M \longrightarrow S$, is a function assigning each node in the super-graph with its corresponding graph in the original set,
- $\nu : V_M \longrightarrow \cup_{i=1}^{|S|} V_i$, is a function assigning each node in the super-graph with its corresponding node on the graphs of the original set,
- $E_M = \{< i, j >, i, j \in V_M : M^{\theta_i \theta_j}_{\nu_i \nu_j} = 1\}$, that is, two nodes will be connected if, and only if, their corresponding nodes in the graphs in the set S are matched, and

- $\xi : E_M \longrightarrow \mathbb{R}^+$ is a weighting function for the edges.

Graph Partitions . Discrete matchings $M^{\theta_i\theta_j}_{\nu_i\nu_j}$ (when applying Softassign-like methods continuous variables before cleanup are even more useful) induce disjoint partitions $P_\alpha = \{i : i \in V_M\}$. In an ideal case, each partition would have at the most one node coming from each graph in S:

$$\forall i \in P_\alpha, \nexists j \in P_\alpha : \theta_i = \theta_j, j \neq i, \forall P_\alpha \subset V_M \tag{1}$$

In this case, the fusion is easy. Each partition corresponds to a node in the prototype graph (see Fig. 1-top-left). However, in a real case, due to the matching ambiguity and errors, a partition could have some nodes from the same graph (see Fig. 1-bottom-left). We must then decide which matches are going to be taken into account in order to build the prototype, and which ones will be discarded. Matches with a higher value in the matching matrix will be preferred, because the higher is this value, the lower is the ambiguity of this match. However, there will be many nodes with the same value in the matching matrix. In order to decide which of them is preferred their kernel values will be used. Therefore, each edge $< i, j >$ from the super-graph will be weighted by a function ξ that is defined as

$$\xi(< i, j >) \longleftarrow M^{\theta_i\theta_j}_{\nu_i\nu_j} + \alpha\Phi^{\theta_i\theta_j}_{\nu_i\nu_j}, \forall < i, j > \in E_M \tag{2}$$

where α is a small value (i.e. $\alpha \sim 0.01$) and Φ is an affinity measure between matched vertices ν_i and ν_j. In this case, we define $\Phi^{\theta_i\theta_j}_{\nu_i\nu_j} = \exp\{-(K^{\theta_i}_{\nu_i} - K^{\theta_j}_{\nu_j})^2\}$ being K the diffusion kernel associated to the Laplacian of the graph $\theta \in S$ containing vertex ν_i (respectively ν_j), that is, $K = \exp\{-(\beta/m)L\}$ being $L = D - A$ where m is the number of vertices of θ, D is the diagonal matrix registering the degree of each vertex and A is the adjacency matrix. Consequently, we have that $K^{\theta_i}_{\nu_i} = K^{\theta_i}_{\nu_i\nu_i}$ is the ν_i-th element of the diagonal (similarly $K^{\theta_j}_{\nu_j} = K^{\theta_j}_{\nu_j\nu_j}$). As it is well known , the values in the diagonal of a diffusion kernel encode the probability that a lazy random walk remains at such vertex, and such probability encodes how the graph structure *is seen* from a given vertex.

The latter weights $\Phi^{\theta_i\theta_j}_{\nu_i\nu_j}$, which encode structural compatibility, will be used to insert all the edges in E_M into a sorted list $\mathcal{L}_e$. The elements with higher weights will be taken first. These edges will be used in order to build the partitions of the graph, taking into account the constraints in 1. For each edge $< i, j >$, there are 4 possible cases:

- *Neither i nor j are assigned to any partition.* In this case a new partition is created, and both i and j are assigned to it.
- *i is assigned but j is not.* Add j to the partition of i if doing this the constraints are satisfied. If not, add j to a new partition.
- *j is assigned but i is not.* Add i to the partition of j if doing this the constraints are satisfied. If not, add i to a new partition.
- *Both i and j are assigned to a partition.* If both i and j are assigned to the same partition, there is nothing to do. In other case, fuse the partitions of i and j if it satisfies the constraints.

2.2 Building the Prototypes

After the process described above, a set of partitions P_S will be obtained, satisfying $\bigcup_{P_i \in P_S} P_i = V_M$. Each partition $P_i \in P_S$ corresponds to a node in the fusion graph (prototype). Such graph is an approximation of the median graph[8] and it is defined by the 6-tuple $\bar{\mathbf{G}} = (\bar{V}, \bar{E}, \bar{\beta}, \gamma, \lambda, \mathcal{M})$, where:

- $\bar{V} = \{P_i \in P_S\}$ and $\bar{E} = \{< i, j >: \exists k \in P_i, l \in P_j \,|< k, l >\in E^{ij}\}$ where $E^{ij} = \{< k, l >: k \in P_i, l \in P_j, \theta_k = \theta_l, < \nu_k, \nu_l >\in E^{\theta_k} \equiv E^{\theta_l}\}$.
- $\bar{\beta} : \bar{V} \longrightarrow \mathbb{R}^n$ are the averaged attributes defined as $\gamma \bar{P_i} = \sum_{k \in P_i} \pi_{\theta_k} \beta_{\nu_k}$, where $\pi_{\theta_k} : S \longrightarrow [0, 1]$ indicate the probability that graph θ_k belongs to the class defined by prototype.
- $\gamma : \bar{V} \longrightarrow [0, 1]$ is the probability density of node P_i in the prototype, and it is defined as $\gamma_{P_i} = \sum_{k \in P_i} \pi_{\theta_k}$. Such probabilities will be properly normalized so that the sum of probabilities of all nodes is unitary.
- $\lambda : \bar{E} \longrightarrow [0, 1]$ are the edge weights defined as $\lambda(< i, j >) = \sum_{<k,l>\in E^{ij}} \pi_{\theta_k}$. Thus, such weights are defined by integrating the weights of the graphs to which the nodes implied in the connections belong.
- $\mathcal{M} : \bar{V} \times S \longrightarrow \cup_{i=1}^{|S|} V_i$ defines the correspondence of a vertex in the prototype and a graph with the matched vertex in the latter graph, that is $\mathcal{M}^{P_i A} = \nu_k, k \in P_i : \theta_k = A$. Having such matches we bypass the solving of a graph matching problem between each graph in S and each prototype.

As stated above, the probabilities π_{θ_k} that a graph belongs to a given prototype are here considered as external information comming from the EM algorithm (see next section) and we define the prototype as the mixture

$$\bar{\mathbf{G}} = \sum_{k=1}^{N} \pi_k \mathbf{G}_{\theta_k} = \pi_1 \mathbf{G}_{\theta_1} + \ldots + \pi_N \mathbf{G}_{\theta_N} \tag{3}$$

where $\pi_k \mathbf{G}_{\theta_k}$ denotes the weighting of each graph by its probability.

3 Graph-Transformation Matching and EM Clustering

3.1 One-to-one Matching

Given two images I_i and I_j, to be clustered, let $\mathcal{L}_i = \{\mathbf{s}_k\}$ and $\mathcal{L}_j = \{\mathbf{p}_l\}$ be their respective sets of salient points. Such salient points are obtained through a Bayesian optimization of the entropy-based Kadir and Brady dectector [17]. However, for matching purposes we consider their SIFT 128−length descriptors $\mathbf{D}$ and for each $\mathbf{s}_k$ we match it with $\mathbf{p}_l$ when $\mathbf{D}_{kl} = \arg\min_{\mathbf{p}_l \in \mathcal{L}_j} \{||\mathbf{D}_k - \mathbf{D}_l||\}$ and $\mathbf{D}_{kl}/\mathbf{D}_{kl^{(2)}} \leq \tau$ being $\mathbf{D}_{kl^{(2)}}$ the Euclidean distance to $\mathbf{s}_{l^{(2)}}$the second best match for $\mathbf{s}_k$, and $\tau \in [0, 1]$ a distinctivity threshold usually set as $\tau = 0.8$. Consequently, we obtain a set of, say M matchings $\mathcal{M} = \{(k, l)\}$, and we denote by $\hat{\mathcal{L}}_i$ and $\hat{\mathcal{L}}_j$ the sets resulting from filtering, in the original ones, features without a matching in the $\mathcal{M}$ set.

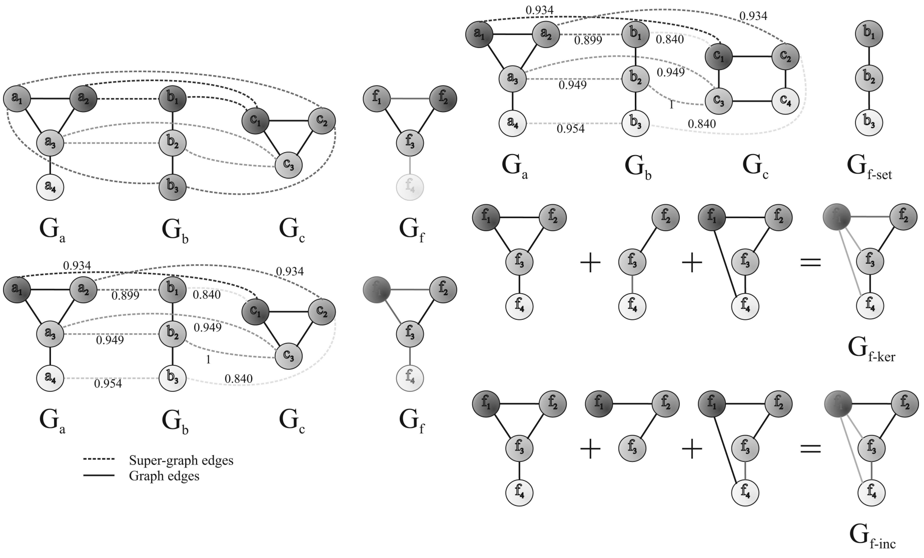

Fig. 1. Illustrating kernelized fusion. Left: Prototype building in an ideal case (top), and a real one where kernels are needed (bottom). Rigth: Step-by-step fusion showing partitions wrt median graph (left) and the difference between kernelized (middle) and incremental (bottom) fusion.

3.2 Iterative Filtering and Consensus Graph

Considering the two sets of M points $\hat{\mathcal{L}}_i \ni \mathbf{s}_k$ and $\in \hat{\mathcal{L}}_j \ni \mathbf{p}_l$, where $\mathbf{s}_k$ matches $\mathbf{s}_l$ we build their associated *median K-NN graphs* as follows. Graph $\mathbf{G}_i = (V_i, E_i)$ is given by vertices V_i associated to the positions of the M points. A non-directed edge $< k, a >$ exists in E_i when $\mathbf{s}_a$ is one of the $K = 4$ closest neighbors of $\mathbf{s}_k$ and also $||\mathbf{s}_k - \mathbf{s}_a|| \leq \eta$, being $\eta = med_{<r,t> \in V_i \times V_i} ||\mathbf{s}_r - \mathbf{s}_t||$ the median of all distances be tween pairs of vertices, which filters structural deformations due to outlying points. If there are not K vertices that support the structure of $\mathbf{s}_k$ then this vertex is disconnected completely. The graph $\mathbf{G}_i$, which is not necessarily connected, has the $M \times M$ adjacency matrix A_{ka} where $A_{ka} = 1$ when $< k, a > \in E_i$ and $A_{ka} = 0$ otherwise. Similarly, the graph $\mathbf{G}_j = (V_j, E_j)$ for points $\mathbf{p}_l$ has adjacency matrix B_{lb}, also of dimension $M \times M$ because of the one-to-one initial matching $\mathcal{M}$.

Graph Transformational Matching (GTM) relies on the hypothesis that outlying matchings in $\mathcal{M}$ may be iteratively removed: (i) Select an outlying matching; (ii) Remove matched features corresponding to the outlying matching, as well as this matching itself; (iii) Recompute both *median K-NN graphs*. Structural disparity is approximated by computing the residual adjacency matrix $R_{ij} = |A_{ka} - B_{lb}|$ and selecting $j^{out} = \arg\max_{j=1...M} \sum_{i=1}^{M} R_{ij}$, that is, the one yielding the maximal number of different edges in both graphs. The selected structural outliers are the features forming the pair $(\mathbf{s}_{j^{out}}, \mathbf{p}_{j^{out}})$, that is, we remove matching (k, j^{out}) from $\mathcal{M}$, $\mathbf{s}_k$ from $\hat{\mathcal{L}}_i$, and $\mathbf{p}_{j^{out}}$ from $\hat{\mathcal{L}}_j$. Then, after decrementing M, a new iteration begins, and the median K-NN graphs are computed from the surviving vertices. The algorithm stops when it reaches the null residual matrix, that is, when $R_{ij} = 0, \forall i, j$, that is, it seeks for finding a *consensus graph* (initial experimental evidence shows that the pruning with the residual adjacency matrix may be too agressive). Considering that the bottleneck of the algorithm is the re-computation of the graphs, which takes $O(M^2 \log M)$ (the same as computing the median at the beginning of the algorithm) and also that the maximum number of iterations is M, the worst case complexity is $O(M^3 \log M)$.

3.3 From Pairwise Matching to EM Clustering

Given N input images $I_1, \ldots, I_N$ to be clustered and characterized by their SIFT descriptors, the first step consists of performing $N \times (N-1)/2$ GTM matchings between all pairs of images, and these matching will be only performed once. The role of the pairwise consensus subgraphs is to yield mappings between the SIFT descriptors. For input image I_i, its graph for clustering purposes will be $\mathbf{G}_i = (V_i, E_i, \beta_i)$ where the vertices V_i are associated to the positions of all the salient points in the image, the edges in E_i are derived from the median K-NN graph considering all salient points, and $\beta_i = \mathbf{D}_i$.

Given N input graphs $\mathbf{G}_i = (V_i, E_i, \beta_i)$, the goal of the Asymmetric Clustering Model (ACM) for graphs [15][16] is to find K (also unknown) graph prototypes $\bar{\mathbf{G}}_\alpha = (\bar{V}_\alpha, \bar{E}_\alpha, \bar{\beta}_\alpha, \gamma_\alpha, \lambda_\alpha, \mathcal{M}_\alpha)$ and the class-membership variables $I_{i\alpha} \in \{0, 1\}$ maximizing the cost function

$$L(\bar{\mathbf{G}}, I) = -\sum_{i=1}^{N}\sum_{\alpha=1}^{K} I_{i\alpha} F_{i\alpha}, \;\; F_{i\alpha} = \sum_{k \in \bar{V}_\alpha} ||\bar{\beta}_{\alpha k} - \beta_{i\mathcal{M}_\alpha^{ki}}|| \tag{4}$$

Alternatively, $F_{i\alpha}$ may be defined in terms of the number of matchings, that is the number of vertices $k \in \bar{V}_\alpha$ satisfying $||\bar{\beta}_{\alpha k} - \beta_{i\mathcal{M}_\alpha^{ki}}|| \leq \tau$ after GTM (the dimension of the consensus graph).

Initialization . For a fixed K, after a greedy process yielding initial prototypes and membership variables, the supergraph $\mathbf{G}_M = (V_M, E_M, \pi, \theta, \nu, \xi)$ (in which all graphs are mapped) is built. As stated above, this step, which implies a quadratic number of GTM processes, will be done only once.

E-step . Membership variables are updated following a deterministic annealing process (with temperature T) and depending on the disparities $F_{i\alpha}$ with respect to the prototypes ($N \times K$ evaluations *without* performing graph matching):

$$I_{i\alpha}^{t+1} = \frac{\rho_\alpha^t e^{-\frac{F_{i\alpha}}{T}}}{\sum_{\delta=1}^{K} \rho_\delta^t \, e^{-\frac{F_{i\delta}}{T}}}, \text{being } \rho_\alpha^t = \frac{1}{N}\sum_{i=1}^{N} I_{i\alpha}^t \;, \tag{5}$$

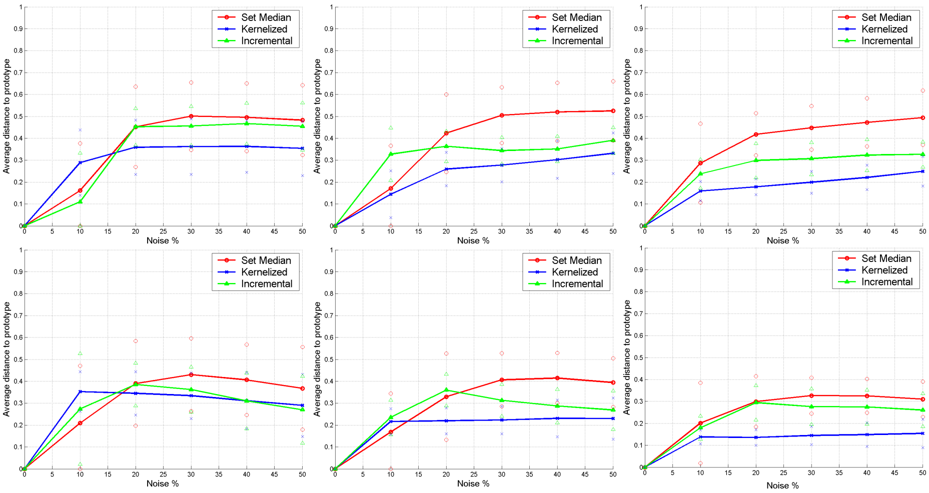

Fig. 2. Robustness with respect to increasing noise levels. Top: Edge noise. Bottom: Node noise. Left: 10% edge density. Center: 30% edge density. Right: 50% edge density.

M-step . After the E-step we have the new membership variables $I_{i\alpha}^{t+1}$ and it is time to update the K prototypes on the basis of graph mixtures whose weights rely on the current membership variables:

$$\bar{\mathbf{G}}_{\alpha}^{t+1} = \sum_{i=1}^{N} \pi_{i\alpha} \mathbf{G}_i \text{ , where } \pi_{i\alpha} = \frac{I_{i\alpha}^{t+1}}{\sum_{k=1}^{N} = I_{k\alpha}^{t+1}} \text{ ,} \tag{6}$$

Modifying weights $\pi_{i\alpha}$ implies changing the configuration (recompute partitions) of the associated fusion graph $\mathbf{G}_{M\alpha} = (V_{M\alpha}, E_{M\alpha}, \pi_\alpha, \theta_\alpha, \nu_\alpha, \xi_\alpha)$ and hence changing the prototypes (but not the supergraph), and hence their attributes $\bar{\beta}_\alpha$. After such recomputation, we proceed to prune the prototypes by discarding vertices (edges) with $\gamma_\alpha < 0.5$ ($\lambda_\alpha < 0.5$) and also their attributes.

Fusion-step . For a variable K, the complete process is started with K_{max} classes and at the end of each EM epoch a statistical test determines whether the two closest prototypes may be fused or not. Then, we compute a fused prototype

$$\bar{\mathbf{G}}_{\gamma} = \sum_{i=1}^{N} \pi_{i\gamma} \mathbf{G}_i \text{ when } h_\gamma < (h_\alpha + h_\beta)\mu \tag{7}$$

being $h_\alpha = \sum_{i=1}^{N} F_{i\alpha}\pi_{\alpha i}$ the heterogeneity of a class, and $\mu \in [0,1]$ a merge factor usually set to 0.6.

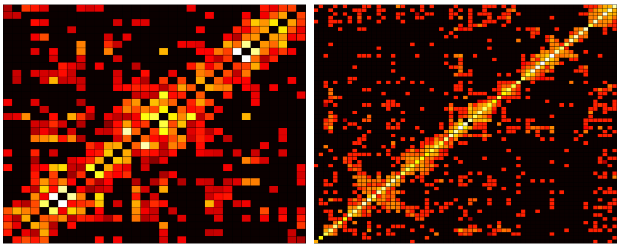

Fig. 3. Results of pairwise matchings using GTM. Left: A small environment with 34 images. Right: A larger one with 64 images.

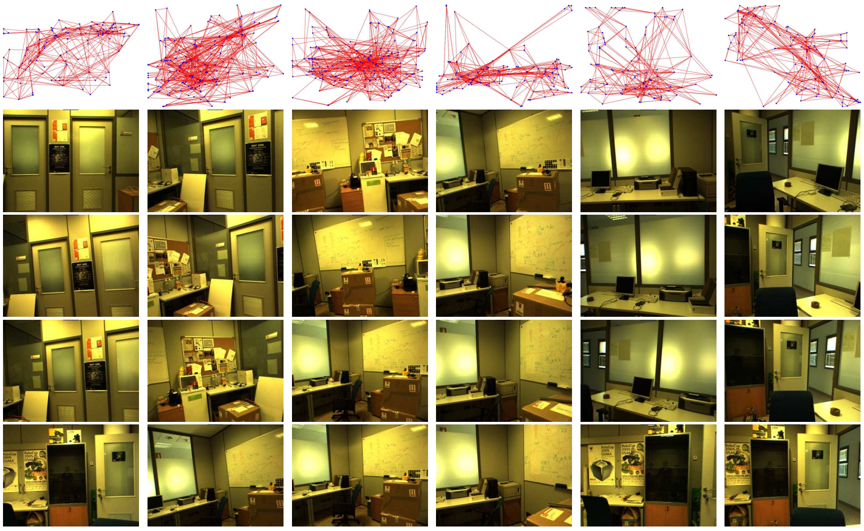

Fig. 4. Graph prototypes and classified images. Each column shows the prototype and sample images of the corresponding class.

4 Experimental Results and Discussion

Experiment 1 . We have performed two kind of experiments: random graphs, and realistic visual localization. In the first case (see Fig. 2) we have evaluated how representative is a prototype by measuring the average distance of the graphs in the class to that prototype in different situations. Compared to the set median graph and the results of our previous incremental method, the new method yields more representative (informative) prototypes: it yields a slower rate of increase of distances with the prototype as the noise level increases.

Experiment 2 . Realistic visual localization experiments where performed by considering two types of indoor environments: a small one ($N = 34$ images), and a larger one ($N = 64$ images). Initial pairwise matchings (confusion matrices) are

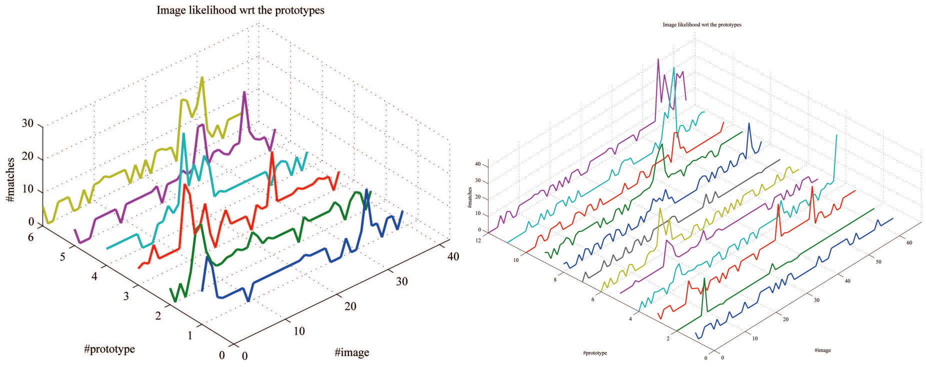

Fig. 5. Likelihoods for both enviroments. Left: Small environment. Right: Larger (medium-size) one.

showed in Fig. 3.3. The obtained prototypes for the first environment ($K = 6$ classes) are showed in Fig. 4, and the likelihood (expressed as the number of matched nodes with the prototype) of each image with respect to each prototype are plotted in Fig. 5-left. Some bimodalities (due to geometric ambiguities) arise, but in general it is possible to find, in this case, a simple threshold (above ≈ 5 matches) to report membership. In addition, we observe that the overlap between classes is minimal. We have also estimated the localization error both for the early version, which does not uses graph clustering, and the new one proposed in this paper. We have found that the percentage of error with respect to the ideal localization is 86.4% in the early version, but 58.13% in the new one. This indicates an improvement of the localization quality besides the computational savings derived from comparing only with images in the same cluster for fine localization.

Experiment 3 . The good results outlined above encouraged us to find the limit of scalability of the approach when the number of images to cluster increases significantly. In this experiment we have tested the method in a larger environment ($N = 64$ images) where our algorithm has unsupervisedly found $K = 12$ classes. Analyzing the pairwise matching matrix (see Fig. 4-right) it has a consistent diagonal with medium-size clusters. On the other hand, the analysis of the likelihoods (Fig. 5-right) reveals few multi-modal classes, and none of them has a unique member. With respect to the localization error, clustering yields a 12.22% less than our early version.

5 Conclusions and Future Works

We have presented a novel method for unsupervised central graph clustering and we have successfully tested it in the context of scene recognition (visual localization). We have found a good generalization conditioning which in turn yields useful structural indexing in coarse-to-fine visual localization provided that the number of ambiguous images does not grow significantly, specially in indoor environments where there are many natural symmetries. We are currently working in building a wearable device for incorporating these elements and also in testing the algorithm in other environments.

References

1. Aguilar, W.: Object recognition based on the structural correspondence of local features. MsThesis, UNAM, México (2006)
2. Bunke, H., Foggia, P., Guiobaldi, C., Vento, M.: Graph clustering using the weighted minimum common supergraph. In E. Hancock and M. Vento (Eds.): IAPR Workshop GbRPR 2003, LNCS **2726** (2003) 235–246
3. Crandall, D., Felzenszwalb, P., Huttenlocher, D.: Spatial priors for part-based recognition using statistical models. In: Proc. Intl. Conf. on Computer Vision and Pattern Recognition, San Diego, CA (2005) 10–17
4. Chung, F.R.K.: Spectral graph theory. In: Conference Board of Mathematical Science CBMS, vol. **92**, American Matematical Society, Providence, RI (1997)
5. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Trans. Pattern Anal. Mach. Intell. **28**(4) (2006) 594–611
6. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. International Journal on Computer Vision **61**(1) (2005) 57–59
7. Jain, B., Wysotzki, F.: Central clustering for attributed graphs. Machine Learning **56** (2004) 169–207
8. Jiang, X., Münger, A., Bunke, H.: On median graphs: properties, algorithms, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **23**(10) (2001) 1144–1151
9. Kadir, T., Brady, M.: Saliency, scale and image description. International Journal of Computer Vision **45**(2) (2001) 83–105
10. Kokkinos, I., Maragos, P., Yuille, A.L.: Bottom-up & top-down object detection using primal sketch features and graphical models. In: Proc. Intl. Conf. on Computer Vision and Pattern Recognition, New York, NY (2006) 1893–1900
11. Kondor, R., Lafferty, J.: Diffusion kernels on graphs and other discrete input spaces. In: Proc. Intl. Conf. on Machine Learning, Los Altos CA (2002) 315–322
12. Leung, T.K., Burl, M.C., Perona, P.: Finding faces in cluttered scenes using random labeled graph matching. In: Proc. IEEE Intl. Conf. on Computer Vision, Cambridge MA (1995) 637–644
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2) (2004) 91–110
14. Lozano, M.A., Escolano, F.: ACM attributed graph clustering for learning classes of images. In E. Hancock and M. Vento (Eds.): IAPR Workshop GbRPR 2003, LNCS **2726** (2003) 247–258
15. Lozano, M.A., Escolano, F.: A significant improvement of softassign with diffusion kernels. In A. Fred et al (Eds.): SSPR&SPR 2004, LNCS **3138** (2004) 76–84
16. Lozano, M.A., Escolano, F.: Protein classification by matching and clustering surface graphs. Pattern Recognition **39**(4) (2006) 539–551
17. Suau, P., Escolano, F.:Bayesian optimization of the Kadir saliency filter. Submitted to Image and Vision Computing (2006)