COMMON SPATIAL PATTERN DISCOVERY BY EFFICIENT CANDIDATE PRUNING

Junsong Yuan[1], Zhu Li[1], Yun Fu[2], Ying Wu[1], and Thomas S. Huang[2]

[1]EECS Dept., Northwestern Univ.
2145 Sheridan Road
Evanston, IL 60208, USA

[2]Beckman Institute, UIUC
405 North Mathews Avenue
Urbana-Champaign, IL 61801, USA

ABSTRACT

Automatically discovering common visual patterns in images is very challenging due to the uncertainties in the visual appearances of such spatial patterns and the enormous computational cost involved in exploring the huge solution space. Instead of performing exhaustive search on all possible candidates of such spatial patterns at various locations and scales, this paper presents a novel and very efficient algorithm for discovering common visual patterns by designing a provably correct and computationally efficient pruning procedure that has a quadratic complexity. This new approach is able to efficiently search a set of images for unknown visual patterns that exhibit large appearance variations because of rotation, scale changes, slight view changes, color variations and partial occlusions.

Index Terms— spatial pattern discovery, image data mining, approximate similarity matching, candidate pruning

1. INTRODUCTION

Recent research of common visual pattern discovery showed many potential applications in image processing and computer vision, such as near-duplicate image detection [1], image categorization [2], object discovery [3] and segmentation [4], and image similarity measure [5]. Given a collection of unlabeled images, the objective is to discover (if there is any) similar spatial patterns that appear repetitively among the images. Such common spatial patterns can be textons, a semantically meaningful part of a category of objects such as wheels of cars, or repetitive objects appearing in the image dataset.

To automatically discover common patterns from images, we need to address two critical issues: (i) measuring the "repetitiveness" of a pattern. This is not a trivial issue because the matching is subject to many possible variations including scale, rotation, viewpoint changes or partial occlusion; and (ii) efficiently discovering the "most repetitive" ones from a huge pool of pattern candidates generated by the set of images.

First of all, it is in general very difficult to define robust similarity measure between two image patches/regions, such that it is invariant to rotation, scale changes or partial occlusion. Global visual feature like color histogram is not robust enough to handle all these variations, and it is also not descriptive enough due to lack of spatial information. Many recent methods thus have aimed at extracting invariant local visual primitives (*e.g.*, corners, interest points, or coarse image regions) and represented an image as a collection of such visual primitives. When further considering the spatial relations among these visual primitives, graph based models (*e.g.*, Attribute Relational Graph) can be applied to measure the similarity. Unfortunately, finding sub-graph matchings is computationally expensive. The widely applied EM-algorithm in solving the matching problem is sensitive to the initialization [6] and has difficulty in handling the case of multiple common patterns [7]. Besides local visual invariants, image segments can also be used as visual primitives. However, they are sensitive to color and scale variations, thus special care needs to be taken.

In addition, even if invariant features can be obtained to match two sub-images, common pattern discovery is still very difficult due to the lack of prior knowledge of the common pattern. For example, it is generally unknown in advance (i) what the appropriate spatial shape of the common patterns is and (ii) where (*location*) and how large (*scale*) they are; or even (iii) whether such repetitive patterns exist. Exhaustive search through all possible pattern sizes and locations is computationally demanding, if not impossible.

We present a robust and efficient method for discovering common spatial patterns from images. Each image $\mathbf{I}_i$ is described by a set of visual primitives, $\mathbf{I}_i = \{p_1, ..., p_m\}$ where $p = \{x, y, \mathbf{d}\}$ represents a visual primitive; (x, y) denotes the spatial location; and $\mathbf{d}$ is the descriptor vector of the visual primitive. A common spatial pattern $\mathcal{P} \subseteq \mathbf{I}_i$ is a set of spatially co-located visual primitives that has good matches in other images. Instead of searching all possible pattern candidates $\mathcal{P}$ in the image dataset, we discover $\mathcal{P}$ by gradually pruning those visual primitives that do not belong to any $\mathcal{P}$. Such pruning process is provably correct since it does not discards qualified solutions. And our method is robust to different pattern variations by using local invariant visual features, and is only of a quadratic complexity of the total number of visual primitives in the database.

2. FEATURE EXTRACTION AND PRE-PROCESSING

In order to handle possible factors that incur variations in the visual pattern, such as rotation, scale and viewpoint changes, the visual primitives p need to be robust under such variations. In our implementation, we apply Scale Invariant Features (SIFT) [8] as our visual primitives although other local invariant features are certainly possible. Each SIFT descriptor $\mathbf{d}_p$ is a 128-d vector which characterizes the local invariance of a visual primitive p. After extracting SIFT points for each image, we build visual primitive database $\mathcal{D}$, whose size is denoted by $N = |\mathcal{D}| = \sum_{i=1}^{k} |\mathbf{I}_i|$, where k is the total image number and $|\mathbf{I}_i|$ denotes the visual primitive number of an image. Each visual primitive is labeled by a unique index number j ($1 \leq j \leq N$) in retrieving it from $\mathcal{D}$.

Initially, each visual primitive $p \in \mathcal{D}$ is a candidate that is a compositional element of a common spatial pattern. For example, it is possible that a common spatial pattern is located around a specific visual primitive. Our idea is to decide which visual primitives really belong to the common spatial pattern by gradually pruning those that do not make any spatial patterns.

To do so, we perform a preprocessing step of the visual primitive database for initial pruning. For each visual primitive $q \in \mathbf{I}_i$, we find all of its matches except those in $\mathbf{I}_i$, *i.e.*, finding $p \in \mathcal{D} \backslash \mathbf{I}_i$ such that $\|\mathbf{d}_p - \mathbf{d}_q\| \leq \epsilon$, where $\epsilon \geq 0$ is a matching threshold and $\|\cdot\|$ denotes the Euclidean distance. This is a typical ϵ- Nearest Neighbors (ϵ-NN) query problem, where ϵ-NN refers to the retrieved points within the distance range ϵ of the query point p in feature space $\mathcal{R}^d$. Performing the NN-query for each $q \in \mathcal{D}$ has two benefits. First, it can quickly identify those uncommon visual primitives that do not match with others in the database. Such visual primitives are mostly non-repetitive, *e.g.*, generated from the unique background of a single image, and thus can be pruned. Second, for the remaining visual primitives, their matches are found, and they will be further used in the next step of discovering spatial patterns (in Sec. 3). We define the retrieved ϵ-NN set (excluding the matches in the same image) of q as its *co-set* $\mathbf{C}_q = \{p : \|\mathbf{d}_q - \mathbf{d}_p\| \leq \epsilon, \forall i, q, p \notin \mathbf{I}_i\}$. A visual primitive q is pruned if its co-set $\mathbf{C}_q = \emptyset$.

The NN-query in large database is computationally expensive as exhaustive search is of complexity $O(|\mathcal{D}|)$ for each query, and we need to query in total $|\mathcal{D}|$ times. Although each query complexity can be reduced to $O(\log |\mathcal{D}|)$ by taking advantage of the data distribution structure in the feature space, most index-based methods such as kd-tree are only feasible in low dimensional feature spaces but cannot extend to high dimensions. Considering our high-dimensional features ($d = 128$), we choose to apply the Locality Sensitive Hash (LSH) method [9] for the approximate NN-query. Generally, LSH provides a randomized solution to the high-dimensional NN search. The query process is accelerated by compromising the results: instead of performing the exact ϵ-NN query, LSH performs approximate ϵ-NN query.

In LSH, each hash function $h(\cdot)$ is a random mapping from vector $\mathbf{d}$ to an integer, $h : R^d \rightarrow \mathcal{N}$

$$h_{\mathbf{a},b}(\mathbf{d}) = \left\lfloor \frac{\mathbf{a} \cdot \mathbf{d} + b}{r} \right\rfloor,$$

where $\mathbf{a}$ is a random vector of d-dimension and b is a random variable chosen uniformly from $[0, r]$. Under a specific hash function, two vectors p and q are matched if their hash values are identical. The closer the two vectors $\mathbf{d}_p$ and $\mathbf{d}_q$ in $\mathcal{R}^d$, the more possible that they will have the same hash value, which is guaranteed by the property of (r_1, r_2, p_1, p_2)-sensitive hash function (see [9] for details). By pre-building a set of hashing functions for the database, each new query vector q can efficiently retrieve most of its neighbors in the features space by only comparing the hash values (*i.e.*, whether they are located in the same interval) instead of calculating the distance in $\mathcal{R}^d$.

3. PATTERN DISCOVERY THROUGH FAST SPATIAL NEIGHBORHOOD MATCHING

After the pre-processing of the visual primitive database $\mathcal{D}$, a graph $\mathcal{G} = \{\mathcal{D}, \mathcal{E}\}$ can be applied to represent the matching relations among the visual primitives. For each $p \in \mathcal{D}$, we denote it as a node and the edge is defined on each pair of nodes $e = \{q, p\} \in \mathcal{E}, \forall q, p \in \mathcal{D}$. The weight of a edge $e(q, p)$ is defined as the similarity measure:

$$e(q,p) = \begin{cases} \exp^{-\frac{\|\mathbf{d}_q - \mathbf{d}_p\|^2}{\alpha}} & if\ p \in \mathbf{C}_q \\ 0 & otherwise \end{cases}, \qquad (1)$$

where $\alpha > 0$ is one parameter and $\mathbf{C}_q$ depends on the other parameter ϵ (matching threshold). This graph is a sparsely connected graph when selecting ϵ appropriately (*e.g.* if ϵ is not too large). We denote the non-isolated node set as $\mathcal{D}^1 = \{p : |\mathbf{C}_p| \neq \emptyset\} \subseteq \mathcal{D}$, while all the isolated nodes are non-repetitive visual primitives that have been pruned. Each $p \in \mathcal{D}^1$ has the potential to act as a compositional element of a common pattern, but how to further evaluate them is a critical problem. In the current stage, the evidence of $p \in \mathcal{D}^1$ being a part of a common pattern is too local, because the pattern is only supported by a single visual primitive. Thus a larger spatial neighborhood needs to be considered. For each $q \in \mathcal{D}^1$, we find its K-nearest spatial neighbors in the image to form a *spatial group* $\mathbf{V}_q$. As we need to check if $\mathbf{V}_q$ can further find matches in other images, it is important to define the similarity measure (or matching) between two groups of visual primitives rather than two individual visual primitives. The larger the size of $\mathbf{V}_q$ and the more matches it can find, the more likely it implies a common pattern.

3.1. Optimal similarity matching

For each spatial group $\mathbf{V}_q$ associated with q, we need to see if it can match with other spatial groups $\mathbf{V}_p$ associated with p. Matching two sets $\mathbf{V}_q$ and $\mathbf{V}_p$ can be formulated as an assignment problem:

$$\mathbf{Sim}(\mathbf{V}_q, \mathbf{V}_p) = \max_f \sum_{i=1}^{K} e(q_i, f(q_i)), \quad (2)$$

where $f(\cdot)$ is the assignment function that specifies which point $q_i \in \mathbf{V}_q$ is matched to a unique point $p_j = f(q_i) \in \mathbf{V}_p$. We define two spatial groups to be matched if their similarity $\mathbf{Sim}(\mathbf{V}_q, \mathbf{V}_p) \geq \lambda$, where $\lambda > 0$ is the matching threshold. In our implementation, we set $\lambda = 0.4|\mathbf{V}_p|$. The exhaustive search to solve such an assignment problem is of complexity $O(K!)$ if one-to-one unique matching is required. In our pattern discovery setting, we need to evaluate the similarity $\mathbf{Sim}(\mathbf{V}_q, \mathbf{V}_p)$ for every valid pair of visual primitives, *i.e.*, $\forall p, q \in \mathcal{D}^1$. Thus the total complexity of the exhaustive search is of $O(M^2K!)$, where $K = |\mathbf{V}_q|$ is the size of the spatial neighborhood, and $M = |\mathcal{D}^1| < |\mathcal{D}|$ is the number of valid visual primitives.

3.2. Fast approximation of similarity score

As it is computationally demanding to obtain the optimal matching score based on formulation in Eq. 2, we present an approximate solution with linear complexity $O(K \times H)$ by taking advantage of the previously built graph $\mathcal{G}$. Here H is the average size of $\mathbf{C}_p$ for $p \in \mathcal{D}^1$. In general H is a small constant depending on ϵ, where in the worst case when $\epsilon = \infty$, $H = N = |\mathcal{D}|$. The approximate similarity score is a positive integer calculated in Eq. 3, by evaluating the intersection size of two sets. Since each edge weight $e(q,p) \leq 1$ according to Eq. 1, it can be proved that the approximate similarity measure is an upper bounded estimation of the optimal matching score:

$$\begin{aligned} \tilde{Sim}(\mathbf{V}_q, \mathbf{V}_p) &= |\mathbf{V}_q \cap \mathbf{C}_{\mathbf{V}_p}| && (3)\\ &= |\{e : e(q_i, f(q_i)) \neq 0\}| && (4)\\ &\geq \max_f \sum_{i=1}^{K} e(q_i, f(q_i)) && (5)\\ &= \mathbf{Sim}(\mathbf{V}_q, \mathbf{V}_p), && (6) \end{aligned}$$

where $\mathbf{C}_{\mathbf{V}_p} = \mathbf{C}_{p_1} \cup \mathbf{C}_{p_2} ... \cup \mathbf{C}_{p_K}$. And $p_i \in \mathbf{V}_p, i = 1...K$ are the K-nearest spatial neighbors of p. Fig. 1 illustrates the matching in the case $K = 6$. A visual primitive $q \in \mathcal{D}^1$ is further pruned if its spatial group $\mathbf{V}_q$ can not find matches *even* with the exaggerated approximate similarity score, namely $\forall p \in \mathcal{D}^1, \tilde{Sim}(\mathbf{V}_q, \mathbf{V}_p) < \lambda$. Since we always have $\tilde{Sim}(\mathbf{V}_q, \mathbf{V}_p) \geq \mathbf{Sim}(\mathbf{V}_q, \mathbf{V}_p)$, the approximate similarity is a safe estimation for pruning because it does not discard qualified solutions. After pruning in $\mathcal{D}^1$, the remaining visual primitives form a new visual primitive database $\mathcal{D}^K$, with $\mathcal{D}^K \subseteq \mathcal{D}^1 \subseteq \mathcal{D}$. Compared with $\mathcal{D}^1$, each element $p \in \mathcal{D}^K$ is more possible to belong to a common pattern due to supports from its K spatial neighbors. We can continue this process until large enough common patterns are discovered.

To calculate the intersection of two sets $\mathbf{V}_q$ and $\mathbf{C}_{\mathbf{V}_p}$, of size K and $K \times H$ (on average) respectively, the complexity can be of $O(K + K \times H)$ by using a binary vector u of length M to index the elements in $\mathbf{V}_q$ and $\mathbf{V}_p$. Therefore by using the approximate similarity matching, we largely reduce the complexity of matching two sets from $O(K!)$ to $O(K \times H)$, and our total complexity is now of $O(M^2K \times H)$.

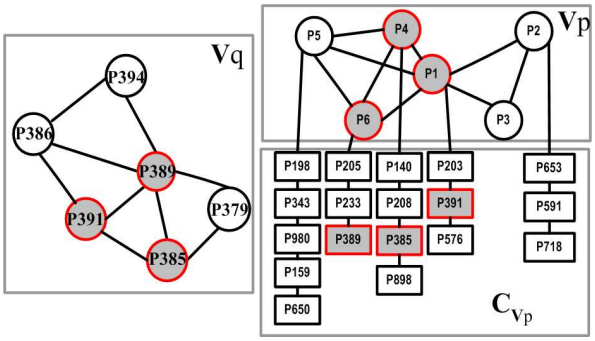

Fig. 1. Illustration of approximate similarity matching between two spatial groups $\mathbf{V}_q$ and $\mathbf{V}_p$. Specifically, here $p = p_1 \in \mathcal{D}$ and $q = p_{389} \in \mathcal{D}$. Each spatial neighbor $p_i \in \mathbf{V}_p$ of p has a matching set (*i.e.* co-set $\mathbf{C}_{p_i}$) which is represented as a string of blocks associated with p_i in the figure. For example node p_1 has a co-set $\mathbf{C}_{p_1} = \{p_{203}, p_{391}, p_{576}\}$. And p_3 is a non-repetitive visual primitive as $\mathbf{C}_{p_3} = \emptyset$. To approximately match $\mathbf{V}_q$ and $\mathbf{V}_p$ is to perform the set intersection between $\mathbf{V}_q$ and $\mathbf{C}_{\mathbf{V}_p}$, $\tilde{Sim}(\mathbf{V}_q, \mathbf{V}_p) = |\mathbf{V}_q \cap \mathbf{C}_{\mathbf{V}_p}| = 3 \geq \mathbf{Sim}(\mathbf{V}_q, \mathbf{V}_p)$.

4. EXPERIMENT

4.1. Experimental setting

The image dataset contains 10 different objects. For each object, we collect 5-10 images that contain it as the foreground object, but with different clutter backgrounds. The foreground objects are under different variations like rotation, partial occlusion, scale and viewpoint changes. For each object, we perform the pattern discovery algorithm on the image set containing that object. It is possible that an image contains multiple objects. The performance is evaluated by using two criteria: precision and recall, where precision refers to the percentage of the correctly discovered visual primitives versus the total discovered ones, and recall refers to the percentage of the corrected discovered area versus the total correct area. The discovered area is the union of the discovered visual primitives, which can be further estimated by the area of a bounding box. All the experiments are performed on a standard P4 @ 3.19G Hz PC (1 G memory). The algorithm is implemented with C++.

4.2. Common spatial pattern discovery

To verify the effectiveness of our approximate similarity measure, Fig. 2 presents one example of our results. The common pattern in the images are under variations like rotation and partial occlusions. Although color images are presented in Fig. 2, our method is not color-sensitive as the SIFT features are extracted from transfered gray-level images. For the 10

Fig. 2. Pattern discovery result. (1) 1st row: SIFT extraction results, each green point represents a visual primitive $p \in \mathcal{D}$ ($|\mathcal{D}| = 3,084$). (2) 2nd row: visual primitive pruning results based on individual SIFT matching. Red points are valid visual primitives $p \in \mathcal{D}^1$ ($|\mathcal{D}^1| = 1,078$) and green points are those pruned. (3) 3rd row: pattern discovery based on fast spatial neighborhood matching with neighborhood size $K = 25$ and matching threshold $\lambda = 10$. Red points are discovered visual primitives $p \in \mathcal{D}^K$ ($|\mathcal{D}^K| = 316$) that belong to the common spatial pattern. (4) 4th row: discovered spatial common pattern by fusing the discovered visual primitives.

objects in the dataset, very good precision can be obtained (from 0.82 to 0.98) while moderate recall can be obtained (from 0.31 to 0.78, depending on the shape and texture of the pattern). It is worth noting that although [3] also speeds up the similarity matching between two sets by quantizing visual primitives into clusters (*i.e.* visual words), our method is more robust as it does not suffer from the quantization errors.

4.3. Computational cost

After SIFT feature extraction (2-5 seconds per image), the computational cost of our mining method is composed of two parts: (1) individual visual primitive similarity query using the LSH algorithm, and (2) pattern discovery through approximate spatial neighborhood matching. Table 1 presents the computational cost comparison between using the proposed algorithm and using the exhaustive search method. The test set contains three images in Fig. 2, with resolution 568×426. The total visual primitive number is $N = 3,084$ ($M = 1,078$ after the initial pruning of visual primitives) and the dimensionality of the feature is $d = 128$. We set $\epsilon = 200$ for ϵ-NN query in $\mathcal{D}$. The query speed of LSH is very fast. Once the hash function index is built for the database (costs a few seconds), the average query time for each $p \in \mathcal{D}$ is only 1 millisecond. For each valid visual primitive $p \in \mathcal{D}^1$, the average number of the matches it can find is $H = 11.2$.

Method	Complexity	CPU cost
LSH query	$< O(N^2)$	5.5 sec
Appr. matching	$O(M^2KH)$	4.7 sec
Total Cost	$< O(N^2KH)$	10.2 sec
Exhaustive search	$O(N^2K!)$	$> 10h$

Table 1. CPU computational cost.

5. CONCLUSION

We present an efficient common spatial pattern mining algorithm for image database. With the visual primitive database size of N, its complexity is around $O(N^2)$. Compared with global image features like color histograms, our method is more robust to the pattern variations and is color-insensitive by using the SIFT points as visual primitives. Moreover, the proposed method does not have the local optimal limitation as the EM-algorithm, and can discover multiple common patterns simultaneously. Discovering such common spatial pattern among images is useful for measuring the similarity between images and for indexing image database through an unsupervised manner.

6. ACKNOWLEDGMENT

This work was supported in part by National Science Foundation Grants IIS-0347877 and IIS-0308222. The authors thank Alexandr Andoni and Piotr Indyk for the E2LSH source code.

7. REFERENCES

[1] Dong-Qing Zhang and Shih-Fu Chang, "Detecting image near-duplicate by stochastic attributed relational graph matching with learning," in *Proc. ACM Multimedia*, 2004.

[2] Josef Sivic, Bryan C. Russell, Alexei A. Efros, Andrew Zisserman, and William T. Freeman, "Discovering objects and their location in images," in *Proc. IEEE Intl. Conf. on Computer Vision*, 2005.

[3] Josef Sivic and Andrew Zisserman, "Video data mining using configurations of viewpoint invariant regions," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[4] Sinisa Todorovic and Narendra Ahuja, "Extracting subimages of an unknown category from a set of images," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[5] Oren Boiman and Michal Irani, "Similarity by composition," in *Proc. of Neural Information Processing Systemson*, 2006.

[6] Kung-Khoon Tan and Chong-Wah Ngo, "Common pattern discovery using earth mover's distance and local flow maximization," in *Proc. IEEE Intl. Conf. on Computer Vision*, 2005.

[7] Pengyu Hong and Thomas S. Huang, "Spatial pattern discovery by learning a probabilistic parametric model from multiple attributed relational graphs," *Discrete Applied Mathematics*, pp. 113–135, 2004.

[8] David Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. Journal of Computer Vision*, 2004.

[9] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab Mirrokni, "Locality-sensitive hashing scheme based on p-stable distribution," in *Proc. DIMACS workshop on Streaming Data Analysis and Mining*, 2003.