FAST COMMON VISUAL PATTERN DETECTION VIA RADIATE GEOMETRIC MODEL

Lingyang Chu[1,2], *Shuqiang Jiang*[1,2], *Qingming Huang*[1,2,3]
[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), China
[2]Institute of Computing Technology, CAS, Beijing, 100190, China
[3]Graduate University of Chinese Academy of Sciences, Beijing 100049, China
{lychu, sqjiang, qmhuang}@jdl.ac.cn

Abstract

In this paper, we propose a novel method to implement fast detection of Common Visual Pattern (CVP). The purpose of CVP detection is to find the correspondences between the common visual regions of two given partial duplicate images. There are two major components of the proposed method which guarantee the good performance. First, we establish the Radiate-Geometric-Model (RGM). The RGM is represented by a set of radiate structures, and each structure is geometrically made up of a group of matched feature pairs. By utilizing the statistical information gained from the radiate structures, the RGM can not only quickly estimate the potential pairs of common regions but also organize the scale relationship between matched pairs into a compact form, hence increase the detection speed substantially. Second, we formulize the Radiate-Geometric-Model (RGM) into a graph optimization problem which could be solved by the method of graph-shift, thus make our algorithm capable of detecting the CVPs of all kinds of correspondences. Experimental results prove that the speed of our algorithm is at least 40 times faster than the state-of-the-art, while achieving a better detection performance at the same time.

Index Terms— common visual pattern detection, Radiate-Geometric-Model, graph-shift, partial duplicate image

1. INTRODUCTION

Given a pair of partial duplicate images, there exist several common regions that are visually similar with each other. These common regions, which are called Common Visual Patterns (CVPs) in this paper, share the same visual-geometric appearances as illustrated in Fig.1. The task of common visual pattern detection is to find out all the CVPs together with their correspondences. A fast and accurate CVP detection algorithm could be used in many applications such as near duplicate image retrieval, object detection and image tagging etc.

There have been solutions trying to solve the CVP detection problem. Shapiro and Brady [1] proposed a spectral technique to find the correspondences between two sets of interest points. Leordeanu [2] proposed another spectral technique by optimizing a quadratic objective function. Caetano *et al.* [3] used markov random field to establish the correspondences between interest point sets. However, the challenges of image distortion and the complexity of CVP-correspondences remains to be a problem. Recently, Hairong Liu et al. [4] established a new algorithm which is robust to these challenges. This method [4] firstly formulate the whole CVP detection problem into a graph optimization problem, and then solve it by the method of graph-shift [5] which aims to find all the local dense sub-graphs of a complete graph and is robust to outliers. However, the detecting speed of their method is relatively slow due to the necessity of scanning all the probable scale-ratios [4].

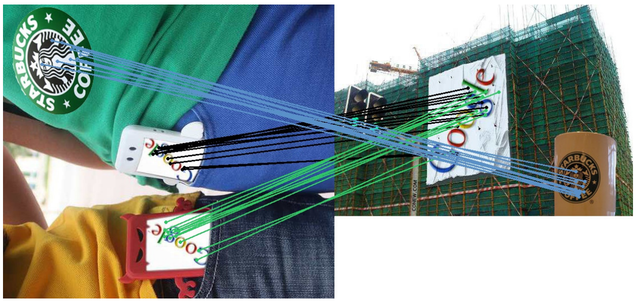


Fig. 1. An example of common visual pattern detection. The correspondence is many-to-many.

In this paper, we propose a fast common visual pattern detection method which is at least 40 times faster than [4] and achieves a better detection performance. The key to fast detection speed of our method is the RGM which is geometrically represented by a set of radiate structures (Fig. 2. (a)) and is mathematically represented by a set of weighted histograms(Fig. 2. (b)). In order to obtain this set of weighted histograms, we firstly detect all the matched pairs of feature points, such as SIFT features [6]. By considering each pair of feature points as the reference center, we can connect all the other feature points to the reference center to construct a radiate geometric structure. Then, the RGM is obtained by transforming each radiate geometric structure into a weighted histogram.

The set of weighted histograms are then used to construct a complete graph whose nodes consist of the set of matched feature points and edge weights are updated by the statistical information extracted from the RGM. In this way, the CVP detection problem is transformed into a graph optimization problem which aims to find the sub dense-graphs of the complete graph and could be solved by the method of graph-shift [5].

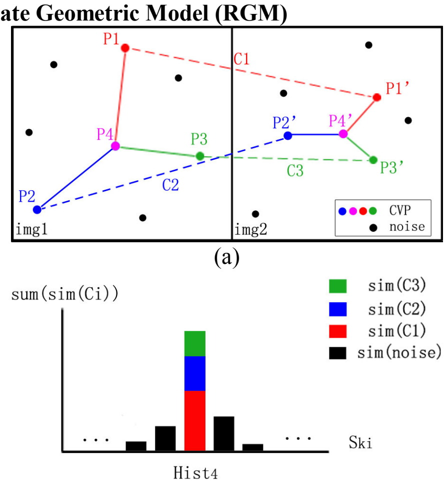
This paper is organized as follows: the proposed algorithm of CVP detection is introduced in section 2. The experimental results are shown in section 3 and section 4 is the conclusion.

978-1-4577-1303-3/11/$26.00 ©2011 IEEE

2. PROPOSED ALGORITHM

In this section, the RGM together with its statistical property is firstly introduced in section 2.1. Then we give a detailed explanation on how to formulize the RGM into a graph optimization problem in section 2.2. At last, in section 2.3, how to solve the optimization problem by graph-shift is discussed.

2.1. Radiate Geometric Model (RGM)



(a)

(b)

Fig. 2. An illustration of radiate geometric structure. (a) is the geometrical form; (b) is the corresponding mathematical form. The solid lines indicate the relations of the structure and the dash lines indicate the matched points. The noisy lines are not drawn.

As is shown in Fig. 2, at first, we coarsely match the local feature points, such as SIFT features [6], between the two partial duplicate images, the coarsely matched feature points are denoted as:

$$C_i = (P_i, P_i'), i \in N \qquad (1)$$

where N is the index set of all the matched feature points, and C_i is called a ***match***. Given a match C_i, a similarity score could be computed as:

$$sim(C_i) = \exp(-feaDist(feature_i, feature_{i'}) / \delta) \qquad (2)$$

where $feaDist(feature_i, feature_{i'})$ is the distance of the two features and δ is a predefined scale parameter.

For a pair of *match* (C_i, C_j), we define the scale ratio parameter as:

$$s_{ij} = dist(P_i, P_j) / dist(P_i', P_j') \qquad (3)$$

where $dist(P_i, P_j)$ is the pixel distance between P_i and P_j, for example, the pixel length of the solid segments in Fig. 2 (a) img1.

Given a set of *matches* $\{C_i \mid i \in M\}$ where M denotes the index of a set of points that lies in the same CVP, a property of the scale ratio parameter has been proved by [4]:

Property1: If $L = \{s_{ij} \mid i, j \in M; i \neq j\}$, then $\sigma(L) \approx 0$

The small standard deviation value of scale ratio parameter means that it stays stable for most of the valid *matches* in the same CVP.

Now, we can use *property1* to formulize the radiate geometric model. Given all the *matches* $\{C_i \mid \forall i \in N\}$ and a single reference *match* $C_k, k \in N$, we can obtain a set of scale ratio parameters:

$$S = \{s_{ki} \mid k \in N, \forall i \in N, i \neq k\} \qquad (4)$$

which correspond to the pairs of solid lines as is shown in Fig. 2. (a). Then S is quantized by a predefined step γ to form a histogram $hist_k$ whose x-axis is the quantized scale ratio parameter and y-axis is the sum of $sim(C_i)$ for all C_i whose corresponding scale ratio parameter S_{ki} belongs to the same bin (Fig. 2. (b)). The subscript k indicates that the $hist_k$ uses C_k as the reference *match*. At the same time, a weight w_k is assigned to the histogram $hist_k$ which is:

$$w_k = \max(hist_k) / mean(hist_k) \qquad (5)$$

This makes the couple of $(hist_k, w_k)$ which is a weighted histogram. Obviously, we can obtain a set of weighted histograms, *which is the mathematical form of our Radiate-Geometric-Model:*

$$H = \{(hist_k, w_k) \mid k \in N\} \qquad (6)$$

According to *property1*, it would be easy for us to infer the following statistical property of H :

Property2: For a weighted histogram $(hist_k, w_k)$ with a large w_k, the corresponding reference *match* C_k is more likely to lie in a CVP. And the *matches* that are assigned to the bin with max y-value are more likely to lie in the same CVP as the one that C_k lies in. On the contrary, for those histograms with a low w_k, the reference *match* C_k is less likely to lie in a CVP and the information it carries is more likely to be noise.

Explanation: According to *property1*, for a reference *match* C_k that lies in a CVP, the scale ratio parameter s_{ki} which corresponds to the couple of $(C_k, C_i), i, k \in M$ would be stable, thus those valid *matches* are most likely to be assigned to the same bin of $hist_k$ which makes the y-value of that bin very large. However, according to experiments, the noise *matches* are more likely to be randomly assigned to all the bins of $hist_k$. Considering the fact that the mean value stays the same for all $hist_k$, the result of a large w_k is obvious. On the contrary, for a reference *match* that does not lie in a CVP, all the other *matches* tend to be randomly assigned to all the bins which is more likely to result in a small w_k.

2.2. The Graph Model

The mathematical definition of a graph is $G(V, E, W)$ where V is the set of vertices, E is the set of edges and W is the set of edge weights. In our model, the vertex set V consists of the *matches* $\{C_i\}$ and is simply defined as $V = \{C_i \mid \forall i \in N\}$; the edge set E is defined as $E = \{(v_i, v_j) \mid i, j \in N, i \neq j\}$ with the set of weights $\{weight_{ij}{}^0\}$ initialized to zero. In the rest of this section, we will focus on how to use the statistical information extracted from the RGM to update the weight of each edge. The updating strategy consists of two parts:

1) For each weighted histogram $(hist_k, w_k)$ in RGM, we obtain the relationship between reference *match* and the other *matches*

which is represented by $R_k : C_k \xrightarrow{w_k} \{C_i \mid i \in Q\}$, where w_k is the weight of $hist_k$, C_k is the reference *match* and Q denotes the index set of $\{C_i\}$ that are assigned to the bin whose corresponding y-value is the maximum in $hist_k$.

2) For each relationship R_k, we update the weight of graph edges by Equ. (7):

$$weight_{ij}^{t+1} = weight_{ij}^{t} + w_k \quad \forall i,j \in \{k \cup Q\}; i \neq j \qquad (7)$$

where $weight_{ij}^{t}$ is the value of the graph edge weight at the t-th iteration.

The graph is completely constructed after all the R_k have been used to update the edge weight. For the set of valid *matches* that lie in the same CVP, the edge weights between each other would all be large. The reason for this phenomenon is that for the relation of $R_k : C_k \xrightarrow{w_k} \{C_i \mid i \in Q\}$ with a large w_k, the *match* set of $\{C_i \mid i \in Q\}$ has a very high probability to lie in the same CVP as the one that C_k lies in. This results in the fact that for every reference *match* C_k that lies in a CVP, it is most likely to collect the other valid *matches* into its set Q with a high weight w_k. In this way, the edge weights between valid *matches* would be updated many times by a set of large w_k, thus become very large finally. These edges with large weights connect the valid *matches* into a dense sub-graph[5].

However, for those invalid *matches* that do not lie in any CVP, the situation is completely the opposite way which means that they are almost impossible to be part of any dense sub-graph. So the task of common visual pattern detection now equals to the optimization problem of finding the dense sub-graphs of graph $G(V,E,W)$. This problem can be solved by the method of graph-shift [5]. In sub-section 2.3, we will discuss the graph-shift method and a tricky estimation procedure to do the initialization which saves a lot of computing time.

2.3. The Graph-Shift Method

There are two vital factors of the graph-shift algorithm: 1) the graph $G(V,E,W)$; 2) an initial vector X. The graph is pre-constructed and the definition of vector X is: $X : V \to \Delta^n$ where $\Delta^n = \{x \in R^n : x \geq 0; |x|_1 = 1\}$ [5], which is the mapping from vertex set V to the standard simplex R^n, each point $x \in \Delta^n$ is a probabilistic combination of vertices, which is called a *probabilistic cluster* [5]. Let x_i denote the i-th component of vector $x \in \Delta^n$ and the value of x_i measures the probability that vertex v_i be contained by the probabilistic cluster. The probability cluster $x \in \Delta^n$ indicates a sub-graph of $G(V,E,W)$, and a graph density function $g(x)$ is defined to measure the connection density between the vertices in $x \in \Delta^n$.

In short, the method of graph-shift aims to find all $x^* \in \Delta^n$ that correspond to a local maximum $g(x^*)$ by an optimized iteration algorithm and the set of $\{x^*\}$ indicate the set of dense sub-graphs which are exactly what we need.

The initialization strategy of $x \in \Delta^n$ affects the computing time and the final result of graph-shift a lot. [4] uses all the vertices, as the init-value each for once. This wastes a lot of computing resources. However, we invent a little trick to estimate the init-values which avoids the necessity of scanning the full set of init-values.

For the relation R_k there is a related bin value that corresponds to the quantized scale-ratio parameter which has the maximum y-value in $hist_k$. Hence the R_k should be modified to be $\tilde{R}_k = (R_k, Bin\_ID)$. Then we can use the Bin_ID to cluster the full set of $\{\tilde{R}_k\}$ into several subsets which are denoted as $\{\tilde{R}_k\}_{bin\_id}$ and simply count the set of $\{C_i \mid i \in k \cup Q; k, Q \in \{\tilde{R}_k\}_{bin\_id}\}$ into a probability cluster which is treated as an init-value $x \in \Delta^n$.

This estimation trick effectively decreases the size of init-value set. The good performance of the final detection result proves the rationality of this estimation method.

3. EXPERIMENTS

In this section, we mainly focus on the comparison of detection performance between our method and the method of [4]. There are two factors of performance to be considered: a) computing time which is in inverse proportion to detection speed; b) detection accuracy which is calculated from the number of valid matches and correct CVPs.

The experiment is implemented on 408 pairs of near duplicate images which is a sub set of [7] (more illustrations of the data set and specified explanation of our method and experimental results is available at http://www.jdl.ac.cn/en/project/mrhomepage/FCVPD/FCVPD.html). We firstly run the two detection methods on each pair of the images, and then manually label the numerical evaluations of the two performance factors. Finally, for a clearer representation, we draw the comparing results for both factors in Fig. 4 and give the numerical results in Table. 1.

Note that there is a very important scale parameter t which is scanned in the method of [4] and the scanning step Δt is vital to their detection performance. For the fairness of comparison, we run their method for three times with the scanning step set to three typical values ($\Delta t = [0.2, 0.5, 0.8]$) and compare our results with each one of them.

Here is how to obtain the ***numerical evaluations*** of the two performance factors:

a) Computing time: for each method, we record the time to complete the detection task on each single pair of images. The computing time for detecting the feature point *matches* is not considered for both methods. Let $time_1(k)$ denote the computing time of our algorithm and $time_2(k)$ for the method of [4], where the subscript k indicates the k-th pair of the 408 near duplicate images.

b) Detection accuracy: for the k-th pair of images, we manually label the number of valid *matches* (denoted by $m_i(k)$) and correct CVPs (denoted by $n_i(k)$). Then, we use the pair of $[m_i(k), n_i(k)]$ to evaluate the performance of detection accuracy, where the subscript $i \in \{1,2\}$ indicates the two compared methods (with $i = 1$ for our method and $i = 2$ for the method of [4]).

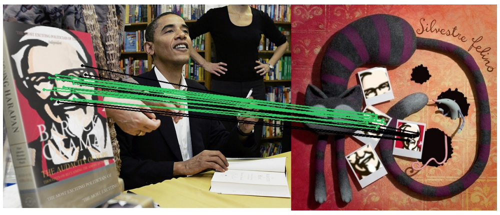


Fig. 4. This is a result of our method. There are two pairs of "KFC" detected in the two partial duplicate images, hence the number of CVPs should be $n_1 = 2$. The number of valid match obtained by counting the number of valid matched lines in the detection result is $m_1 = 20$ and the calculation time recorded by computer is $time1 = 0.043\sec$.

Fig. 4 gives an example of the ***numerical evaluation***. After obtaining the ***numerical evaluations*** of the detection performance, we use Equ. 8 and 9 to compare the detection speed and accuracy of the two methods:

$$time\_ratio(k) = time_1(k) / time_2(k) \qquad (8)$$

$$accuracy\_ratio(k) = \frac{m_1(k)+1}{m_2(k)+1} * \frac{n_1(k)+1}{n_2(k)+1} \qquad (9)$$

where $time\_ratio(k) < 1$ indicates our advantage of detection speed against [4] and $time\_ratio(k) > 1$ means the opposite. $accuracy\_ratio(k) > 1$ indicates our advantage of detection accuracy against [4] and $accuracy\_ratio(k)$ means the opposite.

For the k-th pair of images, we firstly calculate the $time\_ratio(k)$ and $accuracy\_ratio(k)$ by Equ. 8 and 9, then use the Cartesian coordinates of Equ. 10 to draw one point in Fig. 5. (We draw the figure in the metric of $\log_{10}$ for a clearer representation)

$$[\log_{10}(time\_ratio(k)), \log_{10}(accuracy\_ratio(k))] \qquad (10)$$

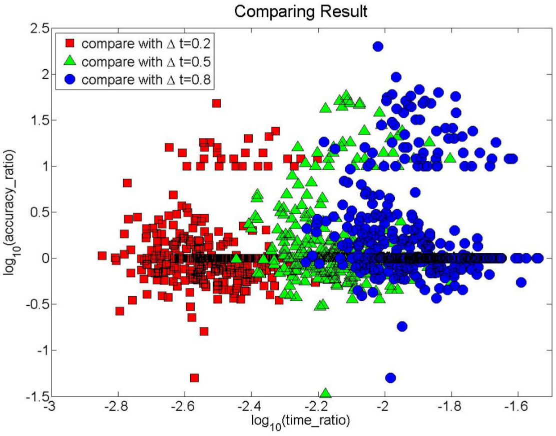


Fig. 5. The comparing result between our method and the method of [4] with $\Delta t = [0.2\ 0.5\ 0.8]$. For a data set of 408 pair of images, there are 408 points for each kind of shape and $408 \times 3 = 1224$ points in all.

As is shown in Fig. 5, each point corresponds to a pair of near duplicate images. According to Equ. 10, the x-coordinate shows the comparing result of computing time and the y-coordinate shows the comparing result of detection accuracy. We compare our method with the method of [4] whose parameter Δt is set to three different values and the comparing results are represented by three kinds of shapes (red square, green delta and blue circle).

The following results can be easily figured out from Fig. 5:

a) *Our method is at least 40 times faster than the method of [4]*: considering all the 1224 points in Fig. 5, the fastest performance of [4] is achieved at the right most blue circle with $\Delta t = 0.8$ which has an x-coordinate value to be about $\log_{10}(time\_ratio) = -1.6$. This is equivalent to a 40-times speed gain.

b) *Our method has achieved a better performance of detection accuracy*: the best detection accuracy performance of [4] is achieved by $\Delta t = 0.2$. However, as it is shown in the comparison plot of Fig. 5, the red squares indicates that the average accuracy of our method is still 1.975 times better according to Table. 1.

Table. 1. This table gives the average time ratio and average accuracy ratio of the points in Fig. 5. (Not considering the factor of $\log_{10}$).

	Our method	
	$average\_time\_ratio$	$average\_accuracy\_ratio$
$\Delta t = 0.2$	0.0033	1.9715
$\Delta t = 0.5$	0.0082	3.2013
$\Delta t = 0.8$	0.0131	5.6529

The comparing results in Table. 1 also prove that our method is better in both speed and accuracy.

4. CONCLUSION

In this paper, we propose a Radiate-Geometric-Model (RGM) and a method to transform the RGM-based CVP detection problem into a graph optimization problem. Our method is at least 40 times faster than the state-of-the-art and achieves a better performance of detection accuracy. The large amount of speed gain is mostly due to the Radiate-Geometric-Model and the graph-shift algorithm ensures the robustness of our method. We are looking forward to developing this method into a similarity measurement which could be used to re-rank the results of image retrieval applications.

5. ACKNOWLEDGEMENT

This work was supported in part by National Natural Science Foundation of China: 61025011, 60833006 and 61070108, in part by National Basic Research Program of China (973 Program): 2009CB320906, and in part by Beijing Natural Science Foundation: 4092042.

6. REFERENCES

[1] L. Shapiro and J. Brady. Feature-based correspondence: an eigen vector approach. IVC, 1992.

[2] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pair wise constraints. ICCV, 2005.

[3] T. Caetano, T. Caelli, D. Schuurmans, and D. Barone, Graphical models and point pattern matching. TPAMI, 2006.

[4] Hairong Liu and Shuicheng Yan. Common Visual Pattern Discovery via Spatially Coherent Correspondences. CVPR, 2010.

[5] Hairong Liu and Shuicheng Yan. Robust Graph Mode Seeking by Graph Shift. ICML, 2010.

[6] D. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 2004.

[7] Zhipeng Wu, Q. Xu, S. Jiang, Q. Huang, P. Cui and L. Li. Adding Affine Invariant Geometric Constraint for Partial-Duplicate Image Retrieval. ICPR, 2010.