Graph Based Image Matching

Hui Jiang
Department of Computer Science
City University of Hong Kong
83 Tat Chee Aveune, Kowloon, Hong Kong
jiangh@cs.cityu.edu.hk

Chong-Wah Ngo
Department of Computer Science
City University of Hong Kong
83 Tat Chee Aveune, Kowloon, Hong Kong
cwngo@cs.cityu.edu.hk

Abstract

Given two or more images, we can define different but related problems on pattern matching such as image registration, pattern detection and localization, and common pattern discovery. These problems have different levels of purpose and difficulties, as a result, often associate with different solutions. In this paper, we propose a novel approach to solve these problems under a unified framework based on graph matching. We first split the images into small blocks and represent each block as a node in a bipartite graph. A maximum weighted bipartite graph matching algorithm is then employed in an iterative way to find the best transformation set. Experimental results show that our approach can handle rotation, scaling and translation, as well as distortion and occlusion. Another virtue of our approach is its efficiency.

1. Introduction

Given two or more images, we may define different but related problems on pattern matching such as image registration (finding the transformation under which an image fits best to another), pattern detection and localization (detecting whether a small image is a subimage of another big one, and when it is, locating the position of it), and common pattern discovery (finding the maximum common subimage of two or more images). In pattern detection, localization and discovery, since the occurrence of the pattern in the image can be a rotated, scaled, translated, and even noisy, distorted, and occlusive version of the original pattern, these problems are in general difficult.

Intuitively, image registration can be regarded as the "easiest" problem, pattern detection and localization are harder than image registration, and common pattern discovery is the "hardest". The reason is that the latter problem has a larger search space for parameters of transformations and subimage locations than the former ones. For example, in the problem of common pattern discovery, we should determine not only the transformations between the unknown common patterns embedded in images, but also the locations of these patterns.

To date, various approaches have been proposed for each of these problems [1, 3, 4, 6]. In [6], color histogram is used to solve the pattern detection and localization problem. The authors introduced a technique called *Histogram Intersection* which is efficient and effective for a large number of images. In [4], an enhanced version of histogram which is called *color correlogram* is introduced. Besides color, correlogram also takes into account spatial information which is proved to be more effective than color histogram for some images. Both histogram and correlogram can be employed to find the location of the pattern, but the size of a pattern need to be known, and the parameters of transformation (rotation angle, scaling factor) can not be implicitly computed after pattern matching.

For common pattern discovery problem, Hong and Huang [3] proposed an approach based on EM algorithm. Their approach builds an attributed relational graph (ARG) by segmenting the images into regions and then employ EM algorithm to "learn" a graph which is actually a common subgraph of all the ARGs of the images. Due to the limitation of image segmentation algorithms, the patterns must be composed of colorful and sharp-edged regions so that it can be easily segmented and represented into a same subgraph in all the images. Furthermore, since the pattern is "learned" from ARGs, the number of the images should be large enough (more than 10 in their experiments) so that enough information of the pattern can be acquired.

In this paper, we propose a unified approach for these problems. We first split the images into small blocks and represent each block as a node in a bipartite graph. An iterative optimization framework based on the maximum weighted bipartite graph (MWBG) matching [5] and flow iteration (FT) algorithm [2] are employed for pattern localization and estimation of the best transformation.

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE

2. Problem Formulation

The problems mentioned in Section 1 (image registration, pattern detection and localization, common pattern discovery) are formulated as image matching problems under our unified framework. Before problem formulation, we briefly introduce the notations.

2.1. Notations

An image can be considered as a function $I : \Omega \mapsto C$, where Ω is the pixel coordinate space, and C is the pixel-value space. We use $C = [0, 1]^3$ for RGB images, and write $I : \Omega \mapsto C$ as I_Ω for convenience.

We say an image $I^*_{\Omega^*}$ is a subimage (or region) of another image I_Ω, if $\Omega^* \subset \Omega$ and $I^* = I|_{\Omega^*}$. We can also regarded $I^*_{\Omega^*}$ as I^*_Ω, with the extra pixel values in $\Omega - \Omega^*$ remained undefined or defined as some special values, depending on the application. So in the rest of this paper, we will omit Ω and write I_Ω as I unless specially needed.

Let $\mathbf{I}$ denotes the set of all the images. A *similarity functions* is defined as a function $F : \mathbf{I}^2 \mapsto [0, 1]$. A typical similarity function is the pixel-wise matching:

$$F(I_\Omega, J_\Omega) = \sum_{p \in \Omega} |I_\Omega(p) - J_\Omega(p)|^2$$

A *transformation* is defined as a function $T : \mathbf{I} \mapsto \mathbf{I}$. The transformation can be taken on either Ω (e.g., translation, rotation or scaling), or C (e.g., lighting variation), or both.

2.2. Image Matching

With the above notations, the image registration, pattern detection and localization, and common pattern discovery problems can be formulated as follows:

Image Registration: given a similarity function F and two images I and J, find the best transformation T which satisfies

$$T = \operatorname*{argmax}_T F(T(I), J)$$

Pattern Detection and Localization: given a similarity function F, and two images I and J, find the best transformation T and subimage J^* of J which satisfy

$$T, J^* = \operatorname*{argmax}_{T, J^*} F(T(I), J^*)$$

Common Pattern Discovery: given a similarity function F and two images I and J (common pattern discovery for more than two images is a little bit troublesome and will be introduced later in Section 5.3), find the best transformation T, subimage I^* of I and subimage J^* of J which satisfy

$$T, I^*, J^* = \operatorname*{argmax}_{T, I^*, J^*} F(T(I^*), J^*)$$

We can find that these problems are defined in a similar manner, matching two images under transformations. We can also see that pattern discovery problem is "harder" because it has more unknown parameters. Even the "easiest" problem, image registration, is actually not easy to solve because the transformation space is usually huge. The naive brute-force approach is feasible only if the transformations are quite simple. For example, when we consider only translations, the problem can be solved by cross correlation.

3. MWBG Matching Algorithm

The similarity function is very important because it should be consistent with human perception on image similarity, and it affects the matching algorithm greatly. For example, the pixel-wise matching is sensitive to noise and requires quite a heavy computation. The histogram intersection approach requires much less computation and is robust to noise, but it often leads to false matches due to its lack of spatial information. Our approach is based on the maximum weighted bipartite graph (MWBG) matching algorithm [5], which can be formulated as follows:

Given two images I and J, we first partition them into small subimages:

$$I = \bigcup_{i=1}^{n} I_i, J = \bigcup_{j=1}^{m} J_j$$

Then we build a weighted bipartite graph $G =< U, V, E >$, where U has n nodes $U = \{U_i | i = 1..n\}$ corresponding to $I_i (i = 1..n)$, and similarly, V has m nodes $V_j (j = 1..m)$ corresponding to J_j. The weight W_{ij} of the edge E_{ij} connecting two nodes U_i and V_j represents the similarity between two subimages I_i and J_j. That is, $W_{ij} = F(I_i, J_j)$, where F is a similarity function. In our implementation, F is calculated by using the *histogram intersection* algorithm [6] on the two subimages I_i and J_j. Finally, if $F(I_i, J_j)$ is less than a threshold δ_F, the edge E_{ij} is deleted from G.

After we built the graph, we can calculate the similarity between two images I and J by using the maximum weighted bipartite graph (MWBG) matching algorithm [5], i.e., the similarity between I and J is defined as the total weight of the edges in the maximum matching.

4. Iterative Optimization Framework

Similar to histogram intersection, the above MWBG matching algorithm actually defines a similarity function. In fact, the MWBG matching can be regarded as an "generalized" version of histogram intersection (To see this, consider an extreme situation of the MWBG matching, where each pixel is regarded as a subimage, then the MWBG matching is just a kind of histogram intersection). Also similar to histogram intersection, no spatial information is taken into consideration in the above MWBG matching. To fix this problem, we introduce spatial information into MWBG matching as follows:

Instead of using only the histogram intersection in the similarity function F on two subimages I_i and J_j, we use a modified version F^*, where F^* is a weighted average of F and another similarity function S, i.e.,

$$F^*(I_i, J_j) = \frac{w_1 F(I_i, J_j) + w_2 S(I_i, J_j)}{w_1 + w_2}$$

where $w_1 = w_2 = 1$ in our implementation. S is a similarity function which takes spatial relation of I_i and J_j into consideration. We use the following Gaussian-like function for calculating S:

$$S(I_i, J_j) = e^{-\frac{d_{ij}^2}{\delta_d}}$$

where d_{ij} is the Euclidean distance between the centers of I_i and J_j, and δ_d is a distance threshold.

Till now, we get our definitions of the image matching problems. For example, image registration problem is now formulated as:

Given F, I and J as before, find the best transformation T which maximizes $MWBG_{F^*}(T(I), J)$, where $MWBG_F(I, J)$ is the maximum matching of the bipartite graph built with image I, J and similarity function F.

To solve this optimization problem, an iterative approach is proposed as follows:

$$M^{(k)} = MWBG_{F^*}(T^{(k)}(I), J) \tag{1}$$

$$T^{(k+1)} = \underset{T \in \Omega}{\text{argmax}} \sum_{E_{ij} \in M^{(k)}} F^*(T(I_i), J_j) \tag{2}$$

where $M^{(k)}$ and $T^{(k)}$ are the optimal matching and optimal transformation respectively at step k, and Ω is the transformation sets including translation, rotation and scaling. The iteration begins with an initial transformation $T^{(0)}$. The Equation 1 and Equation 2 alternate between finding an optimal matching given a transformation and an optimal transformation given a matching. This iterative algorithm is inspired from the FT (flow transformation) algorithm in [2].

5. Applications and Experiments

5.1. Image Registration

The uses of our approach on image registration problem is straightforward, since we can get the best transformation T directly from the proposed iterative optimization framework. We take 10 images for experiments. Each image is rotated with a random angle α between $0 \sim 2\pi$, scaled with a random scaling factor ρ between $0.5 \sim 2.0$, and translated with a random displacement $(\delta x, \delta y)$ between $-200 \sim 200$ pixels. We use the transformed image and original image as two images I and J in our algorithm. Both images are partitioned into 20×20 blocks, and each block is used as a subimage in the MWBG matching. After the iteration, we check whether the transformed image I is close enough to image J, i.e., the distance between matched pair of image blocks is close enough. In all the 10 experiments, the two images are matched well under the transformations found by our algorithm. The mean and standard deviation of the differences between the random generated transformation parameters and the parameters calculated by our algorithm are given in Table 1. We can find that the rotation angle and scaling factor found by our program are very close to the random generated ones. For the displacements, 20 pixels is acceptable since the typical image size used by us is about 800×600 and the block size is 40×30 as a result.

α	ρ	δx	δy
0.01 ± 0.01	0.01 ± 0.01	20 ± 15	16 ± 12

Table 1. Results of image registration.

α	ρ	δx	δy
0.08 ± 0.02	0.18 ± 0.05	11 ± 6	18 ± 8

Table 2. Results of pattern localization.

One of the ten experiments is shown in Figure 1 I, where (a) is the original image and (b) is the random transformed image. (c) and (d) show the matched blocks and the distances between them (darker blocks stand for less similar). The image transformed from (a) by the parameters found by our program is shown in (e). We can see that (e) is very close to (b). In real applications, we can further employ another local search in pixel level rather than block level to optimize the parameters.

To test the robustness of our approach, we also do the following to the transformed images: add random distortions (waves and skewness) up to 20 pixels, and occlude random part of the image up to 20% of the image size. We find that the experimental results are similar to those in Table 1.

5.2. Pattern Detection and Localization

The pattern detection and localization problems are more difficult than the image registration problem, since it allows subimage to be matched for similarity measuring instead of the whole original image. However, what we need is only a small modification of our previous approach for image registration: after we get the transformation and matching, we remove the matched pairs of image blocks which have large distance, and check whether the major part of pattern image is in the matching.

Again, we use 10 images in experiments. This time we cut $10 \sim 20\%$ of the original images as patterns and random transformations are taken on these patterns. Our algorithm successfully detects the occurrence of the patterns in the images, and finds their locations, sizes and poses. Experimental results are given in Table 2 and Figure 1 II. This time (b) is the original image and (a) is the random transformed pattern. Again, our algorithm passed the previous robustness testing.

5.3. Common Pattern Discovery

This time the problem we are facing is even harder, because we have to detect the locations of both of the two subimages inside the original images. However, similar to pattern localization problem, we only need to slightly modify our algorithm: After we get the transformation and matching, we remove pairs of image blocks which have large distance. Then we search and record the maximum connected regions whose component blocks are all in the matching. Two experiments are given in Figure 1 III, where human and animals in different photos are detected.

For more than two images, we first compute common pattern for each pair of the images. Then we count the number of times the image blocks are in the detected patterns, and then select the pattern with the greatest average count per block as the most frequently occurred pattern in the images. Finally we employ our pattern detection and localization algorithm on all the images to detect all the occurrences of the pattern. An experiment is shown in Figure 1 IV, where 5 photos are used in the experiment. Four of them are photos of a doll, while the other one is not. Our algorithm successfully detects the occurrences of the dolls. Figure 1 V gives some intermediate results, for instance V (a) and (b) shows the matching result between IV (a) and (b). For all the 10 pairs of images, our program gets good or acceptable results in 5 of them. The other 5 pairs are $(a) - (d)$ and (e) because (e) has no doll, and (b) and (d) because they have common backgrounds which are larger than the doll as shown in Figure V (b) and (d).

6. Conclusion

Out iterative MWBG matching approach is essentially a gradient descent search. Like other optimization problems, this kind of search can only reach a local optimum instead of a global one. However, by introducing spatial information into MWBG matching, we find that the search space is usually suitable for searching. In most experiments, our program finds the correct parameters in about 10 to 20 iterations, and the average running time of our program is under 5 seconds on a Pentium-III machine with 1G CPU and 256MB memory. Since finding the global optimum is often very hard and the brute-force approach is usually the only one which can guarantee this, the efficiency and effectiveness of our approach make it promising.

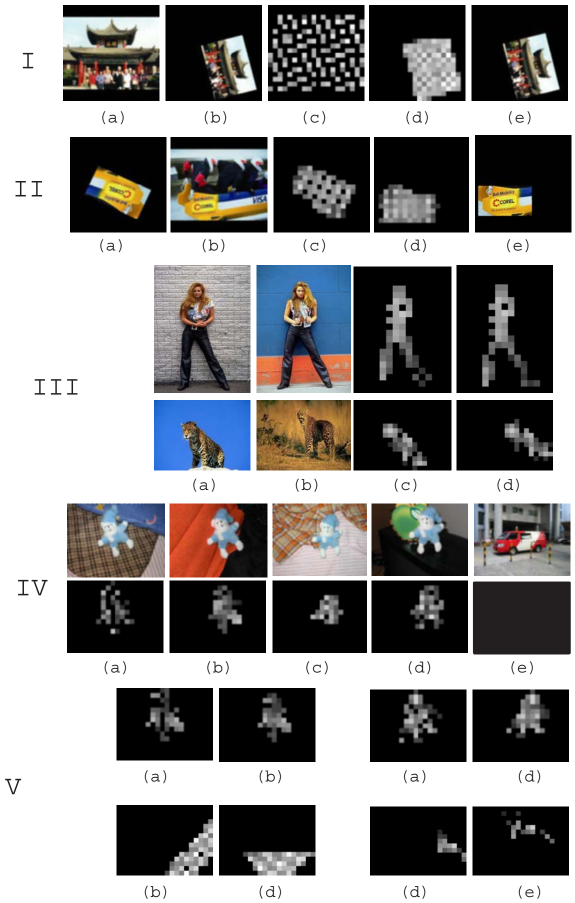

Figure 1. Experimental results.

Acknowledgements

The work described in this paper was fully supported by a grant from City University of Hong Kong (Project No. 7001470).

References

[1] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys (CSUR)*, 24(4):352–376, 1992.

[2] S. Cohen and L. Guibas. The earth mover's distance under transformation sets. In *Int. Conf. Comp. Vision*, 1999.

[3] P. Hong and T. S. Huang. Inexact spatial pattern mining. In *Workshop on Discrete Mathematics and Data Mining*, 2002.

[4] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Spatial color indexing and applications. *Intl. Journal of Computer Vision*, 35(3):245–268, 1999.

[5] J. A. McHugh. *Algorithmic Graph Theory*. Prentice Hall, 1990.

[6] M. Swain and D. Ballard. Color indexing. *Intl. Journal of Computer Vision*, 7(1):11–32, 1991.