A Constrained Alternating Optimization Framework for Feature Matching

Xiabi Liu[1] and Yunde Jia[1]

[1] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
{liuxiabi, jiayunde}@bit.edu.cn

Abstract. This paper proposes a constrained alternating optimization framework to tackle the feature matching problem with partial matching and multiple matching. We model the difference between pairing features as the result of a transformation followed an uncertainty distribution. Based on this modeling, transformation estimation and feature matching are performed alternately from initial matching: the transformation is updated according to the matching, and the matching is updated according to the transformation and the uncertainty distribution. A pruning operation is further presented to reduce the search space of initial matching. In the proposed framework, we develop a B-spline curve feature matching algorithm for hand-gesture based text input, and a line feature matching algorithm which is tested for three applications: model-based recognition, image registration, and stereo matching. The experimental results for two algorithms are reported.

Keywords: Feature matching; feature correspondence; constrained alternating optimization; curve feature matching; line feature matching

1 Introduction

Feature matching is a fundamental problem in computer vision and pattern recognition for a wide range of applications, including model-based recognition, image registration, stereo matching, object tracking, and motion analysis. It's well known that seeking feature correspondence and geometric transformation between two images are coupled problems, knowing either of them will make it much easier to solve the other. Therefore, there are three categories of solutions for feature matching:

- Search the result in only the transformation space. The search can be performed continuously such as the Hausdorff distance based method [1] and deformable template based methods [2-4], or discretely such as geometric alignment and hashing [5-6], random sample consensus (RANSAC) [7], and geometric constraint analysis [8].
- Solve only the feature correspondence. In this category, the similarities between features are often used as heuristic information, for example, in the eigenvector approach [9] and various searching strategies [10-13]. The relationship compatibilities between features are also important for techniques like relaxation labeling [14].

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8

- Estimate feature correspondence and geometric transformation alternately, such as iterative closest point (ICP) algorithm [15] and softassign [16].

Though much progress has been made, feature matching can still be computationally expensive and may also run into difficulties if the feature extraction process yields significant numbers of missing or spurious features or features are perturbed by deformation, occlusion and other factors.

In this paper, we propose a new alternating approach called constrained alternating optimization (CAO) to tackle the feature matching problem where partial matching is considered for robustness and multiple matching is considered for minimal requirement on the segmentation. We model the difference between pairing features as the result of two factors: a transformation between features and a remaining uncertainty which is of the distribution function. Based on this modeling, the transformation and the matching are optimized alternately from initial matching. A pruning operation is further presented to reduce the search space of initial matching. In the proposed CAO framework, transformation group and uncertainty distribution both are variable factors, so various feature matching algorithms can be realized for various applications. We develop a B-spline curve feature matching algorithm for hand-gesture based text input and a line feature matching algorithm which is tested for three applications: model-based recognition, image registration, and stereo matching. The experimental results for two algorithms show the effectiveness of the CAO framework.

The rest of this paper is organized as follows. Section 2 introduces the constrained alternating optimization (CAO) framework. Section 3 describes the B-spline curve feature matching algorithm and the line feature matching algorithm developed in the CAO framework, and reports the experimental results for two algorithms. We conclude in Section 4.

2 Constrained Alternating Optimization

2.1 Problem Statement

Let $\mathbf{I}_R$ and $\mathbf{I}_I$ denote two feature sets extracted from two 2-D images:

$$\mathbf{I}_R = \{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_M\}, \quad \mathbf{I}_I = \{\mathbf{b}_1, \mathbf{b}_2, \cdots \mathbf{b}_N\}, \tag{1}$$

where $\mathbf{a}_i$ and $\mathbf{b}_j$ are feature vectors, M and N are numbers of features in $\mathbf{I}_R$ and $\mathbf{I}_I$ respectively.

In various applications, a result of feature matching reflects either the largest common information between two images or the reproductions of the main information of one image in another image. So feature matching problem is to find out the largest subset or all locally largest subsets of $\mathbf{I}_I$, which is in one-to-one correspondence with one subset of $\mathbf{I}_R$. This description not only considers cases of partial matching for robustness, but also considers cases of multiple matching for minimal requirement on the segmentation. We therefore call $\mathbf{I}_R$ as the reference pattern, the corresponding image as the reference image and the feature in $\mathbf{I}_R$ as the reference feature. Oppositely, we have the input pattern, input image and input feature for $\mathbf{I}_I$.

Suppose a subset $\{\mathbf{a}_i\}$, $i=1,2,\cdots,K$ of $\mathbf{I}_R$ is in one-to-one correspondence with a subset $\{\mathbf{b}_i\}$, $i=1,2,\cdots,K$ of $\mathbf{I}_I$, and $\mathbf{a}_i$ is corresponding with $\mathbf{b}_i$. In most cases, $\mathbf{a}_i \neq \mathbf{b}_i$. We model the difference between them as the result of two factors: (1) a transformation T from $\{\mathbf{a}_1,\cdots,\mathbf{a}_K\}$ to $\{\mathbf{b}_1,\cdots,\mathbf{b}_K\}$; (2) an uncertainty between transformed $\mathbf{a}_i$ and $\mathbf{b}_i$, which is of a distribution function.

Let $s(T(\mathbf{a}_i),\mathbf{b}_i)$ be the similarity between transformed $\mathbf{a}_i$ and $\mathbf{b}_i$, which is computed based on the uncertainty distribution, ε be an uncertainty bound determining whether $\mathbf{a}_i$ and $\mathbf{b}_i$ can be paired, f be a minimum proportion bound for allowable partial matching, then $\{(\mathbf{a}_i,\mathbf{b}_i)\}, i=1,2,\cdots,K$ is a feasible matching if and only if

$$\begin{cases} s(T(\mathbf{a}_i),\mathbf{b}_i) \geq \varepsilon, i=1,2,\cdots,K \\ K/\min(M,N) \geq f \end{cases}. \tag{2}$$

According to the problem statement above, the matching objective should be to find out the optimal feasible matching or all locally optimal feasible matching. The corresponding criterion of matching quality is: between two matching, the better one has either larger pairing number or larger mean similarity if their pairing numbers are equal.

2.2 Alternating Search Paradigm

Eq. (2) indicates that a feasible matching is coupled with a feasible transformation. Given a transformation, the corresponding matching can be estimated, and vice versa. Consequently, the matching and the transformation can be optimized alternately: the transformation is updated according to the matching, and the matching is updated according to the transformation and the uncertainty distribution, they alternate until convergence. An initial matching is used to start this alternating optimization process, in which the pairing number is just big enough to solve transformation parameters. For example, the transformation with 6 parameters makes the pairing number of initial matching be 3 for point feature.

Let the pairing number in initial matching be k, then the search space of initial matching has size $O(M^k N^k)$. We call this space as 'initial matching space' afterwards. From each initial matching $\{(\mathbf{a}_i, \mathbf{b}_i)\}, i = 1, \cdots, k$ in the initial matching space, the following steps are performed to optimize the matching and the transformation alternately:

Step 1. Set the temporary reference pattern $\mathbf{I}'_R = \mathbf{I}_R$ and the current matching $\mathbf{P} = \{(\mathbf{a}_i, \mathbf{b}_i)\}, i = 1, 2, \cdots, k$, let $K = |\mathbf{P}|$, $S = \frac{1}{K}\sum s(\mathbf{a}_i, \mathbf{b}_i), (\mathbf{a}_i, \mathbf{b}_i) \in \mathbf{P}$

Step 2. Transformation update: Get the closed form solution or the least-square solution $\arg\min_T \sum [T(\mathbf{a}_i) - \mathbf{b}_i]^2$ of the transformation parameters according to the current matching.

Step 3. Transform each $\mathbf{a}_i$ in $\mathbf{I}'_R$ using the current transformation.

Step 4. Matching update:

1. Initialize a temporary set $\mathbf{Q} = \phi$.
2. For each feature $\mathbf{a}_i$ in $\mathbf{I}'_R$, seek all $\mathbf{b}_j$ in $\mathbf{I}_I$ satisfying $s_{ij} = s(\mathbf{a}_i, \mathbf{b}_j) \geq \varepsilon$ and add the correspondence $(\mathbf{a}_i, \mathbf{b}_j)$ into $\mathbf{Q}$: $\mathbf{Q} = \mathbf{Q} \cup \{(\mathbf{a}_i, \mathbf{b}_j)\}$
3. Sort $\mathbf{Q}$ in descending order of s_{ij}.
4. Let $\mathbf{P}' = \phi$. Process each $(\mathbf{a}_i, \mathbf{b}_j)$ in $\mathbf{Q}$ orderly: If $\mathbf{a}_i$ or $\mathbf{b}_j$ has been included in $\mathbf{P}'$, go to next one, else add $(\mathbf{a}_i, \mathbf{b}_j)$ into $\mathbf{P}'$: $\mathbf{P}' = \mathbf{P}' \cup \{(\mathbf{a}_i, \mathbf{b}_j)\}$.

Step 5. Let $S' = \frac{1}{|\mathbf{P}'|}\sum s(\mathbf{a}_i, \mathbf{b}_j), (\mathbf{a}_i, \mathbf{b}_j) \in \mathbf{P}'$.

If $|\mathbf{P}'| > K$ or $|\mathbf{P}'| = K \wedge S' > S$, let $\mathbf{P} = \mathbf{P}'$, $K = |\mathbf{P}'|$, $S = S'$, then go to step 2.

Step 6. If $|\mathbf{P}| / \min(M, N) \geq f$, record $\mathbf{P}$ as a feasible matching with corresponding transformation, check the set of feasible matching to keep merely the optimal one or all locally optimal ones.

We call the search paradigm above as constrained alternating optimization (CAO), which provides a flexible framework for defining different feature matching algorithms with different assignments to the transformation group and the uncertainty distribution. The computational complexity of the CAO framework is analyzed as follows. There are $O(M^k N^k)$ initial matching. From each initial matching, the optimization steps iterate at most $\min(M, N)$ times for feasible matching, and $f \min(M, N)$ times for non-feasible matching. In iteration, Step 4 is the most time-consuming with $O(MN)$ time. Therefore, the computational complexity of

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

the CAO framework is $O\left(f \min(M,N) M^{k+1} N^{k+1}\right)$ in the worst cases that none feasible matching can be found. After finding a feasible matching, the following pruning operation can be used to reduce the initial matching space, so that the computation can be accelerated.

2.3 Pruning Operation

A pruning operation is presented to accelerate the computation by reducing the initial matching space according to feasible matching. Let the current feasible matching with m pairs of features be $\mathbf{A} = \{(\mathbf{a}_i, \mathbf{b}_i)\}, i = 1,2,\cdots,m$, an arbitrary initial matching be $\{(\mathbf{a}'_j, \mathbf{b}'_j)\}, j = 1,2,\cdots,k$. Then all initial matching satisfying $\{\mathbf{b}'_1,\cdots,\mathbf{b}'_k\} \subseteq \{\mathbf{b}_1,\cdots,\mathbf{b}_m\}$ will be pruned. For non-multiple matching, all initial matchings satisfying $\{\mathbf{a}'_1,\cdots,\mathbf{a}'_k\} \subseteq \{\mathbf{a}_1,\cdots,\mathbf{a}_m\}$ will be pruned also.

Here, we illuminate that the pruning operation does not lower the veracity of the result. Let a latent feasible matching with n pairs of features be $\mathbf{B} = \{(\mathbf{a}''_i, \mathbf{b}''_i)\}, i = 1,\cdots,n$. Firstly, if $\mathbf{B}$ is a feasible matching in a local area different from that for $\mathbf{A}$, or $n > m$, then at least one reference feature $\mathbf{a}''_i$ and one input feature $\mathbf{b}''_i$ in $\mathbf{B}$ are not included in $\mathbf{A}$, therefore the initial matching including $\{(\mathbf{a}''_i, \mathbf{b}''_i)\}$ will not be pruned by the pruning operation, from which $\mathbf{B}$ can be found. Secondly, if $n = m$ and the mean similarity for $\mathbf{B}$ is larger than that for $\mathbf{A}$, $\mathbf{B}$ could be ignored after the pruning operation. However, the decrease of mean similarity is less than the uncertainty bound. To sum up, the pruning operation prunes neither really better feasible matching nor other local feasible matching, therefore which does not lower the veracity of the result.

Now we analyze the effect of the pruning operation on the computational complexity. Suppose the number of reference features being part of a feasible matching be K_M, and the number of multiple matching be t, then the possibility p of finding an initial matching leading to a feasible matching is $p = \frac{C_{K_M}^k}{C_M^k} \cdot \frac{t}{P_N^k}$, where C denotes combination, P permutation. Thus the time consumed for finding the first feasible matching is $O\left(f \min(M,N)(1-p) M^{k+1} N^{k+1} + \min(M,N) MN\right)$. Let K be the pairing number in this feasible matching. For multiple matching, the pruning operation will reduce the initial matching space to $O\left(M^k N^k - M^k K^k\right)$ and decrease the running time to

$$O\left(f \min(M,N)(1-p) M^{k+1} N^{k+1} + \min(M,N) MN + f \min(M,N) p \left(M^{K+1} N^{K+1} - M^{k+1} K^{k+1}\right)\right).$$

As for non-multiple matching, it is obvious to add $-K^{k+1} N^{k+1}$ in the wake of

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8

$-M^{k+1}K^{k+1}$. Since p is tiny for large N, the pruning operation cannot obviously decrease the complexity as a whole. But if a feasible matching is found rapidly by chance, the computation will be speeded up greatly.

3 Case Studies

The CAO framework includes two variable factors: transformation group and uncertainty distribution. After these two factors are determined, we get the corresponding feature matching algorithm for certain application.

3.1 B-spline curve feature matching

In our hand-gesture based text input method for wearable computers [17], a character is modeled as a set of uniform cubic B-splines, each of which represents a stroke. The key step of character classification is to match candidate strokes in the input character with B-splines in the character model. Let $\mathbf{I}_R = \{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_M\}$ be the set of B-splines, each of which is represented by ordered control points: $\mathbf{a}_i = \{C_{i1,} \cdots, C_{iI}\}$; $\mathbf{I}_I = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_N\}$ be the set of candidate strokes, each of which is a polygonal line represented by ordered vertices: $\mathbf{b}_j = \{p_{j1,} \cdots, p_{jJ}\}$.

In order to find out the correspondence between $\mathbf{I}_R$ and $\mathbf{I}_I$, two variable factors in the CAO framework are assumed as follows.

(1) Transformation group: affine transformation;

(2) Uncertainty distribution: each vertex in $\mathbf{b}_j$ is distributed normally about its projector (closet point) on transformed $\mathbf{a}_i$, i.e. $T(\mathbf{a}_i)$.

The similarity between $T(\mathbf{a}_i)$ and $\mathbf{b}_j$ is measured based on the assumption of the uncertainty distribution. Let p'_{jt} be the projector of vertex p_{jt} of $\mathbf{b}_j$ on $T(\mathbf{a}_i)$, n_j be the number of vertices in $\mathbf{b}_j$, δ be the ratio of the maximum curve length corresponding with $\{p'_{jt}\}, t = 1, \cdots, n_j$ to the length of the curve $T(\mathbf{a}_i)$, which is used to distinguish between a curve and its portions, then we have

$$s(T(\mathbf{a}_i), \mathbf{b}_j) = \frac{\delta}{n_j} \sum_{t=1}^{n_j} \exp\left(\frac{-\left\| p_{jt} - p'_{jt} \right\|^2}{2\sigma^2} \right). \tag{3}$$

The resultant B-spline curve feature matching algorithm was used in Graffiti 2

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

character recognition. In pen-based text input experiments, the recognition rate of 98.64% and 97.31% were achieved for English letters and numerals respectively. Fig. 1 shows some examples of character stroke matching. In Fig. 1, the image (a) shows the input character, the image (b) shows the corresponding character model, where the numbers denote the B-spline indices; the image (c) shows the matching result, not only by drawing each input stroke at a specific color and labeled with the index of the corresponding B-spline, but also by superimposing the transformed character model by the detected affine transformation on the input character.

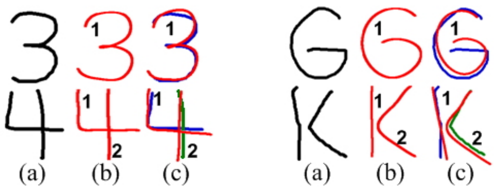


Fig. 1. Examples of character stroke matching for Graffiti 2 numerals and English letters from our collection: (a) the input character; (b) the character model; (c) the matching result.

3.2 Line feature matching

Here, we present a general line feature matching algorithm in the CAO framework. In our algorithm, a line feature is represented by the coordinates of end-points $(x_1, y_1), (x_2, y_2)$ in the image plane:

$$\mathbf{a}_{iE} = ((x_{i1}, y_{i1}), (x_{i2}, y_{i2})) = (X_{i1}, X_{i2}), \tag{4}$$

at the same time, represented by the coordinates of mid-point (x_c, y_c), length l, and orientation θ in 4-D space:

$$\mathbf{a}_{iF} = (x_{ic}, y_{ic}, l_i, \theta_i). \tag{5}$$

There is a bijection between $\mathbf{a}_{iE}$ and $\mathbf{a}_{iF}$.

For solving the line feature matching problem, two variable factors in the CAO framework are assumed as follows.

(1) Transformation group: the affine transformation in the image plane;

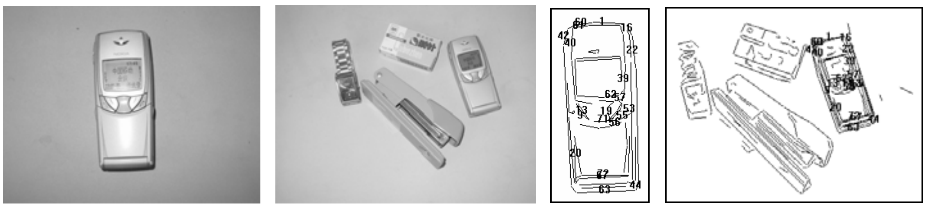
(2) Uncertainty distribution: the Gaussian distribution in 4-D space.

Accordingly, the similarity between two line features is measured as the reciprocal of the Mahalanobis distance between them. If the Mahalanobis distance is zero, a preset maximum similarity is used. Let $\mathbf{\Sigma}$ be the 4×4 covariance matrix of the Gaussian distribution, ms be the preset maximum similarity, then we have

$$d\left(T\left(\mathbf{a}_{iF}\right),\mathbf{b}_{iF}\right)=\left(T\left(\mathbf{a}_{iF}\right)-\mathbf{b}_{iF}\right)^{T}\mathbf{\Sigma}^{-1}\left(T\left(\mathbf{a}_{iF}\right)-\mathbf{b}_{iF}\right), \tag{6}$$

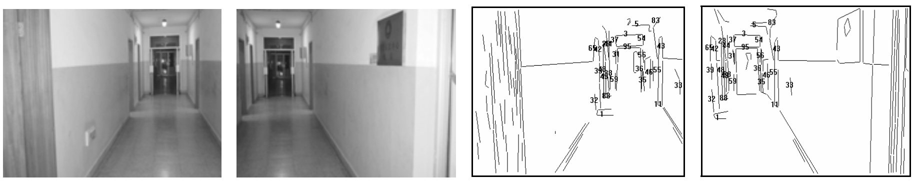
$$s\left(T\left(\mathbf{a}_{iF}\right),\mathbf{b}_{iF}\right)=\begin{cases}\dfrac{1}{d\left(T\left(\mathbf{a}_{iF}\right),\mathbf{b}_{iF}\right)}, & d\left(T\left(\mathbf{a}_{iF}\right),\mathbf{b}_{iF}\right)\neq 0\\ ms, & d\left(T\left(\mathbf{a}_{iF}\right),\mathbf{b}_{iF}\right)=0\end{cases}. \tag{7}$$

The resultant line feature matching algorithm is tested for three applications: model-based recognition, image registration and stereo matching. Fig. 2 shows a matching example for model-based recognition. Fig. 2a is an image of a mobile phone whose line model is shown in Fig. 2c. Fig. 2b shows a collection of objects including the same mobile phone and Fig. 2d shows the corresponding line pattern. The matching result is reflected by labeling same numbers for corresponding line features in Fig. 2c and Fig. 2d, also by superimposing the transformed model by the detected affine transformation on its counterpart in Fig. 2d. A matching example for image registration is shown in Fig. 3. Fig. 3a and 3b are the images of a same scene but from different views. Fig. 3c and 3d show the line patterns extracted from Fig. 3a and 3b respectively, also display the matching result by labeling same numbers for corresponding line features. Fig. 4 shows a matching example for stereo matching. Fig. 4a and 4b are two well-known ‘Valbonne Church’ images from University of Oxford. Fig. 4c and 4d show extracted line patterns from Fig. 4a and 4b respectively, in which the numbers indicate correspondences between line features.



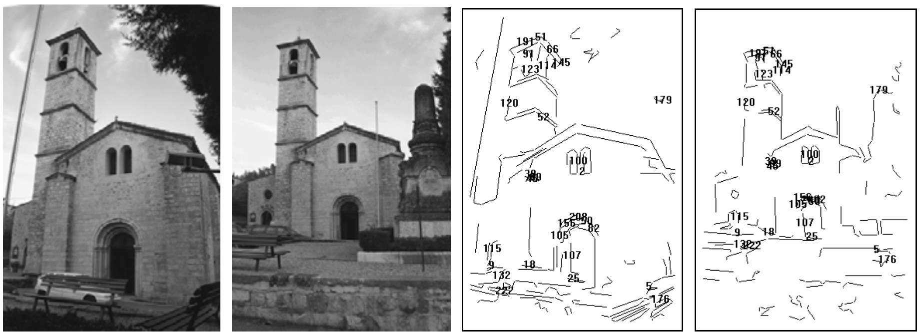
(a) (b) (c) (d)

Fig. 2. A model-based recognition test: (a) the reference image; (b) the input image; (c) the line model extracted from Fig. 2a and numbers indicating correspondences; (d) the line pattern extracted from Fig. 2b, numbers indicating correspondences, and the transformed model superimposed on its counterpart.



(a) (b) (c) (d)

Fig. 3. An image registration test: (a) the left part of a corridor scene; (b) the right part of the same scene; (c) the line pattern extracted from Fig. 3a and numbers indicating correspondences; (d) the line pattern extracted from Fig.3b and numbers indicating correspondences.



(a) (b) (c) (d)

Fig. 4. A stereo matching test: (a) the reference image; (b) the input image; (c) the line pattern extracted from Fig. 4a and numbers indicating correspondences; (d) the line pattern extracted from Fig. 4b and numbers indicating correspondences.

4 Conclusions

In this paper, a new approach called constrained alternating optimization (CAO) has been proposed to tackle feature matching problem with partial matching and multiple matching, which provides a flexible framework for defining different feature matching algorithms with different assignments to variable factors in it. We have implemented a B-spline curve feature matching algorithm for hand-gesture based text input and a general line feature matching algorithm which has been tested for three applications: model-based recognition, image registration and stereo matching. Experimental results for two algorithms show that the proposed CAO framework is effective and promising.

The previous works related to this paper include the alignment method [5], ICP algorithm [15] and geometric constraint analysis [8]. The CAO framework can be treated as the novel one fusing the advantages of the three afore-mentioned approaches. First, it searches initial matching in the similar way with the alignment method to guarantee global optimization in a polynomial time. Second, similar to the ICP algorithm, the matching and the transformation are updated alternately to get more precision and less computation. The difference is that the CAO framework explores all the initial solutions and uses different update methods. Third, the uncertainty of transformed features is taken into account like geometric constraint analysis to achieve high robustness, but as a distribution function instead of a geometric region.

Because of exhaustive search strategy of initial matching and small possibility of finding feasible matching, the CAO framework still has the disadvantage of unsatisfactory efficiency for complex problems, which can be partially solved in two ways. First, the CAO framework is inherently parallel. The alternating optimization process from each initial matching can be performed simultaneously by means of software or hardware. Second, using a more sophisticated strategy to search initial matching can increase the possibility of finding feasible matching; here randomization is a hopeful direction, which is our next work.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

Acknowledgments. This research was supported by the grant 2006CB303105 from 973 Program of China and the grant 2006Y1202 from Excellent Young Scholars Research Fund of Beijing Institute of Technology.

References

1. Huttenlocher, D. P., Klanderman, G. A., Ruchlidge, W. J.: Comparing Images Using the Hausdorff Distance. IEEE Trans. Pattern Analysis and Machine Intelligence 15 (9) (1993) 850-863
2. Jain, A. K., Zhong, Y., Lakshmanan, S.: Object Matching Using Deformable Templates. IEEE Trans. Pattern Analysis and Machine Intelligence 18 (3) (1996) 267-278
3. Wang, Z., Arie, J. B.: Model Based Segmentation and Detection of Affine Transformed Shapes in Cluttered Images. In: Proc. Int'l Conf. Image Processing, Vol. 3. (1998) 75-79
4. Xue, Z., Shen, D. E., Teoh, K.,. A Deformable Template Model Based on Fuzzy Alignment Algorithm. In: Proc. Int'l Conf. Image Processing, Vol. 1. (2000) 296-299
5. Huttenlocher, D. P., Ullman, S.: Recognizing Solid Objects by Alignment with an Image. Int'l J. Computer Vision 5 (2) (1990) 195-212
6. Lamdan, Y., Wolfson, H.: Geometric Hashing: A General and Efficient Model-Based Recognition Scheme. Proc. Int'l Conf. Computer Vision (1988) 238–249.
7. Fischler, M. A., Bolles R. C., Random Sample Consensus: A Paradigm for Model Fitting with Apphcatlons to Image Analysis and Automated Cartography. Communications of the ACM 24 (1981) 381-395
8. Cass, Todd A.: Robust Affine Structure Matching for 3D Object Recognition. IEEE Trans. Pattern Analysis and Machine Intelligence 20 (11) (1998) 1265-1274
9. Scott, G. L., Longuet-Higgins, H. C.: An algorithm for associating the features of two images. Proc. R. SOC. Lond. B (1991) 21-26
10. Beveridge, J. R., Riseman, E. M.: How Easy is Matching 2D Line Models Using Local Search. IEEE Trans. Pattern Analysis and Machine Intelligence 19(6) (1997) 564-579
11. Yip, R. K. K., Ho, W. P.: A Multi-Level Dynamic Programming Method for Stereo Line Matching. Pattern Recognition Letters 19 (1998) 839-855.
12. Aider, O. A., Hoppenot, P., Colle, E.: A Model to Image Straight Line Matching Method for Vision-Based Indoor Mobile Robot Self-Location. In: Proc. Int'l Conf. Intelligent Robots and Systems. (2002) 460-465
13. Suesse, H., Ortman, W.: Robust Matching of Affinely Transformed Objects. In: Proc. Int'l Conf. Image Processing. (2003) 375-378
14. Ranade, S., Posenfeld, A.: Point Pattern Matching by Relaxation. Pattern Recognition 12 (1980) 269-275.
15. Besl, P. J., McKay, N. D.: A Method for Registration of 3-D Shapes. IEEE Trans. Pattern Analysis and Machine Intelligence 14 (2) (1992) 239-256
16. Rangarajan, A., Chui, H., Bookstein, F.: The Softassign Procrustes Matching Algorithm. In: Information Processing in Medical Imaging. Springer, Berlin (1997) 29–42
17. Liu, Y., Liu X., Jia Y.. Hand-Gesture Based Text Input for Wearable Computers. In: Proc. Int'l Conf. Computer Vision Systems (2006) 8-14

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de