MCMC-BASED PEAK TEMPLATE MATCHING FOR GCXGC

Mingtian Ni, Qingping Tao, and Stephen E. Reichenbach

Department of Computer Science and Engineering
University of Nebraska - Lincoln
Lincoln, NE 68588-0115, USA
EMail: {mni | qtao | reich} @cse.unl.edu

ABSTRACT

Comprehensive two-dimensional gas chromatography (GCxGC) is a new technology for chemical separation. Peak template matching is a technique for automatic chemical identification in GCxGC analysis. Peak template matching can be formulated as a Largest Common Point Set problem (LCP). Minimizing Hausdorff distances is one of the many techniques proposed for solving the LCP problem. This paper proposes two novel strategies to search the transformation space based on Markov Chain Monte Carlo (MCMC) methods. Experiments on seven real data sets indicate that the transformations found by the new algorithms are effective and searching with two Markov chains is much faster than searching with one Markov chain.

1. INTRODUCTION

Comprehensive two-dimensional gas chromatography (GCxGC) is a new technology for chemical separation that provides an order-of-magnitude increase in separation capacity over traditional GC [1, 2]. GCxGC separates chemical species with two capillary columns interfaced by two-stage thermal desorption. Given a chemical sample, the GCxGC output can be visualized as a 2D image, with pixels arranged so that the X-axis (left-to-right) and the Y-axis (bottom-to-top) are the elapsed times for the first and second column separation respectively. Each pixel value indicates the rate at which molecules are detected at a specific time. Each chemical substance in the chemical sample produces a small peak or cluster of pixels in the image with values that are larger than the background values.

The goal of GCxGC analysis is to separate, quantify, and identify specific chemicals in a sample. The major image analysis tasks include segmenting the image into individual peaks and background, measuring peaks, and identifying the chemical for each peak of interest. GCxGC images easily contain several thousand chemical peaks. Manually annotating the peaks is tedious and time-consuming. Peak template matching offers a way to speed the annotation process.

A peak template P is a set of peaks whose corresponding chemicals are known. A target peak set Q is a set of peaks whose corresponding chemicals are to be determined. Given P and Q, the objective of template matching is to establish as many correspondences as possible from the peaks in P to the peaks in Q. After the correspondences are established, the information carried by source peaks is copied to target peaks and the chemical identification is achieved.

A peak has many features such as peak location, area, volume, shape, etc. In this paper, only peak location (the coordinates of the pixel with the largest value within the peak) is used for matching. As such, the peak template and the target peak set can be abstractly represented by two point sets in two-dimensional space.

Let $P = \{p_i(x_i, y_i)\}_{i=1}^{m}$ be the point template and $Q = \{q_i(u_i, v_i)\}_{i=1}^{n}$ be the target point set. The peak template matching problem can be posed as the Largest Common Point Set (LCP) problem [3, 4].

> *Given point template P, target point set Q, partial directed Hausdorff distance $\vec{d}_H^k$, transformation space T, and the desired number of points in P to be matched k, compute:*

$$\min_{t \in T} \left\{ \vec{d}_H^k(t(P), Q) \right\}.$$

Generally, P may not be congruent to Q or any subset of Q. The above formulation is merely intended to match a subset of P to a subset of Q and minimize the distance. The solution to the LCP problem is a transformation. From the transformation, the correspondence from P to Q is then computed.

The partial directed Hausdorff from P to Q is defined as [5]:

$$\vec{d}_H^k(P, Q) = \max_{p \in P}{}^k \min_{q \in Q} \|p - q\|$$

where $\|p - q\|$ is the Euclidean distance between point p and point q, and $\max^k$ means taking the k^{th} largest distance. The partial directed Hausdorff distance is a good choice here because it has the effect of matching part of P to part of Q. In addition, it is not required to specify which part of P is to be matched. When $k = |P|$, the partial directed Hausdorff distance becomes the directed Hausdorff distance which is denoted by $\vec{d}_H(P, Q)$. The partial directed Hausdorff distance can be computed in time $O((m + n)\log(m + n))$ [5].

Minimizing Hausdorff distances is one of the many techniques proposed for solving the LCP problem. This technique uses Hausdorff distance (or its variations) as the similarity measure and searches the transformation space for a transformation that minimizes the Hausdorff distance. The search strategies proposed in the literature include exact computation [6, 7], rasterization of the upper envelope of Voronoi surfaces [5], transformation space subdivision [7], multi-instance learning [8], etc.

In this paper, we propose using Markov chain Monte Carlo (MCMC) methods to search the transformation space. MCMC methods are general tools for simulating complex distributions by ergodic Markov chains [9]. When used for solving optimization

0-7803-7997-7/03/$17.00 © 2003 IEEE

problems, MCMC methods map the objective functions to some probability distributions and search the parametric space for a point that optimizes the objective function [9].

2. THE MCMC-BASED SEARCHING ALGORITHMS

In the LCP problem, the goal is to minimize the objective function $\vec{d}_H^k(t(P), Q)$. We define a distribution π on a finite transformation space T as:

$$\pi(t) = \frac{\exp(-\vec{d}_H^k(t(P), Q))}{Z}$$

where $t \in T$ and Z is a normalization factor such that $\int_T \pi(t)dt = 1.0$. Because $\pi(t)$ and $\vec{d}_H^k(t(P), Q)$ are inversely related, if some t maximizes $\pi(t)$, it minimizes $\vec{d}_H^k(t(P), Q)$. So the solution to the LCP problem is $argmax\, \pi(t)$.

2.1. Searching with one Markov chain

In this paper, the Metropolis-Hastings algorithm [10] is used to search the transformation space T by sampling. The algorithm samples T according to π by performing random walk on a Markov chain whose state space is T. The walk starts with some initial transformation (state) and makes each transition as follows: a new transformation t' is proposed from an uncorrelated Gaussian distribution $N(t, \Sigma_t)$, where the mean value t is the current transformation and Σ_t is a diagonal covariance matrix. The new transformation t' will be accepted with the Metropolis-Hastings acceptance probability:

$$A_t(t') = \min\left\{1, \frac{\pi(t')G_{t'}(t)}{\pi(t)G_t(t')}\right\}$$

where $G_{t'}(t)$ and $G_t(t')$ are the pdf's of $N(t', \Sigma_{t'})$ and $N(t, \Sigma_t)$. If $\vec{d}_H^k(t'(P), Q) < \vec{d}_H^k(t(P), Q)$, t' is always accepted ($A_t(t') = 1.0$).

In the experiments presented in Section 3, the same Σ is used for every state. In such a case, $A_t(t')$ is simplified as:

$$A_t(t') = \min\left\{1, \exp(\vec{d}_H^k(t(P), Q) - \vec{d}_H^k(t'(P), Q))\right\}.$$

2.2. Searching with two Markov chains

One difficulty with the above searching algorithm is how to set Σ_t. If standard deviations in Σ_t are too large, the proposed new transformation stays away from the current transformation with high probability. As a consequence, the Markov chain tends to make big jumps in the transformation space, overshooting the global optimal transformation. On the other hand, if standard deviations in Σ_t are too small, the proposed new transformation may oscillate around a local optimal transformation [9].

The selection of Σ_t becomes easier when using two Markov chains instead of one. Then, the searching algorithm runs two Metropolis-Hastings processes, $\Re_g$ and $\Re_l$, simultaneously. Processes $\Re_g$ and $\Re_l$ use two different covariance matrices, Σ_g and Σ_l, with larger standard deviations for Σ_g and smaller standard deviations for Σ_l. The start transformation of $\Re_l$ is set to the best transformation that $\Re_g$ has found so far after each fixed number of steps. The algorithm can be roughly thought of as a two-level multi-resolution searching, where process $\Re_g$ looks through T quickly for a good start point at the course resolution and process $\Re_l$ starts from that point and searches its neighborhood at the fine resolution.

3. EXPERIMENTAL RESULTS

3.1. Data sets

The seven data sets, summarized in Table 1, were acquired at three different laboratories on three different GCxGC instruments. Each data set has several images generated from the same chemical sample or from related samples with the same chemicals. Selected peaks in each data set were annotated using $GCImage^{TM}$ software [11]. The selected peaks form a peak set for each image. Peak correspondences across images in each data set were established for testing the effectiveness of the algorithm. Also, for computational stability, peak locations are normalized. The normalization is done for each data set separately. Let (μ_x, μ_y) and (σ_x, σ_y) be the mean and standard deviation of the peak locations in some data set. Then, the peak location (x, y) in that data set is normalized as:

$$\begin{cases} x' = \frac{x-\mu_x}{(\sigma_x+\sigma_y)/2} \\ y' = \frac{y-\mu_y}{(\sigma_x+\sigma_y)/2} \end{cases}$$

where (x', y') is the new peak location.

Table 1. Data sets

Data set	Number of images	Number of selected peaks
D2287 sdalk	3	15
D2287 sdgas	3	580
Doixin	3	26
GCC2002	12	14
Linearity	5	18
NYSDH	5	10
PCB	4	17

3.2. Estimating T

The transformation model used in this paper is global constrained affine transformation. The global constrained affine transformation from $p(x_p, y_p)$ to $q(u_q, v_q)$ is:

$$\begin{bmatrix} u_q \\ v_q \end{bmatrix} = \begin{bmatrix} s_x & h_x(=0.0) \\ h_y & s_y \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

with h_x set to 0.0 because the x coordinates (first column separation time) are independent of the y coordinates (the second column separation time) in GCxGC images. Experimental results (not reported here) indicate that the above transformations work well for largely removing image-to-image distortions.

Given the global constrained affine transformation model, the complexity of finding a matching primarily depends on the ranges that the transformation parameters vary. If all five parameters vary freely, searching for a solution is expensive. However, experiments show that the least-squares optimal transformations are clustered in the transformation space. Consequently, a search over a small region typically will find a good matching.

Given a training data set, optimal transformations are computed from each peak set to every other peak set based on least-squares estimation. An uncorrelated Gaussian model $N(\mu, \Sigma)$ is then fit to the distribution of the resultant transformations using common techniques such as those in [12]. T is set to be a rectangular region A in the transformation space, where $\int_A N(\mu, \Sigma)dt \geq \mathit{certain\ probability\ threshold}$ and t is a variable defined in transformation space. Figure 1 and 2 illustrate the spatial distributions of the scale parameters and translation parameters of the transformations generated from the seven data sets.

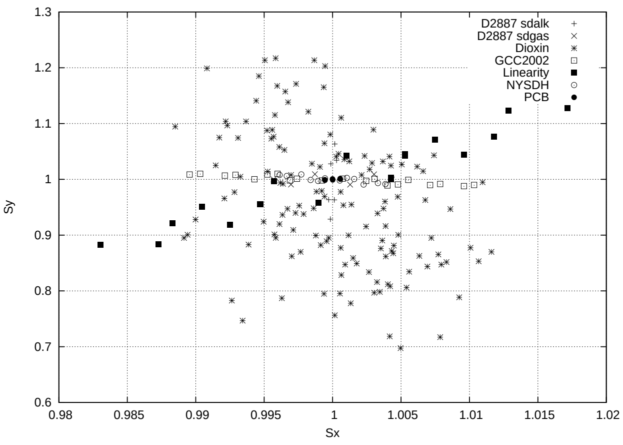

Fig. 1. Scale parameter distribution.

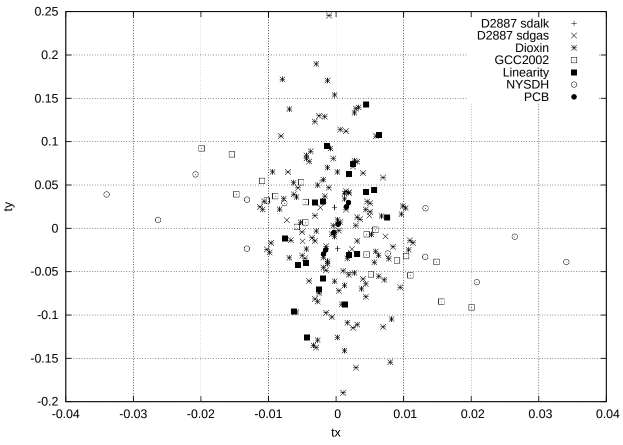

Fig. 2. Translation parameter distribution.

3.3. Selecting the standard deviations

In the experiments described in Section 3.4 and 3.5, when one Markov chain is used, the standard deviations of Σ is set to those of the covariance matrix of the Gaussian distribution that models the transformation space (See Section 3.2). When two Markov chains are used, Σ_g is set to be the Σ. For Σ_l, the standard deviations are selected based on the desired matching accuracy. For example, if the desired matching accuracy is E, we set Σ_l such that $\int_{E(0)} N(0, \Sigma_t)dt \geq \mathit{certain\ probability\ threshold}$. Here, the accuracy is defined as the neighborhood $E(q)$ around a target point q. Template point p is said to be matched to target point q if p lies in $E(q)$. It is clear that the smaller the E, the more accurate the matching.

3.4. Effectiveness of transformations found by the MCMC-based searching algorithms

For point template P and target point set Q, assume that the MCMC-based searching algorithms return transformation t_f, and based on t_f the point correspondences between P and Q are then computed. To evaluate the effectiveness of t_f, $\vec{d}_H(t_f(P), Q)$ is computed and compared to $\vec{d}_H(t_o(P), Q)$, where t_o is the least-squares optimal transformation. The experimental results on the seven data sets are reported in Table 2. Note that when one data set is used for testing, all other six data sets are used as training data for estimating the search range and the standard deviations. Also, within the testing data set, one peak set is selected to be the template, and all others are target sets. Table 2 only reports the average distances for each data set. The average number of steps used to find the transformations are described in Section 3.5. The results show that the transformations found in limited steps by the algorithms work well compared to the least-squares optimal transformations. For four out of the seven data sets, the algorithms found better transformations in terms of $\vec{d}_H$, which is the objective function. For the other data sets, the results of the algorithms are comparable to the least-squares optimal transformations.

Table 2. Effectiveness of the transformations found by the MCMC-based searching algorithm.

Data set	$\vec{d}_H(t_o(P), Q)$	$\vec{d}_H(t_f(P), Q)$
D2287 sdalk	0.0382	**0.0369**
D2287 sdgas	0.0436	**0.0385**
Doixin	**0.0415**	0.0430
GCC2002	0.0902	**0.0728**
Linearity	0.0711	**0.0613**
NYSDH	**0.0404**	0.0422
PCB	**0.0492**	0.0498

3.5. Computational efficiency

The experiments in this section evaluate and compare the computational efficiency of the two MCMC-based searching algorithms. Because the behavior of MCMC methods depends on random number generation and thus varies from one run to another, the experiments run the two algorithms 20 times under the same configuration and report only the average results.

The average numbers of steps that the two algorithms take to find t_f (see Section 3.4) are reported in Figure 3 and 4. For the results in Figure 3, both algorithms start with identity transformation. For the results in Figure 4, both algorithms start with some identical randomly generated transformation in T. The results clearly indicate that searching with two Markov chains is statistically much more efficient than searching with one Markov chain.

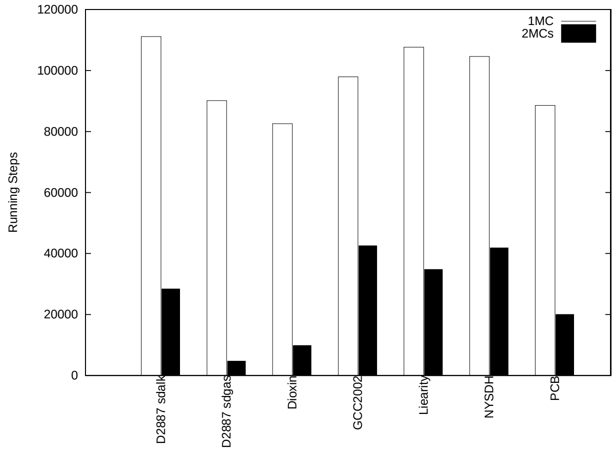

Fig. 3. Comparison of the two algorithms with the initial state being identity transformation.

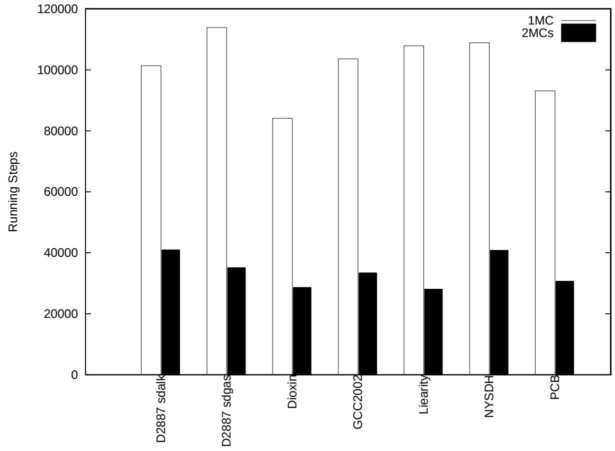

Fig. 4. Comparison of the two algorithms with the initial state being a random transformation in T.

4. CONCLUSION

Peak template matching is an automatic chemical identification method for GCxGC. This paper proposes two novel MCMC-based searching algorithms for solving the problem. Experiments indicate that the algorithms work effectively. On average, the algorithms find transformations with smaller partial directed Hausdorff distances than the least-squares optimal transformations. Experiments also show that searching with two Markov chains is statistically much faster than searching with a one Markov chain.

Our future work includes:

- trying different formulations of the distribution $\pi(t)$,
- using more data sets to test the searching efficiency of the searching algorithms, and
- adjusting standard deviations based on some local properties of the transformation space to accelerate the searching.

5. ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grant No. 0231746 and CCR-0092761. The authors also wish to thank Stephen D. Scott for useful discussion.

6. REFERENCES

[1] W. Bertsch, “Two-dimensional gas chromatography, concepts, instrumentation, and applications — Part 2: Comprehensive two-dimensional gas chromatography,” *Journal of High Resolution Chromatography*, vol. 23, no. 3, pp. 167–181, 2000.

[2] Edward B. Ledford, Jr. and Chris A. Billesbach, “Jet-cooled thermal modulator for comprehensive multidimensional gas chromatography,” *Journal of High Resolution Chromatography*, vol. 23, no. 3, pp. 202–204, 2000.

[3] T. Akutsu, H. Tamaki, and T. Tokuyama, “Distribution of distances and triangles in a point set and algorithms for computing the largest common point sets,” in *Symposium on Computational Geometry*. ACM, 1997, pp. 314–323.

[4] S. Venkatasubramanian, *Geometric Shape Matching and Drug Design*, Ph.D. thesis, Stanford University, 1999.

[5] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge, “Comparing images using the Hausdorff distance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, 1993.

[6] D.P. Huttenlocher and K. Kedem, “Computing the minimum hausdorff distance for point sets under translation,” in *Proc. the sixth annual symposium on Computational geometry*, 1990, pp. 340 – 349.

[7] W.J. Rucklidge, “Efficient visual recognition using the Hausdorff distance,” *Lecture Notes in Computer Science*, vol. 1173, 1996.

[8] S.D. Scott, J. Zhang, and J. Brown, “On generalized multiple-instance learning,” Tech. Rep. UNL-CSE-2003-5, University of Nebraska, 2003.

[9] C.P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer-Verlag, New York, 1999.

[10] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, “Equations of state calculations by fast computing machines,” *Journal of Chemical Physics*, vol. 21, pp. 1087 – 1091, 1953.

[11] S.E. Reichenbach, M. Ni, V. Kottapalli, A. Visvanathan, and J.E.B. Ledford, “Information technologies for comprehensive two-dimensional gas chromatography,” in *International Symposium on Capillary Chromatography*, 2003, p. CDROM:to appear.

[12] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 1999.