Similar-Video Retrieval via Learned Exemplars and Time-Warped Alignment

Teruki Horie, Masafumi Moriwaki*, Ryota Yokote**, Shota Ninomiya,
Akihiro Shikano, and Yasuo Matsuyama***

Waseda University, Department of Computer Science and Engineering,
Tokyo, 169-8555, Japan
{t.horie,masa.m,rrryokote,nino225,a.shikano,yasuo}@wiz.cs.waseda.ac.jp
http://www.wiz.cs.waseda.ac.jp

Abstract. New learning algorithms and systems for retrieving similar videos are presented. Each query is a video itself. For each video, a set of exemplars is machine-learned by new algorithms. Two methods were tried. The first and main one is the time-bound affinity propagation. The second is the harmonic competition which approximates the first. In the similar-video retrieval, the number of exemplar frames is variable according to the length and contents of videos. Therefore, each exemplar possesses responsible frames. By considering this property, we give a novel similarity measure which contains the Levenshtein distance (L-distance) as its special case. This new measure, the M-distance, is applicable to both of global and local alignments for exemplars. Experimental results in view of precision-recall curves show creditable scores in the region of interest.

Keywords: Similar-video retrieval, exemplar, time-bound affinity propagation, M-distance, numerical label.

1 Introduction

Machine learning or computational intelligence has discovered its own new values in the age of big data. Today, various types of unstructured data are continually accumulated. A typical case can be found in videos. Advent of smart phones made users produce and upload their own videos to the Web easily. However, most of them are structured poorly. This hinders users from utilizing rich hidden resources. The tendency is worse than the era of the static image retrieval [1] [2]. Reflecting this situation, efforts have been made on content-based approaches to the video retrieval as is surveyed in [3]. In this paper, we give new machine learning methods for automatic exemplar extraction and novel similarity measures, as well as their applications to similar-video retrieval.

* Currently with NS Solutions Corporation, Tokyo, Japan.

** Currently with Donuts, Co., Tokyo, Japan.

*** This work was supported by the Grant-in-Aid for Scientific Research 26330286, and Waseda University Special Research Projects 2013A-6248, 2013B-094, 2014K-6129.

C.K. Loo et al. (Eds.): ICONIP 2014, Part III, LNCS 8836, pp. 85–94, 2014.

The organization of this paper is as follows. In Section 2, we give a novel learning algorithm called time-bound affinity propagation (TBAP). This has the frame-order awareness which cannot be realized by the original affinity propagation (AP) of [4]. This is an unsupervised learning algorithm which finds representative frames in a video. In Section 3, we give a class of new similarity measures with time-warping which includes the Levenshtein distance (L-distance) [5], the Needleman-Wunsch algorithm [6] and the Smith-Waterman algorithm [7] as its special cases. Such a new measure, the M-distance, is an important part of this paper's similar-video retrieval. In Section 4, a test set of videos is designed. We will prepare a data set which depends on subject's sensibility as less as possible. In Section 5, we give a class of alternative learning methods to the time-bound affinity propagation. That is based on the harmonic competition [8]. Section 6 gives concluding remarks.

2 Problem Description

2.1 Exemplars Reflecting Time Information

Let $\{\boldsymbol{x}_t\}_{t=1}^{n}$ be a given time series. $\boldsymbol{x}_t$ can be any vector. In this paper, this is a feature vector series in terms of the color structure descriptor (CSD) of MPEG-7. The CSD is a patch-based histogram.

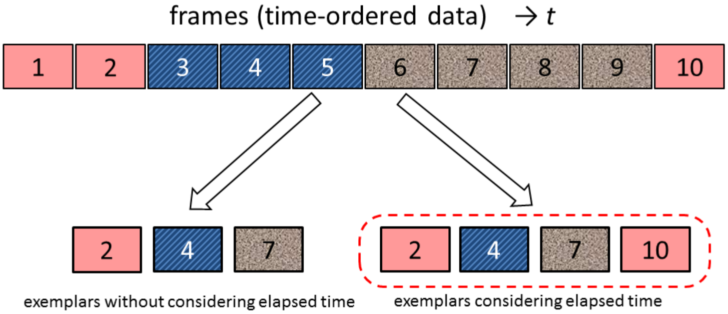

frames (time-ordered data) → t
1
2
3
4
5
6
7
8
9
10
2
4
7
2
4
7
10
exemplars without considering elapsed time
exemplars considering elapsed time

Fig. 1. Exemplars reflecting time information

Figure 1 illustrates a time series of frames $\{\boldsymbol{x}_t\}_{t=1}^{10}$. Let this conceptual video have three similar frames of {1, 2, 10}, {3, 4, 5} and {6, 7, 8, 9}. If time information or frame ordering were not considered, a learning system would choose only frames {2, 4, 7} as exemplars (Fig. 1 bottom left). But, this is not appropriate as a label for the video retrieval. Rather, we want to have a learning algorithm to find {2, 4, 7, 10} as exemplars (Fig. 1 bottom right). In this case, the exemplar set also gives responsible frames or dominant neighbors. For instance, we have {(1, **2**, 0), (1, **4**, 1), (1, **7**, 2), (0, **10**, 0)}. In the next section, we will give a new learning algorithm to obtain an order-aware exemplar set like Fig. 1 bottom right.

2.2 Time-Bound Affinity Propagation

Before going to the algorithm for the exemplar frame learning, it is necessary to have a right understanding about the following items.

(a) The plain affinity propagation algorithm can produce only the case of Fig. 1 bottom left. We need to obtain an algorithm which is aware of the frame ordering.

(b) Since we want to obtain exemplars, i.e., existing frames, the affinity propagation was set as a basic tool. However, the harmonic competition [8] which is a generalization of the vector quantization will also be applicable to the order-aware exemplar extraction.

In this section, we focus on item (a). Item (b) will be discussed in Section 5.

Our method to find order-aware exemplars is as follows.

[Intra-Video Processing: Time-Bound Affinity Propagation (TBAP)]

Step 1: A time series of feature vectors of video frames $\{\boldsymbol{x}_t\}_{t=1}^n$ is given. Each vector is a normalized CSD histogram whose summation is unity. A similarity measure $s(\boldsymbol{x}_i, \boldsymbol{x}_j) \overset{\text{def}}{=} s(i,j)$ is given. Here, $s(k,i) > s(k,j)$ holds if and only if $\boldsymbol{x}_i$ is more similar to $\boldsymbol{x}_k$ than $\boldsymbol{x}_j$. In our experiments, we will use

$$s(i,j) = \bar{D} - \|\boldsymbol{x}_i - \boldsymbol{y}_j\|. \tag{1}$$

Here, $\bar{D}$ is a constant which can be chosen by users.[1] Other design parameters appearing in subsequent steps are set here. A convergence criterion is also specified here.

Step 2: Prepare a matrix $\boldsymbol{A} = [a_{ij}]$ whose initial value is a zero matrix $\boldsymbol{O}$. This is called the availability.

Step 3: Pick up $\boldsymbol{x}_i$, $\boldsymbol{x}_j$, and $\boldsymbol{x}_k$ by considering their temporal ordering in a window.

Window length: Set a sliding window of length $2w - 1$.

Windowing (Order awareness property 1):
The following computation of the responsibility matrix and the availability matrix is computed for

$$\begin{aligned} i &: \quad 1 \leq i \leq n, \\ j &: \quad 1 \leq j \leq n \;\; \text{constrained by} \;\; i - w < j < i + w. \end{aligned} \tag{2}$$

Responsibility matrix update (Order awareness property 2):
$\boldsymbol{R} = [r_{ij}]$, which is symmetric, is updated by

$$\rho_{ij} := s(i,j) - \max_{k:\; k \neq j} \{a_{ik} + s(i,k)\}, \tag{3}$$

$$r_{ij} := (1 - \lambda)\rho_{ij} + \lambda r_{ij}. \tag{4}$$

[1] The choice of $\bar{D}$ does not affect the result of this TBAP which is an intra-video processing. Therefore, it can be set zero here. However, for the inter-video comparison based on the similarity, $\bar{D}$ becomes an important design parameter for users. Its default value will be discussed in Section 3.1.

Availability matrix update (Order awareness property 3): The availability matrix $\boldsymbol{A}$ is updated by

$$\alpha_{ii} := \sum_{k \neq i} \max\{r_{ki}, 0\}, \tag{5}$$

$$\alpha_{ij} := \min\{0,\ r_{jj} + \sum_{k:\ k \neq i,\, k \neq j} \max\{r_{kj}, 0\}\},\ (i \neq j), \tag{6}$$

$$a_{ij} := (1 - \lambda)\alpha_{ij} + \lambda a_{ij} \ \ \text{(including } i = j\text{)}. \tag{7}$$

The design parameter $\lambda \in (0, 1)$ is a dumping factor.

Step 4 (Order awareness property 4): If a convergence criterion is not satisfied for $a_{ij} + r_{ij}$, then Step 3 is repeated. If the convergence is met,

$$\operatorname*{arg\,max}_{j:\ i-w<j<i+w \ \text{for}\ 1 \leq i \leq n} \{a_{ij} + r_{ij}\} \tag{8}$$

is adopted as an exemplar index. The final exemplar set is determined by collecting such indices.

Theoretical Consideration: The affinity propagation (AP) [4] was theoretically derived from the maximization of a similarity measure with a constraint of the node labeling in view of message passing. The labeling stands for the identification of exemplars. In this paper, however, each node has a sequence index. That is, the node is a frame of a video. Therefore, we added a constraint on the message passing so that the set of nodes is a time series. This is our TBAP algorithm.

3 Distance Measure and Similarity Comparison

3.1 Data Normalization and Distance Measure

The similarity measure $s(i, j)$ can be any as long as it leads to the convergence of the algorithm. We found that the form of equation (1) gives the convergence of the algorithm if λ is chosen appropriately. The bias $\bar{D}$ in Equation (1) has effects on inter-video comparison appearing in later sections.

Since each vector $\boldsymbol{x}_t$ is normalized to have only nonnegative elements whose summation makes unity, possible choices of $\bar{D}$ are as follows.

(a) The average of all possible data distances: This is acceptable only if the data size is small.

(b) A fixed choice of $\sqrt{2}$, $\sqrt{(d-2)/(2d)}$, or $\sqrt{1-(1/d)}$: Here, d is the dimension of $\boldsymbol{x}_t$ which resides in a simplex. Note that $\sqrt{2}$ is the edge length of this simplex. $\sqrt{(d-2)/(2d)} \approx 1/\sqrt{2}$ is the radius of the interior sphere. $\sqrt{1-(1/d)} \approx 1$ is the radius of the exterior sphere. In experiments, we will use the exterior radius with $d = 768$ which is our size of CSD bins by the HSV expression of the color space (Hue-Saturation-Value).

3.2 Similarity Comparison 1: M-distance for Global Alignment

Here, we give a method to compare two different videos with different exemplar sets. Although we use the terminology of videos, the method is applicable to any time series with exemplars. Our method generalizes the Levenshitein distance (L-distance) [5] of discrete text processing, and the Needleman-Wunsch algorithm (NW algorithm)[6] for the global alignment in bioinformatics. After the name of the L-distance, the similarity comparison below will be called M-distance for the global alignment.[2]

[Global Alignment and Retrieval]

Step 1: For the video $\boldsymbol{v}_A$, sets of exemplars $\{\boldsymbol{e}_i^A\}$ and accompanied dominance lengths by the relevance $\{E_i^A\}$, $(i = 1, 2, \cdots)$ are given. For the video $\boldsymbol{v}_B$, similar sets are given. The similarity measure (1) is chosen here.

Step 2: Fill a global alignment table, and then backtrack a path by the following dynamic programming procedure.

(2-1) A gap penalty g is chosen as a design parameter.

(2-2) Make a table.
Fill the $\{i = 0\}$-th row by $(0, -gE_1^B, -g\sum_{j=1}^{2} E_j^B, -g\sum_{j=1}^{3} E_j^B, \cdots)$.
Fill the $\{j = 0\}$-th column by $(0, -gE_1^A, -g\sum_{i=1}^{2} E_i^A, -g\sum_{i=1}^{3} E_i^A, \cdots)$.

(2-3) Starting from the position of $(i, j) = (1, 1)$, fill elements by

$$f(i, j) = \max \{f(i-1, j) - gE_i^A, f(i-1, j-1) + r(i, j)s(i, j), f(i, j-1) - gE_j^B\}. \quad (9)$$

To a cell which gave the maximum, an arrow is directed as a pointer. Here, $r(i, j)$ is a weight which reflects the exemplar dominance. We will use $r(i, j) = (E_i^A + E_j^B)/2$ in experiments. If we backtrack from the bottom right element of the value f_{last}, the path gives a global alignment.

Step 3: The similarity between $\boldsymbol{v}^A$ and $\boldsymbol{v}^B$ is computed by

$$u(A, B) = h(f_{\text{last}})/w(\textstyle\sum_i E_i^A, \sum_j E_j^B). \quad (10)$$

Here, $h(x)$ is a monotone increasing function. w is an averaging function. The simplest one is an arithmetic mean.

3.3 Similarity Comparison 2: M-distance for Local Alignment

If video lengths are considerably different, the global alignment might deviate from human sensibility. In such a case, we use a local alignment which can compare the most similar parts. The following algorithm generalizes the Smith-Waterman algorithm for the local alignment [7] in bioinformatics. Only the difference from the global alignment is described.

[Local Alignment and Retrieval]

[2] The M-distances of Section 3.2 and Section 3.3 are due to the last and second authors, Matsuyama and Moriwaki.

Step 1: This step is the same as the global alignment.

Step 2: Fill a local alignment table and backtrack a path by the following dynamic programming procedure.

(2-1) A gap penalty g is chosen as a design parameter.

(2-2) Make a table. Fill elements in the $\{i = 0\}$-th row and $\{j = 0\}$-th column all by zero. Starting from the position of $(i, j) = (1, 1)$, fill elements by

$$f(i,j) = \max \{0,\ f(i-1,j) - gE_i^A, f(i-1,j-1) + r(i,j)s(i,j), f(i,j-1) - gE_j^B\}. \quad (11)$$

To a cell which gives a non-zero maximum, an arrow is directed as a pointer. We backtrack from the position of the largest value $f_{\max}$. This path gives a local alignment.

Step 3: This step is the same as the global alignment.

4 Experiments

4.1 Data Preparation

Since end users of the retrieval are human, the final similarity judgment strongly depends on their sensibility. Therefore, it is desirable that the similarity judgment depends on subjects as less as possible. But, such a simple set would be too easy to judge even by plain machines. Therefore, we designed a data set so that the following is satisfied.

(a) Source video data are totally unlabeled.

(b) Precision and recall on their retrieval can be judged mechanically.

(c) Each video possesses temporal changes of concepts so that the time-dependent property of Fig. 1 bottom right can be identified.

Fig. 2 and Fig. 3 illustrate the generation procedure of the source data.

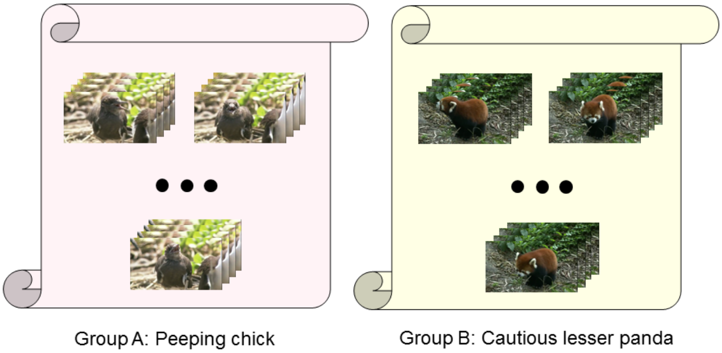


Fig. 2. Groups of videos made from NHK Creative Library

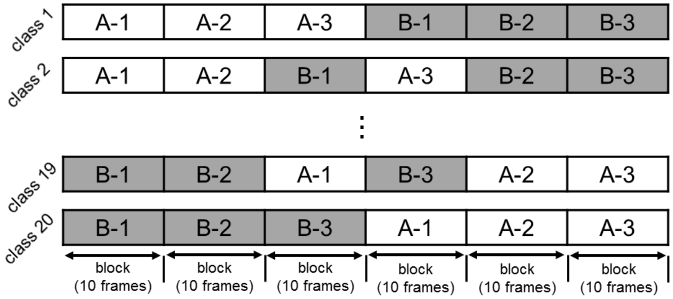


Fig. 3. Twenty classes of videos

Fig. 2 illustrates two groups of videos extracted from the NHK Creative Library which is royalty free [9]. One is a scene of a peeping chick (group A). The other is that of a cautious lesser panda (group B). From these groups, 20 classes of video films were generated as is illustrated in Fig. 3. Each class has 21 videos. Thus, there are 420 test videos, all of which are different each other.

4.2 Alignment Example

On the prepared video set, we tried the time-bound affinity propagation of Section 2.2 to find exemplars accompanied with responsible frames. This is an important step to give a *numerical annotation* to each video. It is equivalent to give a structure to the unorganized video set of Fig. 3 in terms of numerical tags. Such tags can be computed either on-line or off-line.

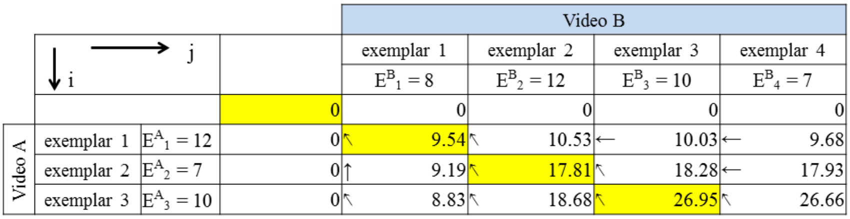
The M-distance is computed by the procedures of Section 3.2 or Section 3.3. Fig. 4 shows a global alignment by $\bar{D} = 1/\sqrt{1-(1/d)}$, $d = 3 \times 256 = 768$, and $g = 0.05$. The backtracking starts from the last element. Fig. 5 illustrates a local alignment by the same $\bar{D}$ and g. Here, the backtracking starts from the largest value. This gave a local matching of Video A to a segment of Video B.

				Video B			
	↓ i → j			exemplar 1	exemplar 2	exemplar 3	exemplar 4
				$E^B_1 = 8$	$E^B_2 = 12$	$E^B_3 = 10$	$E^B_4 = 7$
			0.0	← -0.4	← -1.0	← -1.5	← -1.85
Video A	exemplar 1	$E^A_1 = 12$	↑ -0.6	↖ 9.54	↖ 10.13	← 9.63	← 9.28
	exemplar 2	$E^A_2 = 7$	↑ -0.95	↑ 9.19	↖ 17.81	↖ 17.88	← 17.53
	exemplar 3	$E^A_3 = 10$	↑ -1.45	↑ 8.69	↖ 18.68	↖ 26.95	← 26.60

Fig. 4. A global alignment example

4.3 Evaluation by Precision Recall Curves

Since the video data set was designed deliberately, a mechanical judgment of the precision (11-point interpolated precision) and recall is possible. Numerical



				Video B			
	↓ i → j			exemplar 1	exemplar 2	exemplar 3	exemplar 4
				$E^B_1 = 8$	$E^B_2 = 12$	$E^B_3 = 10$	$E^B_4 = 7$
			0	0	0	0	0
Video A	exemplar 1	$E^A_1 = 12$	0	↖ 9.54	↖ 10.53	← 10.03	← 9.68
	exemplar 2	$E^A_2 = 7$	0	↑ 9.19	↖ 17.81	↖ 18.28	← 17.93
	exemplar 3	$E^A_3 = 10$	0	↖ 8.83	↖ 18.68	↖ 26.95	↖ 26.66

Fig. 5. A local alignment example

values are computed as follows.

$$\text{recall} = |\text{correct videos found}| \, / \, N_{\text{same\_class}} \tag{12}$$

$$\text{precision} = |\text{correct videos found}| \, / \, |\text{top rank videos to be checked}| \tag{13}$$

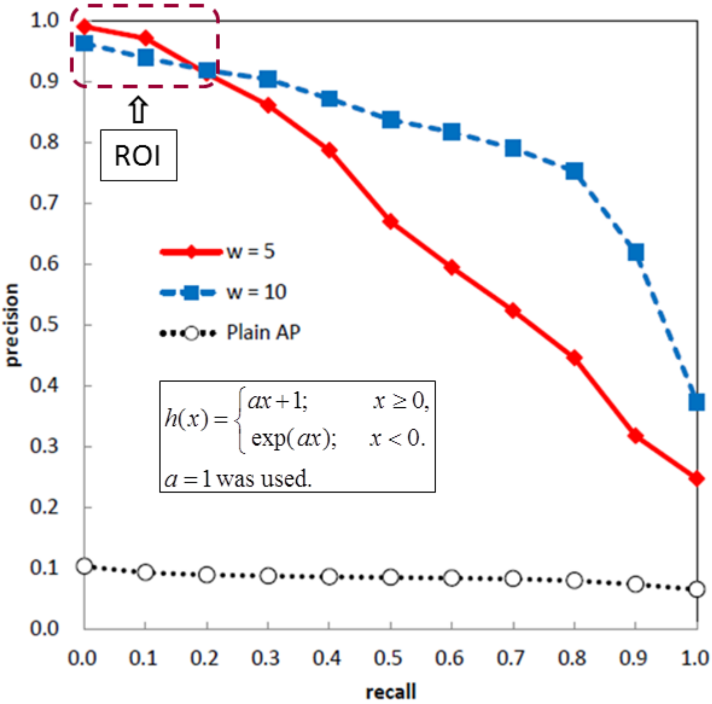


Fig. 6. Precision-recall curves

Fig. 6 illustrates the precision recall curves. We can find that, in its region of interest (recall $\leq 20\%$), the precision is very satisfactory since correct videos are almost always included. On the other hand, the plain AP is unsatisfactory since the time-bound property of Section 2.2 is not considered.

5 Alternative Methods

Here, we consider possibilities of other learning algorithms with the help of Table 1 which summarizes characteristics of exemplar finding methods. This table suggests that a competitive learning approach is possible as a version of the harmonic competition [8].

Table 1. Comparison of methods

Method	Elements	# of exemplars	Mode
time-bound AP	exemplar	variable	successive
harmonic competition	mean vector → exemplar	pre-specified	batch

Step 1: Data set $\{\boldsymbol{f}(\boldsymbol{x}_t, t)\}_{t=1}^{n}$ and the number of clusters are given.
Step 2: Iterations for learning are conducted until the convergence is met.
Step 3: The nearest frame to each centroid is regarded as an exemplar.

Comparison with TBAP: We conducted a set of preliminary experiments by $\boldsymbol{f}(\boldsymbol{x}_t, t) = [\boldsymbol{x}_t, \alpha t]^T$ with $\alpha = 0.015$. Its performance was slightly inferior to the result of Fig. 6. But, the learning speed of a single run was much faster than the affinity propagation family.

6 Concluding Remarks

We presented algorithms and systems for the similar-video retrieval. Since a video has a large size, the set of exemplars and their responsible frames are usually computed off-line. Therefore, they can be used as numerical labels for structure information on a big data set. The reverse direction, or the retrieval, is fast by computing the M-distance.

In [4], it is pointed out that finding the exemplars has a relationship to the labeling by a mechanism of the Hopfield network [10]. After the structural and algorithmic speedup became mature, configurations using such a strategy (e. g., [11]) could be used.

References

1. Katsumata, N., Matsuyama, Y.: Database Retrieval for Similar Images Using ICA and PCA Bases. Engineering Applications of Artificial Intelligence 18, 705–717 (2005)
2. Matsuyama Laboratory: Waseda Image Searchable Viewer (2006), `http://www.wiz.cs.waseda.ac.jp/rim/wisvi-e.html`
3. Hu, W., Xie, N., Li, L., Zeng, X., Maybank, S.: A Survey on Visual Content-Based Video Indexing and Retrieval. IEEE Trans. SMC 41, 797–819 (2011)

4. Frey, B.J., Dueck, D.: Clustering by Passing Messages between Data Points. Science 315(5814), 972–976 (2007)
5. Levenshtein, V.I.: Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics Doklady 10(8), 707–710 (1966)
6. Needleman, S.B., Wunsch, C.D.: A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. J. Mol. Bio. 48, 443–453 (1970)
7. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)
8. Matsuyama, Y.: Harmonic Competition: A Self-Organizing Multiple Criteria Optimization. IEEE Trans. Neural Networks 7, 652–668 (1996)
9. NHK creative library, `http://www1.nhk.or.jp/creative/`
10. Hopfield, J.J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proc. National Academy of Science 79, 2554–2558 (1982)
11. Cheng, L., Hou, Z.-G., Tan, M.: Relaxation Labeling Using an Improved Hopfield Neural Network. Lecture Notes in Control and Information Sciences (345), 430–439 (2006)