Bull. Nat'l Taiwan Nor. Univ.
Vol 40 pp.111-134 1995

師大學報
第40期 第111-134頁 民國84年

# A Transformation-Invariant Relaxation Scheme for Feature Mapping

Sei-Wang Chen 、 Chien-Yun Dai

Department of Information and Computer Education
National Taiwan Normal University
Taipei, Taiwan, Republic of China

## Abstract

A large number of relaxation schemes for feature mapping, claimed to be invariant to transformation, have been reported. However, most of them can deal with transformations involving only rotation and translation, but not scaling. To stay away from the issue of scaling, unrealistic assumptions have to be imposed, such as the conjectures that range data are available so that objects can be rescaled before mapping, and that object shapes are complete so that ratios between object shapes and prototypes can be figured out beforehand. In this paper, we propose a relaxation scheme which is able to be invariant at a time to rotation, translation, as well as scaling. In addition, the proposed scheme can also cope with shapes that may be distorted and incomplete. Our scheme has been tested on both synthetic and real data. Experimental results manifest that the proposed scheme is applicable.

# 1. Introduction

Relaxation processes, typical of iterative operations, deal with a given problem via local refinements in the hope that an overall agreement could finally be attained. This idea can be found in the Waltz work [14] on labeling line drawings of polyhedra. Waltz used an eliminating rule to repeatedly applied to adjacent edges so that incompatible labels between edges are graduately discarded. This process continues until a globally consistent labeling of the line drawings is achieved. Since a complex task fed into a relaxation process will be decomposed into a connected collection of local processes, each of which can be carried out independently, the entire system of processes can thus be performed in a parallel fashion. These characteristics of connectivity and parallelism are, in some sense, analogous to the distributed nature of the human neural network [7]. Yu [13] has implemented his relaxation processes for image thinning using various artificial neural models, including the Hopfield model, the interactive activation and competition neural networks.

In addition to the above characteristics of iteration, connectivity, and parallelism, the capabilities of error absorption and ambiguity resolution inherent in the relaxation processes, have indeed made the relaxation technique a promising problem solving paradigm in the field of computer vision. A variety of relaxation schemes have been proposed for different purposes. They can broadly be categorized, in terms of the defined functionalities, as stochastic (probabilistic) or deterministic (structural) approaches.

Geman et al [5] developed a relaxation scheme for restoring images from degradation of blur and noise. This scheme has been regarded as one of the representative schemes belonging to the class of the stochastic approaches. In their scheme, an image

is formalized as a Markov Random Field in which each pixel is associated with a conditional probability. The probability that an image is in a certain state is in turn specified by the conditional probabilities at pixels. The resultant probability for an image follows a Gibbs distribution. Geman et al used a simulated annealing process to locate a state which maximizes the probability that the image corresponding to the state is a copy of the original image.

Instead of using conditional probabilities, Taxt et al [11] associated each pixel with a physical system inspired from the quantum mechanics. The physical system constitutes a single particle with a certain amount of energy and a potential function. The energy is related to the noise variance at the pixel, whereas the potentials are in connection with the grey levels of the pixel and its first-order neighbors. The particle can only exercise within a restricted space containing the pixel and its neighbors, and behaves in a probabilistic manner. The distribution of the behavior probability can be derived from the Schrodinger quantum mechanic equation. Image restoration and segmentation are then executed by a relaxation process characterized by the piecewise constant or the harmonic oscillator potentials of the physical systems.

An elegant study contributing to the deterministic relaxation approaches may be credited to Hummel, Steven, and Zucker [7]. Unlike many researchers who developed intuitively relaxation techniques based on ad hoc principles and heuristic choices, Hummel et al constructed their relaxation scheme from a theoretical point of view. By such a way, computations within a relaxation process can be fully characterized. A theory of consistency characterized by a set of inequalities of mutual supports between element pairs has been developed in their work. With this theory, the relationship between consistent labeling and optimization can easily be established. Moreover, a number of mathematical concerns about the convergence of processes have also been carefully in-

vestigated. These concerns enable the relaxation processes to automatically determine their stopping criteria.

A lot of researchers have paid their attention to practical uses of the relaxation techniques. Ranade and Rosenfeld [9] consider a problem of matching point patterns by means of relaxation operations. Although their relaxation scheme can handle point patterns with different orientations and translations, scale factors haven't been considered because point patterns to be matched are extracted from the same image. Cucka and Rosenfeld [2] recently extended the above scheme to allow matching of both point and line patterns. Since their compatibility functions rely in part on the attributes of length and distance, their scheme is not scale invariant either. Ton and Jain [12], in their work of registering water regions and oil/gas pads between two Landsat Thematic Mapper (TM) images, extended the Ranade and Rosenfeld scheme by introducing a two-way matching strategy and a mergesort procedure. Unfortunately, Ton and Jain assumed that TM images to be registered are similar in size; the scale factors have also been ignored. Chen and Jain [1] developed a line-pattern relaxation technique for recognizing military vehicles from registered pairs of range and intensity images. Since range data are available in their work, objects are re-scaled before matching.

Except the assumptions made on the availability of scale factors and range values, some even assumed that object shapes are so complete that their ratios to the prototypes could be figured out a priori. Obviously, these assumptions may be unrealistic for some applications. We thus propose, in this paper, a new relaxation scheme capable of dealing with both problems of transformation (including rotation, translation, and scaling) and occlusion (including shape distortion and incompletion). This scheme will be presented in Section 3 after we review the general concept of feature mapping by relaxation in Section 2. Stopping criteria for various situations about input sequences are

discussed in Section 4. The convergence, correctness, and error tolerance of the new scheme are then inspected through experiments in Section 5. Concluding remarks are finally given in Section 6.

## 2. General Concept of Feature Mapping by Relaxation

Let $A_s = \{a_s(i): i = 1,,n_s\}$ and $A_m = \{a_s(j): j = 1,,n_m\}$ be the two patterns of elements to be matched, where $n_s$ and $n_m$ are the numbers of elements, respectively. A deterministic relaxation scheme for mapping elements between two given patterns is generally composed of three components: (1) a similarity function assigning an initial compatibility value for a pair of an element from one pattern and an element from another pattern, (2) a recurrence function iteratively updating the compatibility values of element pairs, and (3) the stopping criteria for terminating the recurrence function. In this section, we address the first two components and leave the last component of stopping criteria to Section 4.

At the beginning of the process, the similarity function h(i,j) assigns a value for each pairing $(a_s(i), a_m(j))$ of an element from $A_s$ and an element from $A_m$. This value reflecting how proximate the two elements are is basically determined from the intrinsic characteristics of the individual elements. Without loss of generality, similarity values are normalized within the range between 0 and 1, which will later be utilized as the initial compatibility values for the recurrence function. Let $Q_s = \{q_s(i):i = 1 \cdots ,n_s\}$ and $Q_m = \{q_m(i):i = 1 \cdots ,n_m\}$ be the respective characteristics of the elements in $A_s$ and in $A_m$. In order to let the similarity values fall within the range between 0 and 1, there are several choices of formulating the similarity function,

$$h(i,j) = (\min\{\frac{q_{st}(i)}{q_{mt}(j)}, \frac{q_{mt}(i)}{q_{st}(j)}\})^{\alpha}, \tag{2.1}$$

or $$h(i,j) = (\frac{1}{1+\beta \cdot |q_{st}(i) - q_{mt}(j)|^{\gamma}})^{\alpha}, \tag{2,2}$$

where $\alpha$, $\beta$, are constants introduced for controlling the rate of convergence. If there is more than one distinct characteristic, a similarity function is constructed for each characteristic and the resultant function may be obtained by either averaging over or producting the similarity functions together. Note that the latter can increase the rate of convergence.

Having defined the similarity function, another function for updating compatibility values of element pairs needs to be determined. We refer to such a function as a recurrence function, denoted by $R^t(i,j)$ at the t-th iteration, and let $R^0(i,j)$ be $h(i,j)$. It is desirable that the compatibility values of the true pairs will become larger and larger, while the values of the false pairs will get smaller and smaller during iterations. Ideally, compatibility values should converge to either 0 (false) or 1 (true). Unlike the similarity function defined in terms of the intrinsic properties of individual elements, the recurrence function is defined in terms of the relational properties between element pairs. To simplify the definition of the recurrence function, we first introduce a function, called a support function. Refer to Figure 1. The support function, denoted by s(i,j;k,l). says that a support is given to an element pair $(a_s(i), a_m(j))$ by the pair $(a_s(k), a_m(l))$. Clearly, there are four possible situations for the pairs $(a_s(i), a_m(j))$ and $(a_s(k), a_m(l))$: both pairs are true pairs, both are false pairs, and either one is false. We hope that only the situation that both pairs are true could obtain a large support, while the rest results in small supports. To this end, the support function can be formulated as:

$$s(i,j;k,l) = (\min\{\frac{\theta_s(i,k)}{\theta_m(j,l)}, \frac{\theta_m(j,l)}{\theta_s(i,k)}\})^{\alpha}, \tag{2.3}$$

or $$s(i,j;k,l) = (\frac{1}{1+\beta \cdot |\theta_s(i,k) - \theta_m(j,)|^{\gamma}})^{\alpha}, \tag{2,4}$$

where $\theta_r(p,q)$'s are some kind of relational property between the two elements $a_r(p)$ and $a_r(q)$ in the same pattern r. Note that both the similarity and the support functions are of the same form. If there is more than one distinct relational property defined, a support function will be constructed for each property and the resultant function may be acquired by either averaging over or producting the support functions together. Again, let computed support values be normalized within the range between 0 and 1.

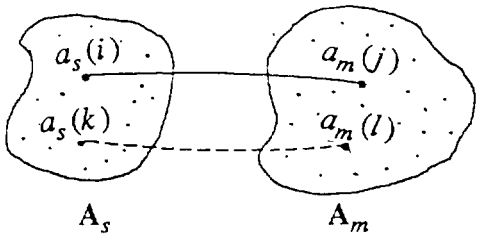



Figure 1. Support s(i,j;k,l) given to $(a_s(i),a_m(j))$ by $(a_s(k), a_m(l))$.

According to the above definition, if a pair $(a_s(i), a_m(j))$ is a true pair, there is only one element, say $a_m(l)$ in $A_m$, which can be correctly paired with a given element, say $a_s(k)$, in $A_s$. In other words, $s(i,j;k,l) \geq s(i,j;k,l')$ for all $a_m(l') \in A_m$. Let

$$s^t(i,j;k) = \max_{l=1}^{n_m} [s(i,j;k,l) \cdot R^t(k,l)]. \tag{2.5}$$

where $R^t(k,l)$ is the compatibility value of pair $(a_s(k), a_m(l))$ at the t-th iteration. The total support $s^t(i,j)$ at the t-th iteration for the element pair $(a_s(i), a_m(j))$ from all other pairs is then given by,

$$s^t(i,j) = \frac{1}{n_s} \sum_{k=1}^{n_s} s^t(i,j;k). \tag{2.6}$$

Finally, the recurrence function, which computes the compatibility value at the next iteration for an element pair $(a_s(i)m\ a_m(j))$, is a function of both the current compatibility value of the pair and the total support given to the pair,

$$R^{t+1}(i,j) = R^t(i,j) \cdot s^t(i,j). \tag{2.7}$$

# 3. Transformation-Invariant Attributes and Functionals

A vision problem in our mind is the assignment of sensed features extracted from images to model features in the database. Our ultimate goal is to recognize, including identifying and localizing, objects present in images. Features used are vertices and polylines resulting from partitioning object silhouettes. In this section, attributes and functionals constituting our relaxation scheme will be defined. They are necessary to be invariant to both spatial transformation and visual occlusion. Transformation includes rotation, translation, and scaling, whereas occlusion includes shape incompletion and distortion.

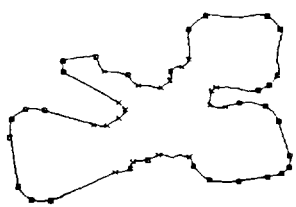


Figure 2. A jumble of keys.

Let $\{v_s(i):i=0,\cdots,n_s+1\}$ and $\{v_m(j):j=1,\cdots,n_m\}$ be the sets of sensed and model vertices, respectively. Model shapes are complete and observed shapes may be occluded. We also assume that the orders of vertices are known a priori. If the observed shape is occluded, after segmentation (see Figure 2), the leading and the trailing vertices, $v_s(0)$ and $v_s(n_s+1)$, of the input sequence of sensed vertices may not correspond to any dominant points of the object silhouette. We should reference them carefully. Let $\{l_s(i):i=1,\cdots,n_s\}$ and $\{l_m(j):j=1,\cdots,n_m\}$ be the respective sequences of sensed and model polylines, formed from connecting the vertices. Define a length ratio r(i) at vertex v(i) as v(i) as $r(i) = l(i+1)/l(i)$. It is realized that length ratios are such a quantity invariant

to both transformation and occlusion. We use them as the first intrinsic attribute for vertices.

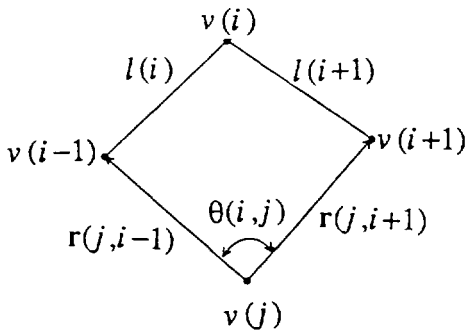



Figure 3. Attributes of length ratio and sweeping angle.

Another potential attribute is the angle of vertices. The angle a(i) of a vertex v(i) is defined as the angle between polylines l(i+1) and l(i) (see Figure 3), and is measured at the interior side of the object. In addition to the above quantitative attributes of length ratio and interior angle, we further introduce a qualitative attribute of vertices, i.e., convex or concave types. Let types of sensed and model vertices be $\{t_s(i):i=1,\cdots,n_s\}$ and $\{t_m(j):j=1,\cdots,n_m\}$, respectively. After determining the intrinsic attributes of features, the similarity function h(i,j) measuring the proximity between a sensed vertex $v_s(i)$ and a model vertex $v_m(j)$ is defined as follows,

$$h(i,j) = \min\left[\frac{a_m(i)}{a_s(j)}, \frac{a_s(j)}{a_m(i)}\right] \cdot \min\left[\frac{r_m(i)}{r_s(j)}, \frac{r_s(j)}{r_m(i)}\right]. \tag{3.1}$$

Here, we product, rather than add, the two min terms on the right hand side of the above equation due to three reasons. The first reason is to increase the rate of convergence; the second reason is to keep the computed value within the range between 0 and 1; the last reason is because length ratios are incompatible with angle ratios. Note that values computed from the similarity function will be used as the initial compatibility values for the later recurrence function $R^t(i,j)$. Define,

$$R^0(i,j) = \begin{cases} h^{\alpha}(i,j) & \text{if } t_s(i)=t_m(j) \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

Here, pairs of vertices with different convexity types are assigned a zero value for their initial compatibility values. This is reasonable because when two vertices have different types it is impossible for them to be matched. The following iterations will not work on vertex pairs with zero compatibility values, so that the processing time can be saved. Finally, $\alpha$ in the above equation is a constant embedded for controlling the rate of convergence.

Unlike the similarity function defined on the basis of the attributes of individual features, the following recurrence function is defined in terms of the relational constraints between features. An angular attribute, being referred to as a sweeping angle, is introduced for this purpose. See Figure 3. Let r(j,i-l) be the vector from vertex v(j) to vertex v(i-1), and r(j,i-l) be the vector from vertex v(j) to vertex v(i+1). Define the sweeping angle $\theta$ (i,j) between vertices v(i) and v(j) as the angle between vectors r(j,i-l) and r(j,i+l). Such angles describing the relationships between vertices are clearly transformation invariant; the recurrence function defined in terms of such an attribute is therefore transformation invariant as well. Define a support s(i,j;k,l) given to a vertex pair ($v_m(i)$, $v_s(j)$) by pair ($v_m(k)$, $v_s(l)$) as,

$$s(i,j;k,l) = \left(\frac{1}{1+\beta \cdot |\theta_s(i,k) - \theta_m(j,l)|^{\gamma}}\right)^{\alpha}, \tag{3,3}$$

where $\alpha$ , $\beta$ and $\gamma$ are constants for controlling the rate of convergence. The total support $s^t(i,j)$ at the t-th iteration for the pair of ($v_s(i)$, $v_m(j)$) from all vertex pairs is then given by,

$$s^t(i,j) = \frac{1}{n_s} \sum_{k=1}^{n_s} \left[\max_{l=1}^{n_m} \{ s(i,j;k,l) \cdot R^t(k,l)\}\right] \tag{3.4}$$

Finally, the recurrence function, which is a function of the current compatibility value and the computed support value, is defined as,

$$R^{t+1}(i,j) = R^t(i,j) \cdot s^t(i,j). \tag{3.5}$$

Furthermore, at each iteration, we apply a two-way normalization to the compatibility matrix. This is inspired by a notion of two-way matching, previously introduced by Ton and Jain [12]. The notion is that if the features in one set select their matches from the other set but not vice versa, this matching is called a one-way matching; however, if features in both sets choose their matches, then the matching is referred to as a two-way matching. Ton and Jain have shown that the two-way matching can generate more consistent feature pairs in the relaxation process. Our normalization proceeds as follows. Suppose we are now at the t-th iteration. First, normalize each row of the compatibility matrix $R^t = [R^t(i,j)]$ such that the row sum equals one. Let $R_r^t = [R_r^t(i,j)]$ be the row normalized matrix, where

$$R_r^t = \frac{R^t(i,j)}{\sum_{k=1}^{nm} R^t(i,k)}$$

Likewise, normalize each column of $R^t$ to obtain the column normalized matrix $R_c^t = [R_c^t(i,j)]$, where

$$R_c^t = \frac{R^t(i,j)}{\sum_{k=1}^{ns} R^t(i,k)}$$

The resultant normalized compatibility matrix $R'^t = [R'^t(i,j)]$ is then defined as the componentwise multiplication of the row normalized matrix $R_r^t = [R_r^t(i,j)]$ and the column normalized matrix $R_c^t = [R_c^t(i,j)]$, i.e.

$$R'^t = [R'^t(i,j)] = [R_r^t(i,j) \cdot R_c^t(i,j)]$$

Since the attributes used to define the functionals constituting our relaxation scheme are all invariant to rotation, translation, and scaling, the proposed scheme is thus invariant to spatial transformations. In addition, since only visible features and local con-

straints are employed, our relaxation scheme also resists the effect of shape incompletion. One difficulty commonly encounted at the end of iterations is how to decide true pairs from the resultant pairing. This is a formidable task for it is equivalent to segmenting objects from a jumble. We propose a strategy to handle this problem, which relies on both a priori information and a principle of transversality [10]. The a priori information is based on the observation that the input sequences of sensed and model features follow the adjacent relationship of their physical counterparts and the intersection between the two sequences retains the relationship of adjacency. Therefore, randomly paired features will violate the relationship and can be removed. The surviving pairs are deemed to be the feasible matches. We further apply a principle of transversality, previously proposed by Richards and Hoffman [10], to the feasible pairs to verify whether they are true pairs. The principle of transversality stems from the observation that when entities are joined to create complex objects concavities almost always are created at the join. A partition rule thus says that segment a curve at concave cusps in order to break the shape into its parts. Clearly, a connected collection of true pairs should correspond to one of such segments.

## 4. Stopping Criteria

Features extracted from an image form a pattern. Different stopping criteria are needed for different patterns. In this section, instead of looking at this problem from a theoretical point of view, we investigate possible patterns and provide suitable stopping criteria from an empirical viewpoint. Let $A_s = \{ \alpha_s(i): i=1,\cdots,n_s \}$ and $A_m = \{ \alpha_m(j): j=1,\cdots,n_m \}$ be the two patterns to be matched. Without loss of generality, think of $A_s$

to be a set of features extracted from an image, and $A_m$ to be a set of features stored in the database. The compatibility matrix at the t-th iteration is denoted by $R^t$. Then, the sequence $\{R^t\} = \{R^0, R^1, R_2, \cdots\cdots\}$ is said to converge to $R^*$ if

$$\lim_{t\to\infty} \| R^t - R^* \| = 0, \tag{4.1}$$

where $\| \cdot \|$ denotes a certain matrix norm. The above expression implies that for each feature pair $(a_s(i), a_m(j))$, $i \leq n_s$, $j \leq n_m$, the sequence $(r^t)_{ij}$ of (i,j)-th components of $R^t$ converges to the (i,j)-th component $(r^*)_{ij}$ of $R^*$. According to the functionals defined in the last section, the components of $R^*$ will ideally be either zero or one. A pair $(a_s(i), a_m(j))$ with its compatibility value close to one (i.e. $(r^*)_{ij} = 1$) will be regarded as a true pair, whereas a pair having its compatibility value close to zero will be regarded as a false pair.

In practice, an ideal $R^*$ is unreachable because of computer round-off and truncation errors. Each component of the matrix should be associated with a small positive value $\varepsilon$, called macheps (machine epsilon), which relies only on the computer precision. Other errors, such as sensing and segmentation errors, are in principle eliminated in the course of iterations. However, $\varepsilon$ should not be too small; otherwise, it may take a long time to finish the job. Furthermore, since $R^*$ is unknown a priori, the convergence condition (4.1) can not be applied directly but is replaced by,

$$\lim_{t\to\infty} | r^{t+1}(i,j) - r^t(i,j) | = \varepsilon \text{ , for all (i,j)'s} \tag{4.2}$$

where $| \cdot |$ is the absolute notation. Since the compatibility value of any feature pair either increases or decreases monotonically to one or zero, it is easy to prove that Condition (4.2) is equivalent to Condition (4.1) [3]. We define, at the t-th iteration, the rate $\gamma$ (t) of convergence for the recurrence function as,

$$\gamma(t) = \frac{\| R^{t+1} - R^t \|}{\| R^t \|} \tag{4.3}$$

It should formally decrease as iterations increase.

The above conditions provide clues for designing the stopping criteria of our relaxation process. However, there may still be unpredictable cases in which the process may infinite loop. In order to avoid this situation, it would better to assign a fixed number of iterations.

## 5. Experiments

We claimed in the previous sections that our relaxation scheme is able to be invariant to both spatial transformation and visual occlusion. However, in light of noise, distortion, and error present in the input data, there are empirical limitations of the proposed scheme. This section investigates the possible limitations of our scheme. First, we use synthetic data to test the correctness of the scheme. Next, noisy data are used to inspect the tolerance of the scheme to noises. Finally, we apply the scheme to real images in which shape distortion and segmentation error are present.

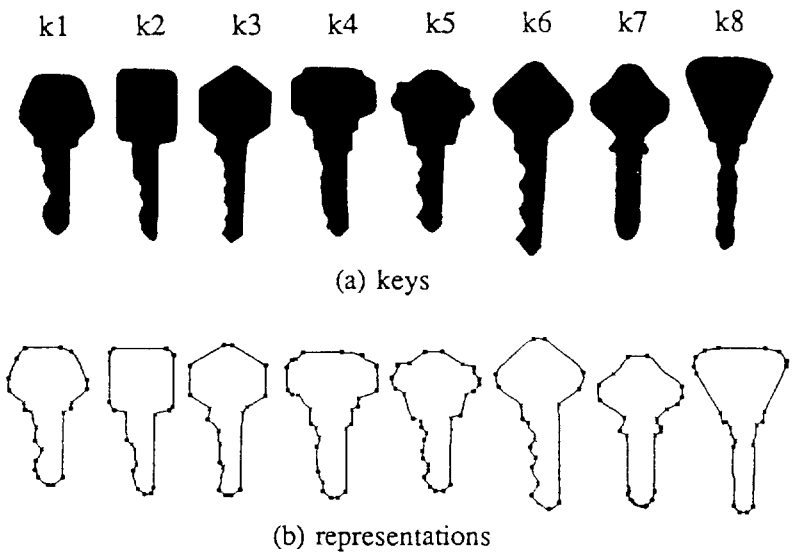



Figure 4. The object domain for the first part of experiments.

Eight keys with different shapes (Figure 4(a)) form the object domain of our experiments. Keys are represented as strings of symbols, each of which corresponds to a polyline of object boundaries. The technique employed to polygonize object silhouettes is the one previously proposed by Fischler and Bolles [Fis86]. Polylines are associated with attributes of length, orientation, and type. Here, a polyline type is defined as follows. Suppose that curvatures along a silhouette can be calculated using some underlying curve fitting method. A polyline consisting of two endpoints can thereby be classified, in terms of the signs of curvature at its endpoints, as either one of the four types: ++, +-, -+, and --. We refer to such an object representation as an attributed string representation (Figures 4(b)).

To test the characteristic of scale invariance of the proposed scheme, a series of synthetic images containing keys with various scales between 0.2 and 2.0 are generated from the key models. There are 8 synthetic key images created for each chosen scale. Synthetic keys are then matched against models using the proposed relaxation scheme. Different numbers of iterations n (5, 10, and 15) and convergence tolerances $\varepsilon$ (0.01, 0.05, 0.1, 0.3, and 0.5) have been employed, from which the reliability of the proposed scheme can be examined. Note that the constants $\alpha$, $\beta$, $\gamma$, and $\zeta$ appearing in Equations (3.2) and (3.3) for controlling the rate of convergence are presently set to 1 for simplicity. Experimental results to be illustrated later are collected in Table 1. Note also that the above experiments assume that the numbers of sensed and model features are the same, which are indeed not the case in practice. We wonder whether incompatible numbers of features will disturb the regularity of convergence. Hence, we generate another set of synthetic keys with incomplete shapes to simulate occlusion cases. The experimental results are collected in Table 1 as well.

TABLE 1 Tests on scale resistance.

| matching rate, (complete shape/incomplete shape) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ε =0.01 | | | ε =0.05 | | ε =0.1 | ε =0.3 | ε =0.5 |
| s n | 5 | 10 | 15 | 5 | 10 | 5 | 5 | 5 |
| 2.00 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 |
| 1.50 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 |
| 1.00 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 |
| 0.50 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 | 1.0/1.0 |
| 0.30 | .88/.88 | .88/.88 | .88/.88 | .88/.88 | .88/.88 | .88/.88 | .88/.88 | .88/.88 |
| 0.25 | .75/.75 | .75/.75 | .75/.75 | .75/.75 | .75/.75 | .75/.75 | .75/.75 | .75/.75 |
| 0.20 | .75/.62 | .75/.62 | .75/.62 | .75/.62 | .75/.62 | .75/.62 | .75/.62 | .75/.62 |

ε :convergence tolerance, n:number of iterations, s:scale factor

In Table 1, each entry contains a pair of values $\nu_l / \nu_r$. The value $\nu_l$ on the left specifies the matching rate resulting from a complete shape, whereas the value $\nu_r$ on the right specifies the matching rate for an incomplete shape. Except the first and the second columns of the table, others indicate only the results for iterations of 5, because after a few iterations (about 5) true pairs already prevail according to the compatibility values. More iterations seem not to influence the matching rate. We also find that varying tolerances of convergence by no means change the final matching rate either. These observations reveal that the proposed relaxation scheme can perform reliably for synthetic data.

However, as shown in Table 1, once scale factors are reduced below 0.3, the matching rates dropped rapidly. This phenomenon can be explained by the following reasons. First, since images are discrete, if sizes of objects present in images are reduced, their relative discrete errors will increase. Secondly, our relaxation scheme heavily relies on angular attributes which have been known to be more sensitive to errors than those attributes of length and distance. Return to Table 1. The performances of the scheme in the cases of complete and incomplete shapes are close to each other. This indicates that incompatible numbers of features, owing to missing and artifact features, will not influence the convergence of compatibility values for both true and false pairs. The above

remarks are made under the assumption that there is no interference of noise which will be considered next.

TABLE 2. Experimental results of noisy data.

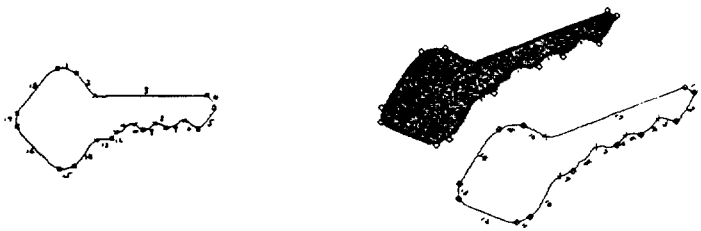
| matching rate (incomplete shapes only) | | | | | |
|---|---|---|---|---|---|
| | levels of noise | | | | |
| s | .03 | .06 | .09 | .12 | .16 |
| 2.00 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 1.50 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 1.00 | 1.0 | 1.0 | 1.0 | 1.0 | .75 |
| 0.50 | 1.0 | .88 | .88 | .88 | .62 |
| 0.30 | .88 | .75 | .62 | .50 | .62 |
| 0.25 | .88 | .88 | .75 | .62 | .38 |
| 0.20 | .62 | .75 | .75 | .62 | .38 |

number of iterations=5, convergence tolerance = 0.05

A uniform random number generator has been employed to generate numbers representing error percentages of vertex positions on object boundaries. Erroneous shapes of objects are created by shifting the vertices of objects according to the generated error percentages. Different levels of noise are defined as the distinct ranges given to the uniform random number generator. Table 2 shows the experimental results of noisy data. In this table, five noise levels (3%, 6%, 9%, 12%, and 16%) are employed. As we expect, the larger the object shapes are, the better the relaxation scheme is to resist noise.

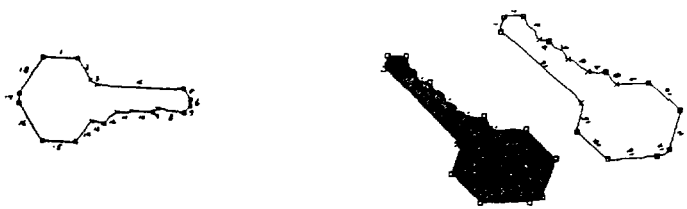
Figure 5 shows two examples of real images and their experimental results. In this figure, models are depicted on the left side of the figure, and observed keys are displyed on the right side. As we can see, observed keys extracted from images are severely distorted. Look at example 1. The key model has 18 polylines. Although only 16 polylines are acquired from segmentation, both missing and artifact features appear in the resulting data. After iterations, the first sensed polyline is mapped to model polyline 5, the second sensed polyline is mapped to model polyline 6, and so forth (refer to

Figure 5(a)). According to this mapping result, 2 out of 16 polyline pairs are mismatched; in other words, 87.5% of sensed polylines are correctly matched. This will be good enough for the ensuing recognition procedure. The mismatched pairs in this example are (12,18) and (13,17). In the second example, 3 out of 17 pairs are mismatched; equivalently, 82.4% of sensed polylines are correctly matched. The mismatched pairs are (8,14), (14,16), and (17,18). The above results are reasonable because of the existence of both segmentation error and shape distortion in the input data.



sensed features: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
model features: 5 6 8 9 10 11 12 13 14 15 16 18 17 1 2 4

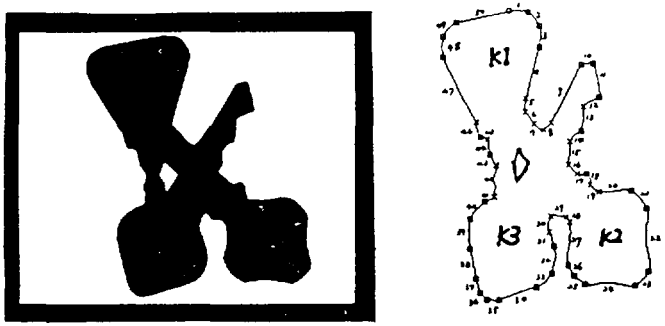
(a) example 1



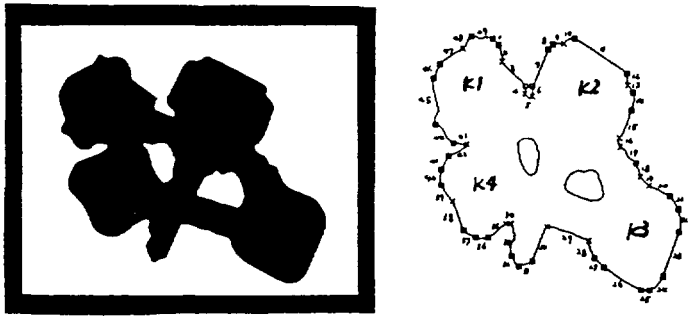
sensed features: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
model features: 4 5 6 8 9 10 14 13 14 15 16 17 16 1 2 18

(b) example 2

Figure 5. Tests of transformation invariance.




k1:6 7 12 13 14 15 17 19 20 21 17 9 2 3
k2:18 19 7 6 21 6 1 22 1 2 14 15 16 17
k3:13 14 15 16 17 10 2 3 9 8 9
(a) scene 1




k1:1 2 1 27 28 15 14 3 4 5 12 13
k2:20 21 10 11 12 13 18 27 1 2
k3:12 13 11 12 15 16 16 17 4 5 1 2 3
k4:17 18 19 12 10 11 2 3 4
(b) scene 2

Figure 6. Jumbles of keys.

Finally, we apply our relaxation scheme to jumbles of keys (Figure 7). In principle, model representations can directly match with a given scene representation. Unfortunately, it doesn't work because there are too many false pairs between model and scene polylines. Our relaxation scheme can not work well as soon as the ratio of the number

of false pairs to the number of true pairs becomes too large. This forces us to partition the scene into individual objects before matching. Refer to Figure 7, where sensed polylines correctly matched are underlined. In scene 1, except key k2, the other two keys k1 and k3 both have more than half of their polylines correctly matched. This is same to the keys k3 and k4 in scene 2. Those keys having poor matching rates are solely due to the fact that silhouette segments used are too short to match.

## 6. Concluding Remarks

A new relaxation scheme for feature mapping has been presented in this paper. Previous researchers either assumed that the range values are available, or assumed that object shapes are complete, once the mapping problems involving scale factors are encountered. To deal with scale factors yet get around unrealistic assumptions, we propose in this paper another scheme capable of invariance to both spatial transformation, as well as visual occlusion. The basic idea is that all functionals constituting the scheme have to be defined in terms of attributes that are invariant to transformation and occlusion. Attributes employed include the length ratio of polylines, the angle of vertex, the sweeping angle between vertices, and the polyline type.

Although our relaxation scheme can work pretty well on synthetic data, the behavior of the scheme is inevitably disturbed if noise, distortion, and error are present in the input data. From our experiments, if defection of data can be controlled within a reasonable range, the disturbance becomes asymptotically stable with iterations and the final results are acceptable as well. This manifests that the relaxation processes do absorb defections from input data. On the other hand, it becomes out of control if the imper-

fection of data increases to a certain extent. There are modifications applicable for the current scheme, for instance, the replacement of production by addition in Equation (3.1), and the use of different constants for controlling the rate of convergence in Equations (3.2) and (3.3). Furthermore, we may formalize relaxation processes as optimization or stochastic processes in order to better understand their behaviors. Finally, because of the parallel and networking nature of the relaxation schemes, it is worth investigating connections between the relaxation processes and the artificial neural models.

# REFERENCES

1. S. W. Chen and A. K. Jain, Strategies of Multiple-View and Multiple-Matching for 3D Object Recognition, *Journal of Computer Vision, Graphics, and Image Processing: Image Understanding*, 1992.
2. P. Cucka and A. Rosenfeld, Linear Feature Compatibility for Pattern-Matching Relaxation, *Pattern Recognition*, Vol. 25, No. 2, pp. 189-196, 1992.
3. J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.
4. M. A. Fischler and R. C. Bolles, Perceptual Organization and Curve Partitioning, IEEE *Trans. Pattern Analysis and Machine Intelligence*, Vol. 8, No. 1, pp. 100-105, 1986.
5. S. Geman and D. Geman, Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images, *IEEE Trans. Pattern Analysis and Machine Intelligence* , Vol. 6, pp. 721-741, 1984.
6. G. E. Hinton, Relaxation and its Role in Vision, *Ph.D. Dissertation, University of Edinburgh*, 1979.
7. R. A. Hummel and S.W. Zucker, On the Foundations of Relaxation Labeling Processes, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 5, pp. 267-287, 1983.
8. S. Peleg, A New Probabilistic Relaxation Scheme, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 2, pp. 362, 1980.
9. S. Ranade and A. Rosenfeld, Point Pattern Matching by Relaxation, *Pattern Recognition*, Vol. 12, No. 4, pp. 269-276, 1980.

10. W. Richards and D. D. Hoffman, Codon Constraints on Closed 2D Shapes, *Readings in Computer Vision*, Ed. by M. A. Fischler and O. Firschein, Morgan Kaufmann Pub. Inc., pp. 700-708, 1987.

11. T. Taxt and E. B $\phi$ eslviken Relaxation Using Models from Quantum Mechanics, *Pattern Recognition*, Vol. 24, No. 7, pp. 695-710, 1991.

12. J. C. Ton and A. K. Jain, Registering Landsat Images using Point Pattern Matching, IEEE *Trans. Geoscience and Remoting Sensing*, Vol. 27, No. 5, pp. 642-651, 1989.

13. S. S. Yu, A Study on the Relaxation Process: Applications and Neural Network Implementations, Ph.D. Dissert., *Institute of Computer Science and Information Engineering, College of Engineering, National Chiao Tung Univ.*, Hsinchu, Taiwan, Republic of China, 1990.

14. D. L. Waltz, Understanding Line Drawings of Scenes with Shadows, *The Psychology of Computer Vision*, Ed. by P. H. Winston, New York, McGraw-Hill, pp. 19-91, 1975.

# 供特徵元件匹配用具有空間轉移不變性之鬆弛法架構

陳世旺、戴建耘
國立台灣師範大學資訊教育學系
國立台灣師範大學工業教育學系

## 摘 要

截至目前爲止已有不少供幾何特徵匹配用之鬆弛法架構被提出來，這些架構雖然宣稱可以對空間轉移具有不變性，但是實際上，大部份只能處理和旋轉及平移有關的轉移，對於具有尺度因素的轉換則儘量避免，因此便有各種不同的假設被加諸於所考慮的問題，例如假設我們已知景深值，因此可以先將物件的尺度正規化後再比對，或者假設物形是完整的，於是物形和模型間的尺度比率事先可以推知。本篇文章提出一種新的架構，它能夠同時對旋轉，平移，和尺度具有不變性，此外，新架構也能處理變形物件及不完整物形。我們的實驗結果顯示新的架構確具有可行性。