Shape Matching Based on Invariants

Stan Z. Li

School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore 639798
http://markov.eee.ntu.ac.sg:8000/~szli

Chapter X: Shape Matching Based on Invariants
In Omidvar (ed), *Shape Analysis*
Progress in Neural Networks, Vol.6, Ablex

INTRODUCTION

Shape matching in computer vision aims to establish correspondences between primitive shape features in the input data such as an image and in model objects. Schemes proposed to date fall into three categories (a) those based on invariant properties, (b) those based on object decomposition into parts and (c) those based on computation of transformation between the shapes in the scene and in a model-base. Two major problems have to be addressed regardless of which scheme is used: how a shape is represented and how matching is carried out.

This chapter addresses two issues in shape analysis: (i) invariant-based shape representation and (ii) optimal shape matching. Shape representation is the basis on which shapes can be compared. Shape invariants, that do not change in values under a class of transformations, are the most effective and efficient for matching. This is because they make it possible to compare shapes directly before transformation parameters are found. On the other hand, shape matching is generally performed in the sense of an optimality criterion, especially in the presence of noise and occlusion when exact matching does not exists. A problem is how to encoded the constraints, provided by the invariants, into the objective function or criterion.

The chapter is organized as follows: Section 2 introduces important transformations in computer vision and describes shape invariants under these transformations. Section 3 presents a general optimization approach for invariant-based shape matching. A structural representation called the Attributed Relational Structural (ARS) is used to systematically encapsulate various types of invariants. A matching is considered as a mapping from the scene ARS to the model ARS. A functional of the mapping is formulated to measure the goodness of matching and is to be maximized to obtain the optimal solution. Section 4 discusses problems in searching for optimal solutions. Two approaches, namely relaxation labeling and Hopfield neural nets, are described and compared. Section 5 presents experiments.

SHAPE INVARIANTS

By using the term invariants, we mean descriptors of shape geometry that do not change under the concerned class of transformations. Examples of transformation classes in computer vision are Euclidean, similarity, affine, and projective. In mathematics, the study of shape invariants has a long history. In computer vision, the importance of geometric invariants has been recognized since the origin of the field; invariance is believed to be the essential property of a shape [1, 2]. Some group theoretic results can be found in [3, 4]. Simple invariants are easily found and have been well utilized. Recent research has led to discovery of less simple, yet important invariants (See [2] for recent developments in this field).

This section introduces basic concepts of transformations and projections in computer vision and describes the most useful invariants under these transformations.

Transformations

A transformation $\mathcal{T}$ is a bijective mapping between two sets. Euclidean transformations, denoted $\mathcal{T}_E$, can be used to model the following situations: two-dimensional objects orthographically projected onto two-dimensional images with known scale factor; three-dimensional object measured by three-dimensional data such as from a range finder or a tactile sensor with known scale factor. Similarity transformations $\mathcal{T}_S$ are useful for similar situations with unknown scale factors. Affine transformations $\mathcal{T}_A$ can model situations in which two-dimensional objects in three-dimensional space are projected onto two-dimensional images under weak perspectives.

Perspective projections $\mathcal{T}_P$ are the most general projections in the visual world. The scaling factor therein depends on the depth, or the distance, of points being viewed. In weak perspective projections $\mathcal{T}_W$, an assumption is made that the depth span of the viewed points is small compared to the average depth of the points. Under this assumption, the scaling factor is approximated by a constant. Projections from a planar object to an image are transformations, whereas projections with dimension reduction, such as those from three-dimensions to two-dimensions, are not because they are not bijective.

Coordinate Systems

Let $\mathbf{P}$ be a point whose coordinate in the n-dimensional Cartesian coordinate system is $\mathbf{P} = (X_1, \ldots, X_n)^T$. Let $\mathcal{T}$ be a class of transformations. $\mathbf{P}$ is transformed to $\mathbf{p} = \mathcal{T}(\mathbf{P})$ under $\mathcal{T}$.

It is sometimes advantageous to represent transformations in terms of homogeneous coordinates. A point $\mathbf{P} = (X_1, \ldots, X_n)^T$ in the n-dimensional Cartesian coordinate system is represented as a point $\tilde{\mathbf{P}} = (\tilde{X}_1, \ldots, \tilde{X}_n, w)^T$ in the $(n+1)$-dimensional homogeneous coordinate system for any scale factor $w \neq 0$. The two are related by $(X_1, \ldots, X_n)^T = (\tilde{X}_1/w, \ldots, \tilde{X}_n/w)^T$. In homogeneous coordinates, a transformation or a projection can be written in a simple matrix form

$$\tilde{\mathbf{p}} = \mathcal{T}(\tilde{\mathbf{P}}) = \mathbf{T}\tilde{\mathbf{P}} \tag{1}$$

where $\mathbf{T}$ is the transformation matrix. In this form, a transformation is carried out between the two homogeneous coordinate spaces and is linear in $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{p}}$. Thus, related computation can be done by simple matrix manipulations.

Euclidean Transformations $\mathcal{T}_E$

This class of transformations is defined as a rotation (orientation change) followed by a translation

$$\mathbf{p} = \mathcal{T}_E(\mathbf{P}) = \mathbf{OP} + \mathbf{t} \tag{2}$$

where $\mathbf{O} = (O_{ij})_{n\times n}$ is the rotation matrix and $\mathbf{t} = (t_1, \ldots, t_n)^T$ is the translation vector. The corresponding transformation matrix is

$$\mathbf{T} = \begin{pmatrix} \mathbf{O} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \tag{3}$$

where $\mathbf{0}$ is the row vector of n zeros. The homogeneous coordinates before and after the transformation are $\tilde{\mathbf{P}} = (\tilde{X}_1, \ldots, \tilde{X}_n, 1)^T$ and $\tilde{\mathbf{p}} = (\tilde{x}_1, \ldots, \tilde{x}_n, 1)^T$, respectively. The rotation matrix under $\mathcal{T}_E$ is orthogonal: $\mathbf{O}^T\mathbf{O} = \mathbf{I}$ where $\mathbf{I}$ is the $n \times n$ identity matrix; and has the unit norm (the unit scaling factor): $\det(\mathbf{O}) = \pm 1$ with -1 representing the reflection. An isotropic size change with a known scaling factor can be treated as $\mathcal{T}_E$ after multiplying $\mathbf{O}$ by that factor.

Similarity Transformations $\mathcal{T}_S$

This class is more general than $\mathcal{T}_E$ in that it is subject to additional non-unit isotropic scale changes and hence is a super-class of $\mathcal{T}_E$. It also has the form as $\mathbf{p} = \mathcal{T}_S(\mathbf{P}) = \mathbf{OP} + \mathbf{t}$. The rotation matrix $\mathbf{O}$ is still orthogonal, but the scaling factor $\det(\mathbf{O})$ is an unknown nonzero constant. Nonetheless, the scaling is isotropic owing to the orthogonality constraint.

Affine Transformations $\mathcal{T}_A$

This is a super-class of $\mathcal{T}_S$ in that the scaling can be anisotropic. It also has the form as $\mathbf{p} = \mathcal{T}_A(\mathbf{P}) = \mathbf{OP} + \mathbf{t}$; but the orthogonality constraint on $\mathbf{O}$ is removed. The only remaining constraint is the non-singularity $\det(\mathbf{O}) \neq 0$. The most important characteristics under $\mathcal{T}_A$ is that the parallel lines remain parallel after the transformation.

These classes of transformations from one space to another do not reduce the dimensionality of spaces. In the case of dimensional reduction, for example when the visual data is a two-dimensional image of three-dimensional objects, we consider the following projections.

Perspective Projections $\mathcal{T}_P$

Let us first model perspective projections from the viewer-centered coordinate system to the image coordinate system. Assume that the image plane is placed at $Z = f$, where f is the focal length and is parallel to the XY plane of the viewer's coordinate. A point $\mathbf{P} = (X, Y, Z)^T$ in the viewer's coordinate is projected onto $\mathbf{p} = (x, y)^T$ in the image plane according to the following equations:

$$x = \frac{fX}{Z}, \quad y = \frac{fY}{Z} \tag{4}$$

Written in the matrix form

$$\mathbf{p} = \frac{f}{Z}\mathbf{A}\mathbf{P} \tag{5}$$

where

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \tag{6}$$

is the orthographic projection matrix and $\frac{f}{Z}$ is the scale factor. The factor is proportional to $\frac{1}{Z}$ and hence is a source of nonlinearity. The parallelism of lines is not preserved; rather, all parallel lines meet at a vanishing point. The most important characteristics under perspective projection is that a straight line remains straight.

This is the projection relationship between the viewer and the image coordinate systems. In many vision tasks, it is useful to express the image in the object-centered coordinate system. This can be done by using two concatenate transformations: An Euclidean transformation, and a subsequent perspective projection

$$\mathbf{p} = \mathcal{T}_P(\mathbf{P}) = \frac{f}{Z'}\mathbf{A}(\mathbf{O}\mathbf{P} + \mathbf{t}) \tag{7}$$

In the above, $\mathbf{P} = (X, Y, Z)^T$ is a point in the object-centered coordinate system, $\mathbf{O}$ a 3×3 rotation matrix, $\mathbf{t}$ a 3×1 translation vector and $Z' = O_{31}X + O_{32}Y + O_{33}Z + t_3$ is the Z coordinate of the rotated and translated $\mathbf{P}$ in the viewer coordinate system. The corresponding transformation matrix is

$$\mathbf{T} = \begin{pmatrix} O_{11} & O_{12} & O_{13} & t_1 \\ O_{21} & O_{22} & O_{23} & t_2 \\ O_{31}/f & O_{32}/f & O_{33}/f & t_3/f \end{pmatrix} \tag{8}$$

and the related homogeneous coordinates are $\tilde{\mathbf{P}} = (\tilde{X}, \tilde{Y}, \tilde{Z}, 1)^T$ and $\tilde{\mathbf{p}} = (\tilde{x}, \tilde{y}, w)^T$, respectively. Assuming $\det(\mathbf{O}) = 1$, the scale factor is $w = f/(O_{31}X + O_{32}Y + O_{33}Z + t_3)$.

Weak Perspective Projections $\mathcal{T}_W$

The perspective is said to be weak when the distance between the camera and the object is much larger than the depth span of the object. In this case, the previous equation can be approximated by

$$\mathbf{p} = \mathcal{T}_W(\mathbf{P}) = \frac{f}{Z'_0}\mathbf{A}(\mathbf{O}\mathbf{P} + \mathbf{t}) \tag{9}$$

where Z'_0 is the average depth of the object. Now, the scale factor is fixed at $\frac{f}{Z'_0}$. This simplifies the analysis. The corresponding transformation matrix can be written

$$\mathbf{T} = \begin{pmatrix} O_{11} & O_{12} & O_{13} & t_1 \\ O_{21} & O_{22} & O_{23} & t_2 \\ 0 & 0 & 0 & Z'_0/f \end{pmatrix} \tag{10}$$

$\tilde{\mathbf{P}}$ and $\tilde{\mathbf{p}}$ are the same as in the general perspective case. Under $\mathcal{T}_W$, the projection from a planar object in three-dimension to an image in two-dimension is equivalent to an affine transformation and the parallelism of lines is preserved.

Orthographic Projections $\mathcal{T}_O$

If the focal length is very large $f \to \infty$, the object is far away from the view point $Z_0' \to \infty$ and both f and Z_0' are in the same order of magnitude, then we have $\frac{f}{Z_0'} \approx 1$. In this case, the projection is parallel or orthographic

$$\mathbf{p} = \mathcal{T}_O(\mathbf{P}) = \mathbf{A}(\mathbf{OP} + \mathbf{t}) \tag{11}$$

with

$$\mathbf{T} = \begin{pmatrix} O_{11} & O_{12} & O_{13} & t_1 \\ O_{21} & O_{22} & O_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{12}$$

In this case, distances in any plane parallel to the image is preserved.

Shape Invariants

Let $\mathbf{Q}$ be the structure of a shape defined by a set of features. Let $\mathbf{q} = \mathcal{T}(\mathbf{Q})$ be the transformed shape under $\mathcal{T}$. A descriptor $I(\cdot)$ of the shape is an invariant under $\mathcal{T}$ if

$$I(\mathbf{Q}) = I(\mathbf{q}) \tag{13}$$

The condition says that the descriptor of the shape remains unchanged after the transformation. An invariant is said to be order one if it is defined on a single feature, order two if defined between two features, three if defined among three features, and so on. An invariant of order one is called an invariant property or an invariant unary relation, that of order two an invariant binary relation, that of order three an invariant ternary relation, and so on.

Invariants under a certain class of transformations are also invariants under its subclasses. Therefore the number of invariants under a certain class of transformations is larger than that under its super-classes. Because $\mathcal{T}_E$ is the smallest class of all those classes that have been previously introduced, it has the largest number of invariants.

On the other hand, the total number of higher order invariants that exist under a certain class of transformations is larger than that of lower order ones. For example, a point or a line by itself has no algebraic invariant unary relations [3] under any transformations except the trivial identity transformation. However, we know that the distance between two points or the angle between two lines (binary relation) are invariant under Euclidean transformations.

The following description of invariants are made in terms of transformation classes, orders of invariants and types of shape primitives (such as points, lines, curves and surfaces).

Under Euclidean Transformations

Although a point by itself has no algebraic invariant unary relations, it is possible to derive some differential unary invariants either for a point on a curve or a surface based on the differential local structure in an infinitesimal neighborhood of the point.

Curvature is an invariant unary relation under $\mathcal{T}_E$. Given a curve $z = z(x)$, its curvature at $\mathbf{p} = (x, z(x))$ is defined by

$$\kappa = \frac{z_{xx}}{(1 + z_x^2)^{3/2}} \tag{14}$$

where z_x and z_{xx} are derivatives of z. Given a graph surface $z = z(x, y)$, the mean curvature H and the Gaussian curvature K of the surface at $\mathbf{p} = (x, y, z(x, y))$ are defined by

$$H = \frac{(1 + z_y^2)z_{xx} + (1 + z_x^2)z_{yy} - 2z_x z_y z_{xy}}{2(1 + z_x^2 + z_y^2\ \ )^{3/2}} \tag{15}$$

$$K = \frac{z_{xx}z_{yy} - z_{xy}^2}{(1 + z_x^2 + z_y^2)^2} \tag{16}$$

The curvatures are invariant to rotations, translations and surface parameterization as long as the Jacobian of the transformation is nonzero. The proof can be obtained based on the fundamental *existence and uniqueness theorem* of curves and surfaces from *differential geometry* [5].

The curvatures are not invariant under $\mathcal{T}_S$ because their magnitudes change, but the signs $\{+, 0, -\}$ of the curvatures are invariant, in theory. The curvature sign of a curve classifies the points on a curve into three different types: convex, flat, and concave. The combination of the sign of mean curvature and the sign of Gaussian curvature of surface points yields eight possible surface types [6]. The signs of the surface curvatures are, therefore, symbolic invariants. In practice, however, a curvature sign is always with respect to a scale-relative threshold [7]. So it is safer not to use it as a scale invariant unless a scale-adaptive threshold can be computed.

The Euclidean distance between two points is an invariant binary relation under $\mathcal{T}_E$. Let $\mathbf{p}_1$ and $\mathbf{p}_2$ are transformed from $\mathbf{P}_1$ and $\mathbf{P}_2$, respectively. The distance

$$(\mathbf{p}_1 - \mathbf{p}_2)^T(\mathbf{p}_1 - \mathbf{p}_2) = (\mathbf{P}_1 - \mathbf{P}_2)^T\mathbf{O}^T\mathbf{O}(\mathbf{P}_1 - \mathbf{P}_2) = (\mathbf{P}_1 - \mathbf{P}_2)^T(\mathbf{P}_1 - \mathbf{P}_2) \tag{17}$$

does not change under $\mathcal{T}_E$. This is due to $\mathbf{O}^T\mathbf{O} = \mathbf{I}$.

Under Similarity Transformations

The length ratio of two lines is an invariant binary relation under $\mathcal{T}_S$. Let two lines I and J be transformed to lines i and j, respectively. Their lengths are scaled by the same factor $|\mathbf{O}| = \det(\mathbf{O})$. The length ratio

$$\frac{len(i)}{len(j)} = \frac{len(I)\ |\mathbf{O}|}{len(J)\ |\mathbf{O}|} = \frac{len(I)}{len(J)} \tag{18}$$

remains unchanged. This is due to the isotropic scaling of $\mathcal{T}_S$.

Another invariant binary relation is the the angle between two lines. The angle can be analyzed via the inner product of the line directions. Let $\mathbf{V}_1$ and $\mathbf{V}_2$ be the two vectors in the directions of the lines, respectively, and $\mathbf{v}_1$ and $\mathbf{v}_2$ are the corresponding vectors after a similarity transformation. The inner product

$$(\mathbf{v}_1, \mathbf{v}_2) = \frac{\mathbf{v}_1^T \mathbf{v}_2}{|\mathbf{v}_1|\ |\mathbf{v}_2|} = \frac{\mathbf{V}_1^T \mathbf{O}^T \mathbf{O} \mathbf{V}_2}{|\mathbf{V}_1|\ |\mathbf{O}|\ |\mathbf{O}|\ |\mathbf{V}_2|} = (\mathbf{V}_1, \mathbf{V}_2) \tag{19}$$

does not change under $\mathcal{T}_S$. Therefore, the angle between lines is a binary invariant.

Three points A, B and C define a triangle. Assume that a triangle $\triangle ABC$ is transformed to $\triangle abc$ under $\mathcal{T}_S$. The two triangles are similar, $\triangle ABC \sim \triangle abc$. This is because each of the three angles remains the same. The three angles give three invariant ternary relations

$$\angle a = \angle A, \quad \angle b = \angle B, \quad \angle c = \angle C \tag{20}$$

The length ratios of every pair of the three sides give another three invariant ternary relations

$$\frac{|\overline{ab}|}{|\overline{AB}|} = \frac{|\overline{cb}|}{|\overline{CB}|}, \quad \frac{|\overline{bc}|}{|\overline{BC}|} = \frac{|\overline{ac}|}{|\overline{AC}|}, \quad \frac{|\overline{ca}|}{|\overline{CA}|} = \frac{|\overline{ba}|}{|\overline{BA}|} \tag{21}$$

where $|\overline{XY}|$ is the length of the side $\overline{XY}$.

Similarity invariants for three-dimensional space curves can be derived based on a representation called the similarity invariant coordinate system (SICS) [8]. Only two curvature extrema of a curve, that are invariant relative to the curve, are necessary to construct an SICS. A reference point $\mathbf{o}$ and a basis $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ can be uniquely determined using the curve segment bounded by the two points under a few mild conditions. All points $\mathbf{p}$ on the curve can be expressed in terms of the SICS $\{\mathbf{o}, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ as

$$\mathbf{p} = \mathbf{o} + u_1 \mathbf{e}_1 + u_2 \mathbf{e}_2 + u_3 \mathbf{e}_3 \tag{22}$$

In this equation, $\mathbf{u} = (u_1, u_2, u_3)$ is the similarity invariant coordinate of $\mathbf{p}$ in the SICS. It can be used as an invariant binary relation, being binary because it is derived based on the two feature points. Other invariants can also be derived based on the SICS.

Under Affine Transformations

The scaling of a shape can be anisotropic under $\mathcal{T}_A$. An equilateral triangle may not remain equilateral after the transform.

Affine-invariant ternary [9] and quaternary [10] relations can be found for planar curves in three-dimensional space projected onto two-dimensional space under weak perspectivity. In [9], each triplet of invariant feature points on a planar curve are transformed to three points $(-1,0)^T$, $(0,1)^T$, and $(1,0)^T$ in a canonic plane. This defines an affine invariant coordinate system. Invariant coordinates can then be computed based on the coordinate system and can be used as a vector of invariant ternary relations.

Divide a line into two segments at an intersecting point. Consider the length of the segments along the direction of the line. The length ratio is preserved under $\mathcal{T}_A$. Two such ratios can be derived from two lines intersecting at the common point. Let $\overline{AB}$ and $\overline{CD}$ be two lines intersecting at E. Assume they are mapped under $\mathcal{T}_A$ to $\overline{ab}$ and $\overline{cd}$ intersecting at e. Then, the ratios

$$\frac{|\overline{ae}|}{|\overline{eb}|} = \frac{|\overline{AE}|S_{AB}}{|\overline{EB}|S_{AB}} = \frac{|\overline{AE}|}{|\overline{EB}|} \tag{23}$$

and

$$\frac{|\overline{ce}|}{|\overline{eb}|} = \frac{|\overline{CE}|S_{CD}}{|\overline{ED}|S_{CD}} = \frac{|\overline{CE}|}{|\overline{ED}|} \tag{24}$$

together give two invariant quaternary relations. In these equations, S_{AB} and S_{CD} are scaling factors in the directions of the lines $\overline{AB}$ and $\overline{CD}$, respectively; for example, $|\overline{ae}| = |\overline{AE}|S_{AB}$. The ratios are invariant due to the fact that the scaling factor is constant along the same direction.

The ratio of the areas of two triangles is also preserved under $\mathcal{T}_A$. Four non-collinear points in a plane define four triangles. The area ratios of the first three triangles to the last one give three invariant quaternary relations.

Under Perspective Projections

Cross-ratio is preserved under perspective projections, although the ratio of distances is not. The cross-ratio is the ratio of ratios of distances. The preservation of cross-ratio is of fundamental importance under general perspective projections. A cross-ratio for four points A, B, C, and D on a line is defined by $\frac{|\overline{AC}|\times|\overline{BD}|}{|\overline{BC}|\times|\overline{AD}|}$. Assume that they are respectively mapped to a, b, c, and d. The following equality of cross-ratio holds

$$\frac{|\overline{ac}| \times |\overline{bd}|}{|\overline{bc}| \times |\overline{ad}|} = \frac{|\overline{AC}| \times |\overline{BD}|}{|\overline{BC}| \times |\overline{AD}|} \tag{25}$$

Therefore, the cross-ratio gives an invariant quaternary relation. The cross-ratio is so essential that many invariants under projective projections can be derived based on it. The interested reader may refer to [2] for recent studies of projective invariants in vision.

Spectrum of Invariants

An invariant can be algebraic, such as an angle or a distance, or differential such as curvature or torsion. These are the two ends in the spectrum of invariants. Invariants for shape patterns of linear features, such as isolated points and lines, must be algebraic; differential properties are not useful in this case. Descriptions of shapes of nonlinear features, such as freeform curves and surfaces, usually have to include differential information in an infinitesimal neighborhood. Differential invariants are local descriptors and hence, are stable under occlusion, whereas algebraic invariants are not.

As the class of transformations becomes more complex from $\mathcal{T}_E$ to $\mathcal{T}_P$, higher order derivatives have to be used to derive differential invariants [11] and a larger number of features have to be used to define algebraic invariants. The higher the orders of algebraic invariants, the more complicated the matching process becomes. The higher the order of derivatives, the more unstable the computation of differential invariants becomes.

Semi-differential invariants [12] offer a compromise between the two extrema. They are derived after trading the order of derivatives for extra points. For example, instead of using the curvature at a single point, one may use angles between two tangents; the curvature involves the second order derivative at one point, whereas the tangents involve the first order derivatives. This also trades locality for stability.

Discovery of invariants are usually difficult. However, if the transformations form a group, systematic methods can be used for constructing invariants [3, 2].

MATCHING BASED ON INVARIANTS

In this section, a representation called attributed relational structure (ARS) is introduced to represent model objects and scenes in input images. Matching of a scene and an object is carried out between two ARS's, one for the scene and the other for the model object. An optimality criterion is defined as the measure of the goodness of ARS matching.

The ARS Representation

A scene or an object can be abstracted by a set of features. These features are not isolated; rather, they are constrained to each other by contextual relations. The features and contextual relations can be represented in a systematic way by a so-called attributed relational structure (ARS) [13, 14, 15, 16, 17]. If all relations are invariants, the ARS is an invariant encapsulation of a shape, an object, or a scene.

When up to N^{th} order relations are considered, an ARS is an $N+1$

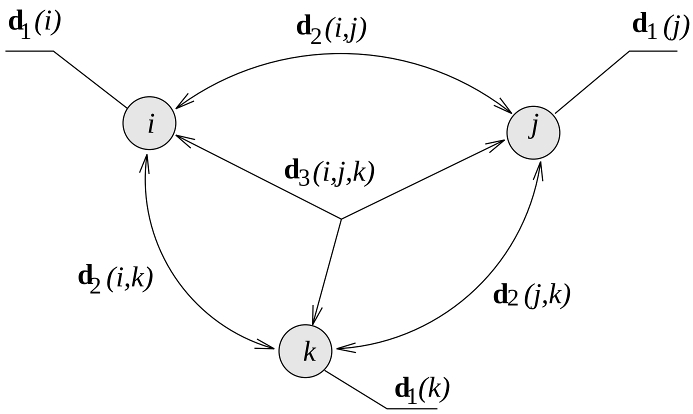

FIGURE 1: Nodes and relations in the ARS representation (Reproduced from [17] with permission).

tuple. The scene ARS is denoted by

$$\mathbf{g} = (\mathbf{d}, \mathbf{r}_1, \ldots, \mathbf{r}_N) \tag{26}$$

or $\mathbf{g} = (\mathbf{d}, \mathbf{r})$ in brief. In this notation,

$$\mathbf{d} = \{1, \ldots, m\} \tag{27}$$

is a set of m scene nodes each of which indexes a feature in the scene;

$$\mathbf{r}_n = \{\mathbf{r}_n(i_1, \ldots, i_n) \mid \forall i_1, \ldots, i_n \in \mathbf{d}\} \tag{28}$$

is the set of n-ary $(1 \le n \le N)$ properties and relations;

$$\mathbf{r}_n(i_1, \ldots, i_n) = [r_n^{(1)}(i_1, \ldots, i_n), \ldots, r_n^{(K_n)}(i_1, \ldots, i_n)]^T \tag{29}$$

is a vector of K_n relations of order n defined among nodes $i_1, \ldots, i_n$. The set of m nodes are therefore attributed by its properties or unary relations and are constrained to each other by inter-relations.

Examples of unary relations are color and size. Examples of binary relations are the distance between two points and the angle between two lines. An example of ternary relations is the vector of the three angles of a triangle. An example of quaternary relations is the two cross ratios of two intersecting lines.

Figure 1 shows three nodes, i, j and k, of an ARS, and their unary, binary and ternary relations. Note that $\mathbf{r}_1(i)$, $\mathbf{r}_2(i,j)$ and $\mathbf{r}_3(i,j,k)$ are vectors of relations with K_1, K_2 and K_3 components, respectively.

A model object is similarly represented by $\mathbf{G} = (\mathbf{D}, \mathbf{R}_1, \ldots, \mathbf{R}_N)$ in which

$$\mathbf{D} = \{1, \ldots, M\} \tag{30}$$

is a set of M model nodes and $\mathbf{R}_n$ is a set of K_n relations of order n $(1 \le n \le N)$. Now introduce a virtual feature, called the NULL and indexed by 0. The NULL node represents everything not modeled by $\mathbf{G}$, such as features due to all other models and noise. Let

$$\mathbf{D}^+ = \{0\} \cup \mathbf{D} = \{0, 1, \ldots, M\} \tag{31}$$

denote the augmented set of object nodes. It is the set of labels to be assigned to the scene features in $\mathbf{d}$. The set of all relations are also augmented so that any relation involving the NULL node assumes a special value, NONE . Now, the model ARS becomes $\mathbf{G}(\mathbf{D}^+, \mathbf{R}^+)$.

Structural Mapping

A matching is denoted by

$$f = (f_1, \ldots, f_m) \tag{32}$$

where $f_i \in \mathbf{D}^+$. It is a mapping from $\mathbf{d}^m$ to $(\mathbf{D}^+)^m$. In terms of the labeling problem, it assigns a label from $\mathbf{D}^+$ to each site in $\mathbf{d}$. Because it maps each node in $\mathbf{d}$ to a node in $\mathbf{D}^+$, it thus maps $\mathbf{r}_n(i_1, \ldots, i_n)$ to $\mathbf{R}_n^+(f_{i_1}, \ldots, f_{i_n})$ and $\mathbf{g}$ to $\mathbf{G}$.

Such structural mapping is called a *morphism*. A morphism is called an *isomorphism* if the mapping bijective. It is called a *monomorphism* if it is just injective. It is called a *homomorphism* if it is many-to-one. One-to-many mappings are not allowed because the one-to-many matching not only increases the complexity of the problem but also contradicts the definition of a mapping. The mapping discussed in this chapter is homomorphism. All nodes in $\mathbf{d}$ due to noise and un-modeled object features should be mapped to the NULL in $\mathbf{D}^+$.

The solution space of f is

$$\mathbf{S} = \underbrace{\mathbf{D}^+ \times \mathbf{D}^+ \cdots \times \mathbf{D}^+}_{m \text{ times}} \tag{33}$$

where "$\times$" is the Cartesian product of sets. Any mapping $f \in \mathbf{S}$ is a feasible solution to the problem[1]. Yet only a few are good in the sense of satisfying the shape constraints. The goodness is measured by a criterion functional $G(f)$ called the gain. The optimal mapping maximizes the gain

$$f^* = \arg\max_{f \in \mathbf{S}} G(f) \tag{34}$$

[1]This holds if all $f_i \in \mathbf{D}^+$ are feasible labels for any $i \in \mathbf{d}$. In some applications, however, matches must satisfy certain unary symbolic constraints such that only some subsets of $\mathbf{D}^+$ are feasible labels for some i.

The Gain Functional

The gain functional is defined as a weighted sum of component gains over all orders (for $n = 1, \ldots, N$). Each component gain has two complimentary terms. In the simplest case, when only the first order constraints are accounted for, the corresponding gain component is

$$G'_1(f) = \sum_{i \in \mathbf{d}: f_i = 0} \Lambda_{10}(f_i) + \sum_{i \in \mathbf{d}: f_i \neq 0} \Lambda_1(i, f_i) \tag{35}$$

The first term Λ_{10} involves only NULL matches and the second term Λ_1 only non-NULL matches. In the case of second order constraints,

$$G'_2(f) = \sum_{(i,j) \in \mathbf{d}^2: f_i = 0 \text{ or } f_j = 0} \Lambda_{20}(f_i, f_j) + \sum_{(i,j) \in \mathbf{d}^2: f_i \neq 0 \text{ and } f_j \neq 0} \Lambda_2(i, f_i; j, f_j) \tag{36}$$

where Λ_{20} involves pairs of matches in which at least one is the NULL and Λ_2 involves pairs of purely non-NULL matches. In general, for n^{th} order constraints

$$\begin{aligned} G'_n(f) \quad = \quad & \sum_{(i_1,\ldots,i_n) \in \mathbf{d}^n: f_{i_j} = 0 \ \exists j} \Lambda_{n0}(f_{i_1}, \ldots, f_{i_n}) + \\ & \sum_{(i_1,\ldots,i_n) \in \mathbf{d}^n: f_{i_j} \neq 0 \ \forall j} \Lambda_n(i_1, f_{i_1}; \ldots; i_n, f_{i_n}) \end{aligned} \tag{37}$$

The global gain is defined as

$$G(f) = \sum_{n=1}^{N} \lambda_n G'_n(f) \tag{38}$$

where λ_n are weights.

The second term Λ_n in G'_n measures relational similarity of order n

$$\Lambda_n(i_1, f_{i_1}; \ldots; i_n, f_{i_n}) = g(\|\mathbf{r}_n(i_1, \ldots, i_n) - \mathbf{R}_n(f_{i_1}, \ldots, f_{i_n})\|) \tag{39}$$

In this equation, $\|\cdot\|$ is a metric, $\|\mathbf{r}_n - \mathbf{R}_n\|$ measures the distance between $\mathbf{r}_n$ and $\mathbf{R}_n$ and $g(\|\mathbf{r}_n - \mathbf{R}_n\|)$ is a function that measures the similarity between the two vectors. The metric can be a weighted distance

$$\|\mathbf{r}_n - \mathbf{R}_n\| = \sum_{k=1}^{K_n} [r_n^{(k)} - R_n^{(k)}]^2 / \sigma_n^{(k)} \tag{40}$$

where $\sigma_n^{(k)} > 0$ $(k = 1, \ldots, K_n)$ are some parameters. The function $0 < g(\rho) < +\infty$ $(\rho \geq 0)$ satisfies the following:

1. It has the maximum value at $\rho = 0$, *i.e.* , $g(0) = \max_\rho g(\rho)$.

2. It monotonously decreases as ρ increases, *i.e.* $g'(\rho) < 0$.

3. It has the asymptote of 0, *i.e.* $\lim_{\rho \to +\infty} g(\rho) = 0$.

Suitable choices of g include

$$g(\rho) = e^{-\rho} \tag{41}$$

$$g(\rho) = e^{-\rho^2} \tag{42}$$

and

$$g(\rho) = 1/(1 + \rho^2) \tag{43}$$

Given the assumption that quantitative relations obey Gaussian distributions, the first choice is the best and this gives

$$g(\|\mathbf{r}_n - \mathbf{R}_n\|) = \exp(-\sum_{k=1}^{K_n} [r_n^{(k)} - R_n^{(k)}]^2 / \sigma_n^{(k)}) \tag{44}$$

When a relation $r_n^{(k)}$ is symbolic, one may set $\sigma_n^{(k)}$ to 0^+ (a small positive number), such that $\exp(-[r_n^{(k)} - R_n^{(k)}]^2/\sigma_n^{(k)})$ is 1 or 0 depending on whether $r_n^{(k)}$ and $R_n^{(k)}$ are the same or not.

The first terms Λ_{n0} is complementary to the first term; it is incurred by n-tuples of simultaneous labels $(f_{i_1}, \ldots, f_{i_n})$ at least one of which is the NULL . It is defined by

$$\Lambda_{n0}(f_{i_1}, \ldots, f_{i_n}) = \begin{cases} H_{n0} & \exists f_{i_j} = 0 \\ 0 & \text{otherwise} \end{cases} \tag{45}$$

where H_{n0} is a constant.

Because Λ_{n0} and Λ_n are complementary, they can be combined into a compact form

$$G_n(i_1, f_{i_1}; \ldots; i_n, f_{i_n}) = \begin{cases} \Lambda_{n0}(f_{i_1}; \ldots; f_{i_n}) & \exists f_{i_j} = 0 \\ \Lambda_n(i_1, f_{i_1}; \ldots; i_n, f_{i_n}) & \text{otherwise} \end{cases} \tag{46}$$

In terms of relaxation labeling, $G_n(i_1, f_{i_1}; \ldots; i_n, f_{i_n})$ is the numerical compatibility of the n matches $(i_1 \to f_{i_1})$, $\cdots$, $(i_n \to f_{i_n})$. Now the global gain incurred by f is written as

$$G(f) = \sum_{n=1}^{N} \lambda_n \sum_{(i_1, \ldots, i_n) \in \mathbf{d}^n} G_n(i_1, f_{i_1}; \ldots; i_n, f_{i_n}) \tag{47}$$

The optimal f is a combination of $(f_1, \ldots, f_m)$ which maximizes the above functional.

Let us illustrate $G_n(i_1, f_{i_1}; \ldots; i_n, f_{i_n})$ using some examples. First, consider unary compatibilities. Assume that each node corresponds to a region and has its area as the unary relation. Then the compatibility for the match $(i \to f_i)$ is $G_1(i, f_i) = e^{-[area(i) - area(f_i)]^2/\sigma}$ if the similarity function is Equation (42). It has the maximum when the two areas are the same. Assume that each region has its color as the unary relation and let $\sigma = 0^+$, then $G_1(i, f_i) = e^{-[color(i) - color(f_i)]^2/\sigma}$ takes a value 1 or 0 depending whether the two colors are the same. Second, consider binary compatibilities. Assume that each node corresponds to a line and each pair of nodes have their angle as the binary relation. The compatibility for the pairs of matches $(i \to f_i)$ and $(j \to f_j)$ is $G_2(i, f_i; j, f_j) = e^{-[angle(i,j) - angle(f_i, f_j)]^2/\sigma}$. The constraints of order three and higher can be encoded in a similar way.

Matching to Multiple Objects

Suppose there are L potential model objects, with object $l \in \{1, \ldots, L\}$ having M_l labels. Let the scene be matched, in turn, against each of the objects by maximizing the corresponding global gain functional, as previously described. The result of matching to model l can be expressed by a set of m quadruples

$$\{(i, l, f_i(l), q_i(l)) \mid \forall i \in \mathbf{d}\} \tag{48}$$

In the above, $[f_1(l), \ldots, f_m(l)]$ denotes the optimal labeling of the m scene features in terms of object l; and $q_i(l)$ is the support of the match $(i \to f_i(l))$ from the other matches.

The final result of matching over all objects is also represented by a set of m quadruples

$$\{(i, l_i, f_i, q_i) \mid \forall i \in \mathbf{d}\} \tag{49}$$

Unlike in Equation (48), the object index l_i in Equation (49) is a function of i instead of a constant. This index is chosen among all $l \in \{1, \ldots, L\}$ to maximize the local support

$$l_i = \arg \max_{l \in \{1, \ldots, L\}} q_i(l) \tag{50}$$

The corresponding quadruple (i, l_i, f_i, q_i) is obtained by letting $f_i = f_i(l_i)$ and $q_i = q_i(l_i)$.

SEARCHING FOR OPTIMAL SOLUTION

Searching for the solution f^* in Equation (34) is a combinatorial optimization problem. Owing to the nature of weighted matching, methods for symbolic matching, such as maximal cliques [18], dynamic programming [19] and constraint search [20, 21], can not be efficiently applied. This is because the efficiency of search in these methods is ascribed to the pruning using symbolic constraints that quickly narrows the scope of search. However, weighted matching can not take this advantage.

Therefore, we turn to relaxation methods. There are basically two categories of relaxation algorithms that can be used to solve the optimization problem: stochastic ones, such as simulated annealing or Boltzmann machines [22, 23, 24], and deterministic ones, such as relaxation labeling (RL) [25, 26, 27]. The stochastic algorithms are proven to find the global solution with probability approaching one but the computational costs currently prevent them from being of practical use. The deterministic algorithms are more efficient but they are by nature local optimizers; the results depend on the property of the functional to be optimized and the initialization.

Hopfield Neural Networks (HNN) [28, 29] have been proposed as yet another deterministic relaxation mechanism for solving a variety of optimization problems. Before the Hopfield networks can be applied to the matching problem, additional terms have to be added to the global criterion to impose the admissibility of the solution.

We have chosen to use a deterministic RL algorithm, more specifically, the Hummel-Zucker algorithm [27] although the Faugeras algorithm [26] is reported to perform just as well [30]. The RL algorithms are more favorable than the Hopfield algorithm for reasons that will be given later.

Relaxation Labeling

The original problem is to find, combinatorially, a homomorphic mapping f^* from $\mathbf{d}^m$ into $(\mathbf{D}^+)^m$ that optimizes the gain functional $G(f)$. In the continuous RL, all combinations are possible, but the possibility of most of them is quickly reduced as the system evolves using contextual information. This forms a trajectory. The endpoint of the trajectory is the final solution.

Problem Reformulation

The confidence of a match $i \to f_i$ is measured by a number in $[0,1]$. Let $I = f_i$. Denote the confidence of $i \to I$ by

$$f(i,I) \in [0,1] \tag{51}$$

and the state of labeling by [2]

$$f = \{f(i,I) \mid i \in \mathbf{d}; I \in \mathbf{D}^+\} \tag{52}$$

The following *consistency* constraint must be satisfied in RL:

$$\sum_{I \in \mathbf{D}^+} f(i,I) = 1 \qquad \forall i \in \mathbf{d} \tag{53}$$

The admissible space of the continuous labeling is

$$\mathbf{S}' = [0,1]^{\mathbf{d} \times \mathbf{D}^+} \tag{54}$$

[2] Without confusion, we will use the same notation f to denote both the mapping function and the confidences.

However, the final solution f^* is required to satisfy the following unambiguity constraint:

$$f^*(i,I) \in \{0,1\} \tag{55}$$

with $f^*(i,I) = 1$ meaning i is unambiguously mapped to I by f^*. Therefore the admissible space of final solution is

$$\mathbf{S}'' = \{0,1\}^{\mathbf{d}\times\mathbf{D}^+} \tag{56}$$

$\mathbf{S}''$ consists of the "corners" of $\mathbf{S}'$. Points in the set difference $\mathbf{S}' - \mathbf{S}''$ are "bridges" to the corners. In terms of $f(i,I)$, the gain functional is redefined by

$$G(f) = \sum_{n=1}^{N} \lambda_n \sum_{\forall (i_1,\ldots,i_n)\in\mathbf{d}^n} G_n(i_1,I_1;\ldots;i_n,I_n) f(i_1,I_1)\cdots f(i_n,I_n) \tag{57}$$

Algorithm

Given an initial labeling, an RL algorithm iteratively updates the labeling towards a maximal un-ambiguous labeling. Introduce a time parameter into f such that

$$f = f^{(t)} \tag{58}$$

At time t, the gradient function $q^{(t)} = \nabla G(f^{(t)})$ is computed and $f^{(t)}$ is updated according to a rule

$$f^{(t+1)} \leftarrow f^{(t)} + \Psi(f^{(t)}, q^{(t)}) \tag{59}$$

where Ψ is an operator. This updates f by an appropriate magnitude in the appropriate direction. In the Hummel-Zucker algorithm [27], f is updated by

$$f^{(t+1)} \leftarrow f^{(t)} + \eta u \tag{60}$$

where u is the direction vector computed by the gradient projection (GP) and η is a factor that ensures that the updated vector $f^{(t+1)}$ to be within the space $\mathbf{S}'$. The convergence to $\mathbf{S}''$ is generally guaranteed by the algorithm [27].

The components of the gradient is computed as

$$q(i,I) = \frac{\partial G(f)}{\partial f(i,I)} = \sum_{n=1}^{N} \lambda_n q_n(i,I) \tag{61}$$

From Equation (57), the following can be obtained

$$q_1(i,I) = G_1(i,I) \tag{62}$$

and

$$q_2(i,I) = \sum [G_2(i,I;j,J) + G_2(j,J;i,I)] f(j,J) \tag{63}$$

In the case of symmetry, *i.e.* $G_2(i,I;j,J) = G_2(j,J;i,I)$, then

$$q_2(i,I) = 2\sum G_2(i,I;j,J)f(j,J) \tag{64}$$

With the symmetry, the following general expression can be obtained

$$q_n(i_l,I_l) = n\sum G_n(i_1,I_1;\ldots;i_n,I_n) \quad f(i_1,I_1)\ldots f(i_{l-1},I_{l-1}) \\ f(i_{l+1},I_{l+1})\ldots f(i_n,I_n) \tag{65}$$

The interested reader may refer to [27] for more details of the algorithm and its convergence properties.

The computation is expensive for sequential implementation of the RL. The main consumption is in computing the gradient and performing the gradient projection operation. However, relaxation algorithms are inherently parallel, which is an advantage for real-time implementation. Due to the parallel nature of the algorithm, the computation of gradient can be done in parallel for all $i = 1,\ldots,m$ and $I = 0,1,\ldots,M$; the gradient projection operation can be performed in parallel for all $i = 1,\ldots,m$ (or $I = 0,1,\ldots,M$). Thus the algorithm could run efficiently on parallel architectures like the Connectionist Machines. See [31] for a discussion of some issues on a parallel distributed implementation of the Hummel-Zucker algorithm for a multiprocessor architecture environment. In parallel implementation, the time complexity depends mainly on the number of iterations required for the algorithm to converge. The expectation of this number is dependent on the ARS sizes m and M and, according to our observations, is a low order polynomial function of m and M.

Local versus Global Optimization

Relaxation algorithms are local optimizers by nature. They do not necessarily find the global optimum. Two factors affect the solution: The initial assignment ($f^{(0)}$) and the interaction function.

It is shown that for certain choices of interaction functions G_2, the RL algorithm can have a *single* convergence point regardless of the initial assignment [32, 33]. In experiments, we initialize $f^{(0)}$ at random and have made the following observations. The less distortion of features and occlusion of objects, the less the result is dependent on the initial assignment. As one extreme, in the case of no distortion or occlusion, that is, when the scene ARS and the model ARS are identical ($\mathbf{d} = \mathbf{D}, \mathbf{r}_1 = \mathbf{R}_1, \mathbf{r}_2 = \mathbf{R}_2$), the solution is always found to be $I = f_i = i$ regardless of the initial assignment. As values in $\mathbf{r}$ are distorted from $\mathbf{R}$ to a considerable extent, the dependence of the result on the initial assignment appear. The more distortion, the more frequently this dependency emerges. The same is true of occlusion.

From the previous observations, we conjecture the following. Assume the optimal solution exists uniquely. Then when there is no distortion or occlusion, for our choice of the interaction function, the RL algorithm has a single convergence point regardless of the initial assignment. Based on this conjecture, we can infer that there exists an extent of allowable distortion and occlusion within which the RL has a single convergence point regardless of the initial assignment. This is because given $\mathbf{G}$, the interaction function continuously depends on the measurements $\mathbf{r}$.

Hopfield Neural Network

Hopfield-style Neural Network (HNN) [28, 29] has been used for image matching and object recognition. Jamison and Schalkoff [34] use a Hopfield-like network for image labeling in which a scene is described by related symbols. Nasrabadi *et al.* [35] use a Hopfield network to perform subgraph isomorphism for matching overlapping two-dimensional objects. Lin *et al.* [36] use Hopfield network for matching the scene to multiple view models. Peterson and Soderber [37] present an alternative solution. They use *graded neurons* to represent a mapping to more than two possible labels and perform the minimization using the mean field theory equation.

Problem Reformulation

The original problem is the constrained *maximization*. This is converted into constrained *minimization* to suit the HNN's notion. An energy functional $E(f) = \sum_n \lambda_n E_n(f)$, where $E_n = -G_n$, is to be minimized with respect to $f = \{f(i,I)\}$ subject to the two global constraints in Equation (53) and Equation (55). The HNN itself does not take care of any of the constraints. The HNN approaches impose the constraints forcibly by adding other terms to the functional. This is crucial.

The un-ambiguity constraint can be imposed by the following term

$$E_a(f) = \sum_{(i,I)} \int_0^{f(i,I)} \theta^{-1}(f) df \tag{66}$$

where θ^{-1} is the inverse of a function θ, shown in the following illustrations. A local state $f(i,I) \in [0,1]$ is related to an *internal variable* $u(i,I) \in (-\infty, +\infty)$ by

$$f(i,I) = \theta(u(i,I)) \tag{67}$$

where $\theta(u)$ is usually a sigmoid function

$$\theta(u) = 1/[1 + e^{-u/T}] \tag{68}$$

controlled by a parameter T. In a very high gain when $T \to 0^+$, $f(i,I)$ is forced to be 0 or 1 depending on whether $u(i,I)$ is positive or negative [29]. The term $E_a(f)$ reaches the minimum value of zero only when all $f(i,I)$'s are either 0 or 1. This means minimizing this term under $T \to 0^+$

in effect leads to unambiguous labeling. On the other hand, the consistency constraints can be imposed by

$$E_b(f) = \sum_i [\sum_I f(i,I) - 1]^2 = 0 \tag{69}$$

This term has its minimum value of zero when the consistency constraint in Equation (53) is satisfied. Now, the constrained optimization minimizes the following functional

$$E'(f \mid T) = E(f) + aE_a(f \mid T) + bE_b(f) \tag{70}$$

where a and b are weights. The energy functional $E'(f \mid T)$ is parameterized by T and approaches $E(f)$ when T approaches 0^+: $\lim_{T\to 0^+} E'(f \mid T) = E(f)$.

Algorithm

The energy is minimized according to a dynamic system. Introduce a time variable into f such that $f = f^{(t)}$. The energy change, due to state change $\frac{df(i,I)}{dt}$, is

$$dE'/dt = -\sum_{(i,I)} \frac{df(i,I)}{dt} \quad \Big\{E_1(i,I) + 2\lambda \textstyle\sum_{(j,J)} E_2(i,I;j,J)f(j,J) - a\, u(i,I) - b[\sum_J f(j,J) - 1]\Big\} \tag{71}$$

where only up to the second order constraints are accounted for. The following dynamic system minimizes the three term energy E':

$$C du(i,I)/dt = E_1(i,I) + 2\lambda \sum_{(j,J)} E_2(i,I;j,J)f(j,J) - a\, u(i,I) - b[\sum_J f(j,J) - 1] \tag{72}$$

where capacitance $C > 0$ controls the convergence of the system. With these dynamics, the energy change

$$\frac{dE'}{dt} = -C\sum_{(i,I)} \frac{df(i,I)}{dt}\frac{du(i,I)}{dt} = -C\sum_{(i,I)} \frac{d\theta^{-1}(f(i,I))}{df(i,I)}[\frac{df(i,I)}{dt}]^2 \tag{73}$$

is non-positive because θ^{-1} is a monotonously increasing function and C is positive. This updating rule will lead to a solution that is unambiguous and consistent if the parameters T, a, and b are chosen properly.

Relaxation Labeling versus Hopfield Network

Minimizing $E'(f \mid T)$ with nonzero T does not necessarily give the same optimum as minimizing $E(f)$. To solve the original problem, T must be gradually reduced to 0^+. However, unless the graduation is purposely performed as in the convex approximation in the GNC [38], the final solution may not be near the global one because of the local optimum problem.

The RL algorithm is much more careful in this regard. It uses gradient projection to choose the best direction and magnitude for the state to evolve. Our experience from experiments is that the HNN algorithm has more difficulties to find the correct solution than the RL. In other words, the problem of local optimal is more significant in the neural approaches. The HNN approaches often result in an unfavorable solution while the RL has a good behavior. Computationally, the RL needs a smaller number of iterations before convergence.

EXPERIMENTS

Two experiments are presented. The first experiment is that of the matching of line pattern. Figure 2(a) shows the model of a line pattern. Figure 2(b) is composed of a scaled, rotated, translated, and noisy subset of the model lines and some spurious lines. It is enlarged to the size of (c) and used as a scene to be matched. Figure 2(c) shows matched model lines (in solid lines) aligned with the scene, in which spurious lines are identified in dashed lines.

The following second order invariant relations are used

1. $r_2^{(1)}(i,j) = angle(i,j)$
2. $r_2^{(2)}(i,j) = \log\left(\frac{len(i)}{len(j)}\right)$
3. $r_2^{(3)}(i,j) = \log\left(\frac{dist_{mid}(i,j)}{dist_{max}(i,j)}\right)$

where $len(i)$ is the length of the line segment i, $dist_{mid}(i,j)$ and $dist_{max}(i,j)$ are the distance between the midpoints of the two line segments and the furthest distance between the endpoints of the line segments, respectively. The ratios are invariants under $\mathcal{T}_S$. The log function is used to convert the ratio of the ratio into the difference of the ratio. For example, the most meaningful comparing factor of the length ratios is $\frac{len(i)}{len(j)} / \frac{len(I)}{len(J)}$. This is converted to the difference $\log\left(\frac{len(i)}{len(j)}\right) - \log\left(\frac{len(I)}{len(J)}\right)$ to suit the form of similarity measures.

The second experiment is that of the matching of three-dimensional space curves. A similarity invariant coordinate system (SICS) [8] can be constructed from a curve segment bounded by two invariant feature points that correspond to two curvature maxima (cf. Section 2). Assume there are a number of such points detected on a curve. Each pair of such points determine a curve segment and thus a SICS. A vector of invariant coordinates can be formed for each curve segment. Let each feature point be

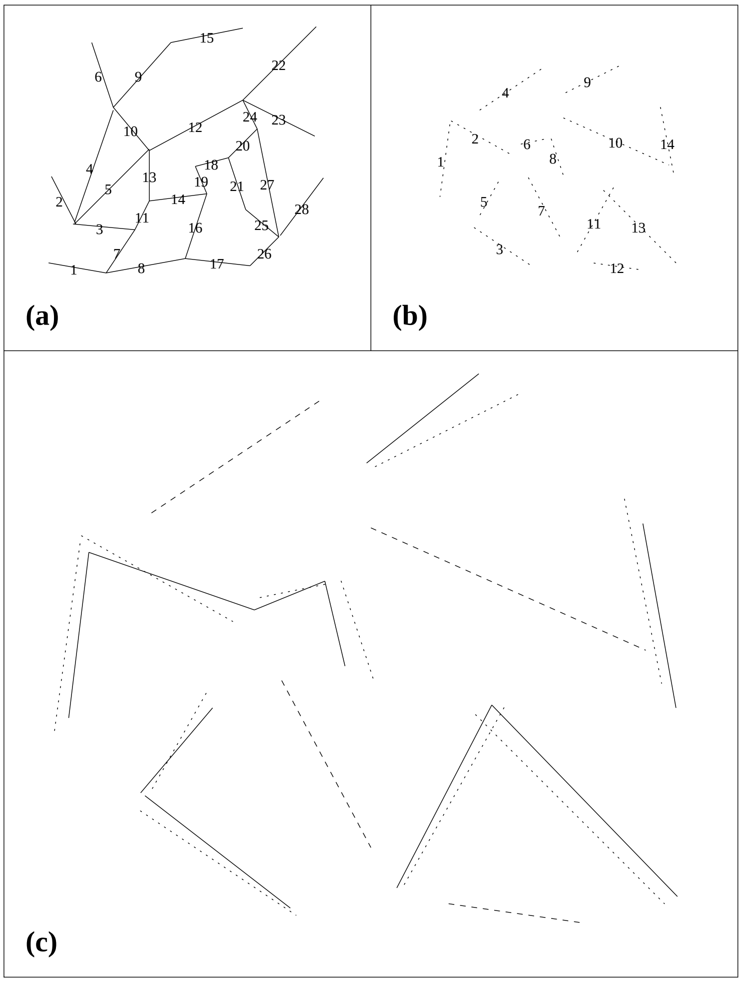

FIGURE 2: Matching of a line pattern under two-dimensional similarity transformation. (Reproduced from [17] with permission)

denoted by a node and regard each vector of the invariant coordinates as the relational vector between the involved points. Then an ARS can be constructed as an invariant encapsulation of the curve. Now the matching problem can be done by ARS matching.

There are 16 synthetic model curves in this experiment (not shown. Figure 3 shows a scene that is composed of rotated, translated, and scaled subparts of three of the model curves with uniformly distributed noise added, and the matched curves aligned with the scene curve.

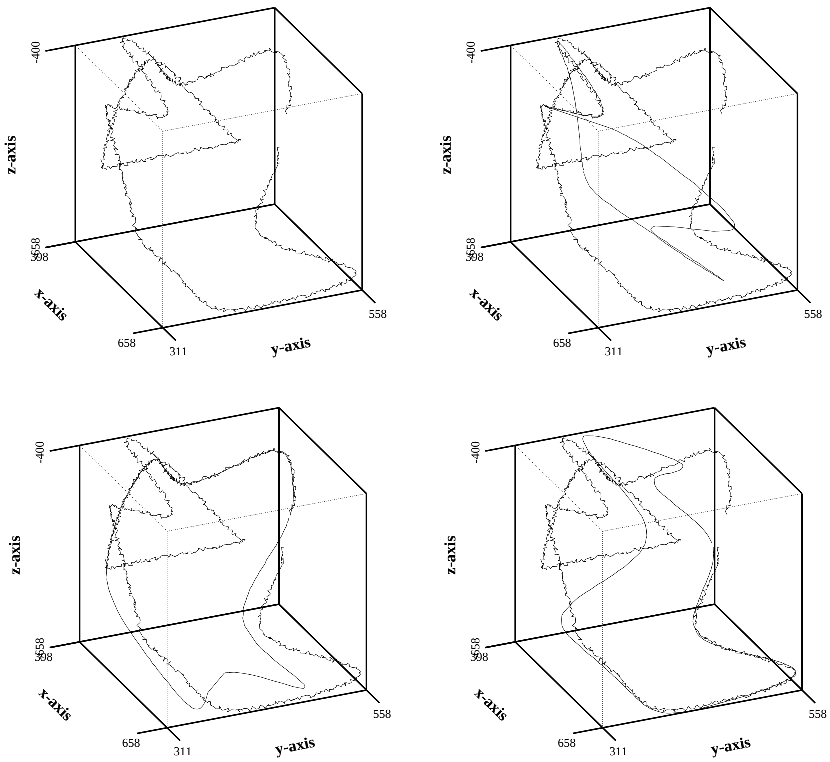

FIGURE 3: A result of matching. Note: From left to right, top to bottom are: (a) the scene curve, (b) a first model curve (shown in smooth curve) matched with scale factor 2.0 and aligned with the scene curve (in noisy curve), (c) a second model curve matched with scale factor 0.5 and aligned and (d) a third model curve matched with scale factor 1 and aligned.

CONCLUSION

This chapter has described some basic shape invariants and presented an optimization approach for shape matching based on invariants. Invariant-based shape representations have an advantage of making direct shape comparison possible before transformation parameters are found. The ARS representation is used to encapsulate various types of invariants in a systematic way and shape matching is reduced to ARS matching. A gain functional is defined to measure the goodness of matching. Algorithms for finding the optimal matching are discussed.

In choosing between algebraic and differential invariants, one needs to consider the following: Algebraic invariants are defined on more than one feature. As the complexity of transformation increases, a larger number of features are required to define an invariant. This increases the complexity of matching. On the other hand, differential invariants are computed from derivatives. As the complexity of transformation increases, higher orders of derivatives are required. This causes instability.

To compromise, semi-invariants are proposed to compromise between the required number of features and the required order of derivatives. This is about the representation. Another possible compromise may be between using invariants and computing transformation parameters during the matching process. One may use invariants of low orders to help compute transformation parameters while performing invariance-based matching.

Recent progress in Markov random field (MRF) based matching models [39, 40] shows that after some modification, the two terms in $G_n'(f)$ can correspond to the joint prior probability and joint likelihood density of a MRF, respectively. Maximizing $G(f)$ is then equivalent to maximizing the posterior probability. Another development is made for improving Hopfield method in energy minimization [41]. There, Lagrange multipliers are incorporated into HNN to stabilize the minimization process, so that both the convergence and solution quality are improved.

Acknowledgement This work was supported by NTU AcRF projects RG 43/95 and RG 51/97.

References

[1] D. Forsyth, J. L. Mundy, and A. Zisserman. "Transformational invariance – a primer. In *Proceedings of the British Machine Vision Conference*, pages 1–6, 1990.

[2] J. L. Mundy and A. Zisserman, editors. *Geometric Invariants in Computer Vision.* MIT Press, Cambridge, MA, 1992.

[3] K. Kanatani. *Group Theoretical Methods in Image Understanding.* Springer-Verlag, Berlin, 1990.

[4] R. Lenz. *Group Theoretical Methods in Image Processing.* Springer-Verlag, Berlin, 1990.

[5] M. Lipschutz. *Differential Geometry.* McGraw-Hill, 1969.

[6] P. J. Besl and R. C. Jain. "Three-Dimensional object recognition". *Computing Surveys*, 17(1):75–145, March 1985.

[7] S. Z. Li. "Invariant surface segmentation through energy minimization with discontinuities". *International Journal of Computer Vision*, 5(2):161–194, 1990.

[8] S. Z. Li. "Invariant representation, recognition and pose estimation of 3d space curved under similarity transformations". *Pattern Recognition*, 30(3), 1997.

[9] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson. "Object recognition by affine invariant matching". In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 335–344, 1988.

[10] C.A. Rothwell, A. Zisserman, A. Forsyth, and J.L. Mundy. "Canonical frames for planar object recognition". In *Proceedings of the European Conference on Computer Vision*, pages 757–772, 1992.

[11] I. Weiss. "Projective invariants of shapes". In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 291–297, 1988.

[12] L. Van Gool, P. Kempenaers, and A. Oosterlinck. "Recognition and semi-differential invariants". In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 454–460, 1991.

[13] L. G. Shapiro and R. M. Haralick. "Structural description and inexact matching". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:504–519, September 1981.

[14] A. Sanfeliu and K. S. Fu. "A distance measure between attributed relational graphs for pattern analysis". *IEEE Transactions on Systems, Man and Cybernetics*, SMC-9:757–768, 1983.

[15] M. A. Eshera and K. S. Fu. "A graph distance measure for image analysis". *IEEE Transactions on Systems, Man and Cybernetics*, SMC-14(3):398–408, May/June 1984.

[16] A. K. C. Wong, S. W. Lu, and M. Rioux. "Recognition and shape synthesis of 3D objects based on attributed hypergraph". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(3):695–703, March 1989.

[17] S. Z. Li. "Matching: invariant to translations, rotations and scale changes". *Pattern Recognition*, 25(6):583–594, June 1992.

[18] A. P. Ambler, H. G. Barrow, C. M. Brown, R. M. Burstall, and R. J. Popplestone. "A versatile computer-controlled assembly system". In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 298–307, 1973.

[19] M. Fischler and R. Elschlager. "The representation and matching of pictorial structures". *IEEE Transactions on Computers*, C-22:67–92, 1973.

[20] O. D. Faugeras and M. Hebert. "The representation, recognition and locating of 3D objects". *International Journal of Robotic Research*, 5(3):27–52, Fall 1986.

[21] W. E. L. Grimson and T. Lozano-Prez. "Localizing overlapping parts by searching the interpretation tree". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):469–482, April 1987.

[22] S. Kirkpatrick, C. D. Gellatt, and M. P. Vecchi. "Optimization by simulated annealing". *Science*, 220:671–680, 1983.

[23] S. Geman and D. Geman. "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[24] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. "A learning algorithm for Boltzmann machines". *Cognitive Science*, 9:147–169, 1985.

[25] A. Rosenfeld, R. Hummel, and S. Zucker. "Scene labeling by relaxation operations". *IEEE Transactions on Systems, Man and Cybernetics*, 6:420–433, June 1976.

[26] O. D. Faugeras and M. Berthod. "Improving consistency and reducing ambiguity in stochastic labeling: An optimization approach". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:412–423, April 1981.

[27] R. A. Hummel and S. W. Zucker. "On the foundations of relaxation labeling process". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(3):267–286, May 1983.

[28] J. J. Hopfield. "Neural networks and physical systems with emergent collective computational abilities". *Proceedings of National Academic Science, USA*, 79:2554–2558, 1982.

[29] J. J. Hopfield. "Neurons with graded response have collective computational properties like those of two state neurons". *Proceedings of National Academic Science, USA*, 81:3088–3092, 1984.

[30] K. E. Price. "Relaxation matching techniques – A comparison". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(5):617–623, September 1985.

[31] B. M. McMillin and L. M. Ni. "A reliable parallel algorithm for relaxation labeling". In P. M. Dew, R. A. Earnshaw, and T. R. Heywood, editors, *Parallel Processing for Computer Vision and Display*, pages 190–209. Addison-Wesley, 1989.

[32] H. I. Bozma and J. S. Duncan. "Admissibility of constraint functions in relaxation labeling". In *Proceedings of Second International Conference on Computer Vision*, pages 328–332, Florida, December 1988.

[33] D. P. O'Leary and S. Peleg. "Analysis of relaxation processes: the two-node two label case". *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13(4):618–623, 1983.

[34] T. A. Jamison and R. J. Schalkoff. "Image labeling: a neural network approach". *Image and Vision Computing*, 6(4):203–214, November 1988.

[35] N. Nasrabadi, W. Li, and C. Y. Choo. "Object recognition by a Hopfield neural network". In *Proceedings of Third International Conference on Computer Vision*, pages 325–328, Osaka, Japan, December 1990.

[36] W-C Lin, F-Y Liao, and C-K Tsao. "A hierarchical multiple-view approach to three-dimensional object recognition". *IEEE Transactions on Neural Networks*, 2(1):84–92, January 1991.

[37] C. Peterson and B. Soderberg. "A new method for mapping optimization problems onto neural networks". *International Journal of Neural Systems*, 1(1):3–22, 1989.

[38] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.

[39] S. Z. Li. "A Markov random field model for object matching under contextual constraints". In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 866–869, Seattle, Washington, June 1994.

[40] S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, New York, 1995.

[41] S. Z. Li. "Improving convergence and solution quality of Hopfield-type neural network with augmented Lagrange multipliers". *IEEE Transactions on Neural Networks*, 7(6):1507–1516, November 1996.