Self Inducing Relational Distance and its Application to Image Segmentation

Jianbo Shi and Jitendra Malik
Computer Science Division
University of California at Berkeley, Berkeley, CA 94720
{jshi,malik}@cs.berkeley.edu

Abstract

We propose a new feature distance which is derived from an optimal relational graph matching criterion. Instead of defining an arbitrary similarity measure for grouping, we will use the criterion of reducing instability in the relational graph to induce a similarity measure. This similarity measure not only improves the stability of the matching, but more importantly, also captures the relative importance of relational similarity in the feature space for the purpose of grouping. We will call this similarity measure the self-induced relational distance. *We demonstrate the distance measure on a brightness-texture feature space and apply it to the segmentation of complex natural images.*

Keywords: Feature distance, grouping, texture segmentation,relational graph matching.

Self Inducing Relational Distance and its Application to Image Segmentation

Abstract

We propose a new feature distance which is derived from an optimal relational graph matching criterion. Instead of defining an arbitrary similarity measure for grouping, we will use the criterion of reducing instability in the relational graph to induce a similarity measure. This similarity measure not only improves the stability of the matching, but more importantly, also captures the relative importance of relational similarity in the feature space for the purpose of grouping. We will call this similarity measure the self-induced relational distance*. We demonstrate the distance measure on a brightness-texture feature space and apply it to the segmentation of complex natural images.*

1 Introduction

In any grouping algorithm, there are two crucial subproblems need to be addressed: (1) the similarity measure between the features, and (2) the grouping criterion and efficient algorithm to solve it.

In our previous paper[20], we have addressed the latter problem with *normalized cuts*. In the normalized cut scheme, the grouping problem is transformed into a graph partitioning problem. The nodes in the graph are feature points, such as pixels, and the weight on each graph edge reflects the similarity between the two nodes connected by that edge. The grouping problem then becomes the problem of finding the best hierarchical sub-partition of the graph according to a global partitioning criterion called *normalized cut*. The *normalized cut* criterion favors sub-partitioning of a graph such that the total similarity among the subgraphs is high and the total dissimilarity across the subgraphs is low. We have shown that this graph partitioning problem can be solved efficiently using a generalized eigenvalue system, and good results have been obtained on segmenting brightness and color images with a simple Euclidean distance in the HSV color space.

In the case of brightness and color segmentation, such simple similarity measures between the pixels are sufficient. Distance measures on combined texture and brightness-color space, however, are harder to define. Figure (1) illustrates some of the difficulties. The difficulties in scenes like those shown in figure (1a) is that although the cheetah has coherent texture, the tree branches in the background are essentially random with each branch differing from the others. One would like the texture similarity measure to not overly emphasize the difference in that region. Defining a simple texture distance in figure (1b) could also be difficult because it is hard to judge whether the "stripe-ness" of the zebra skin is a more important similarity measure than the orientation of the stripes. In principle, one can claim any distance measure can and should be induced by modeling the distribution of all the objects in the world. We will illustrate in this paper, however, that by analyzing relationships between the elements in a single scene, one can induce a very reasonable distance measure. In fact the way such a distance is induced is a direct consequence of the need to find a stable relational structural match of objects.

A relational graph is a graph representation of objects, their parts and the relationships among them. Typically, relational graph matching is used in the object recognition setting,

(a) (b)

Figure 1: Two complex textured scenes in natural world. What is the right texture similarity measure?

where we would like to map the parts of an unknown object to those of a known object in a data base, such that both the properties of the parts as well as the relationship among the parts are preserved. This requirement of preserving structural relationship is particularly useful when the parts themselves are not very descriptive. However, the ambiguities in the matching of the parts can not be completely removed by enforcing relational integrity. One could imagine cases where several parts share the same unary as well as binary relationships among them. One way of solving this problem is through a judicious selection of the parts and the relationship defined between them. In this paper, we will show that given any relational graph, the ambiguity in the graph matching is related to eigenvectors of its relational attribute graph. These eigenvectors can in turn be used to define a similarity measure. By grouping nodes that are similar with respect to this measure, maximum stability of the graph matching can be achieved.

In this paper we will focus mostly on this aspect of defining a new distance measure. In particular, for defining texture similarity in image segmentation, we will build a relational graph for the image by taking each pixel as a node where the attribute values on the nodes are set to be the histograms of the image response to a filter bank between the pixels. We will then use the instability in matching this texture relational graph to itself to define a new texture similarity measure for grouping. We will call this new distance measure the *self induced relational distance*, for it pools not only local feature information, but also more global relationships among them. For image 1(a), one can see that although the tree branches look different from one another, their relative differences to other image parts are similar. In image 1(b), the relationship measured using the "stripe-ness" factor are more stable and prominent in this image than those using the stripe orientation. As we will see in sections 3 and 4(figure 6 and 7), our proposed distance measure can indeed capture this intuition.

This paper is organized as follows. In section 2, we will study the problem of relational graph matching, in particular the instability in the graph matching and how the *self-induced relational distance* can be used to reduce such instability. In sections 3 and 4, we will illustrate how such distance can be used in the case of texture segmentation. We conclude in section 5.

2 Relational Graph Matching

The use of relational graph matching in computer vision starts with the seminal work of Barrow and Popplestone[2], and is subsequently followed up by the works of [1, 8, 19, 17, 3, 6, 24, 9, 23]. For relational structural matching, an object is described by its attribute graph, and the mapping of the object parts can be performed by using relational graph matching. An attribute graph is a graph **G**=(**V**,**E**), with attributes **A** attached to each of the nodes in **V** and edges in **E**. For example, figure (2) shows how an attribute graph can be constructed for an object in figure (2a).

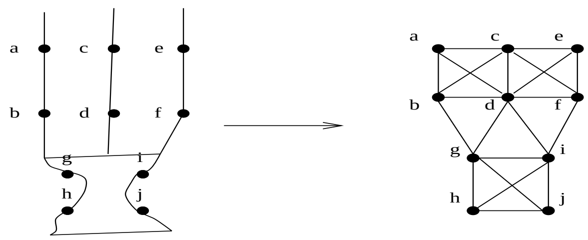

Figure 2: An attribute graph representation(b) of the object on the left(a). The nodes in the graph are the landmark points. An attribute on each of the nodes could be the curvature at each point, and the attribute on edge (i, j) could the the distance between points i and j.

To match two attribute graphs $\mathbf{G} = (\mathbf{V_G}, \mathbf{E_G})$, and $\mathbf{H} = (\mathbf{V_H}, \mathbf{E_H})$, we seek a mapping $f : \mathbf{V_G} \rightarrow \mathbf{V_H}$ of the vertices, such that if $f(v_G) = v_H$, and $f(u_G) = u_H$, for v_G, u_G in **G** and v_H, u_H in **H**, then attributes of the nodes v_G, u_G should match those of nodes v_H and u_H, and the attributes on the graph edge (v_G, u_G) should match those on graph edge (v_H, u_H).

Let us first consider a simpler case where graph **G** and **H** are of the same size, and the attributes on the nodes and edges are real numbers. Given a graph **G** of size N, define $\boldsymbol{A}_G$ to be the $N \times N$ attribute matrix, where $\boldsymbol{A}_G(i, i)$ is the attribute on node i, and $\boldsymbol{A}_G(i, j)$ is the attribute of the graph edge (i, j). Furthermore, we will assume the attribute graph **G** is undirected. Let P be the $N \times N$ permutation matrix with $P(i, j) = 1$ iff $f(v_i) = v_j$ for v_i in **G** and v_j in **H**. We can define the cost function of the relational graph matching to be the L_2 norm of the total error on each of the nodes and edges:

$$\begin{aligned} \boldsymbol{E}_1(P) &= \sum_{i=1}^{N}\sum_{j=1}^{N}(\boldsymbol{A}_G(v_i, v_j) - \boldsymbol{A}_H(f(v_i), f(v_j)))^2 \\ &= \|P\boldsymbol{A}_G P^T - \boldsymbol{A}_H\|^2 \end{aligned}$$

A similar cost function can be also defined using an inner product norm:

$$\begin{aligned} \boldsymbol{E}_2(P) &= \sum_{i=1}^{N}\sum_{j=1}^{N}\boldsymbol{A}_G(v_i, v_j) \times \boldsymbol{A}_H(f(v_i), f(v_j)) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} P\boldsymbol{A}_G P^T(i, j) \times \boldsymbol{A}_H(i, j) \\ &= tr(P^T\boldsymbol{A}_G P\boldsymbol{A}_H). \end{aligned}$$

In fact these two definitions of the error function are equivalent, since:

$$\begin{aligned} \|P\boldsymbol{A}_G P^T - \boldsymbol{A}_H\|^2 &= \|P\boldsymbol{A}_G P^T\|^2 + \|\boldsymbol{A}_H\|^2 - 2tr(P^T\boldsymbol{A}_G P\boldsymbol{A}_H) \\ &= \|\boldsymbol{A}_G\|^2 + \|\boldsymbol{A}_H\|^2 - 2tr(P^T\boldsymbol{A}_G P\boldsymbol{A}_H). \end{aligned}$$

But $\|\boldsymbol{A}_G\|^2 + \|\boldsymbol{A}_H\|^2$ is a constant, thus minimizing E_1 is same as maximizing E_2. However, as we shall see later, some properties of relational graph matching can be more readily seen with one of the cost functions.

The graph isomorphism problem, that is finding the P that minimizes $\boldsymbol{E}_1$, is a difficult problem to solve exactly. In fact, it is one of the open problems which is not known to be **NP**-complete or in **P**[14]. There have been various efforts to find a close approximate solution using relaxation methods[16, 8], as well as variants of such methods in probabilistic settings[6, 24], or in energy minimization settings[9]. Also there are tree-search based methods such as[23]. In defining relational distance, Shapiro and Haralick[19] and Sanfelu and Fu[17] proposed a distance that will accommodate missing parts in the graphs. Boyer and Kak[3] and Vosselman[23] used information-theoretic approaches to take in consideration the different likelihoods of mis-matches in different attributes.

In this paper, however, we will be mostly interested in the possible uncertainties in the matching arising from the structure of the relational graph. Instead of looking at the discrete group of permutation matrices, we will study the continuous group of orthogonal matrices, of which the permutation matrix is a subgroup. As shown in the work of Brockett[4, 5] and Umeyama[21], in the space of orthogonal matrices, the convergence and stability properties of the matching problem are better understood.

2.1 Instability and Self-inducing Distance

The following two theorems capture the key properties of the solutions of the minimization problem (1) over the space of orthogonal matrices.

Theorem 1 [Umeyama 1988, Brockett 1991][21, 4] *Let $\boldsymbol{A}_G = V_G\Sigma_G V_G^T$, and $\boldsymbol{A}_H = V_H\Sigma_H V_H^T$, be the singular value decomposition of $\boldsymbol{A}_G$ and $\boldsymbol{A}_H$. The matrix $Q = V_H D\Pi V_G^T$ minimizes $\boldsymbol{E}_1(Q) = \|QA_GQ^T - A_H\|^2$ over all orthogonal matrices Q, where $D = diag(d_1, d_2, ..., d_n)$ with $d_i = \pm 1$, and Π is some permutation matrix. In the case where the eigenvalues in Σ_G and Σ_H are sorted, $\Pi = I$.*

Theorem 2 [von Neumann 1937, Brockett 1991][22, 4] *Let $\{\lambda_1^G, \lambda_2^G, ..., \lambda_n^G\}$ and $\{\lambda_1^H, \lambda_2^H, ..., \lambda_n^H\}$ be the sorted eigenvalues of $\boldsymbol{A}_G$ and $\boldsymbol{A}_H$. The eigenvalues of the Hessian of $E_2(Q) = tr(Q^T\boldsymbol{A}_G Q\boldsymbol{A}_H)$ at the $Q = V_H D\Pi V_G^T$, are n(n-1)/2 products of form $r_{ij} = (\lambda_j^G - \lambda_i^G)(\lambda_{\pi(i)}^H - \lambda_{\pi(j)}^H)$. In particular, they are all negative for just one choice of π, which is when $\pi(i) = i$.*

Proofs of both theorem can be found in[4, 21], we shall only briefly[1] repeat the proof for Theorem 1. Suppose $Q = V_H D V_G^T$, we have

$$E_1(Q) \;=\; \|Q\boldsymbol{A}_G Q^T - \boldsymbol{A}_H\|^2$$

[1] Space limitations prevent us from developing the motivation and approach behind these results. We urge the reader to study the Umeyama and Brockett paper for a gentler introduction.

$$
\begin{aligned}
&= \|V_H D V_G^T \boldsymbol{A}_G V_G D V_H^T - \boldsymbol{A}_H\|^2 \\
&= \|V_H D V_G^T V_G \Sigma_G V_G^T V_G D V_H^T - V_H \Sigma_H V_H^T\|^2 \\
&= \|V_H D \Sigma_G D V_H^T - V_H \Sigma_H V_H^T\|^2 \\
&= \|D \Sigma_G D - \Sigma_H\|^2 = \|\Sigma_G - \Sigma_H\|^2
\end{aligned}
$$

But since for any orthogonal matrix Q, we also have $\|Q\boldsymbol{A}_G Q^T - \boldsymbol{A}_H\|^2 \geq \|\Sigma_G - \Sigma_H\|^2$, the above Q must also be the minimum of $\boldsymbol{E}_1(Q)$.

Since the orthogonal matrix Q is an approximation of the permutation matrix P, ideally the entry in $Q(i,j)$ should be large if node v_i in **G** is matched to v_j in **H**. Simple inspection of the definition of Q in theorem 1 shows that the entry in $Q(i,j)$ is large if the ith row of V_H matches with jth row of DV_G. In another words, two nodes, where each node is represented by the corresponding row or column in $\boldsymbol{A}_G$ and $\boldsymbol{A}_H$, are more likely to match each other if their projections onto the eigenvectors of the relational graphs are similar. The multiplication of V_G on the left by D is necessary because the eigenvectors of $\boldsymbol{A}_G$ and $\boldsymbol{A}_H$ can take on arbitrary positive or negative direction. The closest permutation P to the orthogonal matrix Q can be then found by maximizing

$$\sum_i \sum_j P(i,j) Q(i,j) = tr(P^T Q).$$

However in general, D is unknown, we have to maximize $tr(P^T |V_H||V_G^T|)$ instead. The problem of maximizing $tr(P^T Q)$ or $tr(P^T |V_H||V_G^T|)$ is an instance of bipartite maximum weighted graph matching, with $Q(i,j)$ being the weight of the bipartite graph edges. The bipartite graph matching can be solved efficiently using max-flow or the Hungarian method[7]. Alternatively, we can use the method of Scott and Longuet-Higgins[18] to produce a quick approximation.

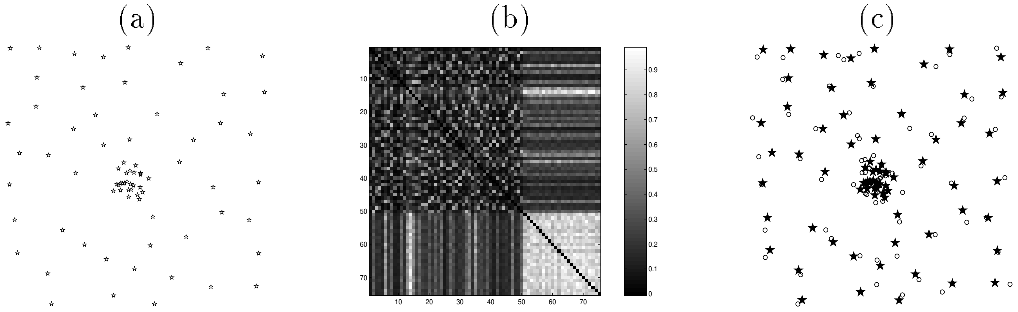

Figure 3: Subplot (a) shows the spatial layout of the points in the graph **G**. The points are formed by overlapping two point sets sampled from Poisson distributions of different densities. The attribute value on the graph edge (v_i, v_j), shown in subplot (b), is defined as $e^{-d(i,j)/\sigma_s}$, where $d(i,j)$ is the spatial distance between the two points, and $\sigma_s = 4.0$ in this case. For purposes of visual illustration, we ordered the point set such the points in the sparse set are numbered 1 to 50, and the points in the denser set are numbered 50 to 75. Note that the similarity measures among the sparse point sets are considerably weaker than those of the denser set. The points in Graph **H**, shown as circles in (c), are obtained by adding random Gaussian noise($\sigma = 0.2$) to the spatial location of points in graph **G**, shown as stars.

To see how such a method would work, we will look at an example shown in figure (3). The nodes in the graph ($\mathbf{G}$) are constructed by overlaying two spatial point sets of different densities. The attribute relationship is a function of the spatial proximity of the nodes.

To test the sensitivity of the graph matching, we will construct a graph $\mathbf{H}$ by adding random Gaussian noise to the spatial location of the points in $\mathbf{G}$. The performance of this matching algorithm as we increase the noise level in the spatial location of the point set can be seen in figure (4). As shown in figure (4), the computed permutation matrix is quite

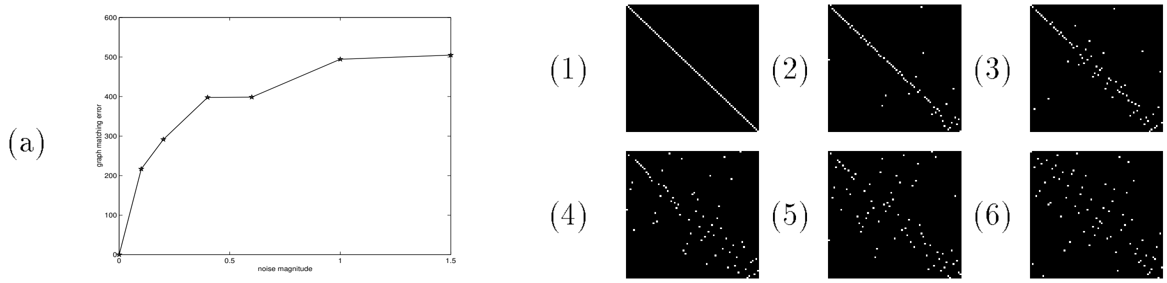

Figure 4: Subplot (a) shows the error cost $\mathbf{E}_1(P)$ as a function of the noise added to points in graph $\mathbf{H}$. Subplots (1) to (6) shows the permutation matrix computed with Gaussian noise of variance of $0, 0.1, 0.2, 0.4, 0.6, 1.0$ added to the spatial location of points in $\mathbf{G}$. Note as the noise level increases, the ambiguity in the matching increases. However, most of the mismatches occur for the points within each their own respective point sets.

sensitive to noise in this particular example.

Upon closer inspection of the graph $\mathbf{G}$, this problem should not be too surprising. The points in the dense cluster have very similar attribute relationships to all other points around it, hence each point can be mapped to other points in the dense cluster without affecting the matching cost. The same statement can also be made about the sparse set of points in the background. Furthermore, from this example, one notes that the uncertainties in matching occur in a structured way– the points in the sparse set are less likely to be mismatched with the points in the denser set.

This degree of uncertainty can in fact be quantified more precisely by theorem (2), which says the instability of Q at its minimum will be high in the dimension where the term $r_{ij} = (\lambda_j^G - \lambda_i^G)(\lambda_i^H - \lambda_j^H)$ is small. The difference in the singular values, $\lambda_i - \lambda_j$, is also often called gap_i.

One way to remove this instability is to use only the set of eigenvectors whose gaps are large. One could define $Q^* = V_H^* D^* V_G^{*T}$, where V_H^* and V_G^* are the first k columns of V_H and V_G such that for all $j > k$, gap_j^G and gap_j^H are all less than some threshold, δ_{gap}. However, Q^* will no longer be an orthogonal matrix, for rows in V_H and V_G are not independent of each other. To solve this problem, we would like to group together nodes whose corresponding rows in V_H and V_G are similar. We can quantify this similarity by the inner product norm of the rows in V_G and V_H:

$$O_G = V_G^* V_G^{*T}. \tag{1}$$

By grouping nodes with this similarity measure, we can optimally remove the uncertainty in the matching of relational graphs.

Note that the similarity measure O_G depends only on the relationship of the nodes in one graph. Given a graph matching criteria we can perform the grouping of the nodes independently. We will call this distance measure the *self-induced relational distance*.

To test our concept of the *self-induced relational distance*, we shall return to the example in figure (3). Figure (5a) shows the value of gap_i for the eigenvalues of the attribute graph $\boldsymbol{A}_G$, normalized by the first eigenvalue. Note that the gap_i becomes very small after the fourth eigenvector. We will define the self induced distance measure O_G by taking the first four eigenvectors of $\boldsymbol{A}_G$, shown in figure (5b). As one can see from figure (5), the new distance measure O_G favors condensing the dense point set in the middle into one node, and closely related points in the sparse set into small numbers of nodes. With this small set of nodes, the relational graph matching become more stable, though at the expense of a reduction of the matching resolution.

(a) (b) (1) (2) (3) (4)

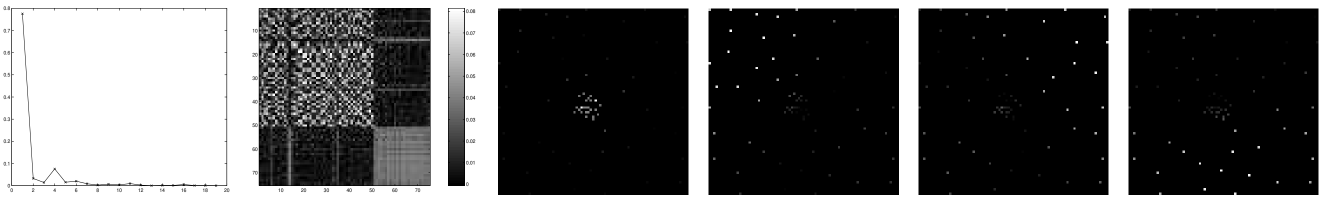

Figure 5: Subplot (a) shows the gap_i in eigenvalues computed from the attribute graph $\mathbf{A}_G$ in 3(b). Subplot (b) shows the self induced distance matrix O_G computed using the first four eigenvectors of $\mathbf{A}_G$. The self induced distances from a point in the dense set and three other points in the sparse set to all other points are shown in subplots (1) to (4) respectively. The brightness value indicates the similarity strength between the points. Note the new distance measure more strongly connects up the sparse points, without blurring the connections from the dense set to the sparse set.

2.2 Relational Graph on Multi-valued Attributes

So far, we have focused on the case where the attribute value on each graph edge is just a scalar. In general, however, we could have a vector of real numbers representing the attribute relationship between two nodes. Let $[\boldsymbol{A}_G(v_i, v_j, 1), \boldsymbol{A}_G(v_i, v_j, 2), ..., \boldsymbol{A}_G(v_i, v_j, L)] = \boldsymbol{A}_G(v_i, v_j)$ denote the vector attribute relating node v_i and v_j. We can define a similar graph matching cost function with this new attribute graph:

$$\boldsymbol{E}_v = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{L} \boldsymbol{A}_G(v_i, v_j, k) \times \boldsymbol{A}_H(f(v_i), f(v_j), k) \tag{2}$$

Although the problem of finding an approximate solution in the space of orthogonal matrices is more complicated, for the purpose of inducing a relational similarity measure, we can define a simpler cost function $\boldsymbol{E}_v^*$ which serve as an upper bound on $\boldsymbol{E}_v$:

$$\begin{aligned}\boldsymbol{E}_v^* &= \sum_{i=1}^{N}\sum_{j=1}^{N}(\sum_{k=1}^{L} \boldsymbol{A}_G(v_i, v_j, k)^2)^{1/2}(\sum_{k=1}^{L} \boldsymbol{A}_H(v_i, v_j, k)^2)^{1/2} \\ &\geq \sum_{i=1}^{N}\sum_{j=1}^{N}(\sum_{k=1}^{L} \boldsymbol{A}_G(v_i, v_j, k) \times \boldsymbol{A}_H(f(v_i), f(v_j), k)) = \boldsymbol{E}_v,\end{aligned}$$

by Schwartz Inequality of $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \leq \|\boldsymbol{x}\|\|\boldsymbol{y}\|$.

For the case where graphs $\mathbf{G}$ and $\mathbf{H}$ are isomorphic, the maxima of each cost function are the same, hence any orthogonal matrix that maximizes $\boldsymbol{E}_v^*$ also maximizes $\boldsymbol{E}_v$. In the general case, however, the orthogonal matrix that maximizes $\boldsymbol{E}_v^*$ is only an optimistic approximation of the one that maximizes $\boldsymbol{E}_v$. Even so, we found in practice, that this approximation often does a good job. We shall show this in the case of texture segmentation,

2.3 Summary of the self induced relational distance

Let us review the main thread of the argument in this section. Relational graph matching is studied in an algebraic setting, where finding the best match reduces to finding a permutation matrix that optimally reorders the nodes of one graph to correspond to the other. For mathematical convenience, the discrete search problem in the space of permutation matrices is transformed to a continuous problem by embedding it in the space of orthogonal matrices. Results derived by Umeyama[21] and Brockett[4, 5] reveal that the optimal matching orthogonal matrix and its instability is related to the eigenvectors(V) and eigenvalues(λ) of the relational attribute graphs. To ensure stability of the graph matching, graph nodes are clustered based their projections in the space spanning the first few eigenvectors(V^*) whose *gap*s are large. We call the distance in this subspace, the self induced distance: $O = V^* V^{*T}$.

To illustrate the usefulness of the *self induced distance measure*, we will apply it to the problem of texture similarity measurement and segmentation by setting up a relational graph on the texture feature space. The nodes in the graph are the pixels in the image, and the attributes on each graph edge are the correlations between the local filter outputs. We will show in the next two sections the detail of how such a relational graph can be set up, and how the self induced relational distance can be used to solve the segmentation problem.

3 Texture Measurements

First we will describe the local texture descriptors. The texture measurements used in the paper is based on image responses to a set of filter banks. As shown in the work of [12, 11, 13], the filter responses contain sufficient information for discriminating different texture patterns. The set of filter bank that we will use are even symmetric elongated difference of Gaussian(DOOG) filters as used in[12]. There are total of 6 orientations repeated over 3 different scales. We will denote by $F_{\sigma,\theta,\alpha}$ an elongated DOOG filter of orientation θ, scale σ, and elongation factor α. Furthermore, to reduce any unwanted bias, all the filters are made zero mean and are normalized so that the L_1 norms equal 1. Let $I_{\sigma,\theta,\alpha}$ denote the filter responses:

$$I_{\sigma,\theta,\alpha} = I * F_{\sigma,\theta,\alpha} \tag{3}$$

For the remainer of the paper, we will fix the elongation factor $\alpha = 3$, and drop the subscript α. For convenience, we will treat the image I itself as the filter response to an impulse function, or $I = I_{0,0,0}$.

Because the filter responses only characterize the intensity variation at a single point in the image, and in describing a texture pattern one would like to integrate information over a spatial local neighborhood, a histogram representation of the filter responses has emerged as an attractive alternative. Theoretical as well as practical benefits of using this histogram representation has been illustrated in the work of [10, 25, 15].

We will use a similar representation which is a *soft* histogram defined over the bins, $\boldsymbol{b}_{\sigma,\alpha} = (b^{min}_{\sigma,\alpha}, ..., b^{max}_{\sigma,\alpha})$ of length K:

$$H^i_{\theta,\sigma,s}(k) = \sum_{j \in N_s(i)} h^j_{\theta,\sigma}(k), \tag{4}$$

where

$$h^j_{\theta,\sigma}(k) = \frac{1}{2\pi} \int_{\boldsymbol{b}(k)}^{\boldsymbol{b}(k+1)} e^{-(I^i_{\theta,\sigma} - x)/\sigma_t^2} dx. \tag{5}$$

The σ_t is related to the degree of uncertainty tolerated in the filter response, and s denotes the spatial radius of the neighborhood center at i. This definition of the histogram softens the effect of binning and uncertainties in the measurement. Since we also don't know a priori the size of the neighborhood in which texture pattern exists, we will compute this soft histogram over several different neighborhood sizes. Note also that the L_1 norm of $H^i_{\theta,\sigma,s} = |N_s(i)|$ the number of nodes in that neighborhood.

4 Relational Brightness-Texture Distance

Given a brightness-texture image, an attributed relational graph can be constructed by taking each pixel as a node, and defining a set of attributes on each graph edge connecting two nodes based on the correlation between their local soft histograms. Although we can take the inner product of the histograms as their correlation, this definition has a bias since the L_2 norm of the histograms, $\|H^i_{\theta,\sigma,s}\|$, varies depending on the distributions in the histogram. To correct for this bias, we let $G^i_{\theta,\sigma,s}(k) = H^i_{\theta,\sigma,s}(k)^{\frac{1}{2}}$, and define the attributes on the graph edge (v_i, v_j) as:

$$\boldsymbol{A}_{\theta,\sigma,s}(i,j) = \frac{1}{|N_s i||N_s j|} {G^i_{\theta,\sigma,s}}^T \cdot G^j_{\theta,\sigma,s}, \tag{6}$$

for each set of parameters θ, σ, and s.

As mentioned in section (2.2), for the purpose of computing the *self induced relational distance*, we can get an approximation with a scalar attribute graph where the attribute on graph edge (v_i, v_j) is

$$\boldsymbol{A}^*(i,j) = (\sum_{\theta,\sigma,s} \boldsymbol{A}^2_{\theta,\sigma,s}(i,j))^{1/2}. \tag{7}$$

Given this relational graph, we can compute the self induced relational distance as described in section 2.1. Experimental tests on brightness-texture image such as the one illustrated in figure (1a) are shown in figure(6). The self induced relational distance clearly depends on the number of eigenvectors of the attribute graph taken in equation (2). In all of our experiments, we set the threshold of the minimum *gap* value to be $\delta_{gap} = 8 \times 10^{-4} \times \lambda_1$, where λ_1 is the largest eigenvalue of $\boldsymbol{A}_G$. This threshold of δ_{gap} determines the number of eigenvectors to be used. Figure (7) shows the effect of using different numbers of eigenvectors in computing the self induced relational distance for the scene shown in figure (7a).

(a)

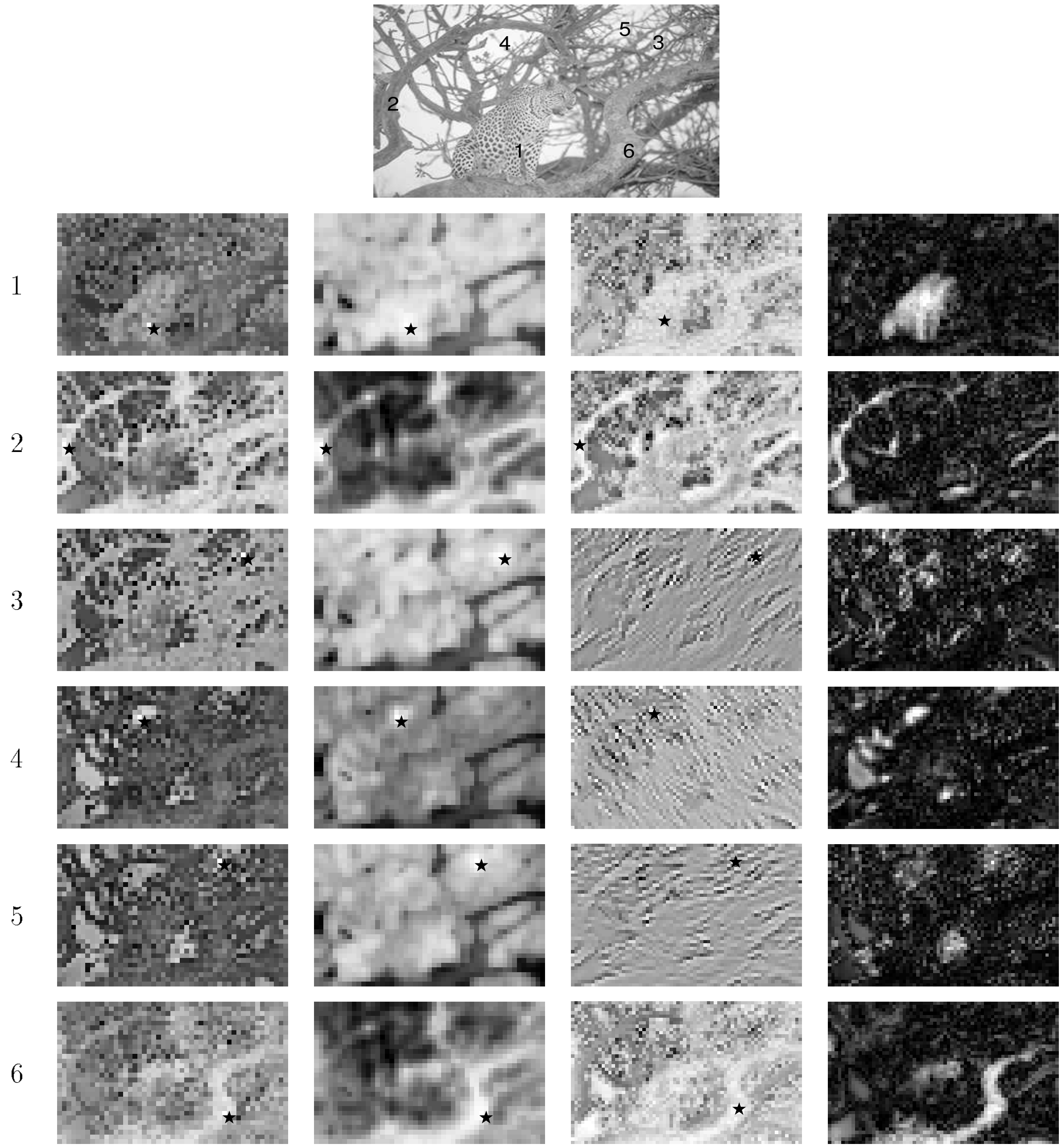

Figure 6: Comparison of different texture distance on a texture image shown in (a). The ith row shows the texture similarity between pixel i(star) labelled in (a) to all other pixels in the image according to (1) χ^2 difference on the local histogram with neighborhood size of 3×3 pixels (column 1), (2) χ^2 difference with neighborhood size of 11×11(column 2), (3) L^∞ norm in filter responses computed with neighborhood size of 5×5 (column 3), and (4) self induced distance(column 4). Note that picking the right neighborhood size could be a hard problem with χ^2 distance– with too small neighborhood size connections in the texture image are sparse and weak, with too large a neighborhood size object boundaries are blurred. Similarly L^∞ could cause over-fragmentation in areas such as the tree branches. With the self-induced texture distance, relational coherence in texture feature space is enhanced, without sacrificing the sharpness of the texture boundaries.

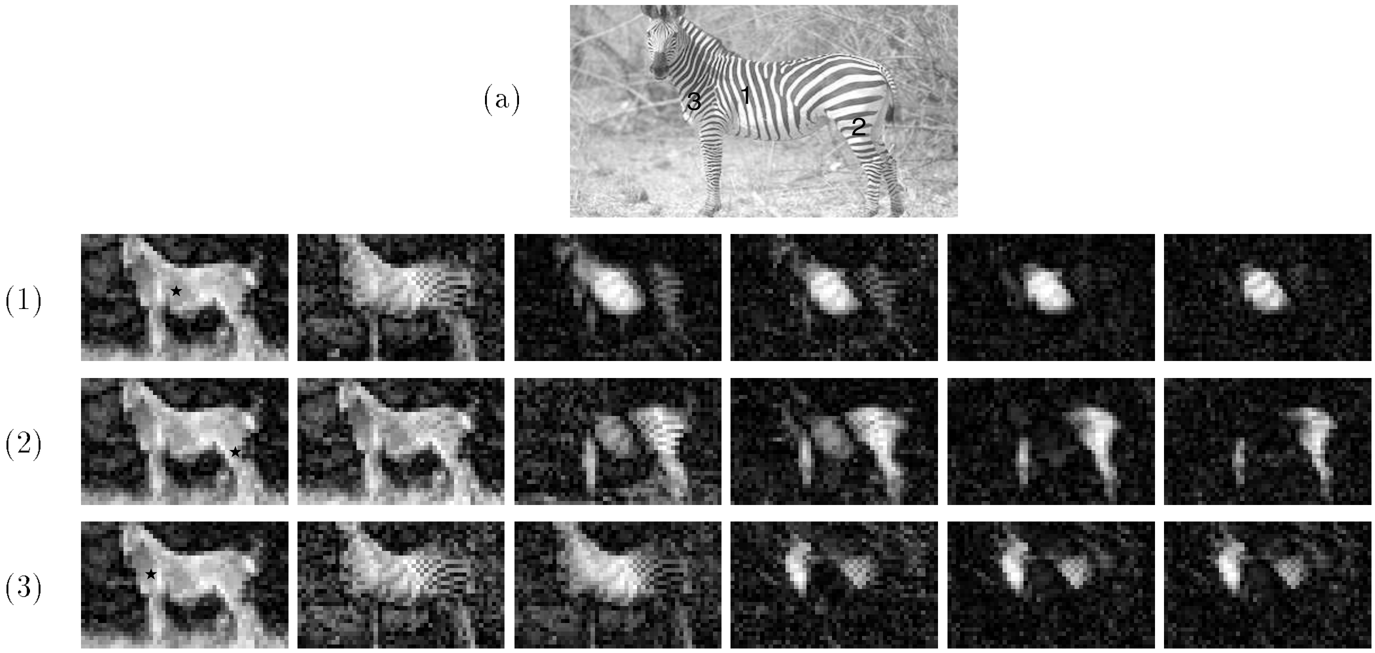

Figure 7: Row i shows the effect of using different number of eigenvectors in computing the self-inducing relational distance $O(i,j)$ between pixel i labelled in (a) and all other pixels j in the image. Each of the columns corresponds to distance computed with 2,3,4,6,9,12 eigenvectors from left to right respectively. With our setting of δ_{gap}, the first 9 eigenvectors will be used in computing the self induced relational distance.

4.1 Normalized Cuts

Using the newly defined texture similarity measure, we can produce a segmentation of a scene using the *normalized cut* algorithm [20]. In the normalized cuts scheme, the image segmentation problem is reduced to a hierarchical graph partitioning problem. In this case, the nodes in the graph are the pixels in the image, and the weights on graph edge connecting two nodes are the self induced relational similarity measures. We then seek a sub-partitioning of the graph, such that the similarities within each the subgraphs are high and similarities between the subgroups are low.

Let $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ be a weighted graph. We can break $\mathbf{G}$ into two disjoint sets, A, B, $A \cup B = \mathbf{V}$, $A \cap B = \emptyset$, by simply removing edges connecting the two parts. The total weights on those edges removed reflect the degree of dissimilarity between the sets A and B. In graph theoretic language, it is called the *cut*:

$$cut(A,B) = \sum_{u \in A, v \in B} w(u,v), \tag{8}$$

where $w(u,v)$ is the weight on the edge connecting u and v. Although there are efficient computational algorithms for finding partitions that minimizes the *cut* value, this criterion favors partitions which have small sizes[20]. To fix this undesirable bias, we will define *normalized cut* (*Ncut*) as unbiased measure of dissimilarity:

$$Ncut(A,B) = \frac{cut(A,B)}{asso(A,V)} + \frac{cut(A,B)}{asso(B,V)}, \tag{9}$$

where $asso(A,V) = \sum_{u \in A, t \in V} w(u,t)$ is the total connection from nodes in A to all nodes in the graph, and $asso(B,V)$ is similarly defined. We can also define an unbiased measure for total similarity within groups for a given partition: $Nasso(A,B) = \frac{asso(A,A)}{asso(A,V)} + \frac{asso(B,B)}{asso(B,V)}$, where $asso(A,A)$ and $asso(B,B)$ are total weights of edges connecting nodes within A and B respectively. A simple calculation shows $Ncut(A,B) = 2 - Nasso(A,B)$. Hence minimizing the dissimilarity between the groups and maximizing the similarity within the groups can be done simultaneously.

Let $\mathbf{W}$ be the graph weight matrix, and $\mathbf{D}$ be the diagonal matrix with $\mathbf{D}(i,i) = \sum_j \mathbf{W}(i,j)$. In [20], we showed that minimizing $Ncut$ can be reduced to minimizing a Rayleigh quotient:

$$min.\, Ncut = min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y}}, \tag{10}$$

with the condition $\boldsymbol{y}_i \in \{1, -b\}$ and $\boldsymbol{y}^T \boldsymbol{D} \mathbf{1} = 0$. By relaxing $\boldsymbol{y}$ to take on real values, we can minimize equation (10) with its constraint by solving for the second smallest eigenvector of the generalized eigenvalue system,

$$(\mathbf{D} - \mathbf{W})\boldsymbol{y} = \lambda \mathbf{D} \boldsymbol{y}. \tag{11}$$

The vector $\boldsymbol{y}$ can be thought of an indicator vector for the partition. Furthermore, the subsequent eigenvectors are the real valued solutions that form the optimal sub-partitions. From those vectors a discrete partition can be obtained as shown in [20].

4.2 Overall Procedure and Texture Segmentation Result

In summary our texture segmentation algorithm can be described as:

1. Compute filter outputs $I_{\theta,\sigma}$ from an input image I,
2. Construct soft histogram at each pixel i, $H^i_{\theta,\sigma,s}$, over a neighborhood of radius $s = [1, 3, 5, 9, 15, 26]$ pixels,
3. Set up attributes $\boldsymbol{A}_{\theta,\sigma,s}$ and $\boldsymbol{A}^*$ from $H^i_{\theta,\sigma,s}$,
4. Compute the distance measure $O = |U^*_A U^{*T}_A|$, where $\boldsymbol{A}^* = U_A \Sigma_A U^T_A$, $U^*_A = U_A(:, 1:k)$ such that $\forall (j > k)\ gap_j < \delta_{gap}$,
5. Produce image segmentation using normalized cut algorithm[20].

The segmentation result using normalized cuts on the self induced relational texture distance are shown in figures (8) and (9). Unlike our previous work, the spatial proximity factor is not used in the segmentation algorithm, hence we obtain somewhat less spatially coherent groups. This is done on purpose to emphasize the texture similarity in our new relational distance measure. Adding spatial proximity factor in our framework is very easy. Note also our segmentation procedure produces a hierarchical partitioning of the scene, some of the under-segmented regions can be sub-divided into smaller parts.

5 Conclusion

In this paper, we have shown how a self induced feature distance can be derived by analyzing its relational graph. In particular, the process of finding a stable matching of the relational

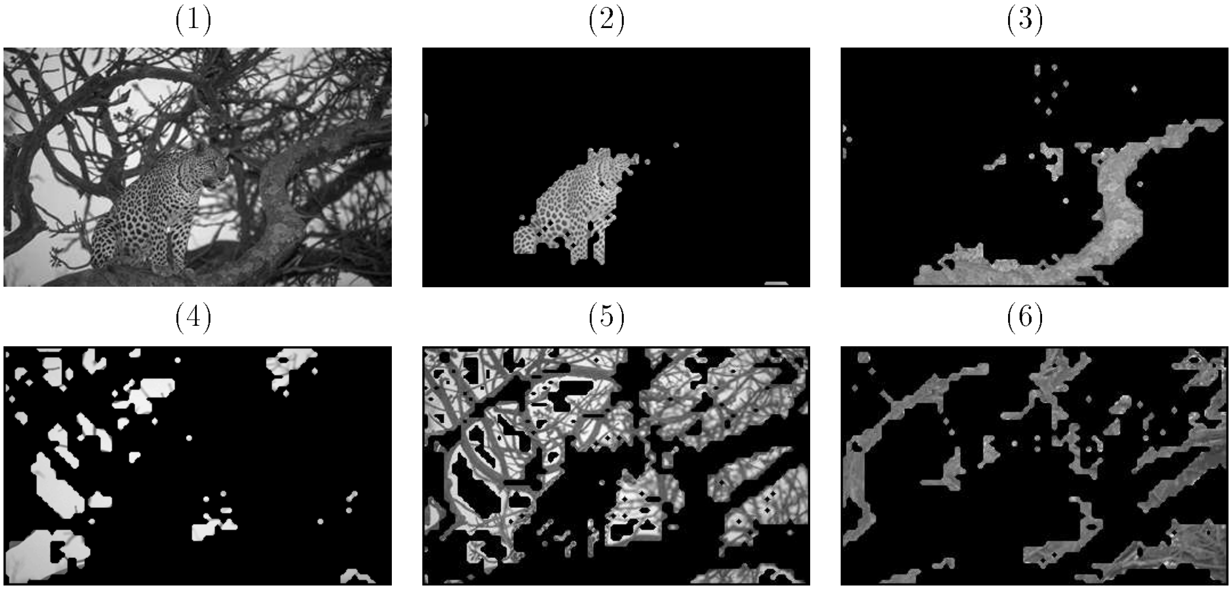

Figure 8: Segmentation results on the image in (a) with the newly defined relational distance. The relational graph is built by taking pixels from every $5th$ row and column as graph nodes. These are also the nodes used in normalized cuts. The full image size is 220×350. Note both the cheetah in (2) and the random tree branches in (5) come out as single coherent groups. The tree trunks in (3) and (6) are in two different groups because the difference in intensity value.

graph produces as a by-product a distance measure which captures the relative importance of the feature relationships. By applying it to the problem of image segmentation in the brightness-texture feature space, very good results are obtained on difficult images of natural scenes.

Currently we are extending this method to include other features such as color and motion, as well as higher level features such as shape. In addition, we are also studying how to handle more complex error functions in the relational graph matching, instead of the simple L_2 norm. We believe our research along this direction will lead to greater understanding of both the grouping and matching problems.

References

[1] H.G. Barrow and R.M. Burstall. Subgraph isomorphism, matching relational structures and maximal cliques. *Information Processing Letters*, 4:83–84, 1976.

[2] H.G. Barrow and R.J. Popplestone. Ralational descriptions in picture processing. *Machine Intelligence*, 6:377–396, 1971.

[3] K. Boyer and A. Kak. Structural stereopsis for 3d vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10:144–166, 1988.

[4] R.W. Brockett. Least squares matching problems. *Linear Algebra and its applications*, pages 761–777, 1989.

[5] R.W. Brockett. Dynamical systems that sorts lists, diagonalize matrices, and solve linear programming problem. *Linear Algebra and its applications*, pages 79–91, 1991.

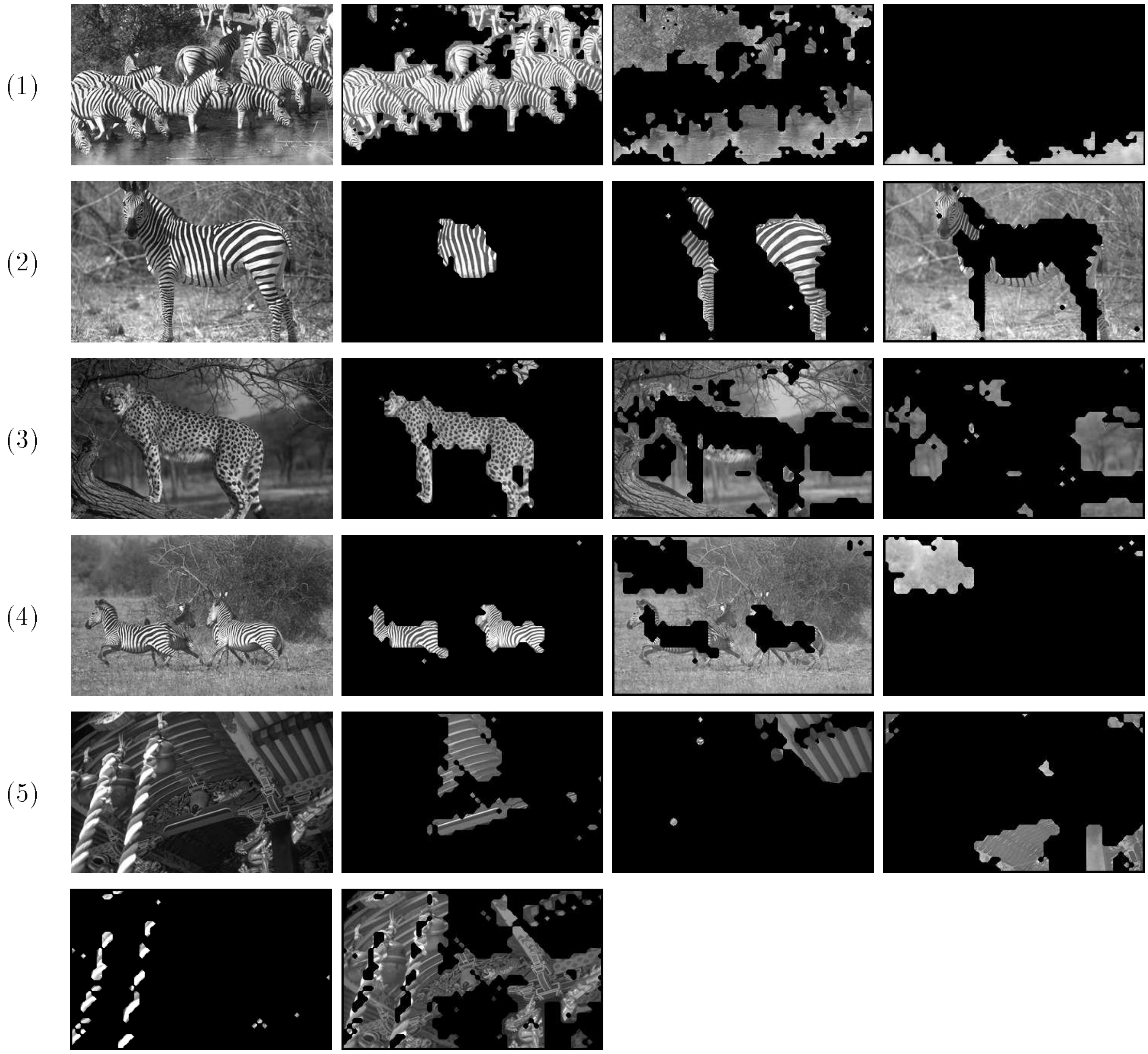

Figure 9: Segmentation results on a set of brightness-texture images. The relational graph is built by taking pixels in every $5th$ row and column as nodes in (1) and every $7th$ row and column as nodes in (2) to (5). The full image size is 220×350. In all examples, images are broken into visually similar texture, brightness patches. Note in (2), the stripes of the zebra are broken into two parts because in this image, the "stripe-ness" feature and the orientations of the stripe are of equal prominence.

[6] W. Christmas, J. Kittle, and M. Petrou. Probabilistic feature labeling schemes – modeling compatibility coefficient distribution. *Image and Vision Computing*, 14:617–625, 1996.

[7] C.E. Cormen, R.L. Leiserson, and L. Rivest. *Introduction to Algorithms.* McGraw-Hill, 1990.

[8] O.D. Faugeras and K.E. Price. Semantic labeling of aerial images using stochastic relaxation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 3:633–642, 1981.

[9] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18:377–388, 1996.

[10] D.J. Heeger and J.R Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH 95*, pages 229–238, 1995.

[11] A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(12):1167–1186, 1991.

[12] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *J. Opt. Soc. Am. A*, 7(5):923–932, 1990.

[13] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.

[14] C.H. Papadimitriou. *Computational Complexity.* Addison-Wesley Publishing Company, 1994.

[15] J. Puzicha, T. Hofmann, and J.M. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 267–272, 1997.

[16] A. Rosenfeld, R. Hummel, and S. Zucker. Scene labeling by relaxation operations. *IEEE SMC*, 6:420–433, 1976.

[17] A. Sanfeliu and K.S. Fu. A distance measure between attributed relational graph. *IEEE SMC*, 13:353–362, 1983.

[18] G.L. Scott and H.C. Longuet-Higgins. An algorithm for associating the features of two images. In *Proc. R. Soc. Lond. B.*, pages 21–26, 1991.

[19] L. Shapiro and R.M. Haralick. A metric for comparing relational descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 7:90–94, 1985.

[20] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 731–737, 1997.

[21] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(5):695–703, 1988.

[22] J. von Neumann. Some matrix-inequalities and metrization of matric-spaces. *Tomsk Univ. Rev.*, 1:286–300, 1937. also in John von Neumann: Collected Works,(A.H. Taub, Ed.) Vol. IV, Pergamon, New York, 1962, pp 205-218.

[23] G. Vosselman. *Relational Matching.* Springer-Verlag, 1992. Lectures in Computer Science, number 628.

[24] R.C. Wilson and E.R. Hancock. Structual matching by discrete relaxation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19:1–2, 1997.

[25] S.C. Zhu, Y. Wu, and D. Mumford. Frame: Filters, random fields, and minimax entropy. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 686–693, 1996.