Layered Graph Matching with Composite Cluster Sampling

Liang Lin, *Member*, *IEEE*, Xiaobai Liu, *Member*, *IEEE*, and Song-Chun Zhu, *Member*, *IEEE*

Abstract—This paper presents a framework of layered graph matching for integrating graph partition and matching. The objective is to find an unknown number of corresponding graph structures in two images. We extract discriminative local primitives from both images and construct a candidacy graph whose vertices are matching candidates (i.e., a pair of primitives) and whose edges are either negative for mutual exclusion or positive for mutual consistence. Then we pose layered graph matching as a multicoloring problem on the candidacy graph and solve it using a composite cluster sampling algorithm. This algorithm assigns some vertices into a number of colors, each being a matched layer, and turns off all the remaining candidates. The algorithm iterates two steps: 1) Sampling the positive and negative edges probabilistically to form a composite cluster, which consists of a few mutually conflicting connected components (CCPs) in different colors and 2) assigning new colors to these CCPs with consistence and exclusion relations maintained, and the assignments are accepted by the Markov Chain Monte Carlo (MCMC) mechanism to preserve detailed balance. This framework demonstrates state-of-the-art performance on several applications, such as multi-object matching with large motion, shape matching and retrieval, and object localization in cluttered background.

Index Terms—Graph matching, graph partitioning, DDMCMC, cluster sampling.

1 Introduction

1.1 Objective and Motivation

Many computer vision tasks can be posed as either a graph partitioning (or coloring) problem, such as image segmentation [1], [32] and scene labeling, or a graph matching (or correspondence) problem, such as wide baseline stereo [2], [20], [8], large motion [20], [26], [33], [35], object, and shape recognition [14], [3], [31], [37]. In this paper, we study a framework called layered graph matching for integrating graph partitioning and matching with their graphs edited. The objective is to find an unknown number of common graph structures in two images (or shapes). Fig. 1 shows an example in our experiments. From the two input images (column 1), we compute their primal sketch graphs (column 2) and partition them into three pairs of objects (columns 3-5): person, cars, and parking meters with graph edited (thick line segments) to achieve common (isomorphism) graph structures. The remaining fragments (column 6) are unmatched.

Our study is motivated by some recent tasks in object categorization, recognition, and unsupervised learning, for example, matching two object instances in the same category, learning object parts in large articulated motion, and detecting and localizing object templates in cluttered scenes. These tasks are different from conventional matching and correspondence problems in three major aspects: 1) The two matched objects or parts are often different instances in a category. They have quite different appearances and undergo large motion or nonrigid deformations, while sharing similar graph structures. This is in contrast to problems such as structure from motion, wide baseline stereo, and motion tracking, where the same instance shows up in two images. 2) The explicit graph structures, by which we mean the connectivity of constituent elements, are important for learning the templates of objects and parts and for localizing objects in clutter and with occlusion. This is in contrast to previous methods that represent objects by isolated points. 3) The number of common objects and parts is often unknown and their graph structures, extracted from images, are imperfect and need graph editing to achieve exact matching (or isomorphism).

These new tasks demand more general representation, and more effective inference algorithms for simultaneously solving the segmentation and matching problems.

1.2 Related Work and Comparison

Graph (or shape) matching has been extensively studied in the literature for numerous vision tasks. We can roughly divide the existing methods into three categories, according to their representations.

1.2.1 Category 1: Single-Layer and Point Based

These methods match local independent features without explicit graph structures, such as Harris corners, KLT features [25], scale invariant features [18], local edge features [2], and geometric blur descriptors [3], [28], [4]. The two sets of feature points are matched under a rigid affine transform plus nonrigid and locally smooth distortions accounted by a thin-plate spline (TPS) model. These point features are often robust against certain geometric distortions and illumination

- *L. Lin is with the School of Software, Sun Yat-Sen University, No. 132, Waihuandong Road, Guangzhou Higher Education Mega Center, 510006, P.R. China. E-mail: linliang@ieee.org.*
- *X. Liu is with the Lotus Hill Research Institute for Computer Vision and Information Science, Jiuzhou Building of Lotus Hill, Ezhou, Hubei Province 436000, China. E-mail: xbliu@lotushill.org.*
- *S.-C. Zhu is with the Statistics Department, University of California, Los Angeles, 8125 Math Science Building, Box 951554, Los Angeles, CA 90095. E-mail: sczhu@stat.ucla.edu.*

Manuscript received 29 Mar. 2008; revised 8 Sept. 2008; accepted 17 July 2009; published online 31 July 2009.
Recommended for acceptance by Y. Ma.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-03-0179.
Digital Object Identifier no. 10.1109/TPAMI.2009.150.

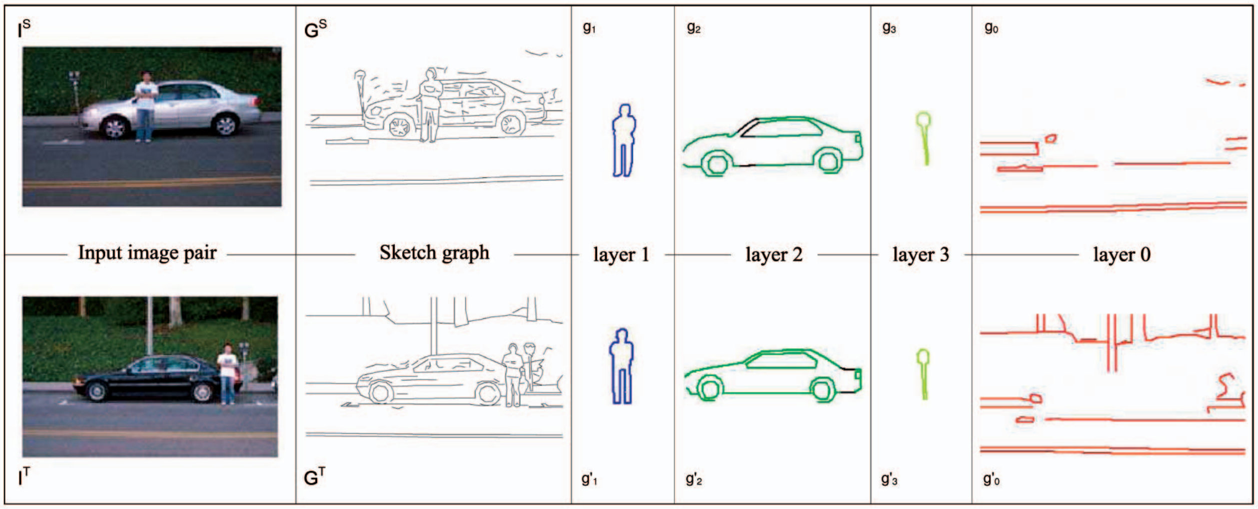


Fig. 1. An example of simultaneous graph matching and partition. Given two input images $\mathbf{I}^S, \mathbf{I}^T$ (column 1), two primal sketch representations G^S, G^T are computed as the source and target graphs, respectively (column 2). They are partitioned into four layers of subgraphs with layers 1-2-3 being the common objects matched (columns 3-5), and layer 0 the unmatched background fragments (column 6). The dark line segments in columns 3-5 are the edited portions on the graphs to achieve isomorphic matching between each pair of subgraphs.

changes, but carry little information about larger object structures. Current state of the art algorithms include the iterative closest point (ICP) algorithm [28] and the soft assignment algorithm [4].

1.2.2 Category 2: Single-Layer Graph Based

Methods in this category match explicit graph structures with graph editing, such as skeleton (medial axis) graphs [37] and shock graphs [23], [27]. For recognizing flexible and articulated objects, the distance is calculated based on not only the similarity of parts, but also their connections and relative positions. Thus, graph operators are introduced with cost to edit the graphs to achieve perfect structural match. In this paper, we use the terms *graph structure* or *graph topology* for connectivity between vertices. Recently, a similar method has also applied to unsupervised learning of object categories through matching parse trees across multiple images [30]. Some recent works on shape recognition, such as shape context [3] and shape matching [31], represent the graph structures implicitly for computational efficiency.

1.2.3 Category 3: Multilayer Point Based

This category includes layered EM clustering [33], [26] for small motion and RANSAC-based methods [9], [20] for large rigid motion. A state-of-the-art algorithm in this category is the recent work by Wills et al. [35] which computes large motion segmentation using RANSAC iteratively based on local texture features.

Our method belongs to *category 4: multilayer graph-based matching with explicit graph editing*. It is aimed at more general cases arising in recent object categorization [30], unsupervised part learning, and object detection and localization in cluttered scenes. In these tasks, many of the traditional assumptions no longer hold, for example, the slow and smooth motion in layered motion [34], rigid transform and static objects in wide baseline stereo, and foreground and background segmentation in the medial-axis-based shape recognition [23], [27].

Our work is built on a series of previous works on image segmentation [32], graph partition [1], and matching [12]. In this paper, we extend the SW-cut method [1] to *composite cluster sampling* on a new candidacy graph representation (to be introduced in the next section) with both positive and negative connections, and in each step, it can move effectively in the joint space of partitioning and matching. Thus, it overcomes a major obstacle—strong coupling between local structures. A similar algorithm, called C4, is studied for scene labeling in conditional random fields and aerial image understanding by Porway and Zhu [21] that outperforms other popular algorithms, such as Belief Propagation, Gibbs sampling, and SW-cut, in general settings.

A preliminary version of this work was introduced by [13]. We provide additional algorithmic and computational details, and extend the framework considering more complete measure distance between graphs. A few examples (e.g., Figs. 1 and 17) in this paper are studied in one of our previous works [12] with a different algorithm which iteratively solves graph matching and partition in two computational dynamics.

1.3 Overview of Our Approach

In this paper, we pose the layered graph matching problem as a multicoloring task and solve it on a candidacy graph representation with a cluster sampling algorithm. Our approach includes three major components: the candidacy graph representation, the matching distance metrics, and the composite cluster sampling algorithm.

1.3.1 Candidacy Graph Representation

As Fig. 2 illustrates, given the source and target images (cropped from Fig. 1 for clarity), we compute the primal sketches and extract a number of discriminative local structural primitives, such as corners, junctions, and small curves. Some primitives are highlighted in thick line segments in Fig. 1a. Each primitive in the source graph has a number of matching candidates in the target graph.

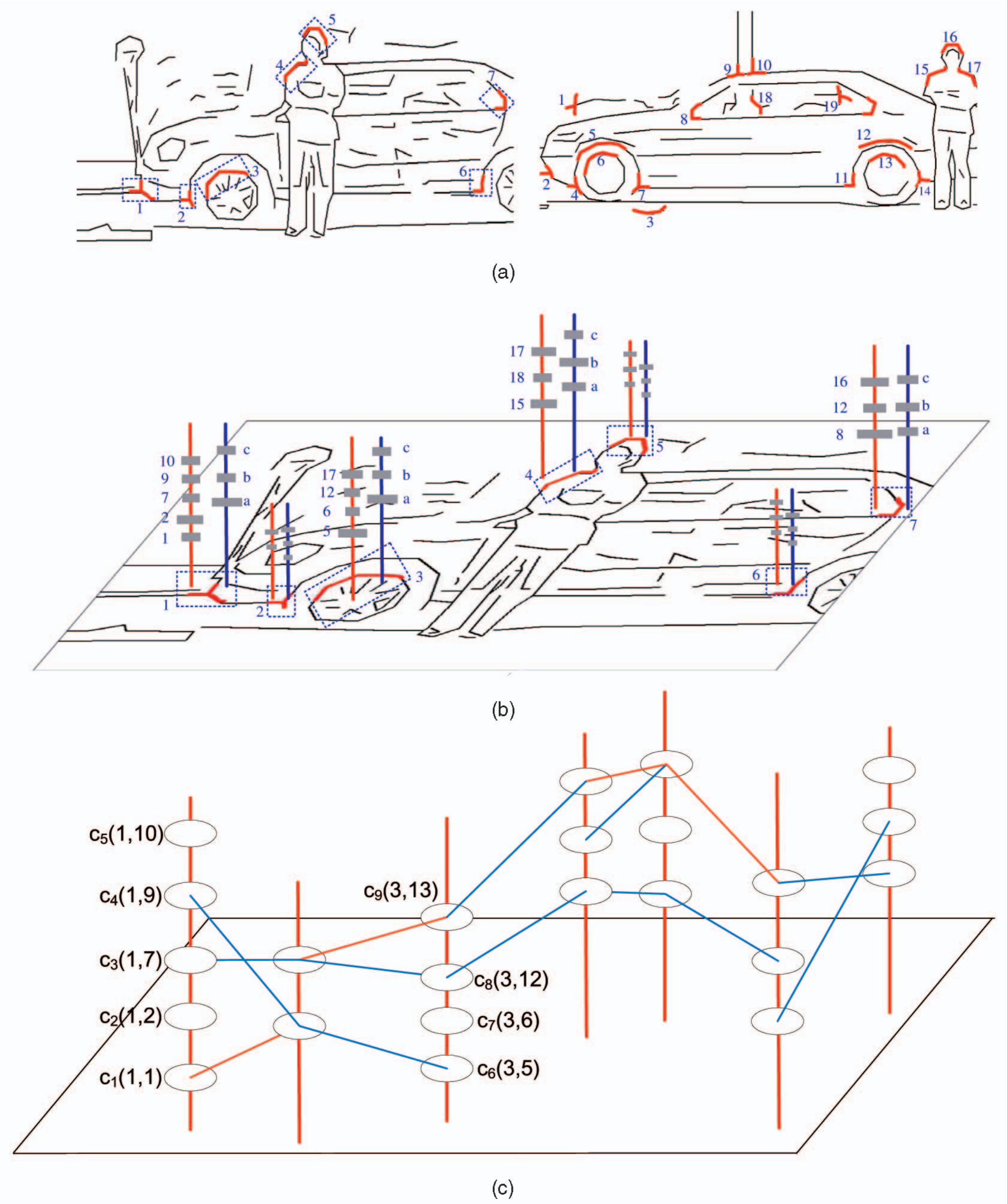


Fig. 2. Simultaneously partitioning and matching in the candidacy graph. (a) From the images cropped from Fig. 1, we compute the primal sketches and highlight some primitives in thick line segments. (b) Each primitive has a layer index (coloring) and a matching index shown by the red and blue bars, respectively, with marks representing the possible assignments. (c) The candidacy graph with positive and negative connections between two candidate matches.

For computational efficiency, we first compute layered matching for these primitives and then propagate the matching to other less discriminative line segments.

Each primitive has a layer index (coloring) and a matching (correspondence) index shown by the red and blue bars, respectively, in Fig. 2b, where the marks on the bars represent possible assignments. This search space is combinatorial. Consider two images with N and M primitives, respectively, which are to be matched in K layers, and each primitive in the source graph has a total of KM possible assignments. Thus, the solution space has $O((KM)^N)$ possible states. In fact, this space has been much reduced because the number of primitives is much smaller than the number of points/line segments in the images.

To further prune the search space, we eliminate a large number of unlikely matches and keep the most promising matching candidates as vertices in a candidacy graph, as shown in Fig. 2c. For example, primitive 1 in the source image has five candidates which are primitives $1, 2, 7, 9, 10$. Thus, we have five vertices as candidates c_1, c_2, c_3, c_4, c_5 in the candidacy graph.

Our objective is to color this graph so that some vertices are assigned K colors representing K matched subgraph for objects or parts, and the remaining vertices are set to be inactive, and thus, eliminated.

Two vertices in the candidacy graph may be linked by either a negative edge or a positive edge, which are illustrated, respectively, by red and blue lines in Fig. 2c.

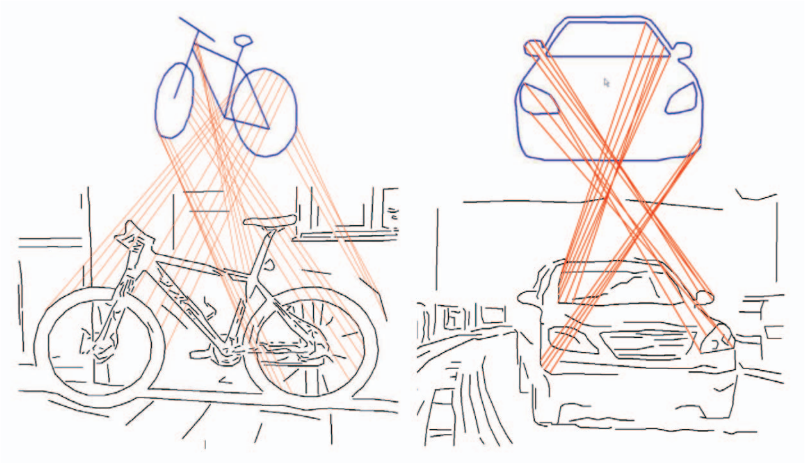


Fig. 3. Repetitive and symmetric structures in the image create strongly coupled but erroneous partial matches. The algorithm needs to simultaneously flip multiple matches to jump out from local minima.

The negative edge indicates that the two candidates are conflicting due to having the same source primitive or their target primitives overlapping, and thus, the two vertices should not both be assigned to the same color. For example, c_1 and c_2 in Fig. 2c are conflicting. The positive edge indicates that the two vertices are collaborative, and thus, enforce each other so that they are likely assigned to the same color if they have similar geometric transformations.

In summary, the candidacy graph is an effective representation since it prunes the vertices by: 1) searching for discriminative primitives and 2) eliminating nonpromising matching candidates; and it is informed by compatibility information on the edges which will be useful for driving the cluster sampling process later.

1.3.2 Distance Metrics

Each pair of matched subgraphs represents common objects or parts under geometric deformations and structural editing for occlusion. In our method, the distance measure between two matched subgraphs includes geometric deformations, appearance dissimilarities, and the cost of graph editing operators. Editing the graph is able to fix the errors caused by interobject occlusion and image clutter. The distance measure will be used to define both the global energy function and probabilities on the edges of the candidacy graph.

1.3.3 Composite Clustering Sampling

With the candidacy graph representation and distance, we study a stochastic Bayesian inference algorithm using the Markov Chain Monte Carlo (MCMC) mechanism. The algorithm iterates the following two steps: 1) Generating a composite cluster by turning on/off the positive and negative edges probabilistically. Vertices (candidates) connected by positive "on" edges form a connected component (CCP) and receive the same color. CCPs connected by negative "on" edges form a composite cluster. 2) Assigning colors to the CCPs in the composite cluster. Vertices in a CCP always receive the same color until the CCP is regrouped in the later iterations. CCPs in a composite cluster receive different colors under the conflicting constraints imposed by the negative "on" edges.

The key contribution of this composite cluster sampling approach lies in the fact that each step is a large MCMC move involving many primitives simultaneously in the joint space of matching and partition. The clustering step identifies strongly coupled local matches in the graph (see examples in Fig. 3) and the color assigning step swaps competing groups of matches. This allows us to quickly move through the search space and jump out of local minima caused by symmetric or cluttered structures and occlusion.

We apply our algorithm to the following four vision tasks and achieve state-of-the-art performance:

1. Multi-object wide-baseline matching (underlying both rigid and nonrigid motion) with occlusion.
2. Shape matching and retrieval against distortions, occlusion, and clutter.
3. Human body matching and learning parts from articulated motion.
4. Detecting and localizing objects from cluttered background.

This paper is organized as follows: We first introduce the representation in Section 2 and Bayesian formulation in Section 3, respectively. Then, Section 4 presents the inference algorithms, and Section 5 discusses a set of experiments with comparisons. Finally, the paper is concluded with discussions in Section 6.

2 Representations

In this section, we introduce three representations which are essential for layered graph matching: 1) the primal sketch graph G and primitives, 2) the mapping function Ψ for graph matching and coloring function Π for graph partition, and 3) the candidacy graph Ω.

2.1 Primal Sketching and Primitives

Given an image, we compute a primal sketch representation [10], which is an attribute graph computed in Bayesian inference:

$$G = \arg\max p(\mathbf{I}|G)p(G). \qquad (1)$$

The sketch graph is similar to an edge map, but is more suitable than the edge maps in our task for two reasons: 1) Texture edges are suppressed and 2) junctions are extracted explicitly. The computed graph G could be imperfect and need editing in the later process.

In our tasks, we are either given two images or we are given an object template (a perfect graph) plus an image. In both cases, we can assume that we have two attribute graphs G^S and G^T, and refer to them as the "source graph" and "target graph," respectively. In the original sketch graph [10], a smooth curve may be represented by a few control points for efficiency. In our work, we sample points densely on the sketches so that each line segment is of 5-7 pixels in length. We call them "linelets" as in the literature. We denote the two sets of linelets by $P = \{x\}$ and $Q = \{y\}$, respectively, and their spatial adjacency information is preserved.

From P and Q, we search for structural primitives, such as junctions, corners, and curve segments, as shown in Fig. 2a. Unlike the junctions in the primal sketch graphs, these primitives are searched simultaneously from both P and Q for potential matches. The computation of these primitives will be introduced in Section 4.1. We denote the two sets of primitives by $U = \{u\}$ and $V = \{v\}$ for P and Q, respectively.

Each primitive u (or v) consists of a few (3-7) linelets. We denote them by $u = \{x_0, x_1, \dots, x_n\} \subset P$ or $v = \{y_0, y_1, \dots, y_n\} \subset V$. These primitives are more discriminative than the single short linelets and thus have fewer matching candidates in the corresponding graph. Like superpixels for image segmentation, they largely reduce the search space.

In the algorithm, we start from matching these primitives between U and V, and then it is straightforward to propagate the matches to other less discriminative linelets in P and Q.

2.2 Graph Partition and Matching

We first define the graph partition Π with respect to the source graph G^S. It divides G^S into $K+1$ disjoint subgraphs with the unknown number of K objects:

$$\Pi = \{g_0, g_1, \dots, g_K\}. \tag{2}$$

Each subgraph g_k is a separate layer of G^S with vertex set U_k, and

$$\cup_{k=0}^{K} U_k = U, \quad U_i \cap U_j = \emptyset, \; \forall i \neq j. \tag{3}$$

All vertices in U_k receive a unique color label $l(u) = l \in \{0, 1, \dots, K\}$, $\forall u \in U_k$. Similarly, the target graph G^T is also divided into $K+1$ layers with $\cup_{k=0}^{K} V_k = V$ and primitives in each vertex set receive the same color $l(v) \in \{0, 1, \dots, K\}$, $\forall v \in V_k$.

We denote the graph matching function from the source graph G^S to the target graph G^T by

$$\Psi : U \mapsto V \cup \emptyset. \tag{4}$$

For each vertex $u \in U$, $\Psi(u) \in V$ or it has no match in V with $\Psi(u) = \emptyset$. The vertices are not matched as independent points, and graph structures (i.e., connectivity) are imposed through the distance measures in the next section.

To couple with the graph partition formulation Π, we rewrite Ψ in K matching functions:

$$\Psi_k : U_k \mapsto V_k \cup \emptyset, \quad k = 1, 2, \dots, K. \tag{5}$$

As the result of matching, both G^S and G^T are partitioned into $K+1$ pairs of subgraphs, as shown in Fig. 1:

$$(g_k, g'_k), k = 0, 1, 2, \dots, K. \tag{6}$$

g_0 and g'_0 are the background layers that are not matched. Each matched pair (g_k, g'_k), $k = 1, 2, \dots, K$, represents a common object or part with g_k transformed into g'_k by a geometric transform, a photometric transform for appearance changes, and topological graph editing operators. We denote these transforms by

$$\Phi_k = \left(\Phi_k^{\text{geo}}, \Phi_k^{\text{pho}}, \Phi_k^{\text{top}}\right). \tag{7}$$

We shall define the matching distances between g_k and g'_k based on the three aspects in Section 3.2. Note that for a matched graph pair, some vertices in U_k, V_k may still be mapped to $\emptyset$ due to partial occlusion.

2.3 Candidacy Graph Representation

For each primitive $u \in U$, as illustrated in Fig. 2b, it has two labels: a coloring index $l(u) \in \{0, 1, 2, \dots, K\}$ for layers and a matching index $\Psi(u) \in V \cup \emptyset$ for correspondence. To reduce the search scope, we prune the set of matching candidates for $u \in U$ to a small set $V(u) \subset V$. For discriminative primitives, $V(u)$ becomes small, say less than eight candidates. The computation of U, V, and $V(u)$, $\forall u \in U$, will be discussed in Section 4.1.

To solve the graph partition and matching problem simultaneously, we propose a candidacy graph representation and formulate the layered matching problem as a multiple coloring problem.

As illustrated in Fig. 2c, we define a candidacy graph $\Omega = <\mathbb{C}, \mathbb{E}>$, where each vertex $c \in \mathbb{C}$ is a possible matching pair of two primitives from the two graphs:

$$\mathbb{C} = \{c_i = (u_i, v_i) : \; u_i \in U, v_i \in V(u_i) \cup \emptyset \; i = 1, \dots, N_C\}. \tag{8}$$

The vertices are either assigned a color to denote the layer that they belong to or made inactive.

In traditional graph matching formulation, for example, Chui and Rangarajan [4], the matching is represented by a binary matrix of $|U| \times |V|$ entries with hard constraints that each row or column sums to one. In comparison, the candidacy graph prunes the number of entries to a small number of promising matching candidates and allow more than two labels.

For a pair of vertices in the candidacy graph, $c_i = (u_i, v_i)$ and $c_j = (u_j, v_j)$, we may link them with an edge $e = <c_i, c_j>$ for either a negative (conflicting) or a positive (consistent) relation. Thus, we divided $\mathbb{E}$ into two disjoint subsets:

$$\mathbb{E} = \mathbb{E}^+ \cup \mathbb{E}^-. \tag{9}$$

Negative edges $\mathbb{E}^-$. As we assume one-to-one matching between U and V, $e = <c_i, c_j>$ is called a negative edge if $u_i = u_j$ or $v_i = v_j$. That is, the two candidates c_i and c_j are mutually exclusive and thus may not be both assigned the same color. Later, we shall define a probability ρ^- on $e = <c_i, c_j>$ for how likely two candidates are to conflict with each other.

Positive edges $\mathbb{E}^+$. $e = <c_i, c_j>$ is called a positive edge if the geometric transforms between (u_i, v_i) and (u_j, v_j) are consistent and thus c_i and c_j are likely to be assigned to the same layer. Later, we define a probability ρ^+ on e based on their geometric properties for how likely they are assigned to the same layer.

2.4 Summary of the Representations

In summary, the representations of the layered graph matching problem can be written as

$$W = (K, \Pi = \{g_0, g_1, \dots, g_K\},\ \Psi = \{\Psi_k\},\ \Phi = \{\Phi_k\}), \quad (10)$$

where (K, Π) represents the partition and (Ψ, Φ) represents the matching. By introducing the candidacy graph, we integrate Π and Ψ into a coloring problem. We denote the labels of the candidacy graph $\Omega = <\mathbb{C}, \mathbb{E}>$ by

$$\mathcal{L} = \{l(c_i) = l_i :\ l_i \in \{0, 1, 2, \dots, K\}, i = 1, \dots, N_C,\ c_i \in \mathbb{C}\}. \quad (11)$$

Since we define a candidate c_i as a pair of possible matching primitives, specifying that $l(c_i)$ indicates activating a matching correspondence as well as assigning these two matched primitives with a layer. $l(c_i) = 0$ means that the candidate match c_i is made inactive. With $\mathcal{L}$, we can derive $\{(g_k, g'_k), k = 0, 1, \dots, K\}$ deterministically. We rewrite W equivalently as

$$W = (K, \mathcal{L}, \Phi = \{\Phi_k\}). \quad (12)$$

Thus, it becomes a coloring problem on the candidacy graph Ω.

3 Bayesian Formulation

In this section, we present the Bayesian formulation of layered graph matching based on the candidacy graph representation.

3.1 Maximizing the Posterior Probability

With the representations defined in the previous section, we solve the layered graph matching problem by maximizing a posterior probability:

$$W^* = \arg\ \max p(W|G^S, G^T) = \arg\ \max p(W)p(G^S, G^T|W). \quad (13)$$

We define the prior probability $p(W)$ and the likelihood $p(G^S, G^T|W)$ in the following:

Prior probability. $p(W)$ penalizes the number of layers K (i.e., complexity) and the number N of unmatched vertices in G^S and G^T so as to avoid degenerated solutions that matches all linelets as separate objects (i.e., K is too large) or inactivates all linelets as unmatched (i.e., N is too large). N can be derived from $\mathcal{L}$ deterministically:

$$p(W) \propto \exp\{-\alpha_K K - \alpha_N N\} \cdot p(\mathcal{L}), \quad (14)$$

where $\alpha_K = 1.4$ and $\alpha_N = 0.08$ are two important scale parameters representing the costs of adding a new layer or leaving a primitive unmatched, and $p(\mathcal{L})$ is a Potts model for the label $\mathcal{L}$:

$$\begin{aligned} p(\mathcal{L}) &\propto \prod_{e\in\mathbb{E}^+} \psi^+(l_i, l_j) \times \prod_{e\in\mathbb{E}^-} \psi^-(l_i, l_j), \\ \psi^+(l_i, l_j) &= \exp\{+\beta_{\mathcal{L}} \mathbf{1}(l_i = l_j \neq 0)\}, \\ \psi^-(l_i, l_j) &= \exp\{-\beta_{\mathcal{L}} \mathbf{1}(l_i = l_j \neq 0)\}, \end{aligned} \quad (15)$$

where $\beta_{\mathcal{L}}$ is set to $[0.10, 0.25]$ in our experiments. $\mathbf{1}(\cdot) \in \{0, 1\}$ is an indicator function. The probability is defined to discourage inconsistent assignments. It is maximized when candidates connected by positive edges are assigned the same label, while candidates connected by negative edges are assigned to different labels.

Likelihood. The likelihood probability of the solution is defined as

$$p(G^S, G^T|W) = \prod_{k=1}^{K} p(g_k, g'_k|\Psi_k, \Phi_k) \propto \prod_{k=1}^{K} \exp\{-E(g_k, g'_k)\}, \quad (16)$$

where $E(g_k, g'_k)$ is a distance measure between the two subgraphs g_k and g'_k given the transform Φ_k, and is defined in the following section.

3.2 Distance Measures

For a pair of graphs g_k, g'_k matched through a transform $\Phi_k = (\Phi_k^{\rm geo}, \Phi_k^{\rm pho}, \Phi_k^{\rm top})$, we define the distance or equivalently energy over the three types of attributes:

$$E(g_k, g'_k) = E^{\rm geo}(g_k, g'_k) + E^{\rm pho}(g_k, g'_k) + E^{\rm top}(g_k, g'_k). \quad (17)$$

In the following, we define the three types of distances:

Geometric distance. The geometric transform $\Phi_k^{\rm geo}$ from g_k to g'_k includes: 1) a global affine transformation S_k, 2) a residual of the matched vertices,

$$E^{\rm geo}_{\rm res}(U_k, V_k) = \sum_{u\in U_k, \Psi_k(u)\neq\emptyset} \lambda^{\rm geo}_{\rm res}((x_u - x_v)^2 + (y_u - y_v)^2), \quad (18)$$

and 3) a TPS warping for local deformation $F_k(\xi, \eta)$ in the 2D domain Λ_i covered by g_k, given the matched vertices,

$$E^{\rm geo}_{\rm TPS}(F_k) = \lambda^{\rm geo}_{\rm tps} \int\int_{\Lambda_k} \left(F^2_{k,\xi\xi} + 2F^2_{k,\xi\eta} + F^2_{k,\eta\eta}\right) d\xi d\eta. \quad (19)$$

(x_u, y_u) and (x_v, y_v) are the center points of the two primitives, which are the mean coordinates of all linelets in the primitives. In our experiments, we set $\lambda^{\rm geo}_{\rm res} = 0.35$ and $\lambda^{\rm geo}_{\rm tps} = 0.25$. This geometric distance accounts for the spatial configuration similarity of two graphs.

Then the overall energy for geometric transform is

$$E^{\rm geo}(g_k, g'_k) = E^{\rm geo}_{\rm Aff}(S_k) + E^{\rm geo}_{\rm res}(U_k, V_k) + E^{\rm geo}_{\rm TPS}(F_k). \quad (20)$$

We may drop the affine term if we allow free rigid affine transforms between objects in the two images.

Photometric distance. Let $u \in U_k, v \in V_k$ with $v = \Psi_k(u)$ be two geometrically aligned primitives in g_k and g'_k, respectively. Each primitive has $n = 3 \sim 7$ linelets. The photometric distance between them is defined by their intensity profiles along the matched linelets. The penalty for unmatched line segments, and thus, inconsistent structures between u and v, is included in the graph topological cost below.

Supposing there are n matched line segments between v and u, we denote the intensity profile perpendicular to each linelet by $\mu_j^{\rm seg}(u)$, $j = 1, \dots, n$, the photometric energy is

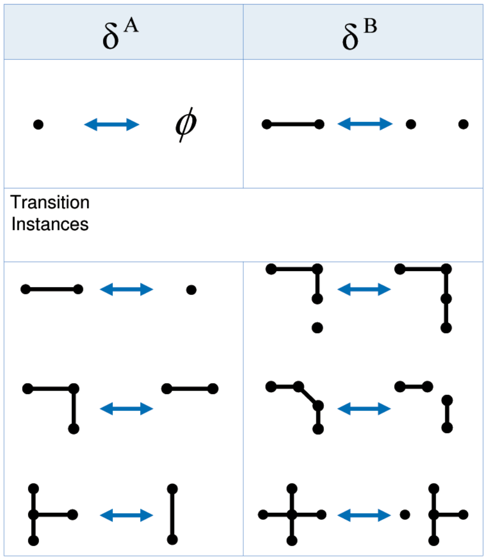


Fig. 4. Two basic graph operators for editing graphs. δ^A: adding/deleting a vertex and δ^B: adding/deleting a link or arm. Some typical examples of editing are shown under each operator.

$$E^{\text{pho}}(U_k, V_k) = \sum_{(u,v)} \sum_j \lambda^{\text{pho}} \| \mu_j^{seg}(u) - \mu_j^{seg}(v) \|^2. \tag{21}$$

The photometric distance provides cues and is weighted differently in experiments for different tasks. For motion estimation, the appearance cue is very strong, and we set $\lambda^{\text{pho}} = 0.15$. For wide-baseline stereo matching, we set $\lambda^{\text{pho}} = 0.05$. For shape matching, it is not used and we set $\lambda^{\text{pho}} = 0$.

Topological distance. Preserving graph connectivity structure is an important aspect in graph matching [33], [12], [4], especially for object localization under occlusions and learning common templates from multiple images. Editing the graphs is also needed for fixing the errors in primal sketch computation.

For a primitive $u \in U_k$ in g_k, we denote its neighbors in the image by ∂u. Suppose $v = \Psi(u) \in V$ is a matched vertex. The match Ψ_k is said to be isomorphic between U_k and V_k, if

$$\begin{aligned} &u' \in \partial u \;\Leftrightarrow\; v' \in \partial v, \quad v = \Psi_k(u), \\ &u = \Psi_k^{-1}(v), \quad \forall u \in U_k, \forall v \in V_k. \end{aligned} \tag{22}$$

If the match is not isomorphic, a number of operators have to be applied, such as adding/deleting points and line segments. Previous graph editing work includes medial axis and shock graphs [37], [23].

For computing efficiency, we define only two basic graph operators δ^A and δ^B, as shown in Fig. 4, and they are associated with costs $\text{cost}(\delta^A)$ and $\text{cost}(\delta^B)$. They are capable of correcting all topological differences.

Suppose the two operators δ^A and δ^B are used m and n times, respectively, between a pair of matched subgraphs (g_k, g'_k). More precisely, we have

$$m = \sum_{u \in U_k} \mathbf{1}(\Psi_k(u) = \emptyset) + \sum_{v \in V_k)} \mathbf{1}\big(\Psi_k^{-1}(v) = \emptyset\big), \tag{23}$$

$$\begin{aligned} n = &\sum_{<u,u'> \in E_k} \mathbf{1}(\Psi(u), \Psi(u') \notin E'_k) \\ &+ \sum_{<v,v'> \in E'_k} \mathbf{1}(<\Psi(v), \Psi^{-1}(v')> \notin E_k). \end{aligned} \tag{24}$$

In the above definitions, E_k and E'_k are the edge sets in g_k and g'_k, respectively. The topological distance for matching g to g' is

$$E^{\text{top}}(g_k, g'_k) = m \cdot \text{cost}(\delta^A) - n \cdot \text{cost}(\delta^B), \quad k = 1, 2, \ldots, K. \tag{25}$$

The costs of editing operators are application-dependent. For shape matching and retrieval with no cluttered background in Experiments I and II, we set $\text{cost}(\delta^A) = 0.8$ and $\text{cost}(\delta^B) = 0.6$. For shape localization in clutter in Experiment IV, we set $\text{cost}(\delta^A) = 0.18$ and $\text{cost}(\delta^B) = 0.10$.

4 Inference

In this section, we first introduce a bottom-up step for constructing the candidacy graph $\Omega = <\mathbb{C}, \mathbb{E}>$ from the primal sketch graphs G^S and G^T, and then present a composite cluster sampling algorithm on Ω for Bayesian inference.

4.1 Bottom-Up: Constructing the Candidacy Graph

In this section, we discuss how we extract two primitive sets $U = \{u\}$ and $V = \{v\}$ from the densely sampled linelets $P = \{x\}$ and $Q = \{y\}$ in G^S and G^T simultaneously. Then from U and V, we construct the candidacy graph.

As mentioned in Section 2.1, each primitive $u \subset U$ (or $v \subset V$) consists of a small number (3-7) of adjacent short line segments. We denote each line segment by its center position x or y in the image domain Λ. Thus, we denote $u = \{x_0, x_1, \ldots, x_a\}$ and $v = \{y_0, y_1, \ldots, y_b\}$. Fig. 5 illustrates the process of extracting U, V from P, Q collectively.

In Fig. 5, the source graph G^S is a car template shown on the top and the target graph G^T is shown to the right side. From an arbitrary line segment $x_0 \in P$, we grow x_0 into a potential primitive u through a Branch-and-Bound algorithm, as shown in Fig. 5b. This algorithm includes two key steps: "branching" to split searching space and "bounding" to prune bad candidates; it was used in matching skeleton (medial axis) graphs for object recognition by [37].

To initialize the growing process, we set $u = \{x_0\}$, and as the linelet is not discriminative, u can be matched to all linelets in Q, i.e., the set of candidate matches is $V(u) = Q$. We grow u by adding one adjacent linelet, say x_1. Then we get a longer line or curve segment, see the top of Fig. 5b. The set of possible matches $V(u)$ is reduced to a smaller set, shown by the darker primitives. With two more line segments added, we obtain an L-junction in the second row and then a Y-junction in the third row. When we grow u, we grow its matching candidates $v \in V(u)$ accordingly. A candidate v is eliminated when it can no longer find an adjacent line segment or the goodness of match falls below a threshold. By the end of this example, $u = \{x_0, x_1, \ldots, x_5\}$ includes six linelets, and the number of candidate matches in Q reduces sequentially to 5, each has six line segments: $v_i = \{y_{i0}, \ldots, y_{i5}\}$, $i = 1, 2, \ldots, 5$.

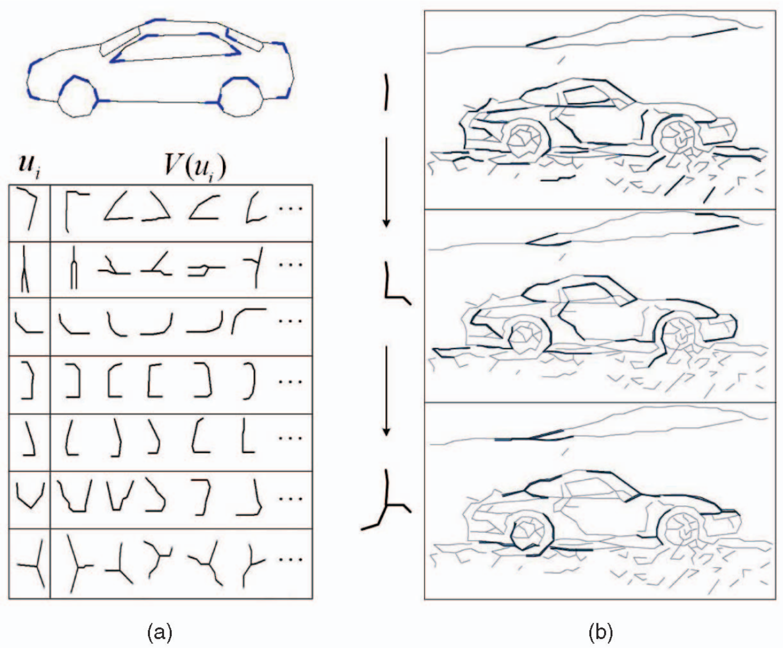


Fig. 5. Primitive searching and candidates pruning. (a) A few structural primitives detected in the source graph (template) (on the top) and detected primitives and possible matches (at the bottom). (b) The branch-and-bound progress to prune candidates.

In the Branch-and-Bound algorithm, we use the squared Procrustes distance [5] to measure the goodness of match between two primitives u and v and prune bad candidates. By writing the coordinates of $x_i = (\xi_i, \eta_i)$ in u and $y_i = (\xi'_i, \eta'_i)$ in v in complex form, namely, X and Y, respectively, we have

$$D(u, v) = 1 - \frac{|Y^* \cdot X|^2}{Y^* \cdot Y \cdot X^* \cdot X}, \tag{26}$$

where X^* and Y^* are the conjugate forms of X and Y.

The Branch-and-Bound algorithm searches for a local compact primitive u so that it is matched to a set of primitives $V(u) \subset V$. In this way, we obtain primitive sets U and V simultaneously. Fig. 5a shows some results. The first column shows examples of $u \in U$ in the car template, and the remaining primitives in each row are the candidate matches $V(u)$ found in the target sketch graph. We prune those primitives which have too many matches (say $|V(u)| > 10$) as they are less discriminative.

Once we finish extracting one primitive u, we remove from P all the linelets in u and repeat the growing process. In this way, the primitives in U do not overlap, but the primitives in V may share common linelets.

The process of primitive searching and candidacy graph construction is summarized in Fig. 6.

Once we have the set of candidates $\mathbb{C} = \{c_i = (u_i, v_i), i = 1, 2, \ldots, N_c\}$, we establish the negative and positive edges and calculate their edge probabilities in the following way. These edges will act as binary "switches" (Boolean variable) that are turned "on" or "off," as in the Swendson-Wang cut algorithm [1] to form connected components.

First, $e = <c_i, c_j>$ is connected as a negative edge in two cases: 1) The two candidates are mutually exclusive: $u_i = u_j$. It is a hard constraint that they cannot be both activated. 2) The two candidates overlap: $v_i \cap v_j \neq \emptyset$. It is a soft constraint that they probably should not be activated together. We define the probability ρ_e^- for how likely e is a negative edge:

$$\rho_e^- = \begin{cases} 1, & u_i \quad = u_j, \\ \frac{1}{Z_e} \exp\{-\lambda \|v_i \cap v_j\|\}, & u_i \neq u_j, v_i \cap v_j \quad \neq \emptyset, \end{cases} \tag{27}$$

where Z_e and λ are constants.

Second, $e = <c_i, c_j>$ is connected as a positive edge if the geometric transforms between $c_i = (u_i, v_i)$ and $c_j = (u_j, v_j)$ are consistent, and c_i and c_j are likely to be assigned to the same layer.

Fig. 7a shows two neighboring candidate matches $c_i = (A, A')$ and $c_j = (B, B')$ on the sketch graphs. We define the positive connecting probability ρ_e^+ between c_i and c_j by a local geometric distance $D_e^+(c_i, c_j)$. This ρ_e^+ is a bottom-up (or data-driven) probability which will be used to form connected components, and it does not have to be very accurate.

We first apply a similarity transformation to align c_i and c_j, as shown in Fig. 7b. Let Γ be the similarity transform between c_i and c_j, the distance between c_i an c_j can be defined as

$$D_e^+(c_i, c_j) = D(v_i, \Gamma(u_i)) + D(v_j, \Gamma(u_j)), \tag{28}$$

where $D()$ is the Procrustes distance [5] used in (26).

Then the connecting probability between c_i and c_j is defined as

$$\rho^+ \propto \exp\{-D_e^+(c_i, c_j)\}. \tag{29}$$

4.2 Composite Cluster Sampling in the Candidacy Graph

Based on the candidacy graph $\Omega = <\mathbb{C}, \mathbb{E}>$, we study a composite cluster sampling algorithm to optimize the Bayesian posterior probability, inspired by the Swendsen-Wang cut algorithm [1].

```
Input: Source Graph $G^S$ with linelets $P = \{x\}$ and target graph $G^T$ with linelets $Q = \{y\}$.
Output: Primitive sets $U = \{u\}$ and $V = \{v\}$ and candidate set graph $\Omega =< \mathbb{C}, \mathbb{E} >$.
Initialize: $C \leftarrow \emptyset$, $U \leftarrow \emptyset$, and $V \leftarrow \emptyset$.
while $P \neq \emptyset$ do
    Select a linelet $x_0 \in P$, and initialize $u = \{x_0\}$ and $V(u) = Q$.
    Grow the optimal $u$ and reduce $V(u)$ by Branch-and-Bound.
    if $|u| \geq 3$ and $V(u) \leq 10$ then
    Set $U \leftarrow U \cup \{u\}$, and $V \leftarrow V \cup V(u)$.
    for each $v \in V(u)$ do
        Set $c = (u, s)$, and $\mathbb{C} \leftarrow \mathbb{C} \cup \{c\}$,
    end
    Set $P \leftarrow P \setminus u$.
end
Establish negative and positive edges $\mathbb{E} = E^+ \cup E^-$,
Calculate edge probabilities $\rho_e^+$, $\forall e \in E^+$ and $\rho_e^-$, $\forall e \in E^-$.
```

Fig. 6. BU: primitive extraction and candidacy graph construction.

Cluster sampling is a powerful MCMC technique proposed by Swendson and Wang [29] and modified by Edwards and Sokal [6] for simulating Ising/Potts models in physics during the 1980s. At each single step, it can flip the label of multiple sites—called a "cluster" or a CCP in the Ising/Potts model. Thus, it moves effectively in the search space. It was extended to general posterior probabilities in vision by Barbu and Zhu [1], who designed the algorithm called Swendson-Wang cut. Most recently, the SW-cut algorithm is further extended to graphical models (MRF or CRF) on scene labeling with both positive and negative connections by Porway and Zhu [21] who called the algorithm $C4$. We adopt this approach to the candidacy graph for layered graph matching. We refer to [1], [21] for the technical background.

This algorithm iterates two steps:

- Step I: Generating composite clusters by turning on the edges in $\mathbb{E}^-$ and $\mathbb{E}^+$ probabilistically. Candidates connected by the positive "on" edges form a CCP. A few CCPs connected by negative "on" edges form a composite cluster.
- Step II: Reassigning colors of the CCPs in the composite cluster guided by the posterior probability and constraints.

4.2.1 Step I: Generating Composite Clusters

We introduce a Boolean variable $\omega_e \in \{1, 0\}$ on each edge $e = <c_i, c_j> \in \mathbb{E}$ as an indicator for whether the edge is turned "on" or "off." According to the cluster sampling algorithms, these variables follow the following Bernoulli probabilities:

$$\omega_e^+ \sim Bernoulli(\rho_e^+ \mathbf{1}(l(c_i) = l(c_j))), \quad \forall e \in \mathbb{E}^+, \tag{30}$$

$$\omega_e^- \sim Bernoulli(\rho_e^- \mathbf{1}(l(c_i) \neq l(c_j))), \quad \forall e \in \mathbb{E}^-, \tag{31}$$

where $\mathbf{1}(\cdot) \in \{0, 1\}$ is an indicator function.

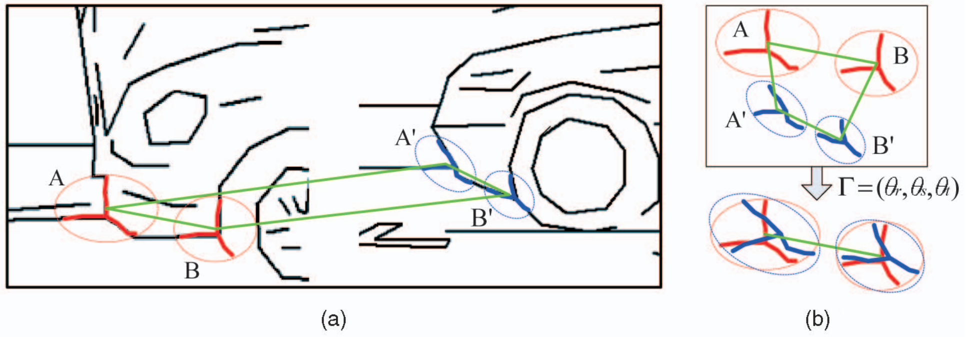


Fig. 7. (a) Measuring consistence between two matching candidates $c_i = (A, A')$ and $c_j = (B, B')$. (b) They are aligned through a similarity transform. The distance is the Procrustes measure between the alignment residuals.

```
Input: Candidacy graph $\Omega = < \mathbb{C}, \mathbb{E} >, \mathbb{E} = \mathbb{E}^+ \bigcup \mathbb{E}^-$.

Output: Composite clusters. Initialize: Turning off all edges in $\mathbb{E}$.

for each $e = < c_i, c_j > \in \mathbb{E}^+$ do
    if $(l(c_i) = l(c_j))$ then $\omega_e = 1$ with probability $\rho_e^+$
    else $\omega_e = 0$
end

Collect all CCPs which are connected by positive edges with $\omega_e = 1$.

for each $e = < c_i, c_j > \in \mathbb{E}^-$ do
    if $(l(c_i) = l(c_j))$ then $\omega_e = 0$
    else $\omega_e = 1$ with probability $\rho_e^-$
end

Collect all the composite clusters $\{V_{cc}\}$ which are connected by negative edges.
```

Fig. 8. BU-II: Generating composite clusters.

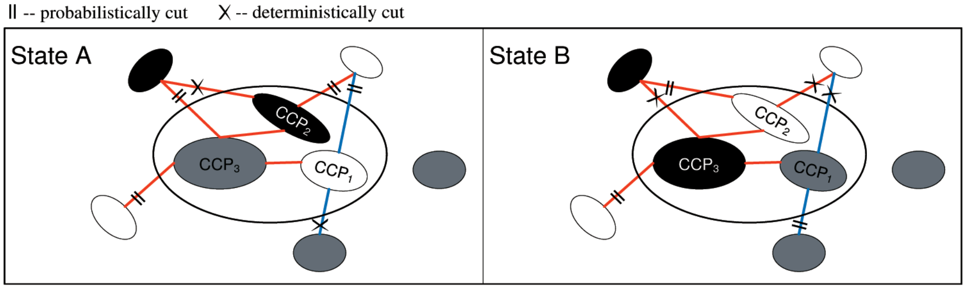


Fig. 9. Two states A and B in coloring the candidate graph. Each ellipse is a CCP within which the candidates are connected by positive edges and receive the same color. A composite cluster V_{cc} consists of multiple CCPs connected by negative edges. The algorithm reassigns colors to the CCPs inside a V_{cc} and therefore realizes a reversible jump between the two states A and B in one MCMC move.

For a positive edge $e \in E^+$, if the two candidates have the same label, i.e., $l(c_i) = l(c_j)$, then the edge is turned on (i.e., $\omega_e^+ = 1$), with probability ρ_e^+. Intuitively, if c_i and c_j are strongly coupled (consistent), they have a high chance of being connected. Otherwise, if $l(c_i) \neq l(c_j)$, then the edge is turned off with probability 1, i.e., $\rho_e^+ = 0$ deterministically. This guarantees that all candidates in a cluster have the same label. A CCP is a set of candidates which are connected by the positive edges that are turned on.

For a negative edge $e \in E^-$, at the beginning of the algorithm, we have $l(c_i) = l(c_j) = 0$; thus, the edge is off $(\omega_e^- = 0)$ deterministically. No negative edges should be turned on between candidates of the same label as they are conflicting. At a certain step, the two candidates may be assigned different colors, then the negative edge becomes active and we have $\omega_e^- = 1$ with probability ρ_e^-. Thus, for a hard constrained negative edge with $\rho_e^- = 1$, the negative edges are always "on" so as to impose a constraint that the two candidates have to be labeled differently to maintain consistency. The negative edges that are turned on will connect some CCPs into a composite cluster, denoted by V_{cc}.

We summarize this clustering step in Fig. 8.

Fig. 9 shows two states A and B in the process of coloring the candidacy graph. Each ellipse represents a CCP, and the three CCPs connected by some negative edges inside the big ellipse form a composite cluster V_{cc}. To generate this V_{cc}, some edges are cut (turned off) probabilistically (denoted by $\|$ in the figure) and some are cut deterministically and denoted by the black crosses.

4.2.2 Step II: Recoloring Candidates Inside the Composite Cluster

We choose at random a composite cluster V_{cc} from step I. For example, $V_{cc} = \{CCP_1, CCP_2, CCP_3\}$ in Fig. 9. In the second step, we reassign colors to the CCPs inside V_{cc} probabilistically, and thus, implement a reversible jump between the two states A and B in one MCMC move.

In both states A and B, candidates inside each CCP_i, $i = 1, 2, 3$, are connected by positive edges and receive the same label, while different CCPs are connected by negative edges which prevent them from receiving the same label for consistency.

The reversible jump between A and B is implemented by a Metropolis-Hastings [19] method. Let $q(A \to B)$ be the proposal probability for moving from state A to state B, and conversely, $q(B \to A)$ is the proposal probability from B to A. The acceptance rate of the move from A to B is

$$\alpha(A \to B) = \min\left(1, \frac{q(B \to A) \cdot p(W = B|G^S, G^T)}{q(A \to B) \cdot p(W = A|G^S, G^T)}\right). \quad (32)$$

```
Input: source graph $G^S$ and target graph $G^T$.

Output: Layered matching configuration $W$.

  1) Construct graph representation $G =< \mathbb{C}, \mathbb{E} >$ by algorithm **BU-I**.
  2) Set an initial state $W$ with all candidates being labelled $0$, i.e. inactivated.
  3) Cluster sampling for $W$
       a) Call algorithm **BU-II** to generate a composite cluster $V_{cc}$.
       b) Re-assign colors to $V_{cc}$ by probability $p(coloring(V_{cc})|W)$.
       c) Accept the new state $W'$ with rate $\alpha(W \to W')$.
```

Fig. 10. Algorithm for layered graph matching.

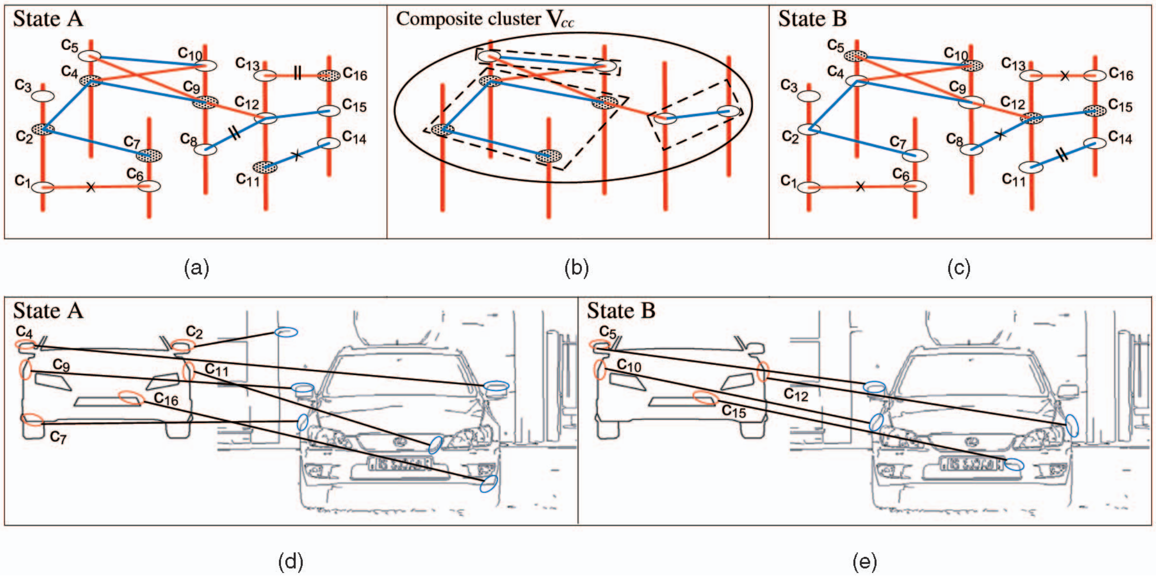


Fig. 11. The illustration of composite cluster sampling. See text for explanation.

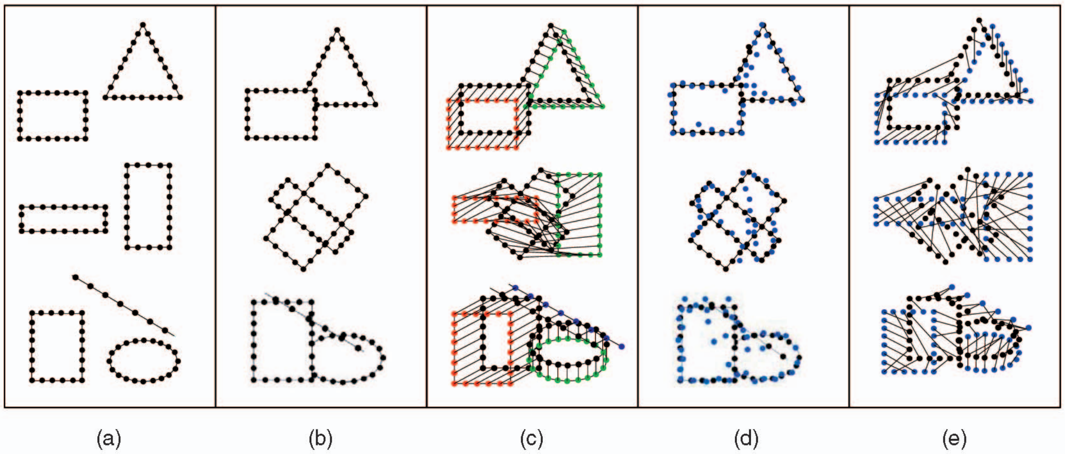


Fig. 12. Graph matching for layered motion. (a) Three source graphs, (b) three target graphs, and (c) the matching results of our method which automatically decomposes the graph into multilayers (indicating by different colors) and matches each pair with geometric and topological transform. The occluded parts are unmatched, and thus, edited. We compare with the state-of-the-art methods: (d) matching results from Chui and Rangarajan [4] and (e) matching results from shape context (Belongie et al. [3]).

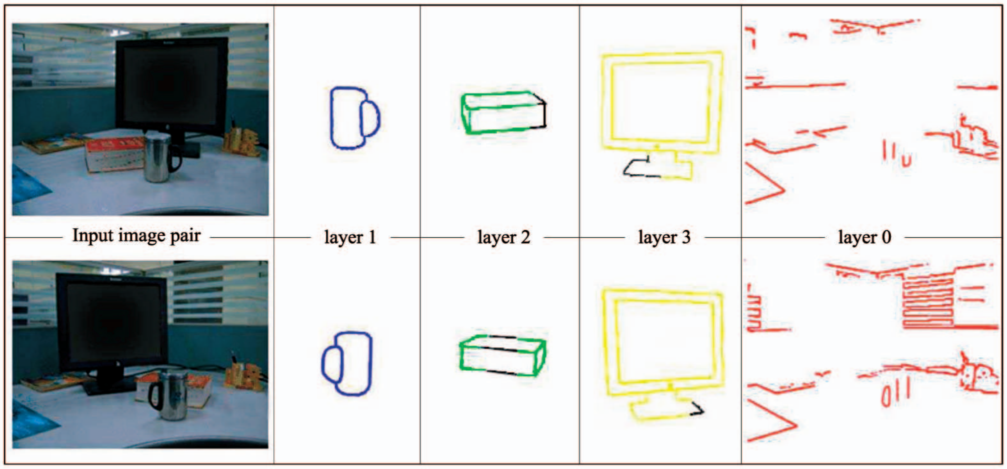
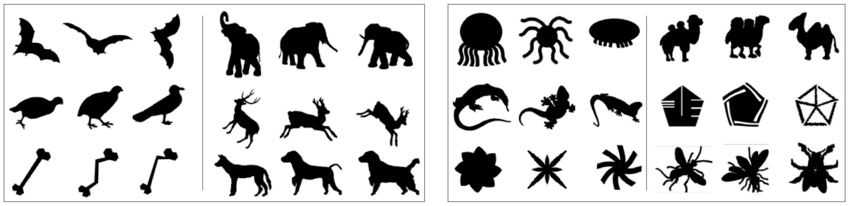


Fig. 13. Another example for simultaneous graph matching and partition. Two input images are partitioned into four layers of subgraphs with layers 1-2-3 being the common objects matched: cup, book, and computer screen, and layer 0 the unmatched backgrounds. The dark line segments are edited portions.

Methods	Recog-rate
Proposed	88.75%
Shape Tree [8]	87.70%
Hierarchical Procrustes [6]	86.35%
IDSC + DP [15]	85.40%
Data-driven EM [32]	80.03%
Curve Edit [22]	78.14%
Shape Context + TPS [3]	76.51%

(a)



(b) (c)

Fig. 14. Results for the MEPG7 CE-SHAPE-1 data set [11]. (a) Classification rate comparison. (b) Examples that the layer graph matching obtains correct results, while other approaches often fail. (c) Failed examples of the layered graph matching method.

In such a Markov chain transition, the computation cost for each move is relatively low since the computation of the posterior probability ratio only involves the recoloring of candidates in V_{cc}. The proposal probability $q(A \to B)$ is the product of two probabilities: 1) $q(V_{cc} \mid A)$—the probability of generating V_{cc} at state A and 2) $q(coloring(V_{cc}) = B(V_{cc}) \mid V_{cc}, A)$—the probability of recoloring the CCPs to state B. Therefore, we have the proposal probability ratio below:

$$\frac{q(B \to A)}{q(A \to B)} = \frac{q(V_{cc}|B) \cdot q(coloring(V_{cc} = A(V_{cc})|V_{cc}, B))}{q(V_{cc}|A) \cdot q(coloring(V_{cc} = B(V_{cc}))|V_{cc}, A))}. \tag{33}$$

In the above equation, $coloring(V_{cc})$ denotes the new colors of V_{cc}, $B(V_{cc})$, and $A(V_{cc})$ denotes the coloring of V_{cc} at states B and A, respectively.

The ratio of generating V_{cc} is similar to the Swendson-Wang cut:

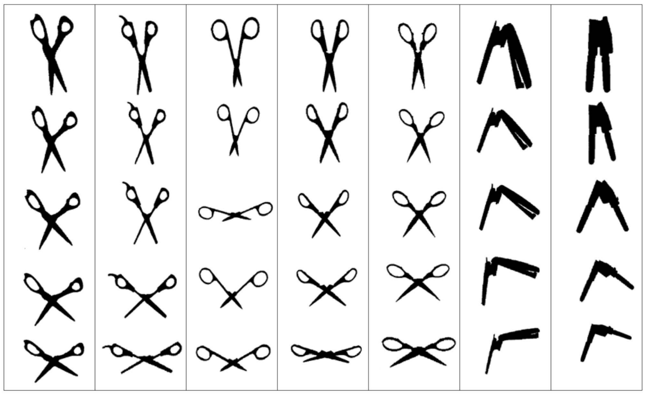
$$\frac{q(V_{cc}|B)}{q(V_{cc}|A)} = \frac{\prod_{e \in Cut_B^+}(1-\rho_e^+)\prod_{e \in Cut_B^-}(1-\rho_e^-)}{\prod_{e \in Cut_A^+}(1-\rho_e^+)\prod_{e \in Cut_A^-}(1-\rho_e^-)}, \tag{34}$$

where Cut_A^+ and Cut_A^- denote the sets of positive and negative edges which are the "cut" probabilistically around V_{cc} on state A:

$$Cut_A^+ = \{e = <c_i, c_j> \in \mathbb{E}^+ : c_i \in V_{cc}, c_j \notin V_{cc}, l(c_i) = l(c_j) \text{ at } A\}, \tag{35}$$

$$Cut_A^- = \{e = <c_i, c_j> \in \mathbb{E}^- : c_i \in V_{cc}, c_j \notin V_{cc}, l(c_i) \neq l(c_j) \text{ at } A\}. \tag{36}$$

Similarly, the sets Cut_B^+ and Cut_B^- are defined for the edges that are cut around V_{cc} probabilistically at state B.

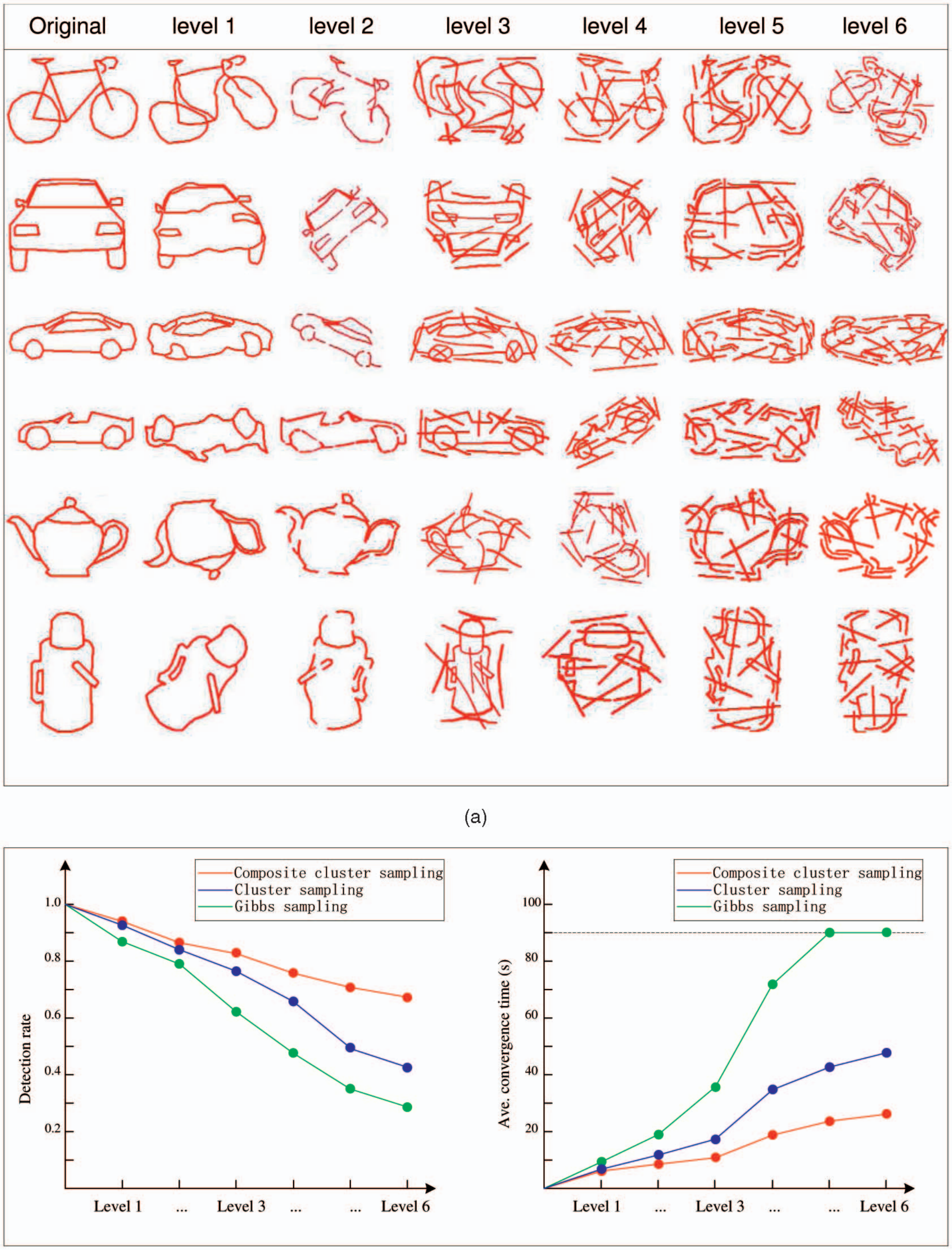


(a)

Method	Rank 1	Rank 2	Rank 3	Rank 4
SC + DP [15]	20/40	10/40	11/40	5/40
IDSC + DP [15]	40/40	34/40	35/40	27/40
Proposed	40/40	38/40	36/40	33/40

(b)

Fig. 15. The articulated shape data set from [15].



(a)

(b)

Fig. 16. Matching test on LHI-Shape data set. (a) The original shape (the first column) and different level of transformed shapes (2-7 columns). (b) The matching rates with distortion, clutter, and occlusion increasing. (c) The convergence time. The red curves denote the proposed composite cluster sampling, and the blue and green curves denote the Swendsen-wang cluster sampling [1] and Gibbs sampling, respectively.

For the detailed derivation of this equation, the reader is referred to [21]. We summarize the overall algorithm in Fig. 10.

Fig. 11 illustrates one step moving from a state A in Figs. 11a and 11d to state B in Figs. 11c and 11e by recoloring a V_{cc} in Fig. 11b, which consists of three CCPs in blue rectangles: $CCP_1 = \{c_2, c_4, c_7, c_9\}$, $CCP_2 = \{c_5, c_{10}\}$, and $CCP_3 = \{c_{12}, c_{15}\}$. In state A, $coloring(CCP_1) = 1$ where candidates are activated and $coloring(CCP_2, CCP_3) = 0$ where candidates are inactivated. The colors are flipped in state B with $coloring(CCP_1) = 0$ and $coloring(CCP_2, CCP_3) = 1$.

By flipping the colors, some local matches due to symmetry are corrected in one step. Thus, it overcomes the coupling problem raised in Fig. 3.

In Fig. 11, the blue edges are positive, while the red edges are negative. The crosses and $\|$ mark the edges that

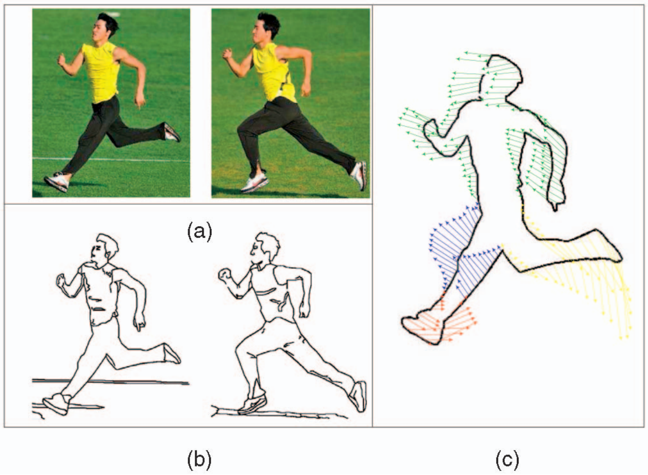


Fig. 17. Articulate motion analysis and part learning. (a) Two far frames in a video sequence, (b) the sketch graphs, and (c) four body parts are matched and segmented in four layers (red, yellow, blue, and green).

are turned off by algorithm **BU-I** in generating the composite cluster.

5 Experiments

We evaluate the layered graph matching algorithm in four tasks:

1. multi-object wide-baseline matching under both rigid and nonrigid motions with occlusion,
2. shape similarity matching and retrieval,
3. human body matching and segmentation of parts from articulated motion, and
4. object detection and localization in cluttered background.

In each task, we compare with the state-of-the art methods in the literature. The algorithm is implemented by C++ on a PC with Core Duo 2.8 GHZ CPU.

Experiment I. We first test the layered graph matching algorithm on some artificial examples of layered large motion with opaque and transparent occlusion. The three pairs of source and target graphs are shown in Figs. 12a and 12b. In these examples, no photometric information is used and the connected components are assigned colors under a uniform likelihood probability during the partition sampling. The matching results are shown in Fig. 12c, where the occluded line segments are recovered. For comparison, Fig. 12d shows the single-layer matching results produced by the state-of-the-art graph matching algorithm [4], and Fig. 12e displays the shape context [3] matching results. As one may expected, these single-layer matching methods do not work in such examples.

Figs. 1 and 13 show three experiment results for layered wide-baseline matching in real images. In Fig. 1, the cars have different appearance and are in slightly different poses occluded by a person. They are matched, and the occluded segments are recovered. Fig. 13 shows another similar example in indoor scene.

Experiment II. In the second experiment, we test our method on three data sets for shape matching and retrieval.

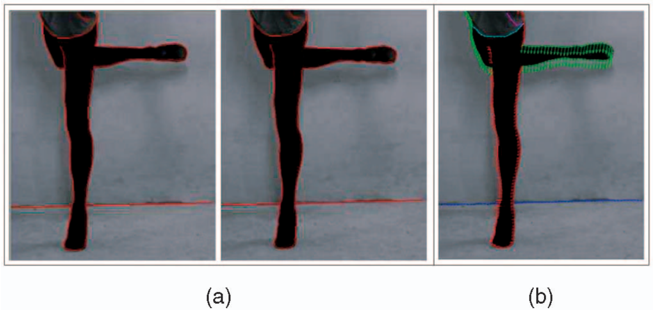


Fig. 18. One contour motion analysis example by Liu et al. [16]. (a) and (b) Two testing images (with sketch graphs) and motion estimation result.

The first data set is the MPEG7 CE-Shape-1 data set [11]. This database contains 70 types of objects, each of which has 20 different instances, giving a total of 1,400 binary silhouettes. According to the Bull's eye criterion [11], we look at the 40 most similar images and count how many of those are in the same class as the query image. The recognition rate is reported in Fig. 14a, and our method outperforms the existing approaches due to the layered representation, which accounts for some articulated deformations, as shown in Fig. 14b. This data set is quite challenging due to the large intraclass variability, and a few failure examples from six classes are shown in Fig. 14c.

The second data set tests the matching of articulated objects. Introduced by [15], it consists of eight objects with five shapes each (Fig. 15a). The criteria follow [15]: For each shape, the four most similar matches are selected and the number of correct hits for ranks 1-4 is counted. The result is presented in Fig. 15b. Our method performs well due to the explicit layer representation that models large articulation.

The third data set tests more difficult shapes with internal edges in contrast to the silhouettes in the two data sets above. This data set contains five categories with three instances each, selected from the LHI data set [36]. The testing shapes are created from original shapes by applying distortion, spurious clutter, and occlusion. The testing shapes are divided into six levels of complexity. Level 1 only adds nonrigid deformation, level 2 adds 15 percent occlusion, and level 3 adds spurious clutter (15 percent compared to the original shape). Levels 4-6 increase occlusion and clutter by both 5 percent in a stepwise manner. Each level contains six testing shapes and the typical shapes are selected in Fig. 16. The task is to match the original sketch shapes ($5 \times 3 = 15$) to the testing shapes ($5 \times 3 \times 6 \times 4 = 360$). Matchings with more than 80 percent correct pairs of points are counted as correct for each pair. Figs. 16b and 16c show the detection (correct matching) rate and convergence time at each testing level. We compare with the Swendsen-Wang sampling algorithm [1] and traditional Gibbs sampler.

Experiment III. We demonstrate that the layered graph matching algorithm can be used in articulate motion analysis and learning object parts. Fig. 17a shows two far frames extracted in a video sequence and the corresponding sketch graphs are shown in Fig. 17b. Four body parts are matched and segmented in different layers (red, yellow, blue, and green) in Fig. 17c. This example runs in 62 seconds.

	Tu [32]		Liu [17]		**Proposed**	
Cate.	T. P.	F. P.	T. P.	F. P.	T. P.	F. P.
Bicycle	0.69	0.17	0.75	0.14	0.88	0.10
F. Car	0.75	0.15	0.81	0.11	0.94	0.08
S. Car	0.69	0.19	0.69	0.13	0.81	0.13
Horse	0.50	0.21	0.62	0.11	0.75	0.07
Teapot	0.62	0.25	0.81	0.21	0.81	0.13

Fig. 19. Comparison results of object localization on real images, as shown in Fig. 20. The true positive rate and false positive rate are reported for each method. The data are selected from LHI data set [36].

For comparison, we also compute the result on an example of contour motion analysis in Fig. 18. This example was originally used in [16].

Experiment VI. We test the algorithm on object detection and localization from natural images. We select 80 images containing five object categories from the LHI data set [36]. We draw one template for each category. For each image, we compute the sketch graph and match it against the five templates. There are thus $5 \times 80 = 400$ matches in total and a match is said to be correct if the template is registered at the correct position in the images. Fig. 20 shows some matching results together with failure examples in the bottom raw. The overall hit rate and false alarm rate are reported in Fig. 19 with comparison to matching algorithms by Tu et al. [31] and Liu et al. [17]. The horse category has relatively lower detection rate due to articulations. A 91 percent detection rate can be achieved if we use three horse templates.

In addition to this data set, an early version of our layered graph matching method was used as an independent module for top-down verification in the object recognition framework [14].

6 Conclusion

In this paper, we study a layered graph matching method for integrating graph partition and matching with graph editing, and demonstrate its applications in a series of vision tasks in comparison with state-of-the-art approaches in the literature. We formulate the problem in a candidacy graph representation and adopt a composite cluster sampling algorithm for inference. The representation is augmented with bottom-up (data-driven) information in terms of both positive and negative links. The cluster sampling algorithm overcomes the problem of combinatorial search space by constructing large MCMC moves and thus can jump from local minimums caused by symmetry, clutter, and occlusion.

Our method can segment and match common structures in an unsupervised way and thus has the potential for automatically learning object categories and their parts. Experiments I and III demonstrate this capability in learning and matching objects (cars, humans, etc.) and parts (human body) from articulated or layered motion. In future research, we will further investigate this problem

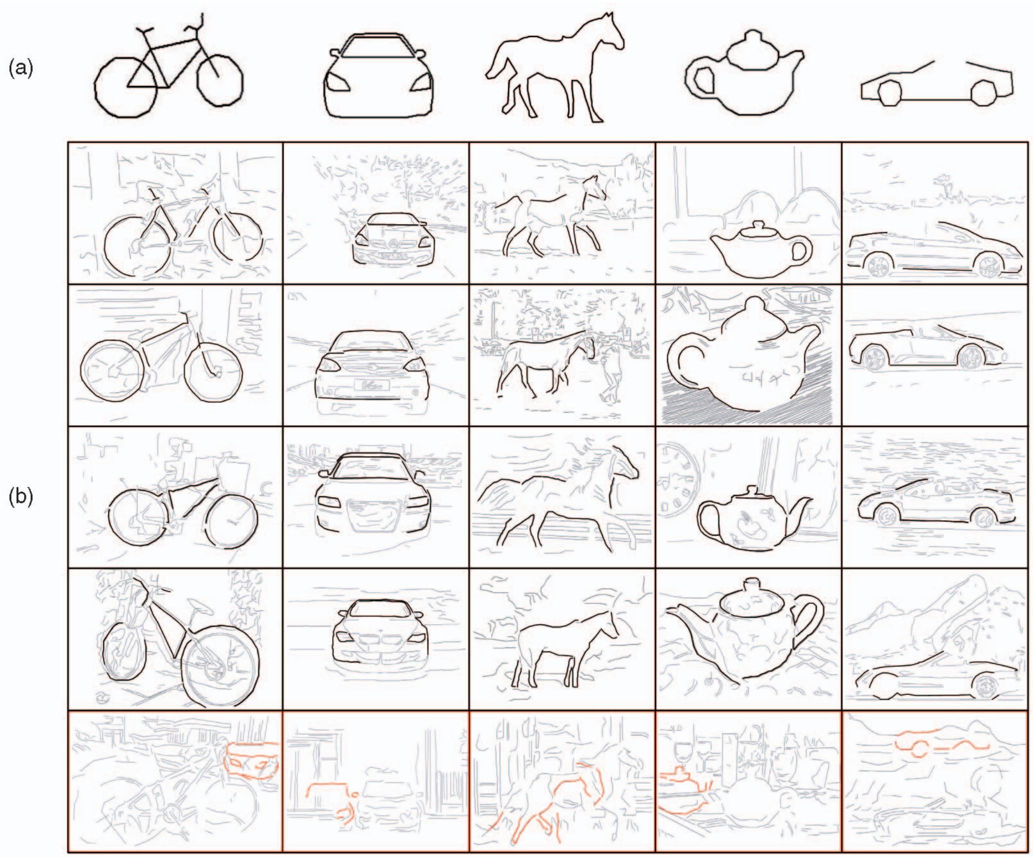


Fig. 20. Object detection and localization from cluttered background. (a) Templates of five objects. (b) Some localization results by a two-layer graph matching method. The failure results are marked by red color in the bottom row.

with a substantial data set and integrate the matching and learning with the hierarchic representation, such as And-Or graph and stochastic image grammar.

ACKNOWLEDGMENTS

The authors would like to thank Ziqiang Liu and Han Lv for their assistance in experiments, and they thank Lotus Hill Institute for the annotated data set. The work at UCLA is supported by the US National Science Foundation (NSF) grant IIS-0713652, and the work at LHI is supported by NSFC grants 60728203, 60970156, and a China 863 grant 2007AA01Z340.

REFERENCES

[1] A. Barbu and S.C. Zhu, "Generalizing Swendsen-Wang to Sampling Arbitrary Posterior Probabilities," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 8, pp. 1239-1253, Aug. 2005.

[2] H. Bay, V. Ferraris, and L.V. Gool, "Wide Baseline Stereo Matching with Line Segments," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 329-336, 2005.

[3] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 4, pp. 509-522, Apr. 2002.

[4] H. Chui and A. Rangarajan, "A New Point Matching Algorithm for Non-Rigid Registration," *Computer Vision and Image Understanding,* vol. 89, no. 2, pp. 114-141, 2003.

[5] I. Dryden and K. Mardia, *Statistical Shape Analysis.* John Wiley and Sons, 1998.

[6] R. Edwards and A. Sokal, "Generalization of the Fortuin-Kasteleyn-Swendsen-Wang Representation and Monte Carlo Algorithm," *Physical Rev. D,* vol. 38, no. 6, pp. 2009-2012, 1988.

[7] P.F. Felzenszwalb and J.D. Schwartz, "Hierarchical Matching of Deformable Shapes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[8] R. Feris, R. Raskar, L. Chen, K. Tan, and M. Turk, "Discontinuity Preserving Stereo with Small Baseline Multi-Flash Illumination," *Proc. IEEE Int'l Conf. Computer Vision,* vol. 1, pp. 412-419, 2005.

[9] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM,* vol. 24, pp. 381-395, 1981.

[10] C.E. Guo, S.C. Zhu, and Y.N. Wu, "Primal Sketch: Integrating Texture and Structure," *Computer Vision and Image Understanding,* vol. 106, no. 1, pp. 5-19, 2007.

[11] L. Latecki, R. Lakamper, and T. Eckhardt, "Shape Descriptors for Non-Rigid Shapes with a Single Closed Contour," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 424-429, 2000.

[12] L. Lin, S.C. Zhu, and Y. Wang, "Layered Graph Match with Graph Editing," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 885-892, 2007.

[13] L. Lin, K. Zeng, X. Liu, and S.C. Zhu, "Layered Graph Matching by Composite Cluster Sampling with Collaborative and Competitive Interactions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[14] L. Lin, S. Peng, J. Porway, S.C. Zhu, and Y. Wang, "An Empirical Study of Object Category Recognition: Sequential Testing with Generalized Samples," *Proc. Int'l Conf. Computer Vision,* vol. 1, pp. 419-426, 2007.

[15] H. Ling and D.W. Jacobs, "Shape Classification Using the Inner-Distance," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 2, pp. 286-299, Feb. 2007.

[16] C. Liu, W.T. Freeman, and E.H. Adelson, "Analysis of Contour Motions," *Advances in Neural Information Processing Systems,* MIT Press, 2006.

[17] X. Liu, H. Li, W. Tao, and H. Jin, "Layered Shape Matching and Registration: Stochastic Sampling with Hierarchical Graph Representation," *Proc. IEEE Conf. Pattern Recognition,* 2008.

[18] D.G. Lowe, "Distinctive Image Features from Scale Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[19] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chemical Physics,* vol. 21, no. 6, pp. 85-111, 1953.

[20] D. Nister, "Preemptive RANSAC for Live Structure and Motion Estimation," *Proc. Int'l Conf. Computer Vision,* vol. 1, pp. 199-206, 2003.

[21] J. Porway and S.C. Zhu, "C4: Stochastic Inference in Graphical Models with Positive and Negative Edges," technical report, Dept. of Statistics, Univ. of California Los Angeles, www.stat.ucla.edu/~sczhu/papers/C4_TR.pdf, Dec. 2008.

[22] T. Sebastian, P. Klein, and B. Kimia, "On Aligning Curves," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 1, pp. 116-124, Jan. 2003.

[23] K. Siddiqi, A. Shokoufandeh, S.J. Dickenson, and S.W. Zucker, "Shock Graphs and Shape Matching," *Proc. Int'l Conf. Computer Vision,* vol. 1, pp. 222-229, 1998.

[24] D.G. Sim, K. Oh-Kyu, and P. Rae-Hong, "Object Matching Algorithms Using Robust Hausdorff Distance Measures," *IEEE Trans. Image Processing,* vol. 8, no. 3, pp. 425-429, 1999.

[25] J. Shi and C. Tomasi, "Good Features to Track," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 593-600, 1994.

[26] P. Smith, T. Drummond, and R. Cipolla, "Layered Motion Segmentation and Depth Ordering by Tracking Edges," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 4, pp. 479-494, Apr. 2004.

[27] T.B. Sebastian, P.N. Klein, and B.B. Kimia, "Recognition of Shapes by Editing Their Shock Graphs," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 5, pp. 550-571, May 2004.

[28] G.C. Sharp, S.W. Lee, and D.K. Wehe, "ICP Registration Using Invariant Features," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 1, pp. 90-102, Jan. 2002.

[29] R.H. Swendsen and J.-S. Wang, "Nonuniversal Critical Dynamics in Monte Carlo Simulations," *Physical Rev. Letters,* vol. 58, no. 2, pp. 86-88, 1987.

[30] S. Todorovic and N. Ahuja, "Unsupervised Category Modeling, Recognition, and Segmentation in Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 12, pp. 2158-2174, Dec. 2008.

[31] Z. Tu, S. Zheng, and A.L. Yuille, "Shape Matching and Registration by Data-Driven EM," *Computer Vision and Image Understanding,* vol. 103, no. 3, pp. 290-304, 2007.

[32] Z. Tu and S.C. Zhu, "Image Segmentation by Data-Driven Markov Chain Monte Carlo," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 657-673, May 2002.

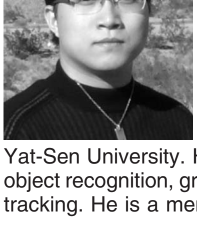
[33] Y. Weiss, "Smoothness in Layers: Motion Segmentation Using Nonparametric Mixture Estimation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 520-526, 1997.

[34] Y. Weiss and E.H. Adelson, "Slow and Smooth: A Bayesian Theory for the Combination of Local Motion Signals in Human Vision," MIT Technical Report AI Memo 1624, 1998.

[35] J. Wills, S. Agarwal, and S. Belongie, "A Feature-Based Approach for Dense Segmentation and Estimation of Large Disparity Motion," *Int'l J. Computer Vision,* vol. 68, no. 2, pp. 125-143, 2006.

[36] B. Yao, X. Yang, and S.C. Zhu, "Introduction to a Large Scale General Purpose Groundtruth Dataset: Methodology, Annotation Tool, and Benchmarks," *Proc. Int'l Conf. Energy Minimization Methods in Computer Vision and Pattern Recognition,* pp. 169-183, 2007.

[37] S.C. Zhu and A.L. Yuille, "Forms: A Flexible Object Recognition and Modeling System," *Int'l J. Computer Vision,* vol. 20, no. 3, pp. 187-212, 1996.



Liang Lin received the BS and PhD degrees from Beijing Institute of Technology (BIT) in 1999 and 2008, respectively. He studied in the Department of Statistics at the University of California, Los Angeles (UCLA), as a visiting scholar during 2006-2007. He was a postdoctoral research fellow at the Center for Image and Vision Science (CIVS) at UCLA, and a research scientist at Lotus Hill Institute (www. lotushill.org). He is now an associate professor at Sun Yat-Sen University. His research interests include but are not limited to object recognition, graph and shape matching, image parsing, and visual tracking. He is a member of the IEEE.

Xiaobai Liu is currently working toward the PhD degree in the Department of Computer Science, Huazhong University of Science and Technology. He was a research associate at the Lotus Hill Research Institute during 2007-2008, and was a research associate in the Department of Electrical and Computer Engineering, National University of Singapore, during 2008-2009. His research is concentrated in the areas of computer and human vision, machine learning, and large-scale image retrieval. He has published more than 10 articles on object recognition, multimedia retrieval, image classification/labeling, visual learning, perceptual organization, and performance analysis. He is a member of the IEEE.

Song-Chun Zhu received the BS degree from the University of Science and Technology of China in 1991, and the MS and PhD degrees from Harvard University in 1994 and 1996, respectively. He is currently a professor in the Department of Statistics and the Department of Computer Science, University of California, Los Angeles (UCLA). Before joining UCLA, he worked in the Division of Applied Math at Brown University, Department of Computer Science at Stanford University, and Department of Computer Science at Ohio State University. His research interests include computer vision and learning, statistical modeling, stochastic computing, vision, and visual arts. He has published more than 100 papers in computer vision. He has received a number of honors, including the J.K. Aggarwal Prize from the International Association of Pattern Recognition in 2008, the David Marr Prize in 2003, the Marr Prize honorary nominations in 1999 and 2007, a Sloan Fellowship in Computer Science in 2001, the US National Science Foundation Early Career Development Award in 2001, and the US Office of Naval Research Young Investigator Award in 2001. In 2005, he founded, with friends, the Lotus Hill Institute for Computer Vision and Information Science in China as a nonprofit research organization (www.lotushill.org). He is a member of the IEEE.