Layered Graph Matching by Composite Cluster Sampling with Collaborative and Competitive Interactions

Liang Lin[1,2,3], Kun Zeng[2], Xiaobai Liu[2], Song-Chun Zhu[2,3]
[1]Beijing Institute of Technology, Beijing, China, linliang@bit.edu.cn
[2]Lotus Hill Research Institute, Ezhou, China, {kzeng, xbliu}@lotushill.org
[3]Dept. of Statistic and Computer Science, University of California, Los Angeles, sczhu@stat.ucla.edu

Abstract

This paper studies a framework for matching an unknown number of corresponding structures in two images (shapes), motivated by detecting objects in cluttered background and learning parts from articulated motion. Due to the large distortion between shapes and ambiguity caused by symmetric or cluttered structures, many inference algorithms often get stuck in local minimums and converge slowly. We propose a composite cluster sampling algorithm with a "candidacy graph" representation, where each vertex (candidate) is a possible match for a pair of source and target primitives (local structure or small curves), and the layered matching is then formulated as a multiple coloring problem. Each two vertices can be linked by either a competitive edge or a collaborative edge. These edges indicate the connected vertices should/shouldn't be assigned the same color. With this representation, the stochastic sampling contains two steps: (i) Sampling the competitive and collaborative edges to form a composite cluster, in which a few mutual-conflicting connected components are in different colors; (ii) Sampling the new colors to this cluster remaining consistency with Markov Chain Monte Carlo (MCMC) mechanism. The algorithm is applied to many applications on many public datasets and outperform the state of the art approaches.

1. Introduction

The objective of layered graph matching in this paper is to find and match an unknown number of common graph structures (objects or parts) in two images (shapes or edge map) – a problem arising in recent object recognition research, such as learning object parts from articulation motion and detecting object-of-interest from background clutters, as Fig. 1 illustrates.

Literature review. We review the related work in two aspects: graph (shape) representation and inference algorithm.

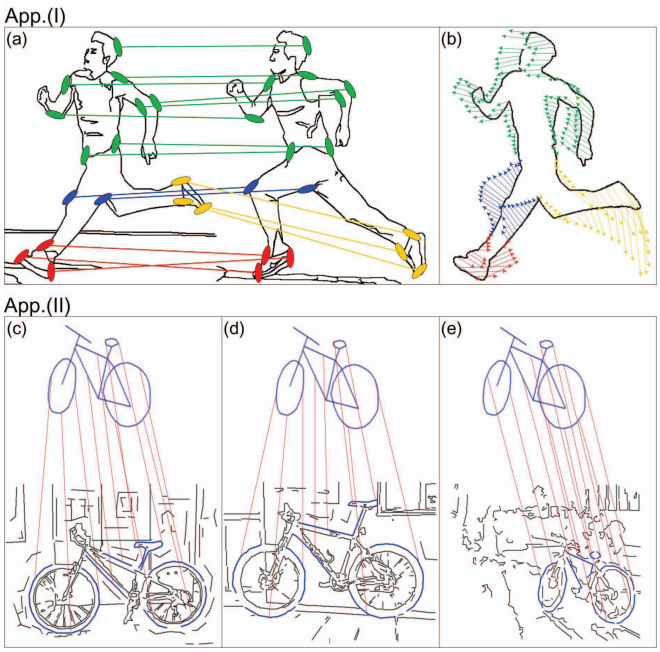

Figure 1. Two representative applications of the proposed layered graph matching. (I): Piecewise matching (shown in (a)) of human body parts based on articulated motion, as shown in (b), and each color denotes a body part. (II): Object-of-interest objects with cluttered background are detected and localized by two layer matching, as shown in (c), (d) and (e).

Representation of graph (shape) in many approaches is usually defined on spatial configurations of a small number of attribute primitives that can be the interval-sampled points from object silhouette or boundary [2, 17, 22], interesting points [10, 3], or small curves and line segments [20, 22, 11]. Some of them measure shape similarity directly with discriminative distances, such as Hausdorff [20], Chamfer matching [4] and Procrustes [6], and these methods work fast but are limited by noise sensitivity or without solving correspondences explicitly. Then some researchers present methods on using flat graph structure

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

with implicit or explicit consistency, such as shape context [2], iterative closest point [17], generative shape matching [22], skeleton (medial axis) graphs [25], and shock graphs [19]. Recently more compact graph structure, such as the layered or hierarchical representation [12, 7, 8, 14] are further proposed to account for complex non-rigid deformable matching. Recently, one promising work [23] learns a better metric through graph transduction by propagating the model through existing shape matching distances.

Inference algorithm with these graph representations in the literature however is complex due to the high instability of such structures and the combinatorial complexity of the search, and they can be roughly divided into heuristic search and statistical procedure. The former drives the exhaustive computing with various initializations, such as the "softassign" [5], data-driven EM [22], and the dynamic programming with its variance [7, 13, 21]; the latter performs sampling via designed solution space, such as the RANSAC [16], Gibbs sampling [19, 12, 14], and Swendsen-Wang Cuts [1].

Overview. In this paper, we pose a layered matching problem as a multi-coloring task in a "**candidacy graph**" by a composite cluster sampling algorithm.

We first search and prune a number of structural primitives (such as corners, junctions, and small curves) on two attribute graphs. Then we define a candidacy graph representation, where each vertex is a possible matching pair of two primitives from two graphs. Each two vertices can be linked by either a **Competitive** edge or a **Collaborative** edge. These edges indicate the connected vertices should/shouldn't be assigned the same color. The competitive edges are defined based on two neighboring vertices sharing the source or target image domains. The collaborative edges account for similar geometric transformation between two connected vertices.

With this representation, we present a stochastic algorithm using the Markov Chain Monte Carlo (MCMC) mechanism [15]. Each move in the Markov Chain contains two iterative steps: (i) Generating a composite cluster by turning on/off the collaborative and competitive edges probabilistically. Each cluster includes a few connected components (CCPs) that are linked by competitive edges, and all vertices (candidates) in a CCP are connected by collaborative edges. (ii) Assigning colors to the composite cluster, where all vertices in a CCP are received the same color, compatible with the neighboring CCPs. This keeps the consistency by the competitive and collaborative interactions.

The key contribution of this paper is a layered graph matching algorithm that employs a compositional clustering sampling for improved inference. Our algorithm identifies strongly coupled matches in the graph and swaps competing groups of matches. This allows us to quickly move through the search space and jump out of local minimums caused by symmetric, cluttered structures and occlusion. We also add some bottom-up methods to narrow the solution space effectively, which greatly improves the algorithm. We apply our algorithm to: (i) shape matching against the distortion, occlusion and background clutters, and (ii) detection and localization for object-of-interest from images. The benefit of using composite cluster is also illustrated by comparison with the Gibbs sampler and cluster sampling method [1].

One graph matching algorithm from our group [12] also employs stochastic sampling for inference. There are two main different aspects: (i) [12] solves graph partitioning and matching independently in two MCMC dynamics, and we integrate these two problems in a joint solution; (ii) [12] uses Gibbs sampler to match correspondence resulting in lower efficiency, and our algorithm is able to assign multiple colors to composite cluster in one sampling step.

The remainder of this paper is arranged as follows. We first present the representation and problem formulation in Sect. 2, and follow with a description of the inference algorithm in Sect. 3. The experimental results are shown in Sect. 4 and the paper concludes with a summary in Sect 5 .

2. Representation and Formulation

2.1. Candidacy graph representation

Given a source graphs $G^S = (\mathcal{X}, A^S)$ and a target graph $G^T = (\mathcal{Y}, A^T)$ to be matched into multiple layer, $\mathcal{X}$ and $\mathcal{Y}$ are dense point sets with attributes (orientation, position and degree) A^S and A^T from two graphs. We first find a number of graph primitives U and V from two graphs respectively, which are local structures, such as junctions, corners, and small curves, as shown in Fig. 2 (a). These primitives are used to reduce solution dimension, just like superpixels for image segmentation. This will be introduced later in Sect. 3.1. Then we define a candidacy graph representation via possible candidates. Each candidate includes a pair of possible matched source and target primitives.

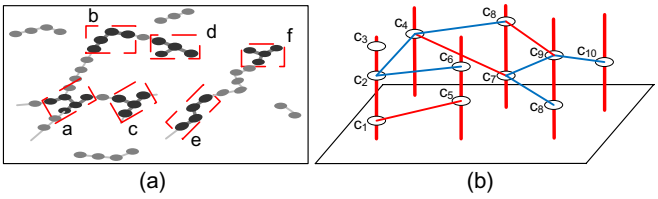

Figure 2. Candidacy graph representation via matching candidates. Each graph vertex is a candidate (indicated by the empty ellipse in the figure), which is a possible match for a pair of source and target primitives. Each edge is a probabilistic connection to account for either collaborative or competitive interactions, illustrated by the blue or red links respectively.

As illustrated in Fig. 2 (b), 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 two graphs, as

$$\mathbb{C} = \{c_i = (u_i, v_i), i = 1, \ldots, N_C\}, \quad u_i \in U, v_i \in \{V(u_i) \cup \emptyset\}, V(u_i) \subset V \tag{1}$$

where N_C denotes the candidate number and $V(u_i)$ is a subset of target primitive set V after pruning. The vertices can be assigned a color to denote which layer of deformation it belongs to, or it is turn off (inactivated). As shown in Fig. 1 (a), each matching link can be viewed as a candidate and its color denotes the layer number of articulated motion. For a pair of vertices, $c_i = (u_i^s, v_i^t)$ and $c_j = (u_j^s, v_j^t)$, we define a probabilistic edge link $e =< c_i, c_j >$ to indicate the competitive or collaborative interactions between them.

Competitive edge: We assume the one-to-one (bijective) matching between U and V, and we thus define competitive edges for the mutual exclusion constraint that the two candidates should not be both "turn on" (activated) when they have the same source primitive or their target primitives overlap in image domain. And the connecting probability ρ_e^- of the competitive edge is defined as

$$\rho_e^- = \begin{cases} 1, & u_i^s = u_j^s \\ \frac{1}{Z_{\rho^-}} \exp\{-|| \wedge (v_i^t) - \wedge(v_j^t)||\}, & u_i^s \neq u_j^s \end{cases} \tag{2}$$

where $\wedge(v_i^t)$ and $\wedge(v_j^t)$ denote the corresponding image domain of the primitive v_i^t and v_j^t. Z_{ρ^-} is a normalize term.

Collaborative edge: We reasonably assume spatial configuration of the source primitives and target primitives are similar in one matching layer. Therefore, we define the collaborative edge by a few pairwise geometric descriptors that encode the pairwise similarity of primitives. The connecting probability of collaborative edge ρ_e^+ is then defined as

$$\begin{aligned} \rho_e^+ &= \rho(e = \text{"on"} | H(u_i^s, u_j^s), H(v_i^t, v_j^t)) \\ &\propto \exp\left(-\frac{Hamm(H(u_i^s, u_j^s), H(v_i^t, v_j^t))}{T}\right) \end{aligned} \tag{3}$$

where $H(u_i^s, u_j^s)$ and $H(v_i^t, v_j^t)$ denote the geometrical descriptors, and $Hamm(\cdot, \cdot)$ is the parametric Hamming distance. T is a temperature factor. In practice, the descriptor H contains 3 types of robust features against background clutters:

- **Relative angles**, illustrated in Fig. 3 (a), is defined on relative orientation (θ_A and θ_B) of the line joining the two mass centers of primitives (A and B) with respect to the incident tangents at the curve of each primitive.
- **Relative replacement of mass center**, illustrated in Fig. 3 (b), is defined on the distance d_M of M_{AB} to the connecting line between primitive A and B, where M_{AB} is the mass center of dense points in the circle. The diameter of the circle is the connecting line. Note the other structure inside the circle is also accounted to calculate the mass center.
- **Inner distance**, proposed by [13], is the length of the shortest path within the silhouette of the shape.

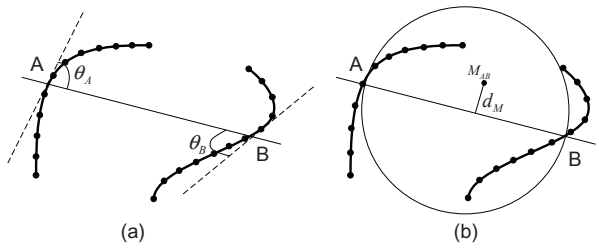

Figure 3. Two typical pairwise features for collaborative connecting probability.

2.2. Bayesian formulation

With the candidacy graph $\Omega =< \mathbb{C}, \mathbb{E} >$, the layered matching problem can be formulated as assigning colors to candidates with their collaboration and competition. We introduce a labeling set $\mathcal{L}$ to record the coloring (number of deformation layer) for all vertices. Letting the candidate number is N_C, we define

$$\mathcal{L} = \{l_i = l(c_i) \in [0, K], i = 1, \ldots, N_C, c_i \in \mathbb{C}\} \tag{4}$$

where K is an unknown number of deformation layer. And a multi-layer transformation is defined based on the coloring as $\Phi = \{\Phi_k, k = 1, \ldots, K\}$. We thus define the solution W in Bayesian framework and solve it by maximizing a posterior probability as,

$$W = \{N, K, \mathcal{L}, \Phi\} \tag{5}$$

$$\begin{aligned} W^* &= \arg\max P(W | G^S, G^T) \\ &\propto \arg\max P(W) P(G^S, G^T | W) \end{aligned} \tag{6}$$

where N denotes the number of unmatched primitives, and it can be calculated based on the number of colored candidates from the labeling set $\mathcal{L}$.

Prior term. Considering the inter-properties of mutual consistency or exclusion (competitive and collaborative interactions) between graph vertices, we introduce a graphical model as a prior distribution for the graph coloring matrix $\mathcal{L}$. We assume K and N are both small numbers and each prior distribution for them can be formed as an exponential term. Letting the other parameters in solution W are uniform in prior, we thus define the prior term as,

$$\begin{aligned} P(W) &= P(K)P(N)P(\mathcal{L}) \\ &\propto \exp\{-\alpha_K K\} \exp\{-\alpha_N N\} \cdot P(\mathcal{L}) \end{aligned} \tag{7}$$

where α_K and α_N are tuning parameters and are set empirically. Letting $e =< c_i, c_j >$ denotes a edge that connects two vertices c_i and c_j (as illustrated in Fig. 2 (b)), we define

$$P(\mathcal{L}) \propto \prod_{e \in \mathbb{E}^+} \psi^+(l_i, l_j) \times \prod_{e \in \mathbb{E}^-} \psi^-(l_i, l_j) \tag{8}$$

$$\begin{aligned} \psi^+(l_i, l_j) &= \exp\{+\beta_{\mathcal{L}} \mathbf{1}(l_i = l_j)\} \\ \psi^-(l_i, l_j) &= \exp\{-\beta_{\mathcal{L}} \mathbf{1}(l_i = l_j)\} \end{aligned}$$

where $\beta_{\mathcal{L}} > 0$ is the tuning parameter. $\mathbf{1}(\cdot) \in \{0,1\}$ is an indicator function for a Boolean variable. The probability is maximized when all collaborative edges have their vertices matched and all competitive edges have differently colored vertices.

Likelihood term. We define the geometric transformations $\{\Phi_k = (A_k, F_k), k = 1, \ldots, K\}$ to measure the similarity of two matched graphs, which are calculated on dense graph points $\mathcal{X}$ and $\mathcal{Y}$ of two graphs G^S and G^T. A_k indicates a global affine transformation and F_k is a smooth non-rigid transformation. Given a pair of matched primitives $c_i = (u_i^s, v_i^t)$, we can deduce two corresponding point sets, as $\Psi : X = \{x : x \in u_i^s\} \rightarrow Y = \{y : y \in v_i^t\}$. For the corresponding point set in each layer, we obtain

$$\Psi_k : X_k = \{x : x \in \cup_{l_i=k} u_i\} \rightarrow Y_k = \{y : y \in \cup_{l_i=k} v_i\} \quad (9)$$

where $X_k \subset \mathcal{X}, Y_k \subset \mathcal{Y}$. Then the likelihood probability of the solution thus can be written as

$$P(G^S, G^T|W) = \prod_{k=1}^{K} P(\Phi_k(A_k, F_k)|\Psi_k) P(\Psi_k|\mathcal{L}) \quad (10)$$

where we ignore the detailed description of affine and TPS transformation in $\Phi_k(A_k, F_k)$.

3. Stochastic Inference

As defined in Eq. 7, we find the joint solution of multiple coloring (including the graph partition and matching) is quite large and complex. Hence we present an efficient composite cluster sampling algorithm for inference. It contains two bottom-up steps and an MCMC sampling step.

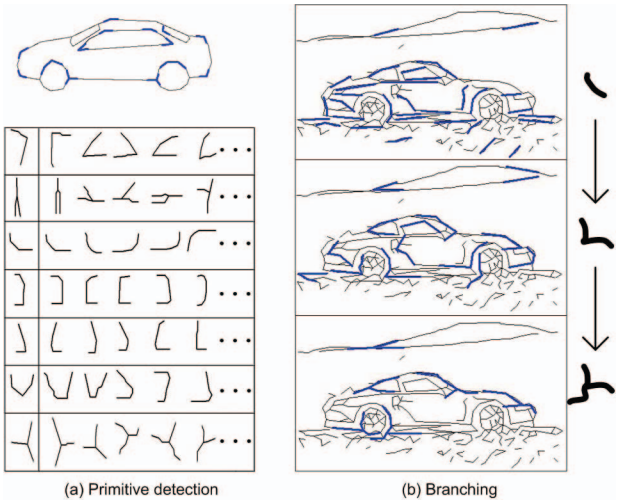

Figure 4. Primitive searching and candidates pruning. (a) shows a few structural primitives detected in the source graph (template) and detected primitives and possible matches; (b) shows the branching-and-bound progress to prune candidates.

3.1. Bottom-up I: Candidate search and pruning

It is straightforward to extract the initial structural primitives set $\hat{U}$ from the template (source) graph G^S, as illustrated in Fig. 4 (a). Then we apply a branch-and-bound algorithm for each initial primitive to output its candidates. Since each source primitive contains a set of dense points $\{x\}$, this algorithm searches from a arbitrary point x of one source primitive and branch to others while pruning the bad matches on the target graph to obtain each candidate target primitives. We assume each point x from a source graph G^S is possibly matched to all points in target graph G^T, as $M(x) = \mathcal{Y}$.

Fig. 4 (b) illustrates three intermediate results of growing a primitive. This algorithm contains three main repetitive steps: (i) **Select** a point x_j randomly from the point set $\mathcal{O}$ of primitive u_i into a seed set $\mathcal{S}$; (ii) **Branch** the seed set to neighboring points and keep all possible matches into a temporary set C_s ; (iii) **Prune** the bad candidates using a bound cost $BCost(\cdot)$, and update the candidate set $\mathbb{C}$. Note the primitives with too few or too many candidates will be also pruned.

We use the squared Procrustes distance [6] as bound cost $BCost(\cdot)$, by representing two matched points set X_s and Y_t into complex form $J(X_s)$ and $J(Y_t)$, as

$$BCost(X_s, Y_t) = 1 - \frac{|J(Y_t)^* \cdot J(X_s)|^2}{J(Y_t)^* \cdot J(Y_t) \cdot J(X_s)^* \cdot J(X_s)} \quad (11)$$

where $J(X_s)^*$ and $J(Y_t)^*$ are the conjugate values of $J(X_s)$ and $J(Y_t)$. The process of candidate searching and pruning is summarized in Alg. 1.

Input: Source Graph $G^S = (\mathcal{X}, A^S)$, target graph $G^T = (\mathcal{Y}, A^T)$
Output: Candidate set: $\mathbb{C} = \{c_i, i = 1, \ldots, N_C\}$
Detect initial primitive set on G^S, denoted as $\hat{U}$;
for *each* $u \in \hat{U}$ **do**
  Obtain point set from u, $X = \{x_j, j = 1, \ldots, n_u\}$;
  $\mathcal{O} = \{x_j, j = 2, \ldots, n_u\}$, $\mathcal{S} = \{x_1\}$, $C_s = \{M(x_1)\}$
  while $\mathcal{O} \neq \emptyset$ **do**
    Select $x \in \mathcal{S}$, $x' \in \partial x$ and $x' \in \mathcal{O}$;
    Branch $\mathcal{S} = \mathcal{S} \cup \{x'\}$, $\mathcal{O} = \mathcal{O} \backslash \{x'\}$,
    $C_s = C_s \times M(x')$;
    for *each* $s \in C_s$ **do**
      if $BCost(s, \mathcal{S}) > \zeta$, then $C_s = C_s \backslash \{s\}$;
    end
  end
  for *each* $s \in C_s$ **do**
    $c = (u, s)$, $\mathbb{C} = \mathbb{C} \cup \{c\}$
  end
end

Algorithm 1: BU-I: Candidate search and pruning.

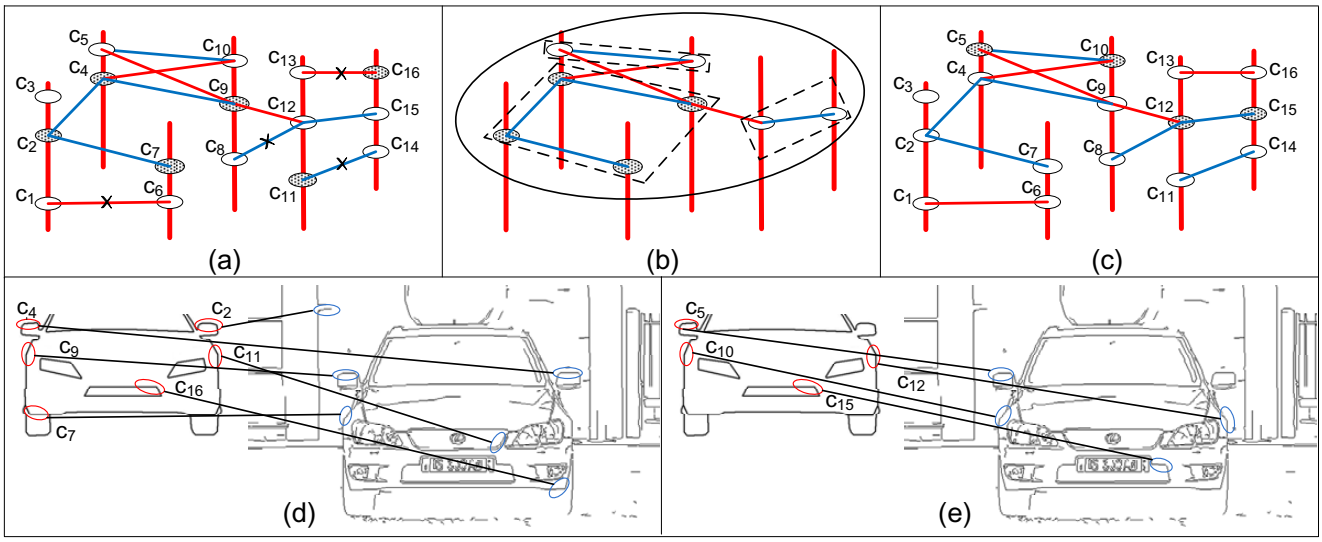

Figure 5. The illustration of composite cluster sampling. Note we only use two-color (white or dashed dark) sampling for illustration, and the color number is K for general layered matching. (a) shows a current state and the edge links are turned off (by the black "cross") deterministically or probabilistically, following the rules defined in Alg. 2; (b) shows a composite cluster (in the black ellipse) including three conflicting connected components (CCPs) (in the blue rectangles), and each one includes vertices connected by collaborative (blue) links while any two neighboring CCPs are connected by a competitive (red) link; (c) is the new state resulting from the re-coloring in the composite cluster, and all vertices in a CCP are received the same color compatible with other neighboring CCPs; (d) and (e) are the real solutions corresponding the state (a) and (c).

3.2. Bottom-up II: Composite cluster generation

Based on the candidacy graph $\Omega =< \mathbb{C}, \mathbb{E} >$, the second step bottom-up computation is to further narrow the solution space by generating a composite cluster, inspired by the Swendsen-wang cluster sampling [1]. But here the important difference are: (i) We incorporate both collaborative and competitive edges, indicating the connected vertices should/shouldn't be assigned the same label; (ii) The composite cluster formed by a few conflicting connected components (as shown in Fig. 5 (b)) can be re-colored at once to remain consistency.

Input: Candidacy graph $\Omega =< \mathbb{C}, \mathbb{E} >, \mathbb{E} = \mathbb{E}^+ \bigcup \mathbb{E}^-$
Output: A composite cluster V_C
for *each* $e =< c_i, c_j > \in \mathbb{E}^-$ **do**
  Select $\omega_{i,j}$ from connecting state set $\mathcal{E}$;
  if $(l_i = l_j)$ then $\omega_{i,j} = 0$;
  else $\omega_{i,j} = 1$ with probability $\rho^-_{i,j}$;
  Update $\mathcal{E}$ with $\omega_{i,j}$;
end
for *each* $e =< c_i, c_j > \in \mathbb{E}^+$ **do**
  Select $\omega_{i,j}$ from connecting state set $\mathcal{E}$;
  if $(l_i = l_j)$ then $\omega_{i,j} = 1$ with probability $\rho^+_{i,j}$;
  else $\omega_{i,j} = 0$;
  Update $\mathcal{E}$ with $\omega_{i,j}$;
end
Randomly select a V_C;

Algorithm 2: BU-II: composite cluster generation.

We introduce an variable $\mathcal{E} = \{\omega_{i,j} :< c_i, c_j > \in \mathbb{E}\}$ and $\omega_{i,j} \in \{1, 0\}$, indicating the edge state "on" or "off". For each probabilistic link $e =< c_i, c_j >$, we define the sampling protocol for edge states as

$$\omega^+_{i,j} \sim Bernoulli(\rho^+_{i,j}\mathbf{1}(l_i = l_j)) \quad (12)$$

$$\omega^-_{i,j} \sim Bernoulli(\rho^-_{i,j}\mathbf{1}(l_i \neq l_j)) \quad (13)$$

That means the collaborative edges connecting two vertices with different colors can be turned off deterministic (like the link $< c_{11}, c_{14} >$ in Fig. 5 (a)), as well as the competitive edges with two same color vertices, (like the link $< c_1, c_6 >$ in Fig. 5 (a)). In the other two cases that collaborative edges with the same color vertices and competitive edges with different color vertices, these edges can be turned by the probability $\rho^+_{i,j}$ and $\rho^-_{i,j}$ respectively (like the link $< c_{13}, c_{16} >$ and $< c_8, c_{12} >$ in Fig. 5 (a)).

Then generating a composite cluster is to sample the edge probability of coloring state $\mathcal{L}$ and then uniformly select a composite group of connected vertices. Thus the bottom-up proposal for V_C can be calculated by the probability of "turning off" the edges (as the "black cross" in Fig. 5 (a)) around the composite cluster, as

$$Q(V_C) = \prod_{e \in \mathbb{E}^+ \bigcup \mathcal{C}} (1-\rho^+_{i,j}) \prod_{e \in \mathbb{E}^- \bigcup \mathcal{C}} (1-\rho^-_{i,j}) \quad (14)$$

where $\mathcal{C}$ is a set of the edges been "turned off" around V_C. The edge connecting probability $\rho^-_{i,j}$ and $\rho^+_{i,j}$ are defined in Eq. 3 and Eq. 2 respectively.

A representative composite cluster V_C with three conflicting connected components (CCPs) is shown in Fig. 5

(b), and all vertices in each CCP should be received the same color together compatible with the other neighboring ones. Fig. 5 (d) and (e) demonstrate a representative example of matching swapping and re-coloring at one sampling step, taking fully advance of composite cluster.

The module of composite cluster generation in our framework is summarized in Alg. 2.

3.3. MCMC sampling

Given a current state A in solution space and a generated composite cluster V_C, we search for the new matching state B by re-coloring the V_C, as illustrated by Fig. 5. In each step, the algorithm assigns colors to each vertices of V_C, subject to the competitive and collaborative constrains. It simulates a ergodic and aperiodic Markov chain which visits a sequence of states of the joint space over time. Hence the basic goal is to realize a reversible jump between any two successive states of graph coloring (like the Fig. 5 (a) and Fig. 5 (c), with the corresponding real cases in Fig. 5 (d) and Fig. 5 (e) respectively), and acceptance of the new state is decided by the Metropolis-Hastings [15] mtehod to guarantee the convergence of the inference algorithm. We thus define the acceptance rate of reversible jump of two successive states A and B, as

$$\alpha(A \to B) = \min\left(1, \frac{Q(B \to A) \cdot P(B)}{Q(A \to B) \cdot P(A)}\right) \tag{15}$$

In such a Markov chain transition, the computation cost for each move is low, since we only need to calculate the posterior probability ratio $P(B)/P(A)$. $Q(B \to A)$ and $Q(A \to B)$ are proposal probability of generating and coloring the composite cluster V_C, as

$$\frac{Q(B \to A)}{Q(A \to B)} = \frac{Q(C(V_C)|V_C, B) \cdot Q(V_C|B)}{Q(C(V_C)|V_C, A) \cdot Q(V_C|A)} \tag{16}$$

where $C(V_C)$ denotes the new labeling of V_C. Since we assume the matching layer number is small, the re-coloring of V_C can be achieved by weighted uniformly sampling to guarantee ergodicity [15, 1]. We can thus assign colors to the composite cluster randomly subject to the consistency of each CCP (as shown in Fig. 5), and $Q(C(V_C)|V_C, A)/Q(C(V_C)|V_C, B)$ can bed canceled. The ratio of generating cluster can be simplified according to Eq. 14 as

$$\frac{Q(V_C|B)}{Q(V_C|A)} = \frac{\prod_{e \in \mathbb{E}^+ \bigcup \mathcal{C}_B}(1-\rho_e^+)\prod_{e \in \mathbb{E}^- \bigcup \mathcal{C}_B}(1-\rho_e^-)}{\prod_{e \in \mathbb{E}^+ \bigcup \mathcal{C}_A}(1-\rho_e^+)\prod_{e \in \mathbb{E}^- \bigcup \mathcal{C}_A}(1-\rho_e^-)} \tag{17}$$

The overall description for this layered graph matching framework is summarized in Fig. 6.

4. Experiments

We evaluate the proposed algorithm on these aspects: (i) Pure shape similarity matching, including close-contour shapes and articulated shapes; (ii) Cluttered sketch shape matching; (iii) Human body learning from articulated motion; (iv) Object-of-interest detection from real images.

Input: source graph G^S and target graph G^T.
Output: Layered matching configuration W.

1. Construct graph representation $G =< \mathcal{V}, \mathbb{E} >$
   (a) Call **Module BU-I 1** for candidate searching and pruning $\mathbb{C} = \{c_i, i = 1, \ldots, N_C\}$.
   (b) Computation for collaborative or competitive connection probability between candidates $\mathbb{E} = \mathbb{E}^+ \bigcup \mathbb{E}^-$.
2. Set an initial state $W = W_0$.
3. Sampling loop for W
   (a) Call **Module BU-II 2** to generate a composite cluster V_C.
   (b) Sample the colors to V_C remaining consistency to achieve a new state W'.
   (c) Calculate the acceptance rate $\alpha(W \to W')$

Figure 6. The sketch of proposed layered graph matching.

Table 1. Recognition rate on MEPG7 CE-SHAPE-1 [11].

Methods	Recog-rate
Proposed	88.75%
Shape Tree [7]	87.70%
Hierarchical Procrustes [6]	86.35%
IDSC+DP [13]	85.40%
Data-driven EM [22]	80.03%
Curve Edit [18]	78.14%
Shape Context+TPS [2]	76.51%

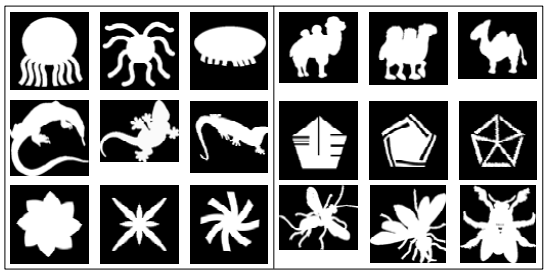

Figure 7. A few failure examples from 6 categories on MEPG7 CE-SHAPE-1 dataset [11].

Experiment I. We first test our method on the MPEG7 CE-Shape-1 dataset [11]. This database contains 70 types of objects each of which has 20 different instances, giving a total of 1400 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 the Tab. 1, and our method outperform the alternatives. This dataset is quite challenging due to the large intra-class variability, and a few failure examples of 6 categories are shown in Fig. 7.

We also test our algorithm using the articulation dataset introduced in [13] which consists of 8 objects with 5 shapes each (Fig. 8). The criteria is followed [13]: for each shape, 4 most similar matches are selected and the number of correct hits for ranks $1 \sim 4$ are counted. As reported in Tab. 4, our method performs well owing to explicitly dealing with large

articulations by layered matching.

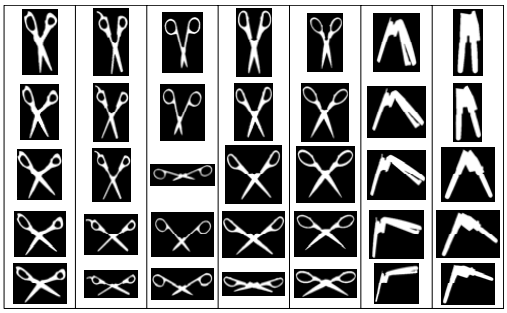

Figure 8. The articulated shape dataset from [13].

Method	Rank 1	Rank 2	Rand 3	Rank 4
SC + DP [13]	20/40	10/40	11/40	5/40
IDSC + DP [13]	40/40	34/40	35/40	27/40
Proposed	40/40	38/40	36/40	33/40

Table 2. Matching results on articulated shape dataset from [13].

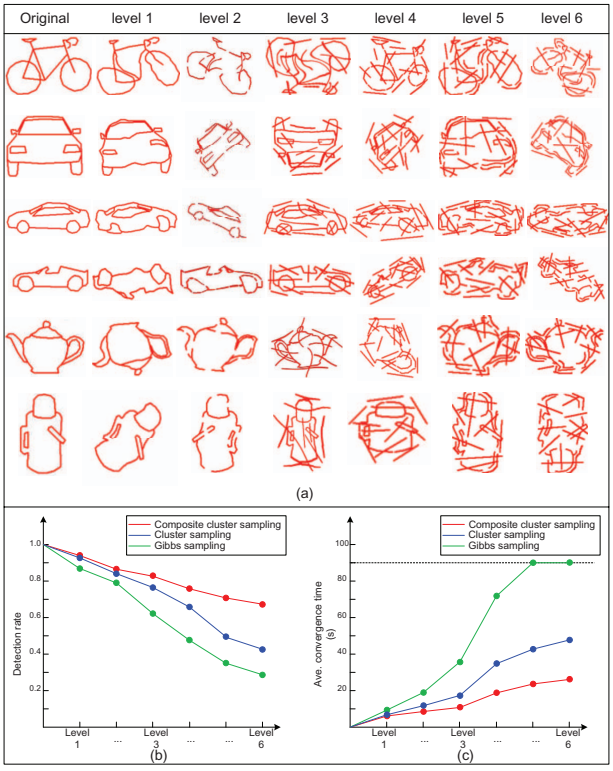

Figure 9. Matching test on LHI-Shape dataset. (a) shows the original shape (the first column) and different level of transformed shapes ($2 \sim 7$ columns); (b) shows the matching rates with distortion, clutters, and occlusion increasing; (c) shows the convergence time. The red curve denotes the proposed composite cluster sampling, and the blue and green curves denote the Swendsen-wang cluster sampling [1] and Gibbs sampling respectively.

Experiment II. We further test the algorithm on more difficult sketch shapes with full inter-structure, using the LHI dataset [24]. This dataset contains 5 categories with 3 source instances each. The testing shapes are created from source shapes with randomly applied distortion, spurious clutters, and occlusion (missing edges). The testing shapes are divided into 6 levels of shape complexity. The level 1 only adds non-rigid deformation, the level 2 adds 15% occlusion, and level 3 adds spurious clutters (15% compared to the original shape). level $4 \sim 6$ increase occlusion and clutters by both 5% stepwisely. Each level contains 6 testing shapes and the typical shapes are selected in Fig. 9. The task is to match the original sketch shapes ($5 \times 3 = 15$) with testing shapes ($5 \times 3 \times 6 \times 4 = 360$). Matching with more than 80% correct pairs of points is counted as correct for each testing. In Fig. 9 (b) and (c), we 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 try to apply this layered graph matching algorithm for human parts learning. Fig. 1 (a) shows an example of body part partitioning based on articulated matching. Then we further test on more cartoon images with various pose. As shown in Fig. 10, a set of body parts are partitioned and matched (in the left) crossing images (in the following). We show the potential of our algorithm and will present more objects/parts learning works in the future.

Experiment IV. The algorithm is also applied on object-of-interest localization from nature images. We select a number of natural images of 5 categories from LHI dataset [24] and manually draw the sketch templates in the top row of Fig. 11. Using a edge detector [9], we match template to corresponding category image and one correct matching is counted when the template is registered at the correct position of images. Some representative matching results and failure examples (marked by red color) are shown in Fig. 11. Object detection using layered graph matching will extensively studied in the future work.

5. Summary

We propose a graph matching framework to match two attribute graphs in multiple layers, by composite cluster sampling in a candidacy graph representation. By forming a composite cluster, our algorithm is able to assign multiple colors in each iteration step, resulting in a reversible "swap" operator. It can be viewed as a "big move" for jumping from the local minimums caused by symmetric, cluttered structures and occlusion, and achieve fast convergence. The advantage of proposed algorithm is demonstrated. We will further study it with object/parts learning in the future work.

6. Accknowledgment

This work is done at the Lotus Hill Institute and is supported by China 863 Program (Grant No. 2007AA01Z340

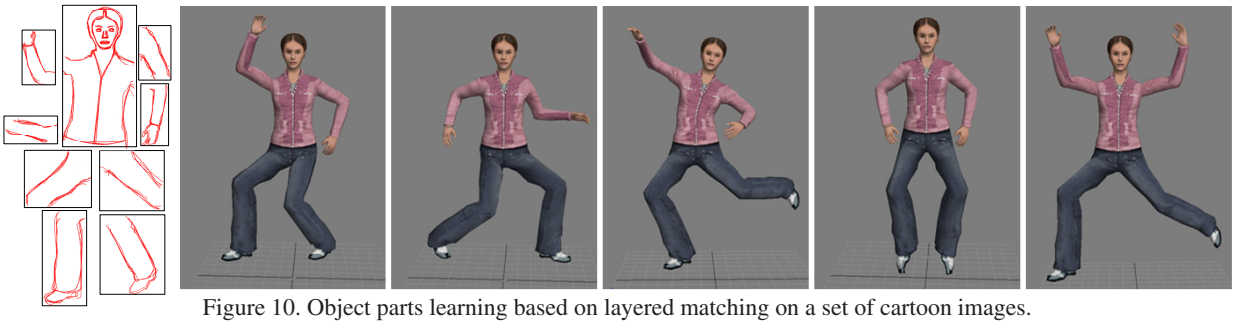

Figure 10. Object parts learning based on layered matching on a set of cartoon images.

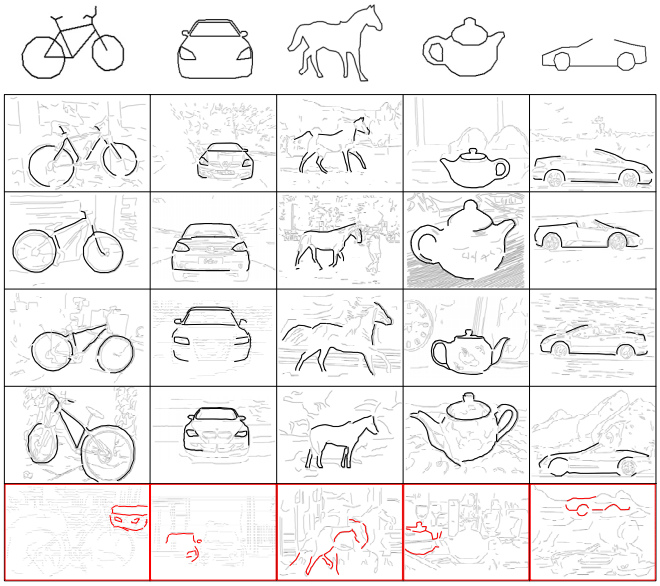

Figure 11. Object-of-interest detection and localization from real images. The top row shows templates of five objects. The remaining rows show some localization results by a two-layer graph matching method. The failure results are marked by red color in the bottom row.

and No. 2009AA01Z331) and National Natural Science Foundation of China (Grant No. 60672162 and No. 60776793).

References

[1] A. Barbu and S.C. Zhu, Generalizing Swendsen-Wang for Image Analysis, *Journal of Computational and Graphical Statistics*, 16(4): 877-900, 2007.

[2] S. Belongie, et al., Shape Matching and Object Recognition Using Shape Contexts. *TPAMI*, 24(4):509-522, 2002.

[3] A.C. Berg, et al., Shape Matching and Object Recognition using Low Distortion Correspondences, *CVPR*, 2005.

[4] G. Borgefors, Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm, *TPAMI*, 10(6): 849-865, 1988.

[5] H. Chui and A. Rangarajan, A new point matching algorithm for non-rigid registration, *Computer Vision and Image Understanding*, 89(2):114-141, 2003.

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

[7] P.F. Felzenszwalb, et al., Hierarchical Matching of Deformable Shapes. *CVPR*, 2007.

[8] K. Grauman and T. Darrell, Pyramid match kernels: Discriminative calssification with sets of image features, Technical Report MIT-CSAIL-TR-2006-020, MIT, 2006.

[9] C. E. Guo, S. C. Zhu, and Y. N. Wu, "Primal Sketch: Integrating Texture and Structure", *Computer Vision and Image Understanding*, 106(1): 5-19, 2007.

[10] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *IJCV*, 60(2): 91-110, 2004.

[11] L. Latecki, et al., Shape descriptors for non-rigid shapes with a single closed contour. *CVPR*, 2000.

[12] L. Lin, S.C. Zhu, and Y. Wang, Layered Graph Match with Graph Editing, *CVPR*, 2007.

[13] H. Ling and D. W. Jacobs, Using the inner-distance for classification of articulated shapes, *CVPR*, 2005.

[14] X. Liu, L. Lin, et al., Layered Shape Matching and Registration: Stochastic Sampling with Hierarchical Graph Representation, *ICPR*, 2008.

[15] N. Metropolis, et al., Equation of State Calculations by Fast Computing Machines. Journal of Chemical Physics, 21(6): 85-111, 1953.

[16] D. Nister, Preemptive RANSAC for live structure and motion estimation, *ICCV*, vol.1: 199-206, 2003.

[17] G.C. Sharp, et al., ICP Registration Using Invariant Features. *TPAMI*, 24(1):90-102, 2002.

[18] T. Sebastian, P. Klein, and B. Kimia, On aligning curves, *IEEE Trans. on PAMI*, 25(1):116C124, 2003.

[19] K. Siddiqi, et al., Shock Graphs and Shape Matching. *IJCV*, 35(1): 13-32, 1999.

[20] D.G. Sim, et al., Object Matching Algorithms Using Robust Hausdorff Distance Measures. *TIP*, 8(3): 425-429, 1999.

[21] M.F. Tappen, el tal, Learning Gaussian Conditional Random Fields for Low-Level Vision, *CVPR*, 2007.

[22] Z. Tu, et al., Shape Matching and Registration by Data-driven EM, *CVIU*, 103(3): 290-304, 2007.

[23] X. Yang, X. Bai, L. Latecki, and Z. Tu, Improving Shape Retrieval by Learning Graph Transduction, *ECCV*, 2008.

[24] B. Yao, et al., Introduction to A Large Scale General Purpose Groundtruth Dataset: Methodology, Annotation Tool, and Benchmarks. *EMMCVPR*, Springer LNCS, 2007.

[25] S.C. Zhu and A. Yuille, Forms: A Flexible Object Recognition and Modeling System, *IJCV*, 1996.