# Layered Graph Match with Graph Editing

Liang Lin[1,2], Song-Chun Zhu[2,3] and Yongtian Wang[1]

[1]School of Info. Sci. and Tech.
Beijing Institute of Technology, Beijing, China
{linliang,wyt}@bit.edu.cn

[2]Lotus Hill Research Institute
Ezhou, China

[3]Dept. of Statistic and Computer Science
University of California, Los Angeles
sczhu@stat.ucla.edu

## Abstract

*Many vision tasks are posed as either graph partitioning (coloring) or graph matching (correspondence) problems. The former include segmentation and grouping, and the latter include wide baseline stereo, large motion, object tracking and recognition. In this paper, we present an integrated solution for both graph matching and graph partition using an effective sampling algorithm in a Bayesian framework. Given two images for matching, we extract two graphs using a primal sketch algorithm [4]. The graph nodes are linelets and primitives (junctions). Both graphs are automatically partitioned into an unknown number of $K + 1$ layers of subgraphs so that $K$ pairs of subgraphs are matched and the remaining layer contains unmatched backgrounds. Each matched pair represent a "moving object" with a TPS (Thin-Plate-Spline) transform to account for its deformations and a set of graph operators to edit the pair of subgraphs to achieve perfect structural match. The matching energy between two subgraphs includes geometric deformations, appearance dissimilarities, and the cost of graph editing operators. We demonstrate its application on two tasks: (i) large motion with occlusion, and (ii) automatic detection and recognition of common objects in a pair of images.*

## 1. Introduction

With graph representations, many vision tasks are formulated as either a *graph partitioning (coloring)* problem [1, 23], such as segmentation, clustering, and grouping, or a *graph matching (correspondence)* problem, such as wide baseline stereo [8, 15, 6], large motion [14, 11], object tracking and recognition [19, 2, 25]. This paper is aimed at an integrated solution for simultaneous graph partition and matching. Fig. 1 shows an experimental result using our method. Given two input images, we compute two attributed graphs by the primal sketch method [4] which may
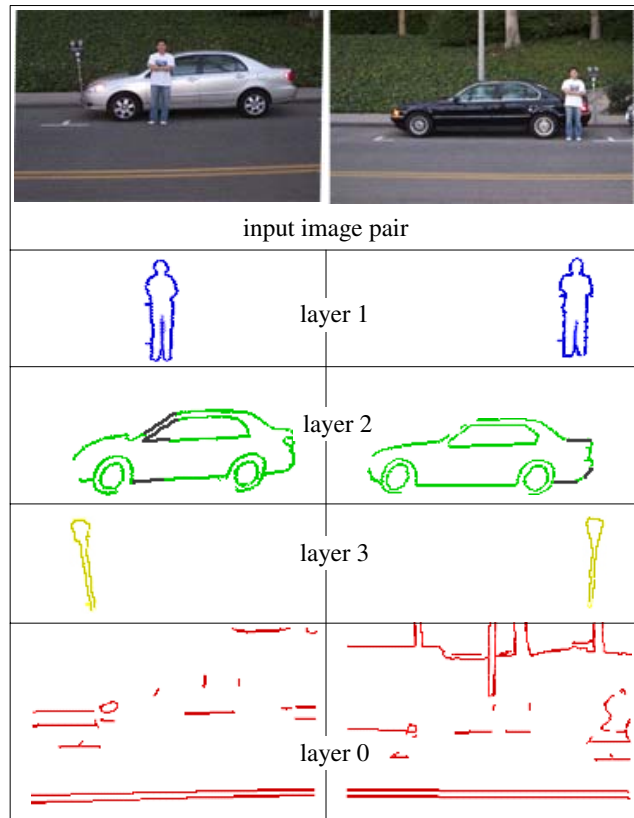


Figure 1. Simultaneous graph matching and partition. Two input images $\mathbf{I}, \mathbf{I}'$ (top) are automatically transformed into attribute graphs $G, G'$ by a primal sketch method[4]. Both $G$ and $G'$ are partitioned into 4 layers of subgraphs with layers 1-2-3 being the common objects matched between $\mathbf{I}, \mathbf{I}'$, and layer 0 the unmatched backgrounds. The dark line segments are edited portions.

be imperfect. Then both graphs are automatically partitioned (colored) into $K = 3$ layers of matched subgraphs for objects of similar structures (see person, cars, and meters) and a remaining layer for the unmatched backgrounds (layer 0).

There is a vast literature on graph matching algorithms

based on a pair of images. These can be divided into three categories.

*Category 1: single layer and point based.* These algorithms match local independent features without explicit graph structures, such as Harris corners, KLT features [10], scale invariant features [5, 13], local edge structure features [22] and geometric blur descriptors [19, 25]. These features are often robust against certain geometric distortions and illumination changes, but carry little information about larger object structures. When a match is performed between two sets of features, a single layer transformation model, e.g. affine or TPS (thin plate spline), is used to account for geometric distortions. Current state of the art algorithms include the iterative closest point (ICP) algorithm [7] and the soft-assignment algorithm [9].

*Category 2: single layer graph based.* These algorithms match explicit graphs with editing, such as, medial axis graphs [20] and shock graphs [12, 21]. They address structural or topological changes by introducing costs for graph operators. The latter edit the graph for perfect structural match (isomorphism). Most recent works include shape matching for object recognition [25] and shape context [19] for bottom-up initializations.

*Category 3: multi-layer point based.* These include layered motion by EM clustering [24, 18, 16] for small motion, and RANSAC based methods [17, 6] for larger motion. An example of the state of the art algorithm is the Wills and Belongie algorithm [11] for large motion segmentation using RANSAC iteratively, based on texture point features.

Our method belongs to a fourth category – multi-layer graph based matching with explicit graph editing. Generally speaking, point features are often good for texture rich areas (high entropy), while graph (lines, curves, axes, sketches) representations are good for textureless or cartoon areas (low entropy). As real world images contain both texture and cartoon areas, in this paper, we adopt a mixed representation which has both graphs and points. The latter are treated as isolated vertices. Fig 8 shows a comparison experiment for matching two images which have both texture and cartoon structures, in comparison to the results of RANSAC and SIFT based matching in Fig. 9.

A key contribution of this paper is a stochastic algorithm for simultaneous graph partitioning, matching, and editing, based on an attributed sketch graph representation. A number of graph operators are developed to edit the subgraphs to achieve perfect structural match. The algorithm samples effectively a huge joint state space, and computes a nearly optimal match and partition solution in a Bayesian framework. The matching energy between each pair of subgraphs includes geometric transformation, appearance dissimilarities, and the cost of operators for editing the graphs.

The joint state space for partitioning and matching is so large and complex that it cannot be searched exhaustively and demands effective sampling techniques. For example, consider a source graph of $N$ nodes and a target graph of $M$ nodes, for $K$-layer partitioning. The solution space is of order $O((KM)^N)$. In this paper, we adopt two types of Markov chain Monte Carlo (MCMC) dynamics to sample the joint space. One key observation in both dynamics is to make use of important proposals generated from bottom-up computation to drive the MCMC search.

**(I)** For graph partitioning, we adopt the SW-cut algorithm [1] for sampling the partition space. Unlike the single site Gibbs sampler, the SW-cut algorithm groups adjacent nodes into connected components and flips the color of a large subgraph in each step. The connected components are drawn through probabilistic sampling by turning on/off edges in the graph according to the local probabilities. These local probabilities define how likely two adjacent nodes in a graph belong to the same color, based on their similarity in appearance and goodness of alignment.

**(II)** For graph matching, each pair of subgraphs are matched by a TPS (thin-plate spline) transform and graph editing operators. This involves more parameters than the rigid affine transformation where traditional sampling algorithms, like RANSAC and MLESAC [17], are applied. To narrow the combinatorial number of possible matches, we start from some strong seed nodes (in source graph) that have fewer possible matches (in target graph), for example, a long curve or a cross is less ambiguous than a short line segment. The seed nodes are then merged hierarchically to form what we call "seed graphs". The nodes in a seed graph are "frozen" and must match. This is similar to the multi-level SW-cut method. A seed graph in source graph and its match in target graph propose a strong candidate for growth into a pair of matched objects.

We demonstrate this algorithm on vision tasks such as large motion with both opaque and transparent occlusions, object detection and matching with similar structure and different appearance, and wide baseline stereo.

The paper is organized in the following manner. Section 2 presents the Bayesian formulation. Section 3 presents the distance measures. Section 4 presents the algorithm. Section 5 shows two experiments: (i) object detection and matching, and (ii) large motion correspondence. Section 6 concludes the paper with a discussion.

## 2. Bayesian formulation

Given an image $\mathbf{I}$, we compute an attribute graph $G$ by the primal sketch algorithm [4] together with the SIFT point detection algorithm. $G$ is denoted by a 3-tuple $G = (V, E, A)$ with $V$ being a set of vertices (nodes) for the image primitives (disconnected feature points and blobs, edges and bars with intensity profiles, junctions), $E$ a set of edges for connectivity of the nodes (these represent adjacency relations and do not correspond to image edges or any image

element), and $A$ a set of attributes for position, orientation and intensity profile crossing the primitive.

In practice we compute the two graphs $G$ and $G'$ from $\mathbf{I}$ and $\mathbf{I}'$ respectively by a primal sketch approach.

$$
\begin{aligned}
G &= \arg\max_G p(\mathbf{I}|G)p(G) \\
G' &= \arg\max_{G'} p(\mathbf{I}'|G')p(G').
\end{aligned}
$$

$G$ and $G'$ are imperfect due to inference uncertainties and errors. Therefore we adopt graph operators to edit the graphs in the matching and partitioning process, and thus to correct some of the errors. The errors occur in both graphs, such as missed detection of a node or edge in both graphs $G$ and $G'$, will not be restored. For clarity, we will focus the discussion on inferring layered matching configuration by assuming that $G$ and $G'$ are given.

## 2.1. Graph partition and coloring

The goal of graph partitioning is to divide all nodes in $V$ into an unknown number of $K + 1$ subgraphs where vertices in each subgraph receive an unique color (label) $l(v) \in \{0, 1, ..., K\}$. We denote a graph partition of $G$ by

$$
\Pi_G = \{g_0, g_1, \cdots, g_k\}
$$

Each subgraph $g_i = (V_i, E_i, A_i)$ is a separate layer and is itself an attribute graph.

We assume $K$ should be small and each subgraph is preferably contiguous. Thus we introduce a Potts model as a prior for the partition $\Pi_G$ plus an exponential term for the number $K$.

$$
\begin{aligned}
p(K) &\propto \exp\{-\alpha K\} \\
p(\Pi_G|K, G) &\propto \exp\{-\beta \sum_{(s,t)\in E} \mathbf{1}(l(s) = l(t))\}.
\end{aligned}
$$

$\mathbf{1}(x) \in \{-1, +1\}$ is an indicator function for a Boolean variable $x$.

## 2.2. Integrating partition with matching

Given a source graph $G = (V, E, A)$ and a target graph $G' = (V', E', A')$, we can denote the graph matching function by

$$
\Psi : V \mapsto V' \cup \{\o\}
$$

For each node in $v \in V$, $\Psi(v) \in V'$ or $\Psi(v) = \o$ indicating no match. In later formulation, the matches on the vertices should not be independent but account for the graph connectivity.

To couple with the graph partition formulation $\Pi_G$, we break $\Psi$ into $K$ matching functions,

$$
\Psi_i : V_i \mapsto V_i' \cup \{\o\}, \quad i = 1, 2, ..., K.
$$

As the result of matching, the target graph $G'$ is also partitioned into $K + 1$ subgraphs $\{g_0', g_1', \cdots, g_K'\}$, where subgraph $g_0'$ has no correspondence in the source graph. The prior over $\Psi_i$ is defined so that two adjacent nodes in a subgraph $g_i$ should either be both matched or both unmatched to minimize the energy.

$$
p(\Psi_i|g_i) \propto \exp\{-\gamma \sum_{(s,t)\in E_i} \mathbf{1}(\Psi_i(s) = \o)\mathbf{1}(\Psi_i(t) = \o))\}
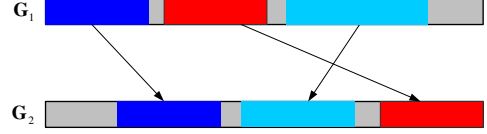$$



Figure 2. Simultaneous graph coloring (partition) and matching.

Now we have $K$-pairs of matched attribute graphs,

$$
(g_i, g_i'), i = 1, 2, ..., K.
$$

$g_i$ is transformed into $g_i'$ by a set of geometric transforms, photometric (appearance) transforms, and topological editing operators, denoted by

$$
\Phi_i = (\Phi_i^{\text{geo}}, \Phi_i^{\text{pho}}, \Phi_i^{\text{top}})
$$

Later, the matching energy terms or distances between $g_i$ and $g_i'$ are defined based on these three aspects. From $g_i, \Phi_i, \Psi_i$, one can predict its match $g_i'$ in the target graph by a prediction probability $p(g_i'|\Phi_i, \Psi_i, g_i)$. It will be defined as a product of the probabilities for the photometric, geometric, and topological transforms in the next section.

By combining graph partitioning and graph matching, our objective is to compute the following representation $W$ from two graphs $G, G'$,

$$
W = \left(K, \Pi_G, g_0, \{g_i = (V_i, E_i, A_i), \Phi_i, \Psi_i\}_{i=1}^K\right),
$$

## 3. Distance measures

As denoted earlier, $\Phi_i = (\Phi_i^{\text{geo}}, \Phi_i^{\text{pho}}, \Phi_i^{\text{top}})$ are the geometric, photometric and topological transforms, we have

$$
\begin{aligned}
&p(g_i'|g_i, \Psi_i, \Phi_i)p(\Phi_i|\Psi_i, g_i) \\
=\ &p(g_i'|\Phi_i^{\text{geo}}, \Psi_i, g_i)p(\Phi_i^{\text{geo}}|\Psi_i, g_i) \cdot \\
&p(g_i'|\Phi_i^{\text{pho}}, \Phi_i^{\text{geo}}, \Psi_i, g_i)p(\Phi_i^{\text{pho}}|\Psi_i, g_i) \cdot \\
&p(g_i'|\Phi_i^{\text{top}}, \Psi_i, g_i)p(\Phi_i^{\text{top}}|\Psi_i, g_i)
\end{aligned}
$$

We define these 3 types of distance in the following section.

### 3.1. Geometrical distance

The geometric transform $\Phi_i^{\text{geo}}$ from $g_i$ to $g_i'$ includes an global affine transformation $A_i$ and a TPS warping for deformation $F_i(x, y)$ in a 2D domain $\Lambda_i$ covered by $g_i$.

$$
\Phi_i^{\text{geo}} = (A_i, F_i)
$$

The affine transformation $A$ is a matrix

$$A = \begin{pmatrix} S_{ix} & 0 \\ 0 & S_{iy} \end{pmatrix} \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} 1 & \gamma_i \\ 0 & 1 \end{pmatrix} \quad (1)$$

where $\theta$ is the rotation angle, $S_x$ and $S_y$ denote scaling, and $k$ is shearing. The energy on $A$ is defined as

$$E_A^{\text{geo}}(A_i) = E_{\text{rot}}(\theta) + E_{\text{scl}}(S_x, S_y) + E_{\text{shear}}(\gamma_i).$$

The energy for the TPS deformation $f$ is

$$E_{\text{TPS}}^{\text{geo}}(F_i) = \lambda \int\int_{\Lambda_i} (F_{xx}^2 + 2F_{xy}^2 + F_{yy}^2)dxdy.$$

Another matching cost is the residue between the mapped position of each vertex $v \in V_i$ in $g_i$ and the position of $v' = \Psi_i(v)$ in $g_i'$. Each unmatched vertex $v \in U_i = \{v : \Phi_i(v) = \varnothing\}$ receives a constant penalty $\alpha$.

$$E_{\text{res}}^{\text{geo}}(V, V') = \sum_{\Psi_i(v) \neq \varnothing} \lambda^{\text{geo}}((x_v - x_{v'})^2 + (y_v - y_{v'})^2) + \alpha^{\text{geo}}|U_i|.$$

In summary, the probability for geometric transform is

$$p(g_i'|\Phi_i^{geo}, \Psi_i, g_i)p(\Phi_i^{geo}|\Psi_i, g_i) \propto e^{-E^{\text{geo}}(\Phi_i^{\text{geo}}, V, V')}$$

$$E^{\text{geo}}(\Phi_i^{\text{geo}}, V, V') = E_{\text{res}}^{\text{geo}}(V, V') + E_A^{\text{geo}}(A_i) + E_{\text{TPS}}^{\text{geo}}(F_i).$$

### 3.2. Photometric distance

Two matched nodes $v, v' = \Phi_i(v)$ in $g_i, g_i'$ correspond to two image primitives (edgelets and bars) in the source and target images $\mathbf{I}$ and $\mathbf{I}_i'$ respectively. After geometric alignment, the photometric distance measures the dissimilarity in two appearance vectors $\mu_v$ and $\mu_{v'}$. The latter are intensity profiles perpendicular to the edges or bars. Again the unmatched nodes receive constant penalty,

$$E^{\text{pho}}(\mu(g_i), \mu(g_i')) = \lambda^{\text{pho}}\|\mu_v - \mu_{v'}\|^2 + \alpha^{\text{pho}}|U_i|.$$

The probability for photometric variance is

$$p(g_i'|\Phi_i^{pho}, \Phi_i^{geo}, \Psi_i, g_i)p(\Phi_i^{pho}|\Psi_i, g_i) \propto e^{-E^{\text{pho}}(\mu(g_i), \mu(g_i'))}.$$

### 3.3. Topological distance

Preserving topological structures is an important issue in graph matching. This is particularly important for recovering from occlusions, and it distinguishes our approach from many previous works that match merely a set of points [24, 5, 10].

For a node $v \in V_i$ in $g_i = (V_i, E_i, A_i)$, we denote its neighbors by

$$\partial v = \{u : <v, u> \in E_i\}$$



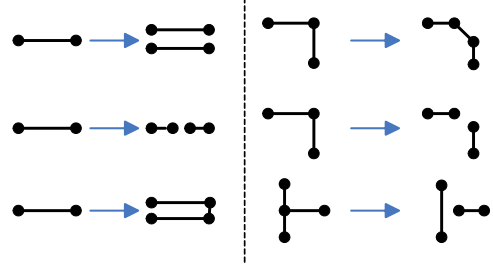Figure 3. Topological operators for editing graphs.

Suppose $v' = \Psi_i(v) \in V'$ is a matched node with neighborhood $\partial v'$. The match $\Psi_i$ is said to be isomorphic between $v$ and $v'$, if

$$u \in \partial v \Leftrightarrow u' \in \partial v', \quad u' = \Psi_i(u), u = \Psi^{-1}(u').$$

If the match is not isomorphic, a number of operators have to be applied, such as death, birth, split or merge of nodes or edges. Previous graph editing work involved the medial axis and shock graphs[12, 20]. Each operator $\delta$ is associated with certain cost $\text{cost}(\delta)$. We denote the whole set of operators between $g_i, g_i'$ to be

$$\text{op}_i(g_i, g_i') = \{\delta_{ij} : j = 1, 2, ..., m_i\}.$$

Thus the topological probability of match $g$ to $g'$ is:

$$p(g_i'|\Phi_i^{top}, \Psi_i, g_i)p(\Phi_i^{top}|\Psi_i, g_i) \propto e^{-\sum_{j=1}^{m_i} \text{cost}(\delta_{ij})}. \quad (2)$$

Fig. 3 shows some typical examples.

## 4. Computing algorithm

We compute $W$ in a Bayesian framework,

$$\begin{aligned} W^* &= \arg\max_W p(W|G, G') \\ &= \arg\max_W p(G'|G, W)p(W|G). \end{aligned}$$

As the $K+1$ subgraphs are matched independently, we have $p(G'|G, W)$ as a product of the prediction probabilities.

$$p(G'|G, W) = p(g_0')\prod_{i=1}^K p(g_i'|\Phi_i, \Psi_i, g_i)$$

Note that the prior on $W$ is given by

$$\begin{aligned} p(W|G) &= p(K, \Pi_G, \{\Psi_i\}, \{\Phi_i\}|G) \\ &= p(K)p(\Pi_G|K, G)p(g_0)\prod_{i=1}^K p(\Psi_i|g_i)p(\Phi_i|\Psi_i, g_i) \end{aligned}$$

The final Bayesian formulation is

$$\begin{aligned} W^* &= \arg\max_W [p(K)p(\Pi_G|K, G)p(g_0)p(g_0')] \cdot \\ &\quad \left[\prod_{i=1}^K p(g_i'|\Phi_i, \Psi_i, g_i)p(\Phi_i|\Psi_i, g_i)p(\Psi_i|g_i)\right] \end{aligned}$$
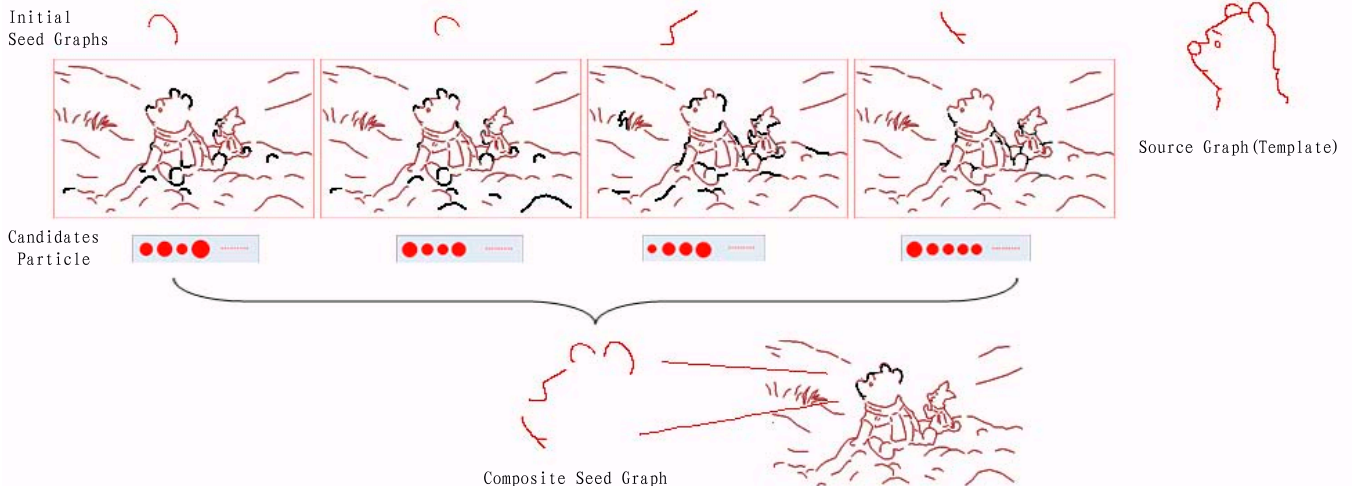
Figure 4. Computing seed subgraphs by hierarchical aggregation. Suppose we are searching for a bear head template in a cluttered cartoon image. The first row shows some initial seeds as candidates. As the number of seed grows, the number of candidates drops to one final candidate at the bottom.

$p(K)$,$p(\Pi_G|K,G)$ and $p(\Psi_i|g_i)$ have been defined previously. $p(g_0)$ and $p(g_0')$ are prior probabilities for the background subgraphs and modeled as

$$p(g_0) \propto \exp\{-\tau|V_0|\}$$
$$p(g_0') \propto \exp\{-\tau|V_0'|\}.$$

Intuitively, each unmatched node receives a penalty $\tau$.

The key variables in $W$ are the partition $\Pi_G$, the match $\Psi = \{\Psi_i\}$, and the transforms $\Phi = \{\Phi_i\}$. In practice, once the partition $\Pi_G$ and the correspondence $\{\Psi_i\}$ are fixed, computing $\{\Phi_i^*\}$ is relatively easy and can be done deterministically.

$$\Phi_i^* = \arg\max_{\Phi_i} p(g_i'|g_i, \Psi_i, \Phi_i)p(\Phi_i|\Psi_i, g_i).$$

Thus we are only concerned with the partition $\Pi_G$ and the matching $\Psi$. From a computational view, each node $v$ in $G$ has a set of label candidates denoted by

$$L_v = \{l_v\} \subset \{0, 1, \cdots, K\},$$

and a set of match candidates denoted by

$$M_v = \{m_v\} \subset V' \cup \{\emptyset\}.$$

Fig. 5 shows the coloring and matching problem. Each node $v$ has two vertical lines indicating the two sets of candidates. The red line is for match candidates $M_v$ and the blue line shows label candidates $L_v$.

The joint state solution space $\Omega_{\Pi,\Psi}$ is combinatorial. Consider a source graph of $N$ nodes and a target graph of $M$ nodes with $K$-coloring. The solution space would have $O((KM)^N)$ possible solutions. To sample this space effectively, we need to narrow the possible combinations using bottom-up information.
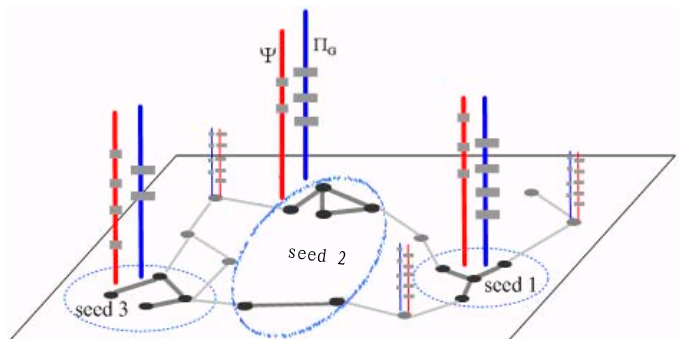


Figure 5. Sampling partition $\Pi_G$ and match $\Psi$. Each node $v$ and a seed graph (composite node) has two 2 sets of candidates: the dots on the red line are the match candidates $M_v$, and dots on the blue line are the label candidates $L_v$. A solution is an assignment of both the blue and red lines.

The general idea is similar in spirit to RANSAC and its variations[17]. We first draw some points and calculate the overall matches, then keep the strong set of points that have the best match. In our graph matching the distance measure includes TPS (thin-plate spline) transform and graph editing operators. This is much more complex than the low dimensional parameters spaces where RANSAC is often applied to. Instead of drawing points randomly, we start with some strong seed nodes and grow them into seed graphs. Each seed graph is then a composite candidate for matching. In the following we briefly discuss the seed graph.

### 4.1. Initializing the seed graphs

The seed graph computation is illustrated in Fig. 4. Suppose we are searching for a target graph $G'$, such as the bear head shown on the right, in a cluttered cartoon image transferred into a large source graph. We first identify a number of nodes as seeds in the source graph. These seeds should

have less ambiguity, such as long curves, junctions, as seen in the top row. In other words, the probability on the red line of a seed node has low entropy and thus is more informative than others. We then grow a seed node into a small seed graph. The number of candidates in $M_v$ drops as the seed graphs grow and merge. The last row in Fig. 4 shows the best candidate found for the bear head template. The merged seed growing and merging process greatly reduces the computation complexity. Since the nodes inside a seed graph are bonded and always receive the same color, they share two common vertical lines (red and blue) in Fig. 5 with fewer candidates. These seed graphs provide context and spatial constraints to matching other nearby nodes.

We adopt a branch-and-bound algorithm for growing and merging the seed graph. This algorithm was used in matching skeleton (medial axis) graphs for object recognition [20]. Suppose $\tilde{g} = (\tilde{V}, \tilde{E}, \tilde{A})$ is an initial seed graph (e.g. a vertex or a curve fragment) identified in $G$. We define a distance function here for $\tilde{g}$ and its corresponding subgraph $\tilde{g}'$ (a matching candidate) in target graph $G'$ by

$$D(\tilde{g}, \tilde{g}') = \min_{\tilde{\Phi}} -\log[p(\tilde{g}'|\tilde{g}, \tilde{\Phi})p(\tilde{\Phi}|\tilde{g})].$$

With this distance function, the branch-and-bound algorithm can quickly prune bad branches and find good candidates in the target graph $G'$.

Identifying good seed graphs in $G$ is a matter of deciding the visiting/search order. A seed graph should be maximally discriminative and robust. For example in textured parts the sketch structures are less reliable and some affine invariant key feature points will be helpful to narrow matching proposals in the bottom-up process. In textureless parts, some structural elements (curves) are more informative, for example the long curves in Fig. 7 and junctions in Fig. 8.

### 4.2. Simultaneously partitioning and matching

The seed graphs serve to narrow the search space in which our algorithm solves a double labeling problem as illustrated in Fig. 5. It simulates a Markov chain which visits a sequence of states in the joint space $\Omega_{\Pi,\psi}$ over time $t$,

$$(\Pi(t), \Psi(t)) \in \Omega_{\Pi,\Psi}, \; t = 1, 2, ....$$

The $\mathcal{MC} = < \nu, \mathcal{K}, p >$ consists of three elements. (1) An initial probability $\nu(\Pi, \Psi)$. (2) Two transition kernels $\mathcal{K}(\Pi, \Pi')$ and $\mathcal{K}(\Psi, \Psi')$ which are conditional probabilities for moving from state $\Pi(t) = \Pi$ to $\Pi(t + 1) = \Pi'$, and from state $\Psi(t) = \Psi$ to $\Psi(t + 1) = \Psi'$ given the other is fixed. (3) An invariance probability $p(\Pi, \Psi)$, which is the Bayesian posterior probability mentioned at the beginning of this section.

Our sampling process iterates between two types of Markov chain Monte Carlo dynamics, which are designed under the Metropolis-Hastings method.

**Dynamics I.** Fixing the match $\Psi$, we sample the partition $\Pi_G$ from a conditional probability $p(\Pi_G|\Psi)$ by the Swendsen-Wang cut algorithm[1].

The SW-cut proceeds in two steps: (i) generating a connected component (subgraph) through turning on/off the edges probabilistically; and (ii) flipping the color of the selected subgraph collectively with an acceptance probability calculated by the Swendson-Wang cut. This step includes splitting (adding a new color), merging (removing a color), and flipping (changing a color).

**Dynamics II.** With fixed $\Pi_G$, sample $\Psi$ from a conditional probability $p(\Psi|\Pi_G)$. For each node $v$ which could be either a single node or a seed graph, we have a set of candidate matches which have passed the bottom-up screening. We then adopt a Gibbs sampler to sample a match from the candidate set. To improve the efficiency, the algorithm visits stronger seed graphs and propagates the matches to nearby nodes.

## 5. Experiments

We present two types of experiments: (i) object detection and recognition, and (ii) large motion and object matching, on both artificial images and real testing images.
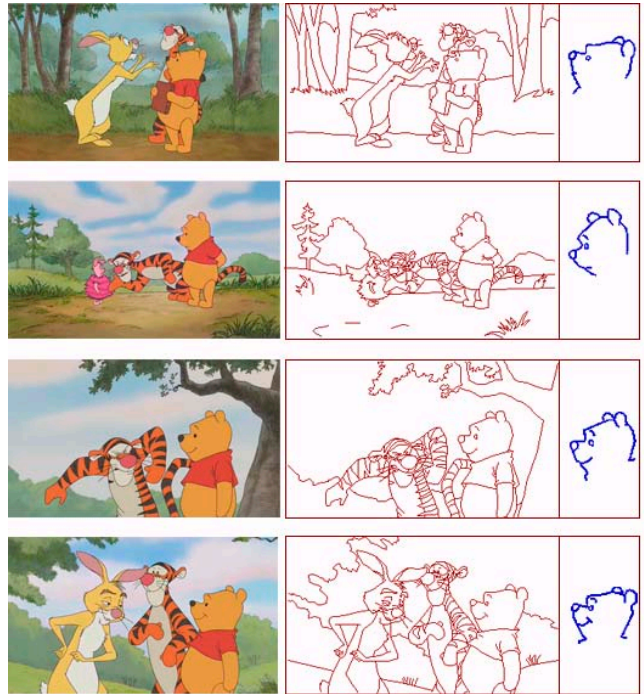


Figure 7. Examples of object detection: finding the bear head in cartoon images. Each row shows (left): input image, (middle): source graph $G$, and (right): the detected object (head at various pose) with editing.

**Example I: Object detection in cartoon images**. The object detection experiments are shown in Fig. 7 and Fig. 4. The task is to find a given template $G'$ – the head of a
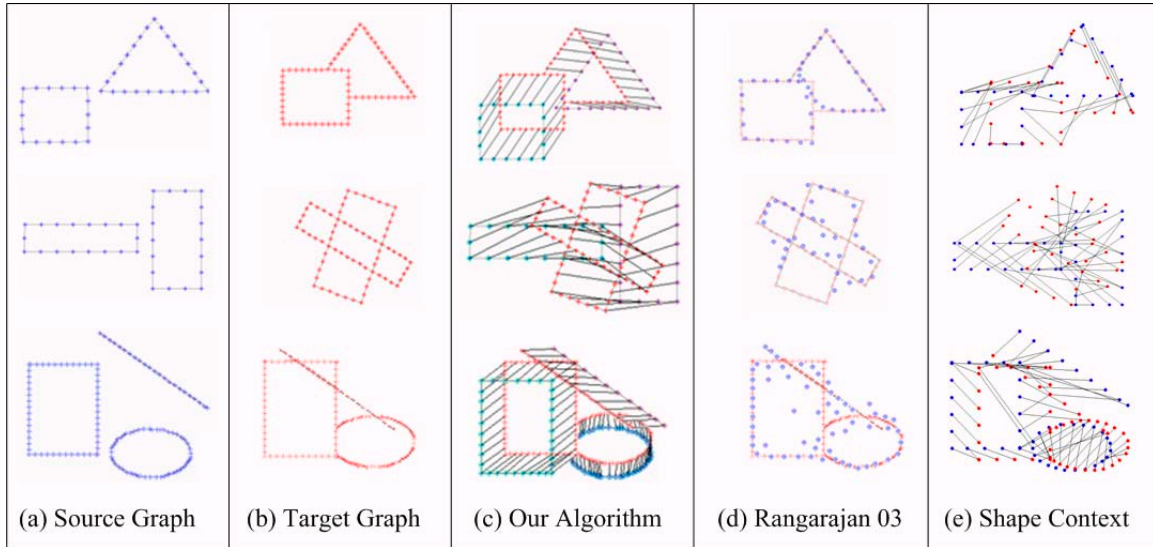
Figure 6. Three simple examples of matching a source graph (a) to a target graph (b). (c) are results of our proposed method which automatically decomposes the graph into multi-layers (subgraphs) and matches each pair with a thin plate spline model. The occluded parts are unmatched and thus edited. We compare with state-of-the-art methods in (d) results from (Rangarajan, 2003) and (e) results from shape context (Belongie 2002)

bear in 5 cartoon images. The images are transferred into a sketch graph $G$ with color and intensity information discarded. The matching is based on the graphs alone. The results show successful matches with the unmatched part being backgrounds. This method tolerates geometric deformations and some structural differences.

**Example II: Layered matching in artificial images.** Fig. 6 shows three pairs of artificial images for large motion (opaque and transparent) with occlusion. Note that no photometric information is used here. The results of our method are shown in the third column. For comparison, Fig. 6).(d) shows the single layer matching results produced by the state-of-the-art graph matching algorithm [9], and Fig. 6).(e) displays the shape context match [19] results.

**Example III: Layered matching in real images.** Fig. 1 and Fig. 8 show two experiment results for simultaneous matching and partitioning 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. In Fig.8, the people inside the picture switched positions with large camera motion. No camera information is given in the experiment. In the two examples, the car and the wall have little texture (low entropy), and thus they are not matched by the feature point based methods.

For comparison, we show the SIFT match results in Fig. 9 (a). The algorithm extracts many interest points on textured surface (picture frame in the background) and matches them quite well. However, on the non-texture surface, it gives very few matches. We show corner detection

with RANSAC match results in Fig. 9 (b). Most of the corners are matched correctly with motion estimation, but it failed in matching the cartoon areas (background and person).
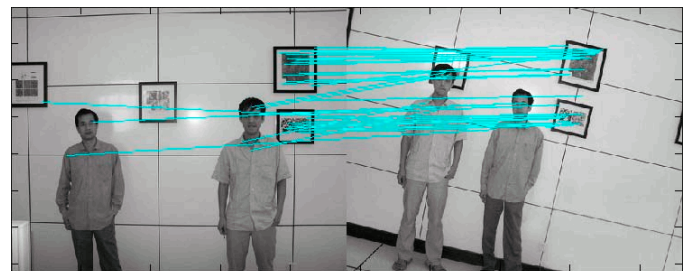


(a) Sift descripter matching
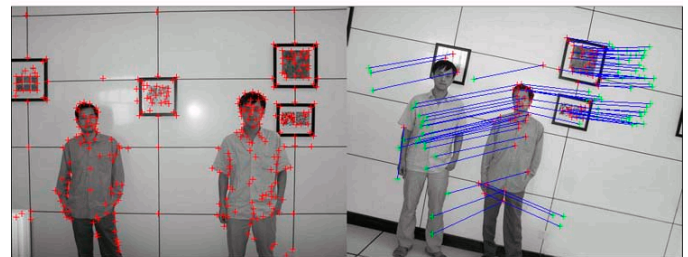


(b) Corner detector with RANSAC matching

Figure 9. For comparison. (a) Matching results by SIFT descriptor for comparison. Dense matches are found around texture regions, but very few matches can be found around texture-less areas. (b) Harris corner detection and robust estimation with RANSAC. Most of the corners are matched correctly with motion estimation, but curves and lines in the textureless areas are not matched well.
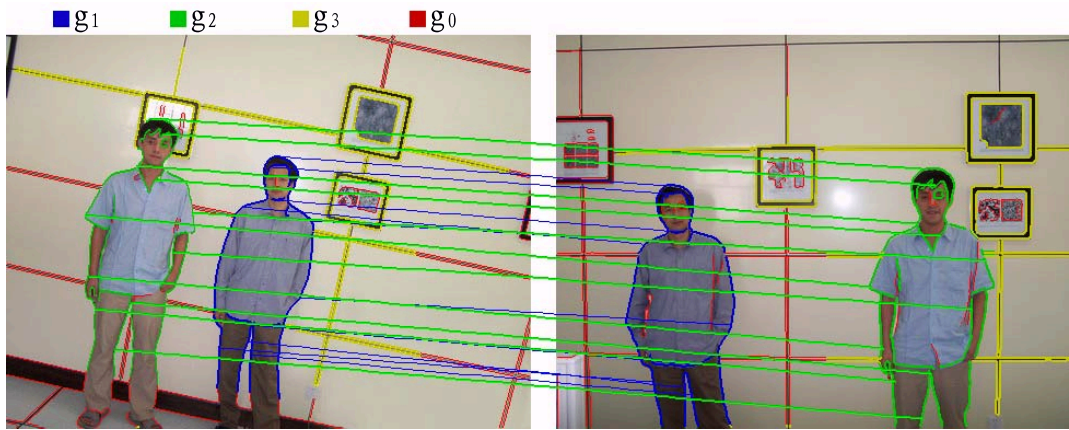
Figure 8. An example of layered graph matching. There are three matched layers ($g_1,g_2,g_3$) between the images, which are denoted by three colors(blue,green and yellow) and unmatched sketches ($g_0$) are highlighted in red. For clarity, only some of the matched contours between images were labeled with different colors to show layered match.

## 6. Summary

In this paper, we present an algorithm for layered graph matching. We compute graphs which include both textural (feature points) and textureless (cartoon) elements. A global probability model incorporates geometric transforms, photometric variations and topological editing costs. The inference process iterates two types of MCMC dynamics in the joint partition and match space. This paper presents a number of examples with large motion matching where occlusions are recovered via graph editing, as well as object detection and matching across two image frames. We are applying this algorithm to object recognition and automated learning object category from a set of images.

## Acknowledgement

## References

[1] A. Barbu and S. C. Zhu. Graph partition by swendsen-wang cuts. *ICCV*, 2003. 1, 2, 6

[2] B. C. Russell, A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. *CVPR*, 2006. 1

[3] B. Yao, X. Yang, and S.C. Zhu. An Integrated Image Annotation Tool and Large Scale Ground Truth Database. EMM-CVPR, 2007. 8

[4] C. E. Guo, S. C. Zhu, and Y. N. Wu. A mathematical theory of primal sketch and sketchability. *ICCV*, 2003. 1, 2

[5] D. G. Lowe. Object recognition from local scale-invariant features. *ICCV*, 1999. 2, 4

[6] D. Nister. Preemptive RANSAC for live structure and motion estimation *ICCV*, 2003. 1, 2

[7] G. C. Sharp, S. W. Lee, and D. K. Wehe. ICP Registration Using Invariant Features. *PAMI*, 24(1):90–102,2002. 2

[8] H. Bay, V. Ferrari, and L. V. Gool. Wide-Baseline Stereo Matching with Line Segments. *CVPR*, 2005. 1

[9] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *CVIU*, 89(2):114–141, 2003. 2, 7

[10] J. Shi and C. Tomasi. Good features to track. *CVPR*, 1994. 2, 4

[11] J. Wills, S. Agarwal, and S. Belongie. A feature-based approach for dense segmentation and estimation of large disparity motion. *IJCV*, 68(2):125–143, 2006. 1, 2

[12] K. Siddiqi, A. Shokoufandeh, et al. Shock graphs and shape matching. *ICCV*, 1998. 2, 4

[13] L. Lin, Y. Liu et al. Registration algorithm based on image matching for outdoor AR system with fixed viewing position. *IEE proc., Vis. image signal process*, 153(1):57–62, 2006. 2

[14] M. Ben-Ezra and S. K. Nayar. What Does Motion Reveal about Transparency? *ICCV*, 2003 1

[15] M. Goesele, S. M. Seitz and B. Curless. Multi-View Stereo Revisited. *CVPR*, 2006. 1

[16] M. J. Black and D. J. Fleet. Probabilistic detection and tracking of motion boundaries. *IJCV*, 38(3):231–245, 2000. 2

[17] P. H. S. Torr. Robust Computation and Parameterization of Multiple View Relation. *ICCV*, 1998. 2, 5

[18] P. Smith, T. Drummond, and R. Cipolla. Layered motion segmentation and depth ordering by tracking edges. *PAMI*, 26(4):479–494, 2004. 2

[19] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002. 1, 2, 7

[20] S. C. Zhu and A. L. Yuille. Forms: A flexible object recognition and modeling system. *IJCV*, 20(3):187–212, 1996. 2, 4, 6

[21] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing shock graphs. In *ICCV*, 2001. 2

[22] T. F. Cootes and C. J. Taylor. On representing edge structure for model matching. *CVPR*, 2001. 2

[23] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision *PAMI*, 26(9):1124–1137, 2004. 1

[24] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. *CVPR*, 1997. 2, 4

[25] Z. W. Tu and A. L. Yuille. Shape matching and recognition using generative models and informative features. *ECCV*, 2003. 1, 2