# An Integrated Framework For Image Segmentation and Perceptual Grouping

Zhuowen Tu
Integrated Data Systems Department
Siemens Corporate Research, Princeton, NJ 08540

## Abstract

*This paper presents an efficient algorithm for image segmentation and a framework for perceptual grouping. It makes an attempt to provide one way of combining bottom-up and top-down approaches. In image segmentation, it generalizes the Swendsen-Wang cut algorithm [1] (SWC) to make both 2-way and m-way cuts, and includes topology change processes (graph repartitioning and boundary diffusion). The method directly works at a low temperature without using annealing. We show that it is much faster than the DDMCMC approach [12] and more robust than the SWC method. The results are demonstrated on the Berkeley data set [7]. In perceptual grouping, it integrates discriminative model learning/computing, a belief propagation algorithm (BP) [15] , and SWC into a three-layer computing framework. These methods are realized as different levels of approximation to an "ideal" generative model. We demonstrate the algorithm on the problem of human body configuration.*

## 1. Introduction

Image segmentation and perceptual grouping are among the key problems in vision. Yet the results of existing methods are not matching human performance in terms of both speed and quality [7]. Discriminative and generative methods are two representative approaches in image segmentation. One promising direction is to integrate discriminative (bottom-up) and generative (top-down) models [16, 13, 8, 10, 9]. It is not clear, though, how to build a generic system to automatically understand natural scenes and segment/recognize a wide class of patterns/objects. Efforts leading to the birth of such a system may include extensive study of low level features/cues (edges, corners, interest points), affinity measures/discriminative models for perceptual grouping, generative models on appearance of complex objects and texture, shape models for a wide variety of objects, and fast algorithms to combine them in making efficient visual inference.

In this paper, we make an attempt to design an efficient algorithm for image segmentation and a general scheme for perceptual grouping. It can be divided into two parts. The first part is about an efficient image segmentation algorithm dealing with low-level patterns. The second part uses the results from the first part and provides a general framework for perceptual grouping. In the future, we will try to combine the two parts into a fully integrated system.

(I) The DDMCMC [12] method is a computational paradigm which combines top-down and bottom-up information in searching for the optimal solution. Compared to the traditional MCMC algorithms [3], it is more efficient due to its use of bottom-up processes in guiding the search. However, it is hard to make large moves since the topology of the regions is maintained on-line when making splitting/merging. The SWC algorithm [1], instead, works on atomic regions (pixel groups), and combines the split and merge dynamics into a single process. This largely improves the speed over the original DDMCMC algorithm. However, SWC only makes 2-way cuts and it works on a graph with fixed topology of atomic regions. Here, we generalize 2-way cuts in SWC to also make m-way cuts. This is particularly useful in perceptual grouping when we want to group multiple atomic regions/components into one part in one step. Also, we allow graph topology changes to include repartitioning and boundary diffusion of atomic regions. The new segmentation algorithm eliminates the need for using annealing and works directly at a fixed low temperature. Intuitively, most of the dynamics in the new algorithm can be understood as doing "region competition". Initially, a set of atomic regions are obtained from a bottom-up process (edge detection). With 2-way and m-way cuts, regions are *competing* for atomic regions. This quickly locates region boundaries according to those of atomic regions rather than moving region boundaries pixel by pixel as in the variational approach. With a variational approach [18], regions are *competing* for the boundary pixels. This corresponds to the boundary diffusion process to locally refine the segmentation.

(II) In the second part of this work, we design a three-layer computing framework for perceptual grouping. It is illustrated on the problem of identifying/configuring an articulated human body. One of the key concepts in SWC is that bottom-up cues/discriminative models are used to probabilistically group elements (atomic regions). Belief propagation algorithms have been shown to be effective methods

for approximating the marginal distributions. This allows us to integrate BP with the SWC. In the first layer, a discriminative method, probabilistic boosting-tree [14] is adopted to learn and compute a multi-class pairwise affinity map for atomic regions. BP is then used to pass the messages to better approximate the marginal distributions in the second layer. These marginals are used as proposals for verification by the high-level knowledge (generative models) in the third layer. We show that discriminative model learning and BP are essentially different levels of approximation to an "ideal" generative model. About 100 images containing some baseball players are manually annotated for learning the pairwise discriminative models of body parts. A PCA shape model is learned from the training set also. We report some results at the end of this paper.

# 2. Overall framework

For an input image, **I**, the task of image segmentation/scene understanding is to infer how many regions/objects there are and their locations. A scene interpretation (solution) $W$ can be denoted as

$$W = (k, \{(R_i, l_i, \theta_i), i = 1...k\}),$$

where $k$ is the number of regions, $R_i$ is the region domain (a set of pixels), $l_i$ is its type, and $\theta_i$ is the parameter for the appearance model of region $R_i$.

*An ideal model*

If we can obtain the model about $W$ and the generation process which generates **I**, then the optimal solution $W^*$ can be inferred from

$$p(W|\mathbf{I}) \propto p(\mathbf{I}|(k, \{(R_i, l_i, \theta_i), i = 1, ..., k\})) \cdot p(k, \{(R_i, l_i, \theta_i, i = 1, ..., k\})). \quad (1)$$

This requires the knowledge about the relations and configurations among all the regions/objects, appearance model for complex objects, and the knowledge about shapes. This "ideal" model is often out of reach, and for each specific problem, we usually seek a certain degree of approximations. [1]

*Independence assumption*

The first common approximation to the ideal model is the independence assumption,

$$p(W|\mathbf{I}) \propto p(k) \prod_i p(\mathbf{I}(R_i)|l_i, \theta_i) p(R_i|l_i) p(\theta_i|l_i) p(l_i) \quad (2)$$

where $p(\mathbf{I}(R_i)|l_i, \theta_i)$ is the appearance model about region/object $i$ and $p(R_i|l_i)$ defines its shape. It assumes independence of each region on appearance and shapes. This is similar to the one defined in [12]. However, this model is also hard to compute due to the complex distribution.

*Limiting the search space*

$R_i$ defines the domain of each region/object and $\bigcup_i R_i = \Lambda$ where $\Lambda$ is the lattice of the input image **I**. $\pi_k = \{R_i, i = 1..k\}$ defines all possible $k$ ways of partitioning $\Lambda$ whose space is enormously large. Drawing samples from this space is computationally prohibitive. Therefore, we try to make another approximation by defining an underlying graph, $G =< V, E >$, in which $V = \{v_1, ..., v_m\}$ includes all the atomic regions as basic elements. $E$ is a set of links $e =< v_i, v_j >$ which define the neighborhood relationship between the $v$s. Such an example can be seen in Fig. (1). The domain of each region $R_i = \{v_{i1}, v_{i2}, ...\}$ is now defined on these atomic regions. Sampling in the partition space of $\Lambda$ is then reduced to draw samples in the new space based on $G$. This largely reduces the search space and also facilitates the use of graph based methods.

# 3. Sampling W

Our goal is to make inference about $W^*$ that maximizes the posterior $p(W|\mathbf{I})$ in eqn. ( 2), which is highly complex. We use the Metropolis-Hasting [3] algorithm to perform sampling in the solution space which consists of the partition space and the parameter space. Suppose we are given a graph $G =< V, E >$ in which $V = \{v_1, ..., v_m\}$ includes all the atomic regions. Each $k$ partition of $\Lambda$ can be denoted as

$$\pi_k = \{R_1, R_2, ..., R_k\}, \ and \ \pi = \bigcup_k \pi_k,$$

where $\pi$ defines the entire the space for all possible partitions. The Swendsen-Wang cut algorithm provides a smart way of sampling in the partition space $\pi$. We briefly discuss the 2-way SWC below, and details can be found in [1].

## 3.1. 2-Way Swendsen-Wang Cut

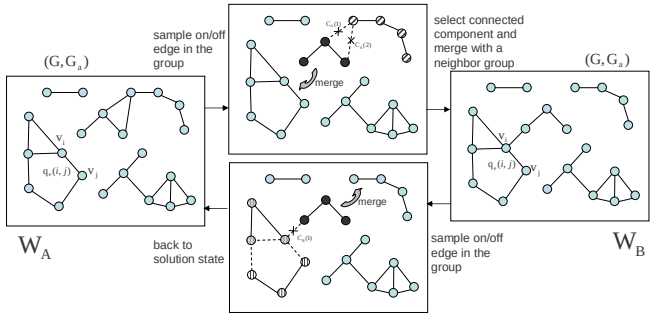



Figure 1: Illustration of a 2-way SWC. Each dot represents an atomic region/element. In each region/group, atomic regions are connected with edges indicating how strongly they are bonded.

[1] Notation $W$ has slightly different meanings in the segmentation part and perceptual grouping part.

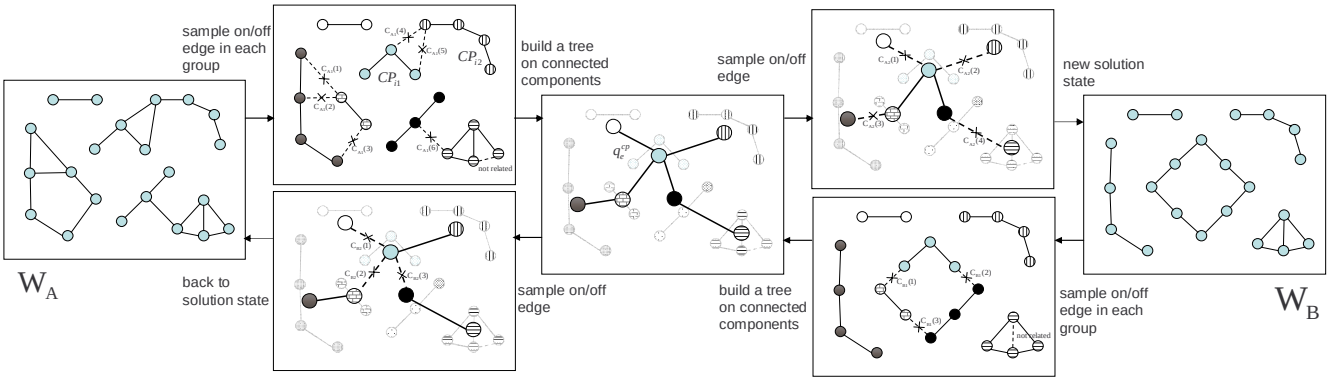



Figure 2: Illustration of the m-way SWC algorithm.

Given a graph $G =< V, E >$, build $Q_e$. For each $e_{ij} \in E$, $q_e(i,j) \in Q_e$ denotes the affinity (bond) between atomic regions $v_i$ and $v_j$. $q_e(i,j) = \hat{p}(e_{i,j} = on|v_i, v_j, \mathbf{I})$.

- Randomly pick a region $R_i$ and sample edges in $R_i$ to be $on$ with probability $q_e$. This gives rise to a set of connected components in $R_i$. See Fig. (1).
- Randomly pick a connected component, and either make it as a new region or merge it with one of its neighboring regions.
- Propose for the region type $l$s and parameters, and compute the $p(W_B|\mathbf{I})$.
- Compute the proposal probabilities from $W_A$ to $W_B$, and from $W_B$ to $W_A$.
- Compute the acceptance rate by the Metropolis-Hasting algorithm to decide whether or not to accept this move.

Figure 3: 2-Way SW-cut algorithm.

Fig.(1) illustrates an example of a 2-way cut. The acceptance rate is computed as follows:

$$\alpha(W_A \to W_B) = min(1, \frac{Q(\pi(W_B) \to \pi(W_A))q_{l,\theta}(W_B)p(W_B|I)}{Q(\pi(W_A) \to \pi(W_B))q_{l,\theta}(W_A)p(W_A|I)}),$$

where $q_{l,\theta}(W_B)$ and $q_{l,\theta}(W_A)$ are proposal probabilities for proposing model type and parameters for the regions changed. The magic of the SWC lies in the fact that when computing the ratio,

$$\frac{Q(\pi(W_B) \to \pi(W_A))}{Q(\pi(W_A) \to \pi(W_B))} = \frac{\prod_{e \in C_B}(1-q_e)}{\prod_{e \in C_A}(1-q_e)},$$

the burden of enumerating all the possibilities leading to the same connected component is removed because they are canceled from the both sides. We only need to compute the cuts. Fig. (1) gives such an example.

## 3.2. Generalizing SW-cut

We generalize the current SW-cut in two ways: (I) to design m-way cut, and (II) to make topology changes to the graph.

**M-way SW-cut**

The 2-way SW-cut algorithm splits a region into two or merges two regions into one at each step. To have a more efficient algorithm, we want to merge multiple regions into one group or split one region into several regions at once. Fig. (2) illustrates a new m-way SWC algorithm. Again, one critical issue is to compute the proposal probability ratio. We obtain:

Given a graph $G =< V, E >$, we construct $Q_e$. This is the same as that in 2-way SWC.

- Randomly pick a set of regions $S = \{R_i, ...\}$ with probability $q(S|W_A)$. This usually can be done by choosing regions that are in a selected focus area.
- Sample only edges in each region $R_i \in S$ with its corresponding edge probability. This gives rise to a set of connected components for each $R_i \in S$, $\{CP_{i1}, , ..., \}$. See Fig. (2).
- Collect all the connected components and deterministically build a tree. Compute the probability $q_e^{cp}$ for each edge in the tree. This can be done efficiently by integrating the edge probability on their atomic regions. See the center figure in Fig. (2).
- Randomly sample all the edges of the tree to be on or off.
- For each set of connected $CP$s, create a region.
- Propose for the region type $l$s and parameter $\alpha$s for each region.
- Compute the acceptance rate by the Metropolis-Hasting algorithm. If the move is rejected, then go back to $W_A$.

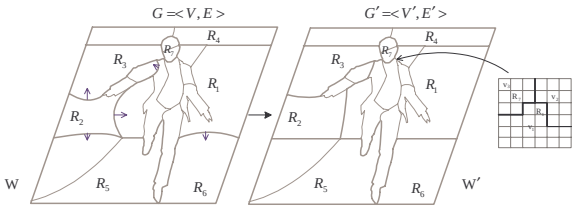
Figure 4: M-way SW-cut algorithm.

$$\begin{aligned}
&\frac{Q(\pi(W_B) \to \pi(W_A))}{Q(\pi(W_A) \to \pi(W_B))} = \\
&= \frac{[\prod_{e \in C_{B1}}(1-q_e)][\prod_i (N(CP_i))][\prod_{e \in C_{B2}}(1-q_e^{cp})]}{[\prod_{e \in C_{A1}}(1-q_e)][\prod_i (N(CP_i))][\prod_{e \in C_{A2}}(1-q_e^{cp})]} \\
&= \frac{[\prod_{e \in C_{B1}}(1-q_e)][\prod_{e \in C_{B2}}(1-q_e^{cp})]}{[\prod_{e \in C_{A1}}(1-q_e)][\prod_{e \in C_{A2}}(1-q_e^{cp})]}.
\end{aligned} \quad (3)$$

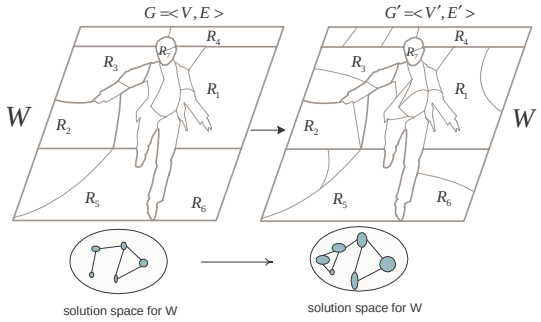
Here $C_{A1}$ are all the edges in cut $A1$, $q_e$ is the bond for the edge connecting atomic regions, $q_e^{cp}$ is the bond for the edge connecting connected components. $N(CP_i)$ sums all the possible probabilities leading to a connected $CP_i$. We only need to compute the cuts as shown in eqn. (3). Fig. (4) illustrates the procedure. Intuitively, we can see that different choices of the edges leading to the same connected components are canceled from both sides. The algorithm first randomly selects a set of regions of interest. Then the edge bonds connecting all the atomic regions are sampled

to be either on or off. This gives rise to a set of connected components. In 2-way cut, only one connected component is picked to either be a stand alone region or merge with a neighboring region. Here, a tree is deterministically built as another layer of edges whose bonds indicate how likely that two connected components $CP_i$ and $CP_j$ should be grouped. The tree is used to make sure that each set of cuts define a unique set of connected $CP$. This is a limitation since the connected components are not fully connected as in the atomic regions case. Computing the proposal ration can be reduced to compute cuts only as in eqn (3). We can see that m-way SWC enlarges the scope of the sampling algorithm. To go from $W_A$ to $W_B$ as shown in Fig. (2), 2-way cut needs at least three steps, among which there may be local minimums, whereas for m-way cut there is only one step. But we need pay a bit more computational price in constructing the tree and building and sampling bonds for connected components.

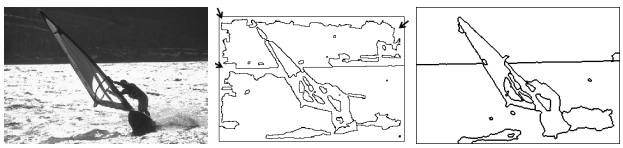
**Changing the topology of the graph**




(a) Boundary diffusion on atomic regions




(b) Coarse-to-fine repartitioning of the graph



(c) Input image (d) By SWC (e)

Figure 5: Two ways of changing the underlying graph topology. In (a) boundaries of atomic regions are diffused resulting in the change of both $G$ and $W$. The topology of atomic regions is maintained and moved explicitly on the image grid. In (b), the graph is repartitioned with a coarse-to-fine strategy. This changes the $G$, but $W$ remains the same. (c) shows an input image. (d) is the segmentation result by SWC [1], (e) shows the result by the algorithm proposed.

In section 2 we show that the introduction of atomic regions is essentially one way of making an approximation. It facilitates the sampling process. But on the other hand, it also limits the scope of the search space. For example, one can not put the region boundary on the place where there are no atomic regions. However, the global optimal is defined on the original lattice, $\Lambda$. Fig. (5d) shows a segmentation result by the SWC algorithm [1]. First, the boundaries are jagged. Second, there are several places where regions have leaks. This causes some big regions to not be proply merged. We introduce two ways of fixing the problem. First, we add the diffusion process on the atomic regions. The purpose is to sample $W$ subject to $p(W|\mathbf{I})$. Region competition [18] is a variational approach which diffuses the neighboring region boundaries in minimizing the energy. We do the same thing here but on the boundaries of regions. Fig. (5a) shows an example. The boundaries of atomic regions which are not shared by the current regions remain unaltered. Second, we adopt a coarse-to-fine segmentation strategy. The sampling process starts from an atomic region map at a coarse scale. After certain steps (2000), we repartition the current graph by a partition map at a fine scale. In theory, this does not change the state of the current $W$. But it enlarges the space that the sampling process can possibly visit. Fig. (5b) shows such an example. Fig. (5e) shows the segmentation by the proposed algorithm.

## 3.3. Summary of segmentation by Generalized SWC

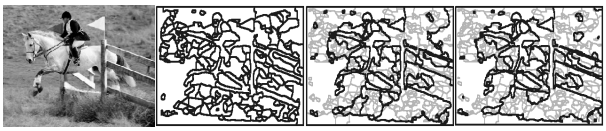
For image segmentation, the prior and likelihood models are the same to those used in [12].

- For an input image, it uses Canny edge detector to obtain a partition map at a coarse scale. This gives us the atomic regions.
- Randomly use 2-way or m-way SWC to group/ungroup atomic regions or diffuse the boundaries.
- After 2000 steps, repartition the atomic regions by a partition map at a fine scale.
- Run the dynamics again.
- Stop the algorithm according to a certain criterion.

Figure 6: Image segmentation by the proposed algorithm.

In Fig. (6), we give a summary of the proposed image segmentation algorithm. The algorithm starts from a segmentation with atomic regions at a coarse level. Due to the use of m-way cuts, these atomic regions are quickly grouped together, whereas it may take many steps for 2-way cuts to merge them one by one. But 2-way cuts are still very useful to split/merge specific atomic regions in the group. A random diffusion process is proposed to move the boundaries of atomic regions for local refinement. With a variational approach [18], regions are *competing* for the boundary pixels. With 2-way and m-way cuts, regions are *competing* for atomic regions. This quickly locates region boundaries according to those of atomic regions rather than moving region boundaries pixel by pixel as in the PDE approach. After a certain stage (2000 steps), a segmentation map from a bottom-up process (Canny edge detector) at a

fine level is used to repartition the current atomic regions. This enlarges the solution space from which the algorithm can draw samples (See Fig. (5)b). The combination of these approaches gives rise to an efficient algorithm which segments an image of $300 \times 300$ in about $2 \sim 3$ minutes. It is much faster than the DDMCMC algorithm and more robust than the original SWC algorithm. Also, it provides a general framework to further perform perceptual grouping. Fig. (7) shows some steps of the algorithm. They are tested on the Berkeley dataset with the outputs being the overlay of segmentations at three scales shown in Fig. (14).



(a) (b) (c) (d)

Figure 7: Illustration of the image segmentation algorithm. (a) is an input image. (b) shows the atomic region map at the beginning. (c) illustrates the status after the repartitioning. Boundaries with light lines are those of atomic regions, and dark ones show the segmented regions. (d) shows the final result.

# 4. Configuration of the human body

The task of segmenting and configuring the human body is very difficult. Human bodies are highly articulated and the appearance of body parts has large variations. Some recent work in this domain includes [6, 17, 5]. Graphical model algorithms are attractive in performing perceptual grouping task. In this section, we show how it is addressed by the proposed framework.

## 4.1. Problem definition

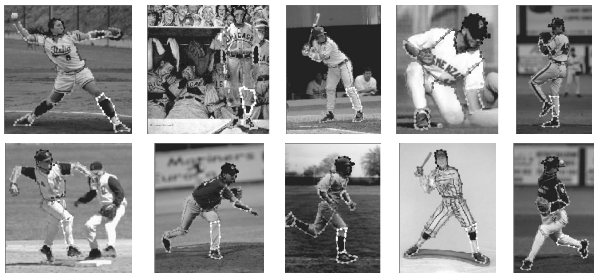


Figure 8: Some of the 100 training images collected from the internet using Google. Each body part is manually annotated.

We model each human body using 14 parts labeled from 0 to 13 and assume there is only one human in each image. The top left figure in Fig. (9) gives such a template. For a given graph $G =< V, E >$, where $V = \{v_1, ..., v_m\}$ includes all the atomic regions, the solution $W$ is defined as

$$W = \{(R_0 = \{v_{(0,1)}, ..\}, R_1 = \{v_{(1,1)}, ..\}, ..., R_{14} = \{v_{(14,1)}, ...\}\},$$

where $R_0$ to $R_{13}$ are the body parts from head to feet. $R_{14}$ includes all the background regions. For each $W$, we denote the part id of each atomic region by $\ell(v_i)$.

The posterior distribution is defined as

$$\begin{aligned} p(W|\mathbf{I}) \quad \propto \quad & p(\mathbf{I}(R_{14})|R_{14})p(R_{14}) \cdot \\ & p(\mathbf{I}(\Lambda/R_{14})|R_0, ..., R_{13})p(R_0, ..., R_{13}) \end{aligned} \tag{4}$$

which says that the background region is different with body part regions, and all the body parts define a joint appearance for a human. Also, there is a joint probability on the shape of each part. Apparently, these parts are highly correlated. We assume the independence of the likelihood model for each body part and apply a simple texture model. Learning the shape model of an articulated object is still an ongoing research topic. Instead of specifically defining the articulation of each part, we adopt a simple PCA model. For each part $R_i$, we fit an ellipse and obtain a vector $(a_i, b_i, x_i, y_i, \theta_i)$, where $a_i$, and $b_i$ are respectively the long and short axis of the ellipse, $x_i$ and $y_i$ are coordinates of the center of the ellipse, and $\theta_i$ is the orientation. These values are normalized w.r.t. the size and center of the head. We gather all the values of each part and align them into a vector

$$A = (a_1, b_1, x_1, y_1, \theta_1, ..., a_{13}, b_{13}, x_{13}, y_{13}, \theta_{13}).$$

Therefore the shape prior is defined by

$$p(R_0, ..., R_{13}) \propto exp\{-\frac{1}{2}(A-\mu)^T \Sigma^{-1}(A-\mu)\}$$

We collect 100 images containing baseball players and we annotate the body parts. A PCA model with 25 components is then learned. Fig. 9 shows some of the samples. This PCA model is adopted to show the need of using top-down information. More work needs to be done to better capture the overall articulation of the parts.

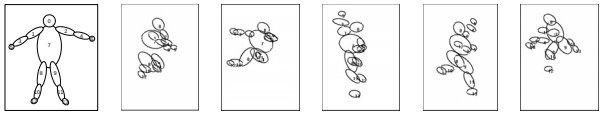


Figure 9: The first image is the template. Others are the samples drawn from the PCA model for the shape of body parts.

## 4.2. Three-layer computing framework

In the first layer, probabilistic boosting-tree [14] (PBT) is adopted to learn and compute pairwise multi-class affinity map for neighboring atomic regions. BP is then used to pass the messages to better approximate the marginal distributions in the second layer. These marginals are used as proposals for SWC to probabilistically group the atomic regions according to the high-level knowledge in the third layer. One of the ideas in SWC is that affinities (discriminative models) between atomic regions/basic elements are

used to guide the grouping process. This largely facilitates the computing process. However, it is hard to obtain the marginals directly form eqn. (4). Instead, we approximate it by

$$\hat{p}(W|\mathbf{I}) = \frac{1}{Z}\prod_{(i,j)} q(\ell(v_i),\ell(v_j)|\mathbf{I}(v_i),v_i,\mathbf{I}(v_j),v_j) \quad (5)$$

where $Z$ is the partition function (normalization), and $v_i$ and $v_j$ are neighboring atomic regions. We learn these discriminative models from examples. The discriminative model $q(\ell(v_i),\ell(v_j)|\mathbf{I}(v_i),v_i,\mathbf{I}(v_j),v_j)$ models the pairwise affinity between $v_i$ and $v_j$. It suggests what are the possible parts for two neighboring atomic regions given their shapes and appearance. This model can be learned based on a large set of annotated images. Once we have the $q(\ell(v_i),\ell(v_j)|\mathbf{I}(v_i),v_i,\mathbf{I}(v_j),v_j)$, what we are really interested is the marginal distribution $b(\ell(v_i),\ell(v_j)|\mathbf{I}(v_i),v_i,\mathbf{I}(v_j),v_j)$ based on $\hat{p}(W|\mathbf{I})$. This fits very well into the BP framework which approximates the marginal distribution in loopy graphs. Since these marginals are all approximations, in the end, we still want to use the high-level knowledge about the shape and overall appearance as verification. The integration of these procedures gives rise to a three-layer computational framework.

*I: Learning the multi-class discriminative model*

Local image patches show various levels of information in identifying a specific object. Fig. (11) illustrates an example in which several zoomed-in parts are displayed. We are able to identify the arm and the head even from local patches. This gives us a hint to use local discriminative model to make inference about object parts. Kumar and Herbert [4] have developed a discriminative Markov random fields model on rectangular image patches to perform scene interpretation. Here, the goal is to learn the discriminative model $q(\ell(v_i),\ell(v_j)|\mathbf{I}(v_i),v_i,\mathbf{I}(v_j),v_j)$ with $\ell(v_i),\ell(v_j) \in \{0,1,..,14\}$. It is a multi-class discriminative learning problem. We use the image segmentation algorithm discussed before to obtain a set of atomic regions on the 100 training examples and collect the pairwise atomic regions for the bodies and background regions. The total number of types of the pairs is 75 since not every two parts are adjacent to each other.

AdbBoost has become a powerful classification algorithm which combines a set of weak classifiers into a strong classifier. In vision, the focus is mostly on its application for detection. We use a new discriminative model learning framework, probabilistic boosting-tree (PBT) [14] , to learn pairwise relationships between neighboring atomic regions. Both appearance and shape contribute to this discriminative model. Intuitively, if two atomic regions are both from left thigh, then they should have similar intensity patterns, and they are most likely located at the bottom-half of the image. These are some weak cues/knowledge. It is up to the learning algorithm to decide what are the important cues and how to combine them. PBT combines these cues into a strong decision maker, and outputs a posterior probability. It learns a unified multi-class discriminative model hierarchically by a divide-and-conquer approach. The top node of the tree outputs the overall posterior probability by integrating probabilities gathered from its sub-trees. The details of how to learn and compute the model by PBT are discussed in [14]. The features used in selection for the learner are edges at various scales, edge orientations, absolute and relative positions, Hu moments for the shape of $v_i$, $v_j$, and $v_i \bigcup v_j$, and mean and variance on filter responses of various Gabor filters. The idea is to make use of both the shape and appearance information about $v_i, v_j, \mathbf{I}(v_i), \mathbf{I}(v_j)$ to tell how the two regions should be probabilistically labeled.

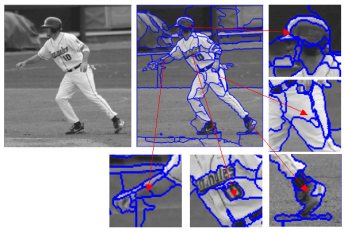


Figure 11: A training image and its zoomed-in versions of various parts. We can see that local image patches still give rich information about what they are.

*II: Approximating the marginals by BP algorithm*

The information computed by the discriminative model is purely local. The first row in Fig. (12) shows some of the saliency maps computed. We marginalize the pairwise probabilities into unary probabilities for better visualization. Our goal is to obtain the marginal distribution based on all the current atomic regions and use them as affinity map to guide the search verified using the high level knowledge. This can be done by message update as shown in [15, 2, 11] The messages are computed by

$$m_{ij}(\ell_j) \leftarrow \sum_{\ell_i} q_{ij}(\ell_i,\ell_j) \prod_{k\in N(i)\setminus j} m_{ki}(\ell_i),$$

and the beliefs are then

$$b_{ij}(\ell_i,\ell_j) = \frac{1}{Z} q_{ij}(\ell_i,\ell_j) \prod_{k\in N(i)\setminus j} m_{ki}(\ell_i) \prod_{l\in N(j)\setminus i} m_{lj}(\ell_j).$$

The second row in Fig. (12) shows some saliency maps for the one-node beliefs obtained. We can see that they eliminates some uncertainty by confirming with each other.

*III: Verification using high-level knowledge*

The local beliefs are used as proposals for the SWC algorithm to make inference based on the generative model's (high-level) knowledge. Therefore, the algorithms in this three-layer framework correspond to low-level, mid-level,

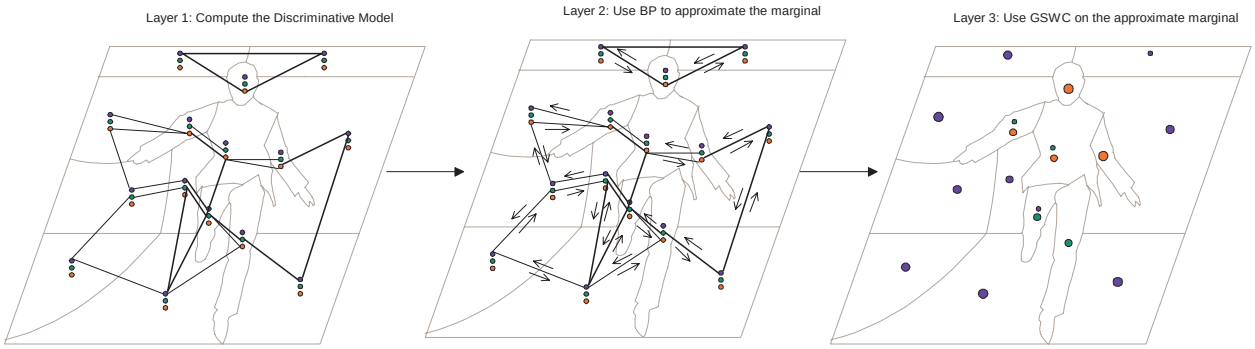



Figure 10: Three-layer computing framework. The first layer computes the affinities of atomic regions based on a multi-class discriminative model learned from an annotated The second layer uses BP algorithm to approximate the marginal distributions. In the third layer, high-level information is used to verification.

and high-level understanding of the scene. The discriminative models give affinities between two atomic regions based on their shape and appearance. This is computed locally. The BP algorithm then passes information around to make confirmation by propagating the messages. This corresponds to the mid-level inference. As we show, both the methods are approximations to the underlying posterior distribution governed. In the third layer, the high-level knowledge defined by generative model is used as verification based on the beliefs gathered from the mid-level.

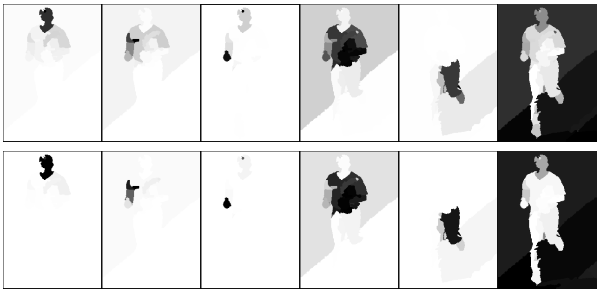


Figure 12: The first row shows some of the saliency maps by the discriminative models for some parts. Dark intensity means the high probability. The second row shows the one-node belief computed by the BP algorithm based on the computed discriminative models. It helps to resolve some local ambiguities and confirms some parts.

### 4.3. Experiments

We test the proposed algorithm on in images in [6]. We use grey scale images instead of color images. Some results are shown in Fig. (13). We show the original images, the segmented parts with each region labeled with the part id, and the corresponding estimation of the ellipse for each part. The results are promising, but they can be improved by learning a better shape model rather than a simple PCA. Also, a joint appearance model may better approximate the ideal generative model than independence assumption.

## 5. Discussions and conclusions

In this paper, we have introduced an integrated framework for image segmentation and perceptual grouping. It generalizes the SW-cut to include 2-way cut, m-way cut, and topology changes. This gives rise to a system which is much faster than the DDMCMC algorithm, more robust than the SW-cut, and makes moves of large scope when searching for the optimal solution. It further generalizes the SW-cut into a three-layer computational framework for perceptual grouping. The scheme integrates discriminative model learning, BP, and SW-cut together as different levels of approximation to an "ideal" model. This computing framework is general and it provides one way of combining top-down and bottom-up information. However, we still don't know the optimal strategy to design general system for making visual inference and we need to investigate more the roles of top-down and bottom-up learning/computing in a general visual inference system.

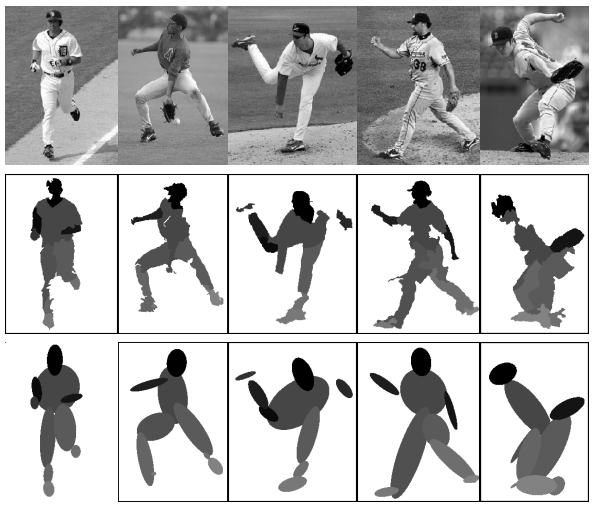


Figure 13: Results by the proposed algorithm on some gray scale images. The first row shows the input image. The second row shows the segmented body with the region labeled with identified part id. The third row are the corresponding estimated ellipses for body configuration.

**Acknowledgment**

Part of the project was done when the author was a post-doc at UCLA supported by NIH (NEI) grant RO1-EY012691-04. I thank Adrian Barbu, Daniel Creamers,

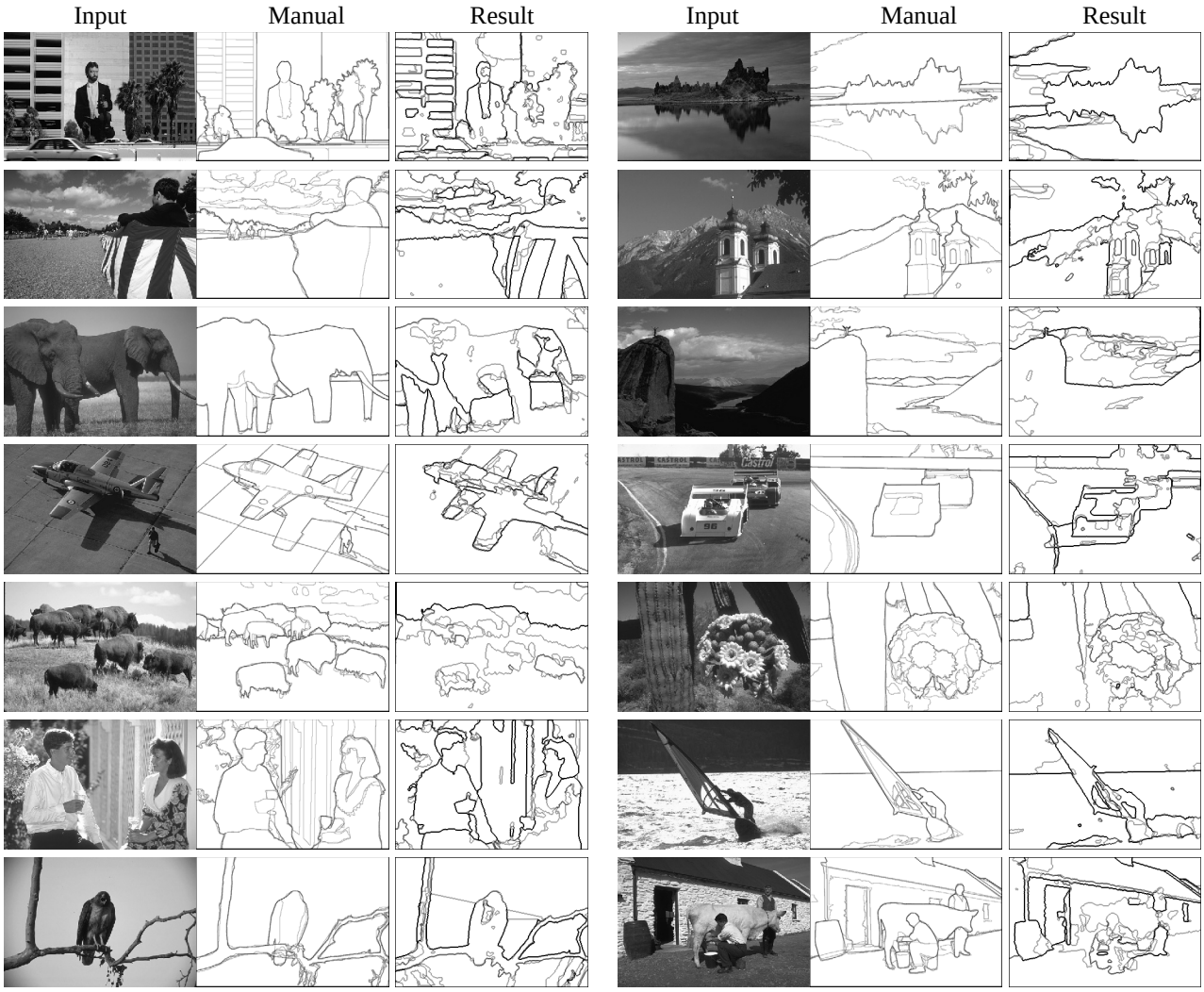



Figure 14: The first column are images from the Berkeley dataset ( http://www.cs.berkeley.edu/projects/vision/grouping/segbench/). The second are the overlayed manual segmentations. Human subjects intend to use high-level knowledge in doing segmentation. The places with dark edges are the places where most people draw the boundaries. The third column are the results by the proposed algorithm at three scales under a batch computation mode. We compute the score using the program developed by the Berkely group for the best precision and recall, and the average score for the above images is 0.71.

Jingbin Wang, and Song-Chun Zhu for stimulating discussions and helpful comments on the paper.

# References

[1] A. Barbu and S.C. Zhu, "Graph Partition by Swendsen-Wang Cut", *ICCV*, 2003.

[2] Y. Gdalyahu, D. Weinshall, M. Werman, "Self-Organization in Vision: Stochastic Clustering for Image Segmentation, Perceptual Grouping, and Image Database Organization", *PAMI*, vol. 23, no. 10, 2001.

[3] P. J. Green, "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination," *Biometrika*, vol. 82, no. 4, pp. 711-732, 1995.

[4] S. Kumar and M. Hebert, "Discriminative Random Fields: A Discriminative Framework for Contextual Interaction in Classification", *Proc. of ICCV*, 2003.

[5] M.W. Lee and I. Cohen, "Proposal Maps driven MCMC for Estimating Human Body Pose in Static Images", *CVPR*, 2004.

[6] G. Mori, X. Ren, A. Efros, and J. Malik, "Recovering Human Body Configurations: Combining Segmentation and Recognition", *CVPR*, 2004.

[7] D. Martin, C. Fowlkes and J. Malik, "Learning to Detect Natural Image Boundaries Using Local Brightness, Color and Texture Cues", *PAMI*, May 2004.

[8] K. Murphy, A. Torralba, and W. Freeman, "Using the Forest to See the Trees:A Graphical Model Relating Features, Objects and Scenes", *NIPS*, 2003.

[9] D. Ramanan and D. Forsyth, "Using Temporal Coherence to Build Models of Animals", *ICCV*, 2003.

[10] E. Sharon, A. Brandt, and R. Basri, "Segmentation and Boundary Detection Using Multiscale Intensity Measurements", *CVPR*, 2001.

[11] N. Shental, A. Zomet, T. Hertz, and Y. Weiss, "Learning and inferring image segmentations with the GBP typical cut algorithm", *ICCV*, 2003.

[12] Z. Tu and S.C. Zhu, "Image segmentation by Data-Driven Markov chain Monte Carlo", *IEEE Trans. PAMI*, vol 24, no 5. May, 2002.

[13] Z. Tu, X. Chen, A. Yuille, and S.C. Zhu, "Image Parsing: Segmentation, Detection, and Object Recognition", *ICCV*, 2003.

[14] Z. Tu, "Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering", *Proc. of ICCV 2005*.

[15] J. Yedidia, W. Freeman, and Y. Weiss, "Generalized Belief Propagation", *NIPS*, 2000.

[16] S. Yu and R. Gross and J. Shi, "Concurrent Object Segmentation and Recognition with Graph Partitioning", *NIPS*, 2002.

[17] J. Zhang, R. Collins, and Y. Liu, "Representation and Matching of Articulated Shapes", *CVPR*, 2004.

[18] S. C. Zhu and A. L. Yuille, "Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multi-band Image Segmentation," *IEEE Trans. PAMI*, vol. 18, no. 9, pp. 884-900, Sept. 1996.