A Hierarchical Markov Random Field Model for Figure-ground Segregation

Stella X. Yu[1,3], Tai Sing Lee[2,3], and Takeo Kanade[1,2]

[1] Robotics Institute
[2] Department of Computer Science
Carnegie Mellon University
[3] Center for the Neural Basis of Cognition
5000 Forbes Ave, Pittsburgh, PA 15213-3890
{stella.yu,tai,tk}@cs.cmu.edu

Abstract. To segregate overlapping objects into depth layers requires the integration of local occlusion cues distributed over the entire image into a global percept. We propose to model this process using hierarchical Markov random field (HMRF), and suggest a broader view that clique potentials in MRF models can be used to encode any local decision rules. A topology-dependent multiscale hierarchy is used to introduce long range interaction. The operations within each level are identical across the hierarchy. The clique parameters that encode the relative importance of these decision rules are estimated using an optimization technique called learning from rehearsals based on 2-object training samples. We find that this model generalizes successfully to 5-object test images, and that depth segregation can be completed within two traversals across the hierarchy. This computational framework therefore provides an interesting platform for us to investigate the interaction of local decision rules and global representations, as well as to reason about the rationales underlying some of recent psychological and neurophysiological findings related to figure-ground segregation.

1 Introduction

Figure-ground organization is a central problem in perception and cognition. It consists of two major processes: (1) depth segregation - the segmentation and ordering of surfaces in depth and assignment of *border ownerships* to relatively more proximal objects in a scene [15, 26, 27]; (2) figural selection - the extraction and selection of a figure among a number of 'distractors' in the scene. Evidence of both of these processes have been found in the early visual cortex [17, 19, 20, 36].

In computer vision, figure-ground segregation is closely related to image segmentation and has been studied from both contour processing and region processing perspectives. Contour approaches perform contour completion based on good curve continuation [11, 12, 24, 32, 33], whereas region approaches perform image partitioning based on surface properties [28, 30, 37, 39].

Here, we focus on the issue of global depth segregation based on sparse occlusion cues arisen from closed boundaries. The importance of local occlusion

cues in determining global depth perception can be appreciated in our remarkable ability in inferring relative depths among objects in cartoon drawings (Fig. 1a). These sparse occlusion cues provide important constraints for the emergent global perception of figure and ground. The formation of global percepts from such local cues and the computation of layer organizations have been modeled as an optimization process with a surface diffusion mechanism [8, 9, 22].

In this paper, we extend these earlier works [8, 9, 22] by embedding explicit decision rules for contour continuation and surface depth propagation in local units of a Hierarchical Markov random field model. The multiscale hierarchy is sensitive to the topology of image structures and is used to facilitate rapid long range propagation of local cues. We also develop a parameter learning method using linear programming to estimate the parameters that encode the relative importance of those decision rules. Results show that parameters learned on a few two-object training samples can generalize successfully to multiple-object images.

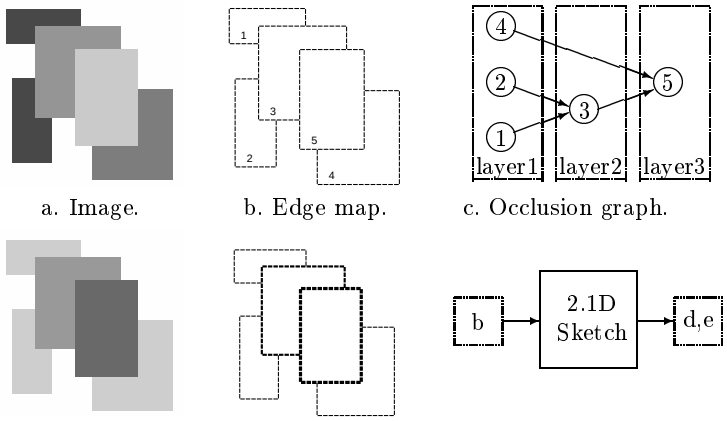
The rest of the paper is organized as follows. Section 2 describes the problem and expands our method in detail. Section 3 shows our results on a new test image. Section 4 concludes the paper with a discussion.

2 Methods

2.1 Problem formulation

For simplicity, we take an edge map (Fig. 1b) with complete and closed contours of rectangular shapes as input to our system. These shapes can overlap and occlude one another. The occluded part of an object is not visible. The system is to produce two complementary maps as output (Fig. 1f): a pixel depth map (Fig. 1d) where a higher depth value is assigned to pixel depth units of a more proximal surface and a lower value to pixel units of a more distant surface; and an edge depth map in which the edge depth units at the border of a more proximal surface assume a higher value. The edge depth units assume the same depth value as the pixel depth units of the surface to which they belong (Fig. 1e). These two representations are sufficient to specify the depth ordering sequence of objects in the scene.

In general, it is not possible to recover the exact depth ordering or overlap sequence in the scene since the solution is not unique. For example, there can be multiple choices when objects do not occlude each other directly (object 1 and 2 in Fig. 1b) and when we cannot tell which object is occluding which (object 3 and 4 in Fig. 1b). If we represent visible pairwise object occlusion relationships in a directed graph (Fig. 1c), these two cases correspond to the existence of unconnected siblings of the same parent. Instead of recovering the overlap sequence, we can sort object depths into layers, ordered by occlusion. This problem is called the 2.1D sketch in [28]. If there is a directed cycle in the graph, then the depth cannot be segregated into layers. We define the depth assignment solution to be the set of smallest depth labels that satisfy all the



a. Image. b. Edge map. c. Occlusion graph.

d. Pixel depth label. e. Edge depth label. f. Goal of our model.

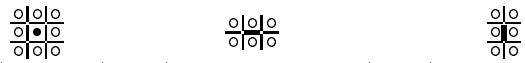
Fig. 1. Segregate depth into layers. Rectangular objects are numbered in b. Darker object surfaces/edges are in front of lighter object surfaces/edges in d/e. Given an edge map as input, our model produces two complementary depth maps as output.

visible occlusion relationships. For example, object 4 in Fig. 1c is on layer 1 rather than layer 2.

2.2 MRF model

Segregating depth into layers is a global process which requires the information to be integrated over the entire image. A change of configuration in a small area can influence the depth labeling at a distance. On the other hand, there exist critical local cues such as T and L junctions which give rise to 3D percepts. If each of these cues can be clearly classified and labeled, and there is an unique association between these cues and 3D depth, depth labeling can be solved by logical inference, for example, using the occlusion graph in Fig. 1c. However, there is always uncertainty in identifying local cues in real images and there is no universal rule of association between a low level cue and a high level percept. The ambiguity in this association is reduced with an increase in the range of integration. For example, two L-junctions can be configured to form a T-junction which is not related to occlusion. The meaning of this T-junction can be disambiguated by gathering information from the origins of the arms and stem of the T-junction.

Long range influence can be mediated by local computation using MRF [10, 21]. An MRF is defined over a graph $\mathcal{G}$, which is determined by its site set $\mathcal{S}$ and neighborhood system η. $\mathcal{S} = Z_m \cup Z'_m$, where Z_m is an $m \times m$ pixel lattice and Z'_m is its dual lattice consisting of an $m \times (m-1)$ and an $(m-1) \times m$ interleaved grids for *line sites* [10]. The *coupled neighborhood* of a site includes both its peer sites and dual sites, as illustrated in Fig. 2.



a. η(pixel site) b. η(horizontal line site) c. η(vertical line site)

Fig. 2. The neighborhood system η used in the model.

Given an edge map, $g : Z'_m \mapsto \{0,1\}^{2m(m-1)}$, with 1 and 0 indicating the presence and absence of an edge respectively, we would like to find a depth map on both pixel and line sites, $h : Z_m \cup Z'_m \mapsto \{0,\infty\}^{m^2} \cup \{1,\infty\}^{2m(m-1)}$, with the depth layer numbered from 0 (for background). To model the depth segregation by MRF, we need to specify *clique potentials* $V_c(\omega)$, ω being a particular configuration of an MRF, and c being a *clique* defined as a subset of sites, which consists of either a single site or more sites where any two of them are neighbors. The probability $P(\omega)$ can be written as

$$P(\omega) = \frac{e^{-U(\omega)}}{Z}, \quad Z = \sum_{\omega \in \Omega} e^{-U(\omega)}, \quad U(\omega) = \sum_{c \in \mathcal{C}} V_c(\omega),$$

where Z is called the *partition function* and $U(\omega)$ the *energy function*.

MRF's have been widely used in texture modeling [5], as well as in image segmentation [10, 13]. In texture modeling, the clique potentials are used to model the probability of co-occurrence of subsets of pixels [5] or capture marginal probability distributions in terms of filter responses [38]. In image segmentation, it is closely related to the energy functional approaches [4, 10, 25] and the clique potentials are used to encode smoothness priors [10]. In our formulation below, we generalize the idea of multi-level logistic models, and suggest a broader view that clique potentials can be more general so that they can encode arbitrary local decision rules.

2.3 Encoding Local Decision Rules

To model depth segregation process in MRF, we seek to make correct depth labeling correspond to the most probable configurations or equivalently configurations of the minimum energy.

Let χ and γ denote two indicator functions, which map from {True, False} to $\{1,0\}$ and $\{-1,1\}$ respectively. $\gamma(\cdot) = 1 - 2\chi(\cdot)$. Let ζ denote the *sign* function, which takes on $-1, 0, 1$ for negative, zero and positive numbers respectively. The line site a between pixel i and j is denoted by $a = i \circ j$ and conversely, the set of pixels associated with the line is denoted by $a^\circ = (i,j)$, with i and j ordered from left to right or from top to bottom. In particular, $(i,j) \circ (i,j+1)$ and $(i,j) \circ (i+1,j)$ are abbreviated as $(i, j\circ)$ and $(i\circ, j)$ respectively. Using these symbols and notations, we can define $V_c(h|g)$ to encode our prior knowledge in

terms of 10 local rules.

$$
\begin{aligned}
& V_c(h|g) \\
&= \textstyle\sum_{a=(i\circ j)\in c} \beta_1 \cdot \gamma(h_i = h_j) \cdot \chi(g_a = 0) && \text{(rule 1)} \\
&+ \textstyle\sum_{a=(i\circ j)\in c} \beta_2 \cdot \gamma(h_i \neq h_j) \cdot \chi(g_a = 1) && \text{(rule 2)} \\
&+ \textstyle\sum_{a=(i\circ j)\in c} \beta_3 \cdot \gamma\Big(h_a = \max(h_i, h_j)\Big) \cdot \chi(g_a = 1) && \text{(rule 3)} \\
&+ \textstyle\sum_{(a=i\circ j, b=k\circ l)\in c^l} \beta_4 \cdot \gamma(h_a = h_b) \cdot \chi(g_a = g_b = 1) && \text{(rule 4)} \\
&+ \textstyle\sum_{(a=i\circ j, b=k\circ l)\in c^l} \beta_5 \cdot \gamma\Big(\zeta(h_i - h_j) = \zeta(h_k - h_l)\Big) && \text{(rule 5)} \\
&\quad \cdot \chi(h_i \neq h_j, h_k \neq h_l) \cdot \chi(g_a = g_b = 1) \\
&+ \textstyle\sum_{(a=i\circ k, b=j\circ k)\in c^c} \beta_6 \cdot \gamma(h_a = h_b) \cdot \chi(g_a = g_b = 1) && \text{(rule 6)} \\
&+ \textstyle\sum_{(a=i\circ k, b=j\circ k)\in c^c} \beta_7 \cdot \gamma\Big(\zeta(h_i - h_k) = \zeta(h_j - h_k)\Big) && \text{(rule 7)} \\
&\quad \cdot \chi(h_a = h_b) \cdot \chi(g_a = g_b = 1) \\
&+ \textstyle\sum_{(a=i\circ j, b=k\circ l, u=j\circ l, v=i\circ k)\in c^t} \beta_8 \cdot \Big(\gamma(h_a > h_u) + \gamma(h_b > h_u)\Big) && \text{(rule 8)} \\
&\quad \cdot \chi\Big(\zeta(h_i - h_j) = 1 \cup \zeta(h_k - h_l) = 1\Big) \cdot \chi(g_a = g_b = g_u = 1 \cap g_v = 0) \\
&+ \textstyle\sum_{(a=i\circ j, b=k\circ l, u=j\circ l, v=i\circ k)\in c^t} \beta_9 \cdot \Big(\gamma(h_i > h_j) + \gamma(h_k > h_l)\Big) && \text{(rule 9)} \\
&\quad \cdot \chi\Big(\zeta(h_i - h_j) = 1 \cup \zeta(h_k - h_l) = 1\Big) \cdot \chi(g_a = g_b = g_u = 1 \cap g_v = 0) \\
&+ \textstyle\sum_{(a=i\circ j, b=k\circ l, u=j\circ l, v=i\circ k)\in c^t} \beta_{10} \cdot \gamma(h_i = h_l) && \text{(rule 10)} \\
&\quad \cdot \chi(g_a = g_u = 0 \cup g_b = g_v = 0)
\end{aligned}
$$

where c^l, c^c, c^t are the sets of cliques for aligned lines, corners and crosses:

$$
\begin{aligned}
c^l &= \{(a,b) : a = (i\circ, j), b = (i\circ, j+1); a = (i, j\circ), b = (i+1, j\circ), a, b \in c\}, \\
c^c &= \{(a,b) : a = (i\circ, j), b = (k, l\circ), |i-k| \leq 1, |j-l| \leq 1, a, b \in c\}, \\
c^t &= \{(a,b,u,v) : (a,b) \in c^l, (u,v) \in c^l, \{a,b\} \cap \{u,v\} = \emptyset, a^\circ \cup b^\circ = u^\circ \cup v^\circ\}.
\end{aligned}
$$

The two indicator functions, χ and γ, enable us to embed the conjunction of *if* conditionals into the clique potentials. Let us decode rule 1 as an example. Consider the line site a between pixel i and j. If the clause $(g_a = 0)$ is not true, i.e. there is an edge between the two pixels, then this first term is zero, no action will be taken; otherwise, if the clause $(h_i = h_j)$ is also true, i.e. the pixel depth values at the two sites are equal, then the term produces a reward of $-\beta_1$, lowering the energy. However, if it is not true, i.e. the depth values at the two pixel sites are different, then $V_c(h|g)$ gets β_1 on this term as a punishment, increasing the energy. Here we require all βs to be positive. These 10 rules are summarized in Table 1 and they can be classified into 6 groups as follows.

Group 1: *Depth continuity within surface.* Rules 1 and 10 assert that surface depth units in adjacent locations should be continuous. Adjacency is defined on two kinds of neighborhood. Rule 1 is concerned with the first order neighborhood (up, down, left and right neighbors), and rule 10 is concerned with the second order neighborhood (diagonally adjacent pixels).

Group 2: *Depth discontinuity across edges.* Rule 2 asserts that when there is an edge between two adjacent locations, the surface depth units in those two locations must have different depth values.

Configuration	Condition A	Pattern B	Score C	#	Meaning
$\circ$ | $\circ$ i a j	$g_a = 0$	$h_i = h_j$	β_1	1	Depth continues in surface.
	$g_a = 1$	$h_i \neq h_j$	β_2	2	Depth breaks at edges.
	$g_a = 1$	$h_i \neq h_j$	β_3	3	Edges belong to surface in front.
k b l $\circ$ | $\circ$ $\circ$ | $\circ$ i a j	$g_a = g_b = 1$	$h_a = h_b$	β_4	4	Depth continues along contour.
	$g_a = g_b = 1$ $h_i \neq h_j$ $h_k \neq h_l$	$\zeta(h_i - h_j)$ = $\zeta(h_k - h_l)$	β_5	5	Depth polarity continues along contour.
j $\circ$ | $\circ$ b i a k	$g_a = g_b = 1$	$h_a = h_b$	β_6	6	Depth continues around corners.
	$g_a = g_b = 1$ $h_a = h_b$	$\zeta(h_i - h_k)$ = $\zeta(h_j - h_k)$	β_7	7	Depth polarity continues around corners.
k b l v $\circ$ | $\circ$ u $\circ$ | $\circ$ i a j	$g_a = g_b = 1$	$h_a > h_u$	β_8	8	Depth breaks on edges at T-junctions.
	$g_u = 1, g_v = 0$	$h_b > h_u$	β_8		
	$\zeta(h_i - h_j) = 1$ or	$h_i > h_j$	β_9	9	Depth breaks in surface at T-junctions.
	$\zeta(h_k - h_l) = 1$	$h_k > h_l$	β_9		
	$g_a = g_u = 0$ or $g_b = g_v = 0$	$h_i = h_l$	β_{10}	10	Depth continues in surface.

Table 1. Encoding rules in clique potentials. Each of these β terms encodes a logic rule, which in general reads like this: if current clique configuration does not satisfy condition A, it gets a score of 0; otherwise, if condition A is satisfied, pattern B is expected; if B is also satisfied, then it gets a negative score $-C$; otherwise it gets a positive score C. a, b, u and v are labels for line sites while i, j, k, l are labels for pixel sites in the cliques.

Group 3: *Border-ownerships.* Rule 3 specifies that an edge depth unit shares the same depth value as the surface that owns it.

Group 4: *Depth continuity along contour.* Rules 4 and 6 specify the edge depth value along contour or corners should be continuous.

Group 5: *Depth polarity continuity along contour.* Rules 5 and 7 specify the depth polarity of surface units across an edge unit should be continuous along contour and corners.

Group 6: *Occlusion relationships at T-junctions.* At those T-junctions, rule 8 and 9 specify that the arms of the T are in front of the T stem.

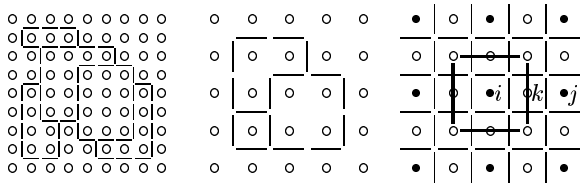
In this formulation, the clique potentials no longer simply specify local co-occurrence, smoothness constraints or filter response histograms as in other MRF models, but are generalized to encode a set of local decision rules. From neural modeling perspective, the units in the network are not neurons with linearly weighted inputs and sigmoidal activation functions, but are capable of performing complicated logical computations individually. Recent findings and models in cellular neurophysiology [1, 18, 23] suggest neurons are capable of computations more sophisticated than previously assumed.

The relative importance of the weights βs in the depth segregation can be estimated using a variety of methods. We will describe a particular supervised learning method we use in a later section.

2.4 Multiscale Hierarchy

The MRF model described above suffers from being myopic [14] in local computation and sluggish at propagating constraints between widely separated processing elements [31]. This problem can be overcome by embedding the MRF in a hierarchy using multigrid techniques.

We build an edge map pyramid by down-sampling with a factor of 2(Fig. 3). Assuming $m = 2^k + 1$, we preserve spatial locations at the center and the boundary of the lattices throughout the levels of the hierarchy. Let η^l and Z_m^l denote the neighborhood and lattice at level l. Let $\hat{}$ and $\check{}$ address the correspondence between pixels at level l and $l+1$, such that $\hat{i} \in Z_m^{l+1}$ and $i \in Z_m^l$, or, $\check{i} \in Z_m^l$ and $i \in Z_m^{l+1}$ point to the same spatial location on the sampling grid (Figure 3c). The edge map at a high level is determined by $g^l_{\hat{i}\circ\hat{j}} = \zeta\big((g^{l-1}_{i\circ k} + g^{l-1}_{k\circ j}) \cdot |\zeta(h^{l-1}_k - h^{l-1}_i) + \zeta(h^{l-1}_j - h^{l-1}_k)|\big)$.



a. g^l b. g^{l+1} c. sampling

Fig. 3. Hierarchical edge maps and illustration of sampling. a. Edge map at some level l. b. Edge map at a higher level $l+1$. c. The hierarchy is built by downsampling pixels (filled circles) by a factor of two and inferring new lines (darker lines) between sampled pixels based on the depth and edge maps at a lower level.

If an edge is considered to disconnect two neighboring pixels, the above operation preserves connectivity when there is only one edge separating two sampled pixels. However, when there are two edges in a local neighborhood, the depth polarity of the edges has to be considered (Fig. 4). When two nearby edges have the same polarity, they can be merged into one edge of the same polarity as in (Fig. 4a). When the two edges have opposite depth polarities as in (Fig. 4b), they would disappear at the next level of the hierarchy. In this way, relaxation at each resolution deals with topologically equivalent diffusion processes and thus the same procedure can be applied.

The intergrid transfer functions involve *restriction* $\Uparrow$ and *extension* $\Downarrow$.

$$h^l = \Uparrow (h^{l-1}, g^{l-1}), \qquad h^l = \Downarrow (h^l, g^l, h^{l+1}).$$

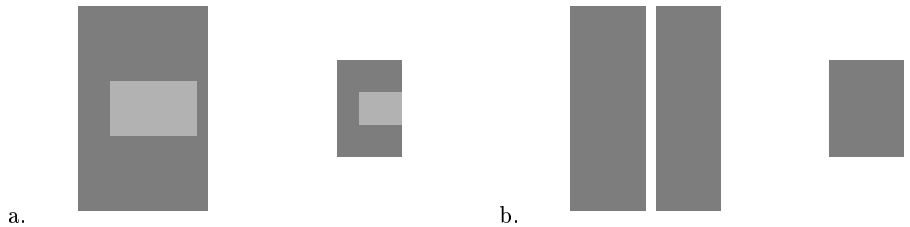


Fig. 4. The operation in the multiscale hierarchy takes edge depth polarity into consideration. a. Edges of overlapping shapes have the same depth polarity and are preserved at a coarser resolution. Edges of abutting shapes that have opposite depth polarities will disappear at a coarser resolution, as indicated by the disappearance of the two edges between the two shapes at the coarser scale. In this way, relaxation at each resolution deals with topologically equivalent diffusion processes and thus the same procedure can be applied. In both a and b, the pictures on the left and right indicate the images at a fine and coarse resolution respectively.

During the restriction, *smoothing* is carried out on *connected* pixel sites. For line sites, the smoothing on aligned horizontal (vertical) edges is blocked by vertical(horizontal) edge neighbors:

$$h_{\hat{i}}^l = \max\left\{h_k^{l-1} : g_{i\circ k}^{l-1} = 0, k \in Z_m^{l-1} \cap \eta_i^{l-1}\right\}$$

$$h_{(\hat{i},\hat{j}\circ)}^l = \max\left\{h_{(p,q\circ)}^{l-1} : g_{(p,q)\circ(i,q)}^{l-1} = g_{(p,q+1)\circ(i,q+1)}^{l-1} = 0, p \in [i-1, i+1], q \in [j, j+1]\right\}.$$

$h_{(\hat{i}\circ,\hat{j})}^l$ can be defined in a similar fashion. Median filtering can also be used in the above. During the extension, the information is selectively transferred to a fine grid. The dual operation of smoothing is *diffusion*, which is subject to boundary blockage:

$$\begin{aligned}
h_{\check{i}}^l &= h_i^{l+1} \\
h_{(\check{i},\check{j}\circ)}^l &= h_{(i,j\circ)}^{l+1}, \text{if } g_{(i,j\circ)}^{l+1} = 1, g_{(i,j\circ)}^l = 1, g_{(i,(j+1)\circ)}^l = 0 \text{ or } h_{(i,j)}^l > h_{(i,j+2)}^l \\
h_{(\check{i},(\check{j}+1)\circ)}^l &= h_{(i,j\circ)}^{l+1}, \text{if } g_{(i,j\circ)}^{l+1} = 1, g_{(i,(j+1)\circ)}^l = 1, g_{(i,j\circ)}^l = 0 \text{ or } h_{(i,j)}^l < h_{(i,j+2)}^l \\
h_i^l &= \max\left\{h_{\hat{k}}^{l+1} : g_{i\circ k}^l = 0, k \in Z_m^l \cap \eta_i^l, \hat{k} \in Z_m^{l+1}\right\} \\
h_{(i,j\circ)}^l &= \max\left\{h_{(\hat{p},j\circ)}^{l+1} : g_{(p,j)\circ(i,j)}^l = g_{(p,j+1)\circ(i,j+1)}^l = 0, p \in [i-1, i+1]\right\}.
\end{aligned}$$

Finally, to complete our HMRF model, we provide site visitation and a multi-level interaction scheme. A complete *sweep* of all the sites includes four checker board update schemes on first pixel sites and then line sites. The separate visitation to pixel sites and line sites allows each of the two MRF's to develop fully in itself so that the resultant configuration provides enough driving force for the other to change accordingly. The hierarchy is visited bottom-up through

restriction and then top-down through extension. The MRF at each level carries out a relaxation process until its configuration converges. When the configuration at the lowest level does not change after visiting the entire hierarchy, that configuration is the final result.

In summary, multiscale not only helps to speed up computation, but also helps propagating sparse depth cues at boundary to the interior of the surface by longer range interactions at higher levels of the hierarchy. In addition, at each level of the hierarchy, we repeat the same relaxation operation of local decision rules. This relies on the consistency of topology in the restriction and extension operations.

2.5 Parameter Estimation

The above HMRF model has unknown parameter $\beta = [\beta_1, \cdots, \beta_{10}]^T$. The major difficulty in estimating MRF parameters lies in the evaluation of the partition function. There are several approaches to deal with the problem [21]. One way is to avoid the partition function in the formula, such as pseudo-likelihood [3] and least squares(LS) fit [5]. Another way is to use some estimation techniques such as the coding method, mean field approximation [35] and Markov Chain Monte Carlo maximum likelihood [6]. The approach we take here is to derive a set of constraints on β using a method called *learning from rehearsals* and use linear programming to obtain the β that satisfy these constraints.

This perturbation-based method is most closely related to the LS fit approach [5]. Let $U_k(\omega)$ denote the sum of clique potentials $V_c(\omega)$ over all cliques containing site k. Since $V_c(\omega)$ is a linear function of β, so is $U_k(\omega)$. In general, it can be written as $U_k(\omega) = x(\omega, k) \cdot \beta$, where $x(\omega, k)$ can be obtained by evaluating clique potentials on the configuration ω confined to the neighborhood of k. In the LS approach, the probabilities of training samples are utilized to derive a set of equalities based on the formula below.

$$\ln\Big(\frac{P(\omega_k = i|\omega_{\eta_k})}{P(\omega_k = j|\omega_{\eta_k})}\Big) = -[U_k(\omega) - U_k(\omega')] = -\Big[x(\omega, k) - x(\omega', k)\Big] \cdot \beta,$$

where $\omega_k = i, \omega'_k = j, \omega_{S\setminus\{k\}} = \omega'_{S\setminus\{k\}}$ are given. However, this is only applicable to the case where $P(\omega_k = j|\omega_{\eta_k}) > 0$. This condition may not be very restrictive in texture modeling, but it is in our model because when ω_{η_k} is set, ω_k is often determined as well. Another problem concerns numerical stability. When $P(\omega_k = j|\omega_{\eta_k})$ is small, the estimation is not accurate. To relax this condition, we derive inequality constraints on β instead:

$$\Big[x(\omega, k) - x(\omega', k)\Big] \cdot \beta < 0, \text{if } P(\omega_k = i|\omega_{\eta_k}) > P(\omega_k = j|\omega_{\eta_k}).$$

We do not need to know the exact sizes of the two probabilities, but rather the relative order of the two quantitities. In other words, for a given neighourhood configuration ω_{η_k}, if we know label i is preferred to label j for site k, we obtain a constraint which ensures that site k assuming value of i leads to a lower energy.

We obtain two sets of constraints on β in the form of above inequalities. We generate a set of images which have two randomly positioned rectangular shapes. Both the edge map g and the final depth map h are known for each training image. The first set of constraints come from the fact that given neighbors of a site assuming correct labels, this site prefers its own correct label. This will map the correct labeling into a local minimum in the configuration space. We summarize all such constraints into $A \cdot \beta < 0$, where the rows of A come from the perturbation on the teacher map h at all sites:

$$\Big[x(\omega, k) - x(\omega', k)\Big] \cdot \beta < 0, \text{for } P(\omega_k|\omega_{\eta_k}) > P(\omega'_k|\omega'_{\eta_k}),$$

where $\omega_k = h_k, \omega'_k = h_k \pm 1, \omega_{\mathcal{S}\setminus\{k\}} = \omega'_{\mathcal{S}\setminus\{k\}} = h_{\mathcal{S}\setminus\{k\}}$. An example on an L-junction is given in Fig. 5. As can be seen in the example, the first set of constraints are usually satisfiable as the correct label is far better than any other choices according to the rules we encode in the energy function.

a. Configuration.

$$\begin{bmatrix} -4 & -4 & -4 & 0 & -2 & 0 & -2 & 0 & 0 & -2 \\ -4 & 0 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \end{bmatrix} \cdot \beta < 0$$

b. Constraints.

Fig. 5. Derive the first set of constraints from teacher depth maps. a. An L-junction at a pixel site's neighborhood. The teacher depth map in this neighborhood is 0 for unfilled circles and 1 for all the line sites and filled circles. b. Two constraints obtained by perturbation on the depth value of the center pixel site. The first constraint comes from the difference in the energy functions for labeling 1 and 0 at the center pixel. The second constraint comes from the difference in the energy functions for labeling 1 and 2 at the center pixel, all its neighbors assuming correct labels. These two constraints on β are trivial as any $\beta > 0$ is feasible.

The first set of constraints only guarantee local behaviors when the system is close to the optimal configuration. They may not be enough to drive an initial configuration toward that final optimal configuration. A second set of constraints are derived for this purpose. This is not easy because there are many possible different paths of evolution from one configuration into another, and we do not necessarily know the intermediate configurations that the system has to go through in order to arrive at the final state. We develop a method called *learning from rehearsals* to overcome this difficulty. Not knowing β in advance or teacher depth maps at intermediate steps, we use the following principle to choose a preferred label during the learning process and to establish its validity by rehearsing. The principle is that a site's depth value should be as close as possible to its final target value at that site subject to the dragging force from its current neighborhood configuration. That is, the derivation of the second set of constraints is based on finding the most effective intermediate states that

will move the system from the initial state to the final state with a minimum number of steps. Once a preferred depth label is chosen, we can derive plausible constraints in a similar way as we did in Fig. 5. We build a constraint database during learning. Whenever a new constraint is to be added into the database, we check its own feasibility as well as its compatibility with those already in the database. We implement two simple checks on these two properties by testing if new constraint $\alpha \cdot \beta < 0$ leads to $\beta < 0$, or some other constraint requiring $-\alpha \cdot \beta < 0$ already exists in the database. If either of these conditions is true, the constraint is removed and accordingly the hypothesized teacher is abandoned and next candidate depth value, which is not so close to the target value as this one, is chosen. When new constraints can be checked into the database, the intermediate teacher is instantiated. We make the depth assignment at the site and continue the learning process as if all the conditions were satisfied. We call this process *rehearsal* because we carry out the relaxation without knowing whether there is a feasible set of β. We summarize the second set of constraints in $\widetilde{A} \cdot \beta < 0$.

The system will rehearse and practice, like a baby learning to walk, trying to reach the final goal from an initial state, while generating constraints on its gaits at each step along the way. Having obtained these two sets of constraints on β, we can proceed to find the set of β that satisfy most constraints by optimizing the following linear programming problem,

$$LP: \text{ minimize: } \xi \sum_i \delta_i + \sum_j \tilde{\delta}_j,$$
$$\text{subject to: } A \cdot \beta - \delta \leq -1, \tilde{A} \cdot \beta - \tilde{\delta} \leq -1, \delta \geq 0, \tilde{\delta} \geq 0, \beta \geq 1,$$

where $\xi \geq 1$ is a weighting factor between the two sets of constraints, here we simply set it to 1. Since not every constraint can be satisfied, we introduce slack variable δ and $\tilde{\delta}$ to turn them into soft constraints. Linear programming is used to find the set of β that minimizes the total amount of violation of the constraints.

Once LP yields a set of β, we examine the constraints' slack variables to see which constraint is most severely violated (the largest positive δ or $\tilde{\delta}$). We find that a bad constraint is typically generated by making a hasty jump before the condition is mature, putting an unnecessarily harsh constraint on β. We go back to the constraint database and remove this constraint and choose alternative teachers for all the patterns that give rise to this constraint. This prevents that constraint to be selected again in subsequent rehearsals. We remove enough bad constraints till a feasible β is found. We test its validity by relaxation using this β to see if it can actually drive the system from the initial state to the final state for each training example. The learning and checking processes are iterated until final configurations for all the training images are correct. The learning proceeds from simple to complex images, to gradually build up a set of reasonable constraints. Most time when a new image is learned, only a couple of iterations is sufficient to obtain a new β such that all δ and $\tilde{\delta} = 0$.

3 Results

Learning on a small set of training images containing *two objects* singles out a unique value for β, where $\beta = [18, 9, 97, 23.3, 3.2, 86.7, 3.35, 16.5, 42.5, 137, 20.8]$. With this set of parameters, the model produces reasonable results for a set of test images that the system has never been exposed to before.

Figure 6 shows how the system responds to a test image with *five overlapping rectangles* in the scene. The system generalizes very well in its response to this new input configuration. A sequence of 8 snap shots are taken at different time points during the evolution of the system. Snap shot 1 shows the system detecting T-junctions and starting propagating its initial result one level up the hierarchy. Snap shot 2 shows the information has propagated to the third level, and propagation of depth information within surface is now evident at the second level. Snap shots 3 and 4 show the information has propagated to the fourth and fifth levels respectively. Snap shot 5 shows the information starts to propagate down the hierarchy, introducing rapid filling-in of surface depth and depth segregation in snap shot 6. Snap shots 7 and 8 show the completion of surface/contour depth interpolation and segregation. All these are completed very rapidly in two iterations up and down the hierarchy.

4 Discussion

In this paper, we present a hierarchical MRF model to perform depth segregation of region edge maps. The model is hierarchical rather than simply multiscale because its fine-to-coarse transform is topology-dependent. In this work, we propose a broader view that clique potentials in MRF can be used to encode any local logical decision rules. By introducing a set of rules that asserts continuity of depth assignment values along contour and within surfaces, and discontinuity of depth assignment values across contours, we demonstrate a system that automatically integrates sparse local relative depth cues arisen from T-junctions over long distance into a global ordering of relative depths. Interestingly, because the rules we set are encoding relative relationships between objects, the system trained on scenes containing two objects can actually generalize and perform correctly when a scene containing five objects is first encountered.

We also propose a new method called *learning from rehearsals* for estimating MRF parameters. In this method, we derive a set of constraints based on perturbation of target solutions and the rehearsals of relaxation processes, and then use linear programming to obtain feasible solutions. Conflicting constraints are removed and constraint derivation by rehearsals and parameter solving are repeated until there is a set of parameters that work correctly for every test image. We do not have a theoretical proof that the learning of this system will actually converge. We have restricted our domain of investigation to a world of simple shapes so that we can gain a better understanding of the system and associate constraints with their origins.

Another assumption we made is that the input edge maps are closed contours. There is no technical difficulty here in so far as there exist a number of algorithms

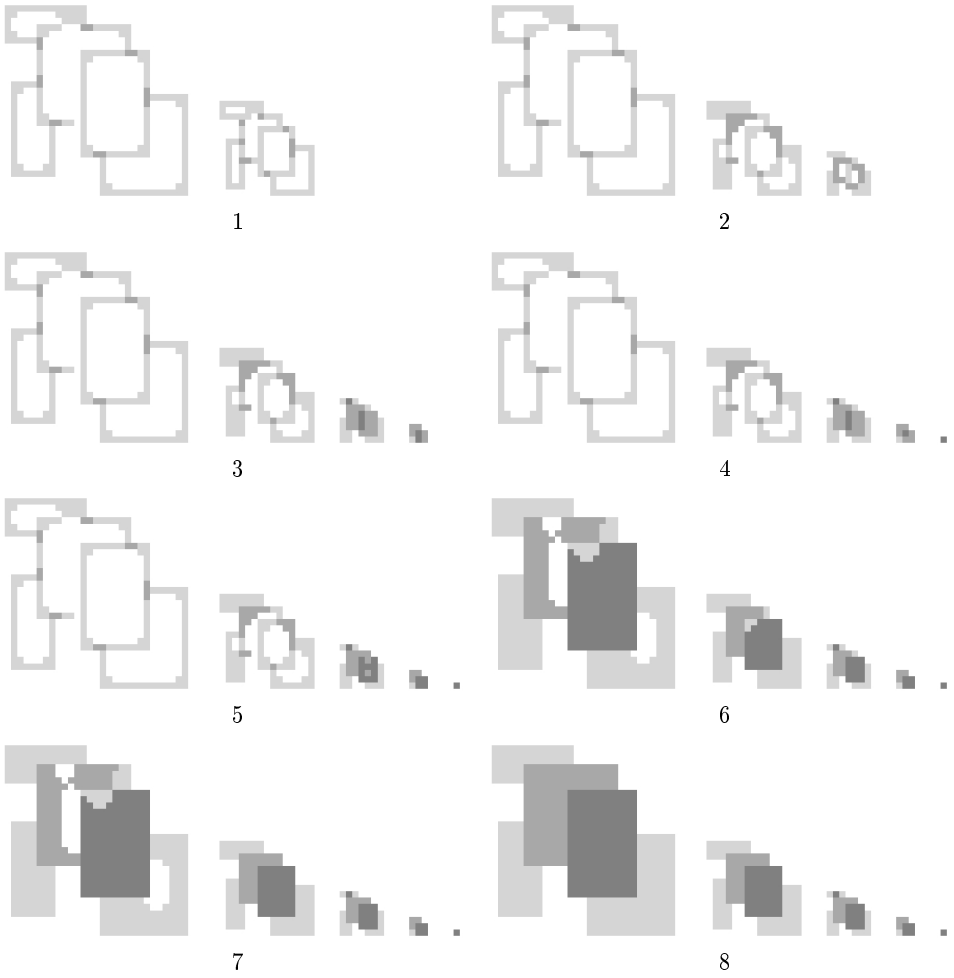


Fig. 6. Dynamics of the HMRF's response to a 5-object test image. The parameters are learned on a few 2-object images. Shown here are a number of snaps shots taken at different time points during the depth segregation computation. The hierarchy is traversed twice till its complete convergence to the correct labeling.

such as active contours [16] and region competition algorithms [37, 39] that can produce complete and closed contours. However, depth segregation and ordering can potentially help segmentation by feeding back additional constraints to organize the contour detection and completion process itself. Earlier work by Belhumeur [2] and recent work by Yu and Shi [34] are examples of how depth cues and intensity cues can be integrated simultaneously into the segmentation process. These are potential directions for future research.

We think this HMRF model for depth segregation might provide a plausible computational framework for reasoning about and understanding the basic computational constraints and neural mechanisms underlying local and global integration and figure-ground segregation in the brain. This work provides us with several insights to some psychological and neurophysiological phenomena.

First, brightness has been observed to propagate in from the border in the psychophysical experiment by Paradiso and Nakayama [29]. Such phenomenon has been postulated to be mediated by horizontal connections in V1, for example in Grossberg and Mingolla's model [11]. Here, we show that a hierarchical framework can speed up the diffusion of depth assignment process considerably. In fact, traversing up and down the hierarchy twice is sufficient to complete the computation. This suggests that both the brightness perception and the depth segregation could be mediated by the feedback from V2 and V4, which are known to have receptive fields two and four times larger than those of V1 respectively.

Second, while Paradiso and Nakayama's experiment suggests diffusion in the brightness domain, the similarity in dynamics between brightness diffusion and our depth assignment suggests depth segregation and assignment might be the underlying process that carries the brightness diffusion along. By the same reasoning, one would expect other surface cues such as color, texture and stereo disparity should also be accompanying, if not following, the depth assignment process. It will indeed be interesting to examine experimentally whether the propagation of surface cues follows the depth assignment process or occurs simultaneously. That Dobbins et al. [7] found a significant number of V1, V2 and V4 cells sensitive to distance even in monocular viewing conditions suggests that depth assignment might be intertwined with many early visual processes.

Finally, the hierarchy presented is not simply a multiscale network in that, when the information travels up, the topological relationships between different objects are taken into consideration in such a way that the same relaxation procedure can be applied at each level. For example, edges of overlapping shapes are kept (Fig. 4a), whereas the edges of two nearby shapes appearing side by side would disappear at a coarser resolution(Fig. 4b). This operation can be achieved by taking the sum of depth polarities during the down-sampling process. In order to accomplish this in the network, depth polarity of edges needs to be computed and represented explicitly. This might provide a computational rationale for the existence of the depth-polarity sensitive cells von der Heydt and his colleagues found in V1, V2 and V4 [36].

Acknowledgements

Yu and Lee have been supported by NSF LIS 9720350, NSF CAREER 9984706 and NIH core grant EY 08098.

References

[1] L. F. Abbott. Integrating with action potentials. *Neuron*, 26:3–4, 2000.

[2] P. Belhumeur. A Bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3):237–260, 1996.

[3] J. Besag. Efficiency of pseudo-likelihood estimation of simple Gaussian fields. *Biometrika*, 64:616–8, 1977.

[4] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.

[5] H. Derin and H. Elliott. Modeling and segmentation of noisy and textured images using Gibbs random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):39–55, 1987.

[6] X. Descombes, R. D. Morris, J. Zerubia, and M. Berthod. Estimation of Markov random fields prior parameters using Markov chain Monte Carlo maximum likelihood. *IEEE Transactions on Image Processing*, 8(7):954–62, 1999.

[7] A. C. Dobbins, R. M. Jeo, J. Fiser, and J. M. Allman. Distance modulation of neural activity in the visual cortex. *Science*, 281:552–5, 1998.

[8] D. Geiger and K. Kumaran. Visual organization of illusory surfaces. In *European Conference on Computer Vision*, Cambridge, England, April 1996.

[9] D. Geiger, H. kuo Pao, and N. Rubin. Salient and multiple illusory surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1998.

[10] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–41, 1984.

[11] S. Grossberg and E. Mingolla. Neural dynamics of form perception: boundary completion, illusory figures, and neon color spreading. *Psychological Review*, 92:173–211, 1985.

[12] F. Heitger and R. von der Heydt. A computational model of neural contour processing: Figure-ground segregation and illusory contours. In *International Conference on Computer Vision*, pages 32–40. 1993.

[13] T. H. Hong, K. A. Narayanan, S. Peleg, A. Rosenfeld, and T. Silberberg. Image smoothing and segmentation by multiresolution pixel linking: further experiments and extensions. *IEEE Transactions on Systems, Man, and Cybernetics*, 12:611–22, 1982.

[14] T. H. Hong and A. Rosenfeld. Compact region extraction using weighted pixel linking in a pyramid. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):222–9, 1984.

[15] G. Kanizsa. *Organization in vision*. Praeger Publishers, 1979.

[16] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.

[17] J. J. Knierim and D. van Essen. Neuronal responses to static texture patterns in area v1 of the alert macaque monkey. *Journal of Neurophysiology*, 67(4):961–80, 1992.

[18] C. Koch. Computation and the single neuron. *Nature*, 385:207–210, 1997.

[19] V. Lamme. The neurophysiology of figure-ground segregation in primary visual cortex. *Jounral of neuroscience*, 10:649–69, 1995.

[20] T. S. Lee, D. Mumford, R. Romero, and V. Lamme. The role of primary visual cortex in higher level vision. *Vision Research*, 38:2429–54, 1998.

[21] S. Z. Li. *Markov random field modeling in computer vision*. Springer-Verlag, 1995.

[22] S. Madarasmi, T.-C. Pong, and D. Kersten. Illusory contour detection using MRF models. In *IEEE International Conference on Neural Networks*, volume 7, pages 4343–8. 1994.

[23] H. Markram, J. Lubke, M. Frotscher, and B. Sakmann. Regulartion of synaptic efficacy by coincidence of postsynpatic APS and EPSPs. *Science*, 275:213–215, 1997.

[24] D. Mumford. Elastica and computer vision. In C. L. Bajaj, editor, *Algebraic geometry and its applications*. Springer-Verlag, 1993.

[25] D. Mumford. The bayesian rationale for energy functionals. In B. Romeny, editor, *Geometry-driven diffusion in computer vision*, pages 141–53. Kluwer Academic Publishers, 1994.

[26] K. Nakayama and S. Shimojo. Experiencing and perceiving visual surfaces. *Science*, 257:1357–63, 1992.

[27] K. Nakayama, S. Shimojo, and G. H. Silverman. Stereoscopic depth: its relation to image segmentation, grouping, and the recognition of occluded objects. *Perception*, 18:55–68, 1989.

[28] M. Nitzberg. *Depth from Overlap*. PhD thesis, The Division of Applied Sciences, Harvard University, 1991.

[29] M. A. Paradiso and K. Nakayama. Brightness perception and filling-in. *Vision Research*, 31(7/8):1221–36, 1991.

[30] J. Shi and J. Malik. Normalized cuts and image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 731–7, June 1997.

[31] D. Terzopoulos. Image analysis using multigrid relaxation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):129–39, 1986.

[32] S. Ullman. Filling-in the gaps: the shape of subjective contours and a model for their generation. *Biological Cybernetics*, 25, 1976.

[33] L. R. Williams and D. W. Jacobs. Stochastic completion fields: A neural model of illusory contour shape and salience. *Neural Computation*, 9(4):837–58, 1997.

[34] S. X. Yu and J. Shi. Segmentation with pairwise attraction and repulsion. International Conference on Computer Vision, 2001.

[35] J. Zhang. The mean field theory in EM procedures for Markov random fields. *IEEE Transactions on Image Processing*, 40(10):2570–83, 1992.

[36] H. Zhou, H. Friedman, and R. von der Heydt. Coding of border ownership in monkey visual cortex. *Journal of Neuroscience*, 20(17):6594–611, 2000.

[37] S. C. Zhu, T. S. Lee, and A. Yuille. Region competition: unifying snakes, region-growing and mdl for image segmentation. *Proceedings of the Fifth International Conference in Computer Vision*, pages 416–425, 1995.

[38] S. C. Zhu, Y. N. Wu, and D. Mumford. Filters, random field and maximum entropy: — towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):1–20, 1998.

[39] S. C. Zhu and A. Yuille. Unifying snake/balloons, region growing and Bayes/MDL/Energy for multi-band image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9), 1996.