Learning Coupled Conditional Random Field for Image Decomposition with Application on Object Categorization

Xiaoxu Ma
xiaoxuma@csail.mit.edu,gmail.com

W. Eric L. Grimson
welg@csail.mit.edu

Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Abstract

This paper proposes a computational system of object categorization based on decomposition and adaptive fusion of visual information. A coupled Conditional Random Field is developed to model the interaction between low level cues of contour and texture, and to decompose contour and texture in natural images. The advantages of using coupled rather than single-layer Random Fields are demonstrated with model learning and evaluation. Multiple decomposed visual cues are adaptively combined for object categorization to fully leverage different discriminative cues for different classes. Experimental results show that the proposed computational model of "recognition-through-decomposition-and-fusion" achieves better performance than most of the state-of-the-art methods, especially when only a limited number of training samples are available.

1. Introduction

Generic object recognition has been a challenging task in computer vision for years. Many approaches have been proposed for modeling shape, appearance or a combination of both for object classes [3, 8, 10, 12, 25]. While this work has significantly advanced the field of object categorization, relatively less work has been done in decomposing various visual stimuli, such as contour and texture in images, and adaptively combining them for object recognition. Many popular approaches based on local image patches treat each local patch as an integral entity, which means visual information such as contour, texture and color are mixed together in one descriptor, and given uniform or fixed weights for recognizing objects. However, it has been shown in psychophysical studies [1, 2, 24] that when human observers see an object, the visual stimuli such as contour and texture in an object image are first separated into different channels, processed individually, and then recombined at a later stage in the human visual system for object recognition. Moreover, it is intuitively more sensible to assume that different visual cues should play different roles in discriminating different class pairs. Figure 1 gives a schematic illustration of this idea. For instance, to classify *beaver* versus *emu*, the shape information may be more important since both classes have similar texture and color; while for *laptop* versus *skate*, the keyboard texture of laptops may be a more salient cue for discriminating the two classes.

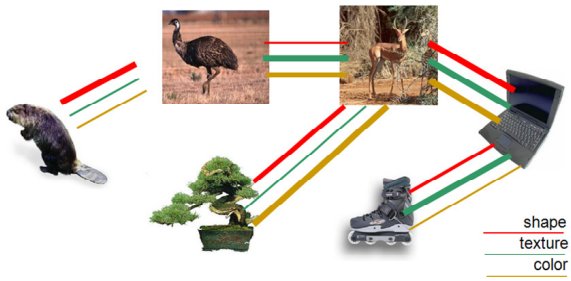


Figure 1. An illustration that various visual cues should have different weights in discriminating different class pairs.

Some work similar in spirit is image segmentation by combining contour and texture [22], and learning the probability of boundaries in natural images with local brightness, color and texture cues [23]. Also related is a range of work in perceptual grouping. Many researchers have applied perceptual grouping as an automatic and preattentive precursor to object recognition [19, 26].

In this paper, we develop a computational system of visual information decomposition and fusion for object recognition. A coupled Conditional Random Field is proposed to model and decompose the interactive processes of contour and texture. The importance of the coupled CRF model is demonstrated by comparing to a single-layer CRF. The decomposition of contour and texture naturally enables methods of matching different visual stimuli separately to fully leverage each visual cue. At the top level of the system, multiple visual cues are adaptively combined, by weighing different visual cues for recognizing different classes. The key concept of this computational model is to achieve better recognition performance by decomposing and recombining multiple disparate visual cues in object images.

2. Contour process and texture process

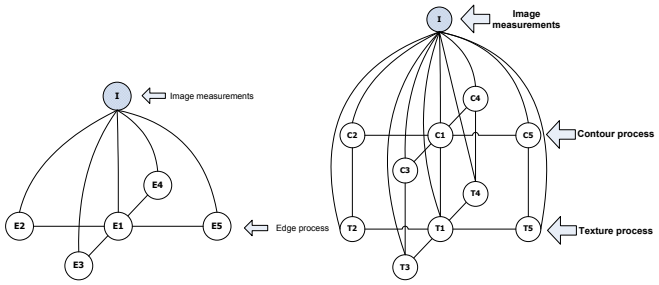
Our first goal is to separate image information into contour and texture. We focus on labeling of edge pixels, assigning to each edge pixel a contour or texture label, based on local context. Once we have learned labelings for edge pixels, we can use the different cues as a complementary basis for classification of objects, including learning which cues are more salient for distinguishing classes.

To decompose contour and texture, we must first define our image processes. In this work, a contour process and a texture process are defined on edge pixels. Each label in the contour process will be one of {1, -1}, signaling an edge pixel as a contour pixel or not; similarly for the texture process. We have some flexibility in how contour pixels are defined. One possibility is that only occluding contours are regarded as contour pixels. An alternative includes both occluding and internal contours as contour pixels. In the latter definition, depending on the scale, some internal edge pixels may be considered as texture flow or an internal contour. For instance, at a large scale, zebra stripes or soccer-ball patches have a repeating pattern and appear to be texture; at a smaller scale, edge pixels from stripes or patches are well-aligned and appear to be internal contours. In this work, we choose the latter definition.

3. Coupled Conditional Random Field for contour and texture interaction

A popular way of labeling image processes is to use a single layer random field grid. Such a model for labeling an edge process, with one node for each edge point, is shown in Figure 2(a). The underlying idea is that labels for each edge point are influenced by nearby labels as well as local measurements, and thus local context helps propagate labels throughout a region. However, this single-layer random field is limited in modeling power. The two processes, contour and texture, could have disparate characteristics and dynamics in their respective inter-point interactions. One distinction lies in the angular alignment of points. In the contour process, compatible contour points are mostly aligned in local neighborhoods. For the texture process, edge points are seldomly well-aligned and often random (recall that locally well-aligned patterns such as zebra stripes are defined as internal contours, part of the contour process.) Hence, it is logical to postulate that contour points are compatible only when they are locally continuous and aligned, while the compatibility of texture points could allow a random layout. This means the compatibility functions of the two processes will exhibit disparate dependencies on an angular alignment parameter. Other different dynamics may also exist in the measurements of coarseness, anisotropy, homogeneity and entropy.

Under these situations, using single-layer random field models inevitably has to introduce a trade-off between distinct dynamics of different processes. To accommodate different characteristics of interactions, the compatibility function in a one-layer model will be forced to compromise between the two otherwise distinct compatibility functions of different processes. A better model is to explicitly capture different dynamics of processes, with more than one layer of random field grids. In the proposed model, one grid layer is used for a contour process and a separate grid is used for a texture process. The dependency between the two processes is modeled with coupling links between the layers. To reduce the complexity, each node in one layer is only coupled with the same node in the other layer. This leads to the proposed coupled Conditional Random Field (cCRF) model, shown in Figure 2(b). The importance of using coupled rather than single layer Conditional Random Fields to address different image processes will become more evident in the experimental results (Section 3.4).



(a) single-layer CRF (b) coupled Conditional Random Field

Figure 2. Single-layer CRF and coupled CRF models.

3.1. Parametrization of coupled Conditional Random Field

The proposed functional forms of the cCRF model in Figure 2(b) are shown in Table 1. Five image measurements are used in the current work. Contourness cm_i and textureness tm_i (defined in Section 3.3) are used for local evidence functions, which have a form of logistic regression. This discriminative form of local evidence was originally proposed in Discriminative Random Field [17]. Unlike the log-linear compatibility in [17], the proposed cCRF uses a form of logistic regression for compatibility functions too. The corresponding measurements used are: (1) $\delta\theta_{ij}$, angular difference between the orientation of i and the line joining i with a neighboring pixel j; (2) $\delta cm_{ij} = |cm_i - cm_j|$, absolute contourness difference between i and j; (3) $\delta tm_{ij} = |tm_i - tm_j|$, absolute textureness difference between i and j. The compatibility functions of the two processes will have different parameters, capturing the distinct interaction dynamics of the two processes stated in Section 3.

In both processes, the compatibility between a negative labeling and a neighboring negative labeling, *e.g.*, for the labeling pair (c_i, c_j)=$(-1, -1)$, is fixed to 0.5. For each of the two layers, this negative-to-negative compatibility is represented in the positive-to-positive compatibility in the opposite layer and is already coupled into the current

- Variables for pixel i

  c_i: labeling variable in the contour layer:

  c_i = 1: contour pixel; $c_i = -1$: non-contour pixel.

  t_i: labeling variable in the texture layer:

  t_i = 1: texture pixel; $t_i = -1$: non-texture pixel.

  cm_i: contourness measurement. tm_i: textureness measurement.

  $\delta\theta_{ij}$: angle between orientation of i and the line joining i and a neighboring pixel j.

  δcm_{ij}: absolute difference between the contourness of i and j.

  δtm_{ij}: absolute difference between the textureness of i and j.

- Evidence function $\Phi_c(c_i|I)$

$$\Phi_c(c_i|I) = \frac{1}{1+e^{-c_i(\alpha_0+\alpha_1 cm_i+\alpha_2 tm_i)}}$$

- Compatibility function $\Psi_c(c_i, c_j|I)$

$$\Psi_c(c_i,c_j|I) = \begin{cases} 0.5, & \text{if } (c_i,c_j)=(-1,-1) \\ A, & \text{otherwise} \end{cases}$$

where $A = \dfrac{1}{1+e^{-c_i c_j(\tau_0+\tau_1\delta\theta_{ij}+\tau_2\delta cm_{ij}+\tau_3\delta tm_{ij})}}$

- Evidence function $\Phi_t(t_i|I)$

$$\Phi_t(t_i|I) = \frac{1}{1+e^{-t_i(\beta_0+\beta_1 cm_i+\beta_2 tm_i)}}$$

- Compatibility function $\Psi_t(t_i, t_j|I)$

$$\Psi_t(t_i,t_j|I) = \begin{cases} 0.5, & \text{if } (t_i,t_j)=(-1,-1) \\ A, & \text{otherwise} \end{cases}$$

where $A = \dfrac{1}{1+e^{-t_i t_j(\gamma_0+\gamma_1\delta\theta_{ij}+\gamma_2\delta cm_{ij}+\gamma_3\delta tm_{ij})}}$

- Compatibility function $\Psi_{ct}(c_i, t_i|I)$

$$\Psi_{ct}(c_i,t_i|I) = \begin{cases} 0 & \text{if } c_i = t_i \\ 1 & \text{if } c_i \neq t_i \end{cases}$$

i.e., contour and texture processes are mutually exclusive.

Table 1. Evidence and compatibility functions of the proposed coupled Conditional Random Field. See text for details.

layer through the coupling link. The compatibility matrix $\Psi_{ct}(c_i, t_i|I)$ for the coupling links is fixed to make the two processes mutually exclusive. Note, however, this could be extended to allow non-mutually-exclusive labeling.

For clarity, the graphical model in Figure 2(b) shows a cCRF with a 4-neighborhood system for each of the contour and texture processes. In practice, models defined on higher order neighborhood systems, capturing more information from neighboring pixels, are used.

3.2. Learning and inference of coupled Conditional Random Fields

Assuming only up to pairwise clique energies are nonzero, with the functional forms in Table 1, the posterior of the cCRF is given by the following factorized form:

$$P(C,T|I,\Theta) = \frac{1}{Z}\left\{\prod_i \Phi_c(c_i|I)\Phi_t(t_i|I)\Psi_{ct}(c_i,t_i|I)\right\} \cdot \left\{\prod_{(i,j)\in C_{edge}} \Psi_c(c_i,c_j|I)\right\} \cdot \left\{\prod_{(i,j)\in T_{edge}} \Psi_t(t_i,t_j|I)\right\} \quad (1)$$

where C_{edge} indicates the set of inter-node links in the contour layer and T_{edge} for the texture layer, Z is the partition function and Θ is the set of parameters of the cCRF. In principle, parameters can be learned with maximum-likelihood. Maximum-likelihood learning is complicated by the partition function Z. Exact maximum-likelihood in this case is intractable thus the model learning has to resort to approximation techiniques. For homogeneous random fields such as the proposed cCRF, maximum pseudolikelihood [4] can be used, which simplifies learning by approximating the likelihood with a factorization of local conditional likelihood:

$$\Theta^*_{ML} \simeq \text{argmax}_\Theta \log \prod_{m=1}^M \prod_i P(c_i^m, t_i^m | C_{N_i}^m, T_{N_i}^m, I^m, \Theta) \quad (2)$$

where $C_{N_i}^m$ is the contour labeling of edgels in the neighborhood of i for the mth training sample; similarly for $T_{N_i}^m$.

Each local conditional likelihood has the form

$$P(c_i,t_i|C_{N_i},T_{N_i},I,\Theta) = \frac{P(c_i,t_i,C_{N_i},T_{N_i}|I,\Theta)}{Z_i} \quad (3)$$

$$Z_i = \textstyle\sum_{c_i\in\{+1,-1\},t_i\in\{+1,-1\}} P(c_i,t_i,C_{N_i},T_{N_i}|I,\Theta) \quad (4)$$

Now each of the partition functions Z_i only sums over 4 combinations of labels, making the computation tractable.

The proposed cCRF is a complex image model, prone to over-fitting. To avoid this, in practice a tempered maximum pseudolikelihood is used for learning the parameters. Tempered maximum likelihood is also used in tempered EM for learning the pLSA model to improve generalization capability [15]. At each step, instead of maximizing the original pseudolikelihood (2), the tempered maximum pseudolikelihood maximizes a modified pseudolikelihood:

$$\Theta^*_{ML_\beta} \simeq \text{argmax}_\Theta \log \prod_{m=1}^M \prod_i P^\beta(c_i^m, t_i^m | C_{N_i}^m, T_{N_i}^m, I^m, \Theta) \quad (5)$$

The tempered pseudolikelihood is equivalent to discounting the corresponding free energy by a multiplicative constant β. When β is small, or equivalently, when temperature is high, the parameter learning is encouraged to move around the feasible space more freely. This has the effect of discounting each of the conditional probabilities in Equation (5) to make each of them contribute more evenly to the joint distribution. The tempered maximum pseudolikelihood used for learning cCRF proceeds as in Table 2:

1. Initialize β with a small constant and perform maximum pseudolikelihood to estimate parameters.
2. Using previous steps as initialization, increase β with a small step and perform maximum pseudolikelihood.
3. Iterate step 2 until β reaches 1.

Table 2. Learning with tempered maximum pseudolikelihood.

To initialize the parameters for learning, the logistic functions in Table 1 are first trained separately with maximum likelihood estimation for logistic regression, assuming points are independent. The learned parameters are then used as a starting point for the non-linear optimization on the joint pseudolikelihood. Gradient Ascent is used for each step of the tempered maximum pseudolikelihood learning. For labeling in test images, Maximum a Posteriori (MAP) inference is carried out using loopy Belief Propagation.

3.3. Measure of contourness and textureness

Quadrature filters are used to detect edges. To measure contourness and textureness of edge pixels, we adopt an approach similar to the method by Martin *et al.* [23].

3.3.1 Edge extraction and contourness

The quadrature filter bank used here are the even and odd pairs as in [22, 23]. The base symmetric filter is the second derivative of an elongated Gaussian, and the base odd-symmetric filter is its Hilbert transform. The entire filter bank consists of 8 rotated versions of the base even/odd pair. Orientation energy is computed as the sum of squares of even/odd filter responses. For color images, orientation energy is computed by summing responses of the Lab color channels. A Canny's hysteresis thresholding [6] is applied to the orientation energy image to extract edge points. Both the lower and higher thresholds are set to be relatively small in order to minimize misses at true edges with low contrast. Rectification will be postponed until inference on cCRF. For extracted edges, contourness cm_i is measured by corresponding orientation energy.

3.3.2 Textureness

We use texture gradients [22] to measure textureness. The filter banks are the 8 rotated quadrature pairs used in Section 3.3.1, plus 3 Difference of Gaussians and 3 Gaussians at the scales of $\sigma = \{1.5, 2, 3\}$. First, all filter responses of all points in an image are clustered to form 50 textons. As in [22], for each edge point, a 20-pixel wide circular region around the point is extracted and divided into three parts: a 10-pixel wide center strip D_0 along the orientation of the edge point of interest, and D_+ and D_- which are the pixels to the left and right of D_0 respectively. χ^2-distances are computed between the histograms of textons in D_+ and $D_0 \bigcup D_-$ and between D_- and $D_0 \bigcup D_+$. The larger of the two distances is kept as a measure of textureness of the edge point. The larger the distance, the smaller the textureness. Further details can be seen at [21]. In the current work, textures are only modeled on edge processes. Adding texture information from homogeneous regions could also be beneficial.

3.4. Model learning and evaluation

In the current implementation, the neighborhood is set to 5×5 for both the contour and texture CRF. A set of ground truth data are labeled to train the cCRF. To show the advantage of the proposed cCRF vs. the single-layer CRF, the single-layer CRF in Figure 2(a) is also trained for comparison purposes. The parametrization of the single-layer CRF is shown in Table 3. The evidence and compatibility functions have the same logistic regression forms as in the cCRF.

In the full models in Table 1 and 3, the compability functions depend on three image measurements - $\delta\theta_{ij}$, δcm_{ij} and δtm_{ij}. Compatibility functions in different models

- Edge process:

  e_i: labeling variable for an edge pixel i:

  e_i = 1: contour pixel; $e_i = -1$: texture pixel.
- Evidence function of $\Phi_e(e_i|I)$= $\frac{1}{1+e^{-e_i(\lambda_0+\lambda_1 cm_i+\lambda_2 tm_i)}}$
- Compatibility function of $\Psi_e(e_i, e_j|I)$= $\frac{1}{1+e^{-e_i e_j(\eta_0+\eta_1\delta\theta_{ij}+\eta_2\delta cm_{ij}+\eta_3\delta tm_{ij})}}$

Table 3. Parametrization of a single-layer CRF.

a. Model-$\delta\theta$: compatibility depends on angular difference: $\Psi_e(e_i, e_j|I) = \frac{1}{1+e^{-e_i e_j(\eta_0+\eta_1\delta\theta_{ij})}}$

b. Model-δcm: compatibility depends on contourness diff.: $\Psi_e(e_i, e_j|I) = \frac{1}{1+e^{-e_i e_j(\eta_0+\eta_2\delta cm_{ij})}}$

c. Model-δtm: compatibility depends on textureness diff.: $\Psi_e(e_i, e_j|I) = \frac{1}{1+e^{-e_i e_j(\eta_0+\eta_3\delta tm_{ij})}}$

d. Model-all: compatibility depends on all: $\Psi_e(e_i, e_j|I) = \frac{1}{1+e^{-e_i e_j(\eta_0+\eta_1\delta\theta_{ij}+\eta_2\delta cm_{ij}+\eta_3\delta tm_{ij})}}$

Table 4. Different compatibility functions of a single-layer CRF.

(cCRF and single-layer CRF) could have quite distinct dependencies on the three measurements, as discussed in Section 3. To better evaluate these different dependencies in different models, each model is also trained with the compatibility function dependent on only one of the three measurements. For instance, the single-layer CRF is trained with different implementations of compatibility function as shown in Table 4. The evidence function remains the same for all instances. Similarly for the cCRF model. For simplicity, the four cases in Table 4 are referred as Model-$\delta\theta$, Model-δcm, Model-δtm and Model-all respectively. The learned parameters are listed in Table 6. There are several noticeable facts in the learned parameter of different models:

1. For Model-$\delta\theta$ where compatibility functions only depend on the angular difference $\delta\theta_{ij}$:

   for the cCRF, the learned parameters of compatibility functions for contour process and texture process are different, with τ_0=5.4240 and τ_1=$-$5.3993 for contour and γ_0=6.7777 and γ_1=$-$3.7558 for texture, which captures the disparate dynamics of the two processes;

   whereas the single-layer CRF, with learned compatibility parameters of η_0=0.7571 and η_1=$-$0.5324, makes a forced compromise while using only one function to account for both dynamics in the two otherwise distinct processes.

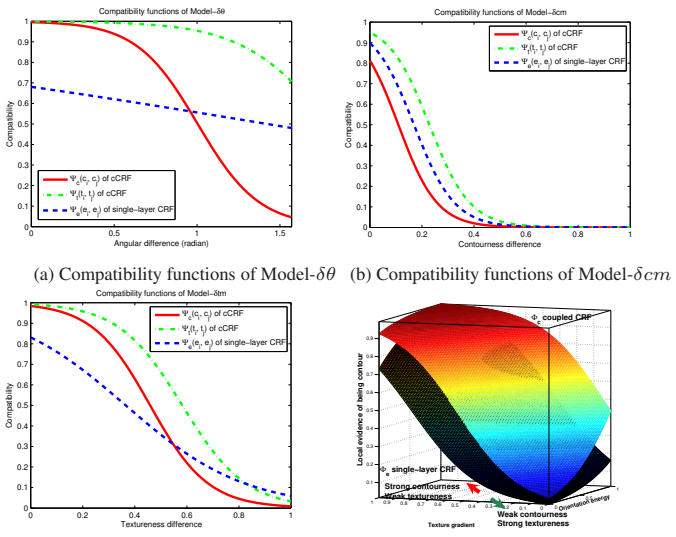
2. For Model-δcm and Model-δtm:

   for the cCRF, the learned parameters of contour and texture processes are slightly different; while for the single-layer CRF, the learned parameters are again compromises to those in the cCRF.

3. For Model-all where compatibility functions depend on all three measurements:

   for the cCRF, for contour compatibility, the learned parameter for dependency on angular difference $\delta\theta_{ij}$, which is τ_1=-2.3727, is quite different from the corresponding parameter for texture which is γ_1=-0.9708. Other learned parameters of the compatibility functions are comparable for both processes; while for the single-layer CRF, the learned parameters are an apparent compromise, with η_1=-1.3580, which lies between τ_1 and γ_1.

To visualize these differences, the learned compatibility functions of Model-$\delta\theta$, Model-δcm and Model-δtm are drawn in Figure 3. Figure 3(a) clearly shows that $\Psi_c(c_i, c_j)$ (red curve) and $\Psi_t(t_i, t_j)$ (green curve) are distinct. Roughly speaking, for small angular differences, *e.g.*, less than 0.5 radian (28.6 degree), $\Psi_c(c_i, c_j)$ gives high compatibility of more than 0.9; whereas for large angular difference, *e.g.*, larger than 1 radian (57.3 degree), the compatibility is smaller than 0.5. This means the contour compatibility function $\Psi_c(c_i, c_j)$ encourages local alignment of edge points, which is consistent with intuition. On the contrary, the compatibility function $\Psi_t(t_i, t_j)$ of texture process remains large for nearly all angular differences ($0\sim\frac{\pi}{2}$). As a comparison, the compatibility function $\Psi_e(e_i, e_j)$ (blue curve) of the single-layer CRF is forced to account for two different interaction dynamics, hence lies between the two different compatibility functions.



(a) Compatibility functions of Model-$\delta\theta$ (b) Compatibility functions of Model-δcm

(c) Compatibility functions of Model-δtm (d) Evidence functions of Model-all

Figure 3. Comparisons of different models (better view in color).

Figure 3(b)(c) reveals similar observations for dependencies on contourness difference δcm and textureness difference δtm. Again, the compatibility functions of the single-layer CRF are compromises of the two processes.

Figure 3(d), which plots the learned evidence function Φ_c of the cCRF and Φ_e of the single-layer CRF (whose parameters are shown in Table 6(d)), shows another important difference between the cCRF and the single-layer CRF. In Figure 3(d), the evidence function Φ_c of the cCRF lies above Φ_e of the single-layer CRF. This indicates that the single-layer CRF is stricter in assigning local evidence of contourness, *i.e.*, only edge points with strong enough contourness and weak enough textureness measurements are given large local evidence of being contour, whereas the cCRF relaxes this compared with the single-layer CRF, allowing a much wider range of measurements to be considered as possible contour. This is also intuitively correct, since the single-layer CRF has no other strong cues of detecting contour while the cCRF is able to rectify contour with local angular alignment in the compatibility function.

To quantitatively evaluate the proposed cCRF and compare it with the single-layer CRF, another set of labeled images are used as a test set. The performance of different models is evaluated with precision-recall rates and corresponding F-measure. The results of the evaluation are shown in Table 5. The cCRF is better than the single-layer CRF for contour and texture processes respectively, as measured by F-measure. For contour and texture jointly, the cCRF also performs much better than the single-layer CRF.

	$\text{Recall}_{contour}$	$\text{Precision}_{contour}$	$\text{F}_{contour}$
coupled CRF	**83.93%**	80.97%	**0.8243**
single-layer CRF	58.33%	**91.69%**	0.7130

(a) Performance of models on contour process

	$\text{Recall}_{texture}$	$\text{Precision}_{texture}$	$\text{F}_{contour}$
coupled CRF	89.65%	**90.07%**	**0.8986**
single-layer CRF	**97.67%**	80.42%	0.8821

(b) Performance of models on texture process

	Recall_{decomp}=$\text{Precision}_{decomp}$=$\text{F}_{decomp}$
coupled CRF	**87.53%**
single-layer CRF	83.20%

(c) Performance of models on contour texture decomposition.

Table 5. Performance evaluation of different models.

Figure 4 shows a visual comparison of contour-texture decomposition by the cCRF and the single-layer CRF on some of the test images used for the evaluation. The cCRF clearly does a much better job in separating contour and texture processes, with fewer mis-detections of contour edgels while keeping most texture edgels in the texture process. On the contrary, the single-layer CRF, due to the lack of modeling power for the two distinct processes, compares inferiorly in decomposition. Other decomposition results by the cCRF are shown in Figure 5. Note the contours of these objects are well-extracted as shown in the second and fifth columns, and their textures, such as fur, spines, grass and keyboard patterns, are clearly separated as shown in the third and sixth columns.

4. Object recognition from decomposition

With the coupled CRF model, visual information such as contour and texture in images are decomposed into separate

Model-$\delta\theta$							
coupled CRF				single-layer CRF			
Φ_c	α_0 -7.4036	α_1 2.7973	α_2 7.1600	Φ_e	λ_0 -4.8556	λ_1 2.9285	λ_2 6.1034
Φ_t	β_0 7.4036	β_1 -2.7973	β_2 -7.1600				
Ψ_c	τ_0 5.4240	τ_1 -5.3993		Ψ_e	η_0 0.7571	η_1 -0.5324	
Ψ_t	γ_0 6.7777	γ_1 -3.7558					

(a) Learned parameters of Model-$\delta\theta$ for cCRF and single-layer CRF.

Model-δcm							
coupled CRF				single-layer CRF			
Φ_c	α_0 -4.4558	α_1 1.9039	α_2 7.5505	Φ_e	λ_0 -5.4363	λ_1 1.5867	λ_2 7.5817
Φ_t	β_0 4.4558	β_1 -1.9039	β_2 -7.5505				
Ψ_c	τ_0 1.4656	τ_2 -13.3836		Ψ_e	η_0 2.1681	η_2 -12.7884	
Ψ_t	γ_0 2.9286	γ_2 -12.8005					

(b) Learned parameters of Model-δcm for cCRF and single-layer CRF.

Model-δtm							
coupled CRF				single-layer CRF			
Φ_c	α_0 -6.3037	α_1 3.6045	α_2 5.5762	Φ_e	λ_0 -4.9371	λ_1 4.1724	λ_2 4.1971
Φ_t	β_0 6.3037	β_1 -3.6045	β_2 -5.5762				
Ψ_c	τ_0 4.1161	τ_3 -8.9647		Ψ_e	η_0 1.5933	η_3 -4.3612	
Ψ_t	γ_0 4.7566	γ_3 -8.1626					

(c) Learned parameters of Model-δtm for cCRF and single-layer CRF.

Model-all								
coupled CRF					single-layer CRF			
Φ_c	α_0 -2.9588	α_1 2.7014	α_2 5.5123		Φ_e	λ_0 -4.1046	λ_1 2.4582	λ_2 5.1092
Φ_t	β_0 2.9588	β_1 -2.7014	β_2 -5.5123					
Ψ_c	τ_0 2.7942	τ_1 -2.3865	τ_2 -8.8175	τ_3 -3.1492	Ψ_e	η_0 2.7313	η_1 -1.3540	η_2 -8.8621	η_3 -1.9071
Ψ_t	γ_0 3.7624	γ_1 -0.9594	γ_2 -8.6021	γ_3 -2.1545				

(d) Learned parameters of Model-all for cCRF and single-layer CRF.

Table 6. Learned parameters for different models of cCRF and single-layer CRF.

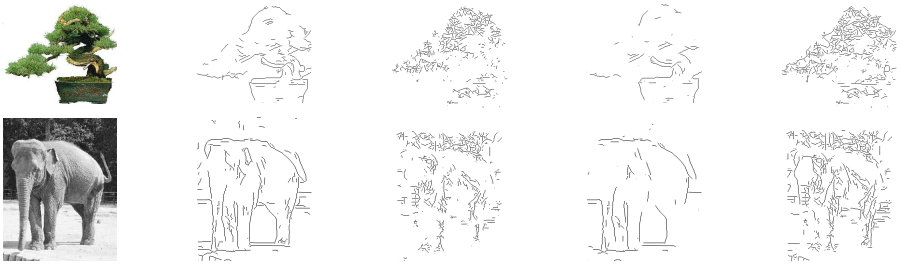


Figure 4. Comparisons of contour-texture decomposition by cCRF and single-layer CRF. The first column are images of test objects. The second and fourth columns are extracted contours by cCRF and single-layer CRF respectively. The third and fifth columns are texture.

channels. Each of the decomposed visual channels captures a distinct perceptual aspect of objects. In the following sections, we employ some existing methods for matching the decomposed contour and texture channels, and adaptively combine them at the kernel level for object categorization.

4.1. Matching kernels

Spatial pyramid matching [18] is used on both contour and texture channels. To extract features, dense sampling on a grid with spacing of 8 pixels is used. For each point on the grid, we extract two patches of 50×50 and 25×25 pixels and compute SIFT [20] descriptors on edge points. Next, visual dictionaries of 400 words for the contour channel and 200 words for texture are learned through clustering on a random subset of images. Then spatial pyramid matching is applied to compute pair-wise similarity.

Color information is also used for matching on both contour and texture channels. Features are again defined on a dense sampling grid. For each sample point, a small patch is extracted and average Hue-Saturation-Value is computed. Hue and Saturation are discretized into 10 bins, and Value into 2 bins, giving 200 visual words. Then spatial pyramid is applied for matching.

In the contour channel, shape correspondence is also a salient visual cue, which is not explicitly addressed by spatial pyramid matching. In this work, we leverage shape matching with robust chamfer matching in the contour channel, which works reasonably well for rigid or slightly deformable objects. Usually chamfer matching performs poorly in cluttered images. This shortcoming is mitigated, to a large extent, by the fact that contour is decomposed by the cCRF into a cleaner channel. Similar to [27], we define a robust oriented chamfer distance as follows:

$$d(X,Y) = \frac{1}{N_x} \sum_{x_i \in X} \max(\min_{y_j \in Y} ||x_i - y_j||, \tau) + \lambda \frac{1}{N_x} \sum_{x_i \in X} (1 - \mathrm{e}^{-\delta\theta^2_{x_i y_j}/(2\sigma^2_\theta)}) \quad (6)$$

where x_i and y_j are positions of edge pixels in image X and Y respectively, N_x is the number of edge pixels in X, and $\delta\theta_{x_i y_j}$ is the difference between the orientation of x_i and its closest match y_j in image Y. The first term in Equation 6 is the truncated chamfer distance, and the second term is a Gaussian penalty for orientation differences. To account for misalignment, X is slightly translated and rotated and the best match to Y is kept. To make a symmetrical distance, $d(Y,X)$ is also computed and the average of $d(X,Y)$ and $d(Y,X)$ is taken as similarity between X and Y.

To compute kernel entries from pair-wise chamfer distances, another Gaussian form is used:

$$K(X,Y) = \mathrm{e}^{-\frac{[(d(X,Y)+d(Y,X))/2]^2}{2\sigma_k^2}} \quad (7)$$

where $K(X,Y)$ is chamfer matching kernel entry. In practice, we find that adding robust chamfer matching is complementary to bag-of-features and able to address the distinct characteristics of some classes, and helps to achieve better recognition performance.

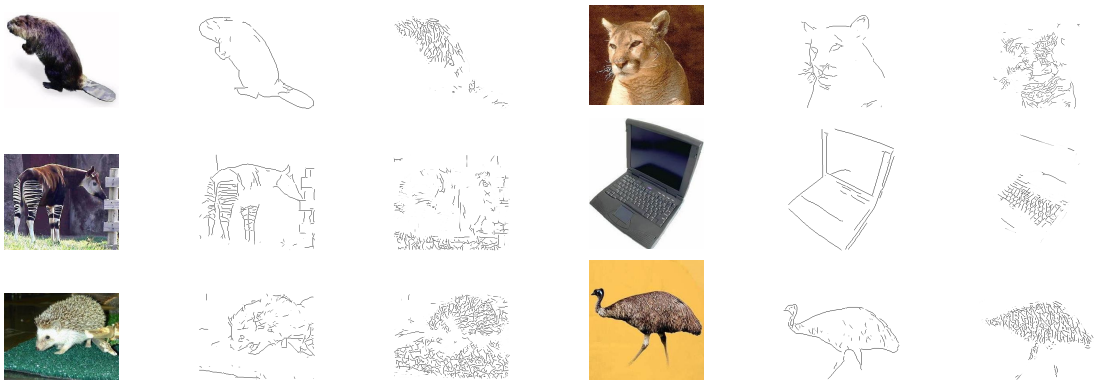


Figure 5. Example of contour/texture decomposition. The first and fourth columns are images of objects. The second and fifth columns are extracted contours of the images to their left. The third and sixth columns are extracted textures of the corresponding images.

4.2. Adaptive kernel combination

As discussed earlier, it is more sensible to assume that different visual cues play different roles in discriminating different classes. To this end, we use adaptive kernel combination based on kernel alignment [9, 16]. This technique finds a weighted linear combination of constituent kernels which optimizes the alignment with an "ideal" kernel. The learned combination weights for each class pair effectively capture the relative importance of different visual cues. With the adaptively combined kernel, we use a one-vs-one multi-class SVM for classification. Further details are available at [21].

4.3. Classification results

The proposed method is evaluated on the widely used dataset CIT-101 [11]. The evaluation runs with different numbers of training samples and tests on up to 50 images per class. The algorithm is run 10 times with different randomly selected training and test samples, and the average of per-class recognition rates is reported.

Our first experiment is run on a subset of 28 classes from CIT-101, which is used by Cao and Fei-Fei [7]. These 28 classes present a good balance of texture-rich, contour-rich and color-rich classes, hence this subset is a good test bed for the proposed adaptive method. With the kernels from the decomposed channels, a simple average combination of kernels gives an average recognition rate of **81.69**% for 30 training samples per class. With adaptive combination, the performance is boosted to **85.02**%. These results suggest that decomposing different visual cues and adaptively combining them to fully leverage their potential is a promising scheme for object categorization.

On the entire 101 classes of CIT-101, the recognition rates of the proposed method are shown in Table 7. Figure 6 shows the comparisons to some of the state-of-the-art methods on CIT-101. Compared with one of those top methods [14], the proposed method in this paper achieves recognition improvement of about {7.24%, 5.75%, 4.31%, 2.24%,} for {5,10,15,30} training samples per class respectively. Comparison to another top method [13], where the 'Background' class is excluded and the 'Faces_easy' class is included, shows 5.42% and 6% recognition rate improvements for 5 and 10 training samples per class respectively. These comparisons demonstrate the effectiveness of the proposed visual decomposition and recombination scheme for object recognition. The performance improvements are more significant when only a few training samples, *e.g.*, 5, 10 or 15, are available for each class. This suggests that when there are not enough training samples, it is more important to decompose various visual cues, leverage each of them to their full potential, and recombine them for a better understanding of image contents.

On the CIT-4 classes (Face, Leopard, Motorbike, Airplane), the proposed method achieves a recognition rate of **99.8**% with only 30 training samples per class, with no errors for the classes of Face, Leopard, and Motorbike, and occasionally 1 to 2 misclassified test samples for Airplane. Many current methods use many more training samples to achieve comparable performance.

Figure 7 draws the learned weights when adaptively combining multiple visual cues for classifying different class pairs of CIT-101. It is clear that different visual cues have different weights when discriminating different class pairs. For most classes, contour is the most useful cue for recognition. Texture and color are not as important, with exceptions to some classes.

5. Conclusion and discussion

We propose a coupled CRF and demonstrate its advantages to model and decompose contour and texture in images. Adaptive combination is applied at the kernel level to weigh different visual cues for different classes. The proposed method performs well on challenging data sets, especially when only a small number training samples are available. We expect the "recognition-through-decomposition-and-fusion" scheme to be a promising direction for building effective and efficient object categorization systems.

At the time of this work, some researchers are also investigating methods of combining multiple matching schemes, with improved features and enhanced adaptive combination methods, which are shown to achieve significant performance improvements [5, 28]. It is expected that incorporating these enhanced elements into the proposed model will be able to achieve further improvements.

Training sample	5	10	15	30
With Background class	48.74(±0.86)	58.75(±0.67)	63.31(±0.65)	69.84(±0.98)
No Background class	49.32(±0.89)	59.30(±0.65)	63.82(±0.65)	70.38(±0.96)

Table 7. Recognition rates on CIT-101 of the proposed method. Number are in percentile, with standard deviation in parenthesis.

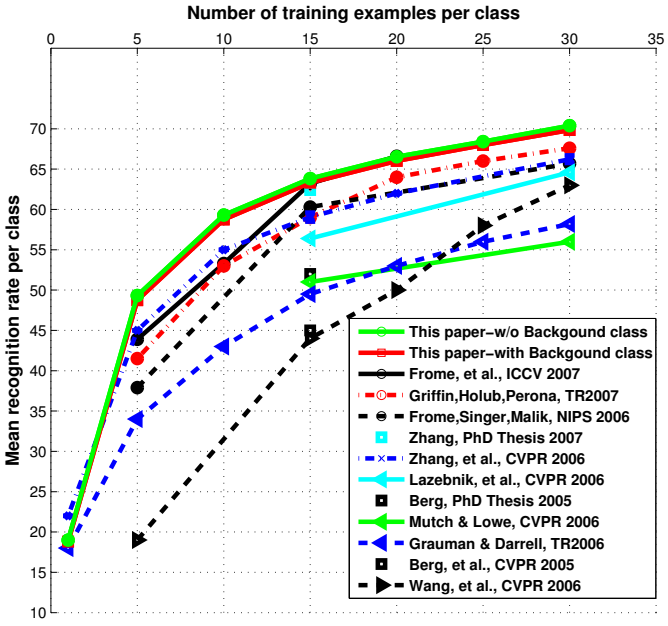
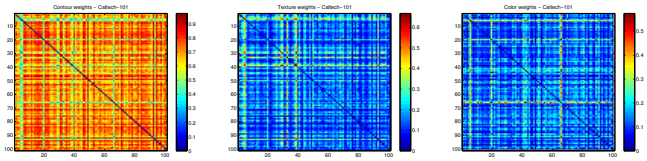


Figure 6. Comparison of performance on CIT-101.



(a) contour weights (b) texture weights (c) color weights

Figure 7. Learned adaptive weights for different visual cues when combining kernels for classifying CIT-101. (better view in color)

References

[1] R. Abadi, J. J. Kulikowski, and P. Meudell. Visual performance in a case of visual agnosia. In *Functional Recovery from Brain Damage*. 1981. 1

[2] L. Battelli and G. Sartori. Dissociation between contour-based and texture-based shape perception: A single case study. *Visual Cognition*, 4(3):275–310, 1997. 1

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI*, 24(24):590–522, April 2002. 1

[4] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24:179–195, 1975. 3

[5] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *ICCV*, 2007. 8

[6] J. Canny. A computational approach to edge detection. *IEEE Trans. PAMI*, 8(6):679–698, 1986. 4

[7] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. In *ICCV*, 2007. 7

[8] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *PAMI*, 23(6):853–857, 2001. 1

[9] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In *NIPS*, 2001. 7

[10] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka. Visual categorization with bags of keypoints. *ECCV Workshop on Statistical Learning in Computer Vision*, 2004. 1

[11] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples. In *IEEE. CVPR 2004, Workshop on GMBV*, 2004. 7

[12] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003. 1

[13] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *ICCV*, 2007. 7

[14] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, CIT, 2007. 7

[15] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(2), 2001. 3

[16] S. Hoi, M. Lyu, and E. Chang. Learning the unified kernel machines for classification. In *KDD*, 2006. 7

[17] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *ICCV*, 2003. 2

[18] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 6

[19] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 1987. 1

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91 – 110, November 2004. 6

[21] X. Ma. *http://people.csail.mit.edu/xiaoxuma*. PhD thesis, MIT, 2008. 4, 7

[22] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 2001. 1, 4

[23] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. PAMI*, 26(5):530–549, 2004. 1, 4

[24] M. J. Riddoch and G. W. Humphreys. A case study of integrative visual agnosia. *Brain*, 110(6):1431–1462, 1987. 1

[25] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. PAMI*, 19(5):530–535, 1997. 1

[26] A. Sha'ashua and S. Ullman. Structural saliency: The detection of globally salient structures using a locally connected network. In *ICCV*, 1988. 1

[27] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, 2005. 6

[28] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, 2007. 8