SuperParsing: Scalable Nonparametric Image Parsing with Superpixels

Joseph Tighe and Svetlana Lazebnik

Dept. of Computer Science, University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175
{jtighe, lazebnik}@cs.unc.edu

Abstract. This paper presents a simple and effective nonparametric approach to the problem of image parsing, or labeling image regions (in our case, superpixels produced by bottom-up segmentation) with their categories. This approach requires no training, and it can easily scale to datasets with tens of thousands of images and hundreds of labels. It works by scene-level matching with global image descriptors, followed by superpixel-level matching with local features and efficient Markov random field (MRF) optimization for incorporating neighborhood context. Our MRF setup can also compute a simultaneous labeling of image regions into semantic classes (e.g., tree, building, car) and geometric classes (sky, vertical, ground). Our system outperforms the state-of-the-art nonparametric method based on SIFT Flow on a dataset of 2,688 images and 33 labels. In addition, we report per-pixel rates on a larger dataset of 15,150 images and 170 labels. To our knowledge, this is the first complete evaluation of image parsing on a dataset of this size, and it establishes a new benchmark for the problem.

Key words: scene understanding, image parsing, image segmentation

1 Introduction

This paper addresses the problem of image parsing, or segmenting all the objects in an image and identifying their categories. The literature contains diverse proposed image parsing methods, including ones that estimate labels pixel by pixel [1, 2], ones that aggregate features over segmentation regions [3–6], and ones that predict object bounding boxes [7–10]. Most of these methods operate with a few pre-defined classes and require a generative or discriminative model to be trained in advance for each class (and sometimes even for each training exemplar [5]). Training can take days and must be repeated from scratch if new training examples or new classes are added to the dataset. In most cases (with the notable exception of [2]), processing a test image is also quite slow, as it involves operations like running multiple object detectors over the image, performing graphical model inference, or searching over multiple segmentations.

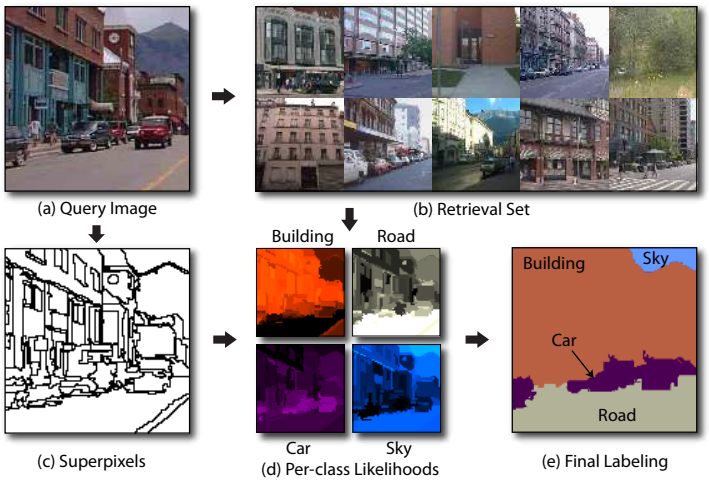


Fig. 1. System overview. Given a query image (a) we retrieve similar images from our dataset (b) using several global features. Next, we divide the query into superpixels (c) and compute a per-superpixel likelihood ratio score for each class (d) based on nearest-neighbor superpixel matches from the retrieval set. These scores, in combination with a contextual MRF model, give a dense labeling of the query image (e).

While most existing methods thus remain trapped in a "closed universe" recognition paradigm, a much more exciting paradigm of "open universe" datasets is promising to become dominant in the very near future. For example, the LabelMe dataset [11] is composed of complex, real-world scene images that have been segmented and labeled (sometimes incompletely or noisily) by multiple users. There is no pre-defined set of class labels; the dataset is constantly expanding as people upload new photos or add annotations to current ones. In order to cope with such datasets, vision algorithms must have much faster training and testing times, and they must make it easy to continuously update the visual models with new classes or new images.

Recently, several researchers have begun advocating nonparametric, data-driven approaches to breaking out of the "closed universe" [12–15]. Such approaches do not do any training at all. Instead, for each new test image, they try to retrieve the most similar training images and transfer the desired information from the training images to the query. Liu et al. [15] have proposed a nonparametric image parsing method based on estimating "SIFT Flow," or a dense deformation field between images. This method requires no learning and in principle, it can work with an arbitrary set of labels. However, inference via SIFT Flow is currently very complex and computationally expensive. While we agree with [15] that the nonparametric philosophy currently holds the most promise for image parsing in large-scale, dynamic datasets, there is a lot of room for improvement over their method in terms of efficiency.

We set out to implement a nonparametric solution to image parsing that is as straightforward and efficient as possible, and that relies only on operations that can easily scale to ever larger image collections and sets of labels (see Figure 1 for a system overview). Similarly to [15], our proposed method requires no training (just some basic computation of dataset statistics), and makes use of a *retrieval set* of scenes whose content is used to interpret the test image. However, unlike the approach of [15], which works best if the retrieval set images are very similar to the test image in terms of spatial layout of the classes, we transfer labels at the level of *superpixels*, or coherent image regions produced by a bottom-up segmentation method. The label transfer is accomplished with a fast and simple nearest-neighbor search algorithm, and it allows for more variation between the layout of the test image and the images in the retrieval set. Moreover, using segmentation regions as a unit of label transfer gives better spatial support for aggregating features that could belong to the same object [16].

The current consensus among recognition researchers is that image parsing requires *context* (see, e.g., [3, 4, 9, 10]). However, learning and inference with most existing contextual models is slow and non-exact. Therefore, due to our goal of developing a scalable system, we restrict ourselves to efficient forms of context that do not need training and that can be cast in an MRF framework amenable to optimization by fast graph cut algorithms [17, 18]. We show that our system equipped with this form of context can achieve results comparable to state-of-the-art systems based on more complex contextual models [3, 6]. We also investigate geometric/semantic context in the manner of Gould et al. [6]. Namely, for each superpixel in the image, we simultaneously estimate a semantic label (e.g., building, car, person, etc.) and a geometric label (sky, ground, or vertical surface) while enforcing coherence between the two class types.

Our system exceeds the results reported in [15] on a dataset of 2,688 images and 33 labels. Moreover, to demonstrate the scalability of our method, we present per-pixel and per-class rates on a subset of LabelMe with 15,150 images and 170 labels. To our knowledge, we are the first to report complete recognition results on a dataset of this size. Thus, one of the contributions of this work is to establish a new benchmark for large-scale image parsing. Our code, data, and output can be found at `http://www.cs.unc.edu/SuperParsing`.

2 System Description

2.1 Retrieval Set

Similarly to several other data-driven methods [7, 12, 14, 15], our first step in parsing a query test image is to find a relatively small *retrieval set* of training images that will serve as the source of candidate superpixel-level annotations. This is done not only for computational efficiency, but also to provide scene-level context for the subsequent superpixel matching step. A good retrieval set should contain images of a similar scene type to that of the test image, along with similar objects and spatial layouts. To attempt to indirectly capture this kind of similarity, we use four types of global image features (Table 1(a)): spatial

Table 1. A complete list of features used in our system

(a) Global features for retrieval set computation (Section 2.1)		
Type	Name	Dimension
Global	Spatial pyramid (3 levels, SIFT dictionary of size 200)	4200
	Gist (3-channel RGB, 3 scales with 8, 8, & 4 orientations)	960
	Tiny image (3-channel RGB, 16×16 pixels)	768
	Color histogram (3-channel RGB, 8 bins per channel)	24
(b) Superpixel features (Section 2.2)		
Shape	Mask of superpixel shape over its bounding box (8×8)	64
	Bounding box width/height relative to image width/height	2
	Superpixel area relative to the area of the image	1
Location	Mask of superpixel shape over the image	64
	Top height of bounding box relative to image height	1
Texture/SIFT	Texton histogram, dilated texton histogram	100×2
	SIFT histogram, dilated SIFT histogram	100×2
	Left/right/top/bottom boundary SIFT histogram	100×4
Color	RGB color mean and std. dev.	3×2
	Color histogram (RGB, 11 bins per channel), dilated hist.	33×2
Appearance	Color thumbnail (8×8)	192
	Masked color thumbnail	192
	Grayscale gist over superpixel bounding box	320

pyramid [19], gist [20], tiny image [13], and color histogram. For each feature type, we rank all training images in increasing order of Euclidean distance from the query. Then we take the minimum of the per-feature ranks to get a single ranking for each training image, and take the top 200 images as the retrieval set. Taking the minimum of per-feature ranks amounts to taking the top fifty matches according to each global image descriptor, and it gives us better results than, say, averaging the ranks. Intuitively, taking the best scene matches from each of the global descriptors leads to better superpixel-based matches for region-based features that capture similar types of cues as the global features.

2.2 Superpixel Features

We wish to label the query image based on the content of the retrieval set, but assigning labels on a per-pixel basis as in [1, 14, 15] would be too inefficient. Instead, like [3–5], we choose to assign labels to superpixels, or regions produced by bottom-up segmentation. This not only reduces the complexity of the problem, but also gives better spatial support for aggregating features that could belong to a single object than, say, fixed-size square patches centered on every pixel in the image. We obtain superpixels using the fast graph-based segmentation algorithm of [21] and describe their appearance using 20 different features similar to those of [5], with some modifications and additions. A complete list of the features is given in Table 1(b). In particular, we compute histograms of textonsand dense

SIFT descriptors over the superpixel region, as well as that region dilated by 10 pixels. For SIFT features, which are more powerful than textons, we have also found it useful to compute left, right, top, and and bottom boundary histograms. To do this, we find the boundary region as the difference between the superpixel dilated and eroded by 5 pixels, and then obtain the left/right/top/bottom parts of the boundary by cutting it with an "X" drawn over the superpixel bounding box. All of the features are computed for each superpixel in the training set and stored together with their class labels. We associate a class label with a training superpixel if 50% or more of the superpixel overlaps with the segment mask for that label.

2.3 Local Superpixel Labeling

Having segmented the test image and extracted all its features, we next obtain a likelihood ratio score for each test superpixel and each class that is present in the retrieval set. Making the Naive Bayes assumption that features are independent of each other given the class, the likelihood ratio for class c and superpixel s_i is

$$L(s_i, c) = \frac{P(s_i|c)}{P(s_i|\bar{c})} = \prod_k \frac{P(f_i^k|c)}{P(f_i^k|\bar{c})}\,, \tag{1}$$

where $\bar{c}$ is the set of all classes excluding c, and f_i^k is the feature vector of the kth type for s_i. Each likelihood ratio $P(f_i^k|c)/P(f_i^k|\bar{c})$ is computed with the help of nonparametric density estimates of features from the required class(es) in the neighborhood of f_i^k. Specifically, let $\mathcal{D}$ denote the set of all superpixels in the training set, and $\mathcal{N}_i^k$ denote the set of all superpixels in the retrieval set whose kth feature distance from f_i^k is below a fixed threshold t_k. Then we have

$$\frac{P(f_i^k \mid c)}{P(f_i^k \mid \bar{c})} = \frac{n(c, \mathcal{N}_i^k)/n(c, \mathcal{D})}{n(\bar{c}, \mathcal{N}_i^k)/n(\bar{c}, \mathcal{D})} = \frac{n(c, \mathcal{N}_i^k)}{n(\bar{c}, \mathcal{N}_i^k)} \times \frac{n(\bar{c}, \mathcal{D})}{n(c, \mathcal{D})}\,, \tag{2}$$

where $n(c, \mathcal{S})$ (resp. $n(\bar{c}, \mathcal{S})$) is the number of superpixels in set $\mathcal{S}$ with class label c (resp. not c). To prevent zero likelihoods and smooth the counts, we add one to $n(c, \mathcal{N}_i^k)$ and $n(\bar{c}, \mathcal{N}_i^k)$. In our implementation, we use the ℓ_2 distance for all features, and set each threshold t_k to the median distance to the 20th nearest neighbor for the kth feature type over the dataset. The superpixel neighbors $\mathcal{N}_i^k$ are currently found by linear search through the retrieval set.

At this point, we can obtain a labeling of the image by simply assigning to each superpixel the class that maximizes eq. (1). As shown in Table 2, the resulting classification rates already come within 1.5% of those of [15]. We are not aware of any comparably simple scoring scheme reporting such encouraging results for image parsing problems with many unequally distributed labels.

2.4 Contextual Inference

Next, we would like to enforce contextual constraints on the image labeling – for example, a labeling that assigns "water" to a superpixel completely surrounded

by "sky" is not very plausible. Many state-of-the-art approaches encode such constraints with the help of conditional random field (CRF) models [1, 6, 4]. However, CRFs tend to be very costly both in terms of learning and inference. In keeping with our nonparametric philosophy and emphasis on scalability, we restrict ourselves to contextual models that require minimal training and that can be solved efficiently. Therefore, we formulate the global image labeling problem as minimization of a standard MRF energy function defined over the field of superpixel labels $\mathbf{c} = \{c_i\}$:

$$J(\mathbf{c}) = \sum_{s_i \in SP} E_{\text{data}}(s_i, c_i) + \lambda \sum_{(s_i, s_j) \in A} E_{\text{smooth}}(c_i, c_j)\,, \tag{3}$$

where SP is the set of superpixels, A is the set of pairs of adjacent superpixels and λ is the smoothing constant. We define the data term as $E_{\text{data}} = -w_i \log L(s_i, c_i)$, where $L(s_i, c_i)$ is the likelihood ratio score from eq. (1) and w_i is the superpixel weight (the size of s_i in pixels divided by the mean superpixel size). The smoothing term E_{smooth} is defined based on probabilities of label co-occurrence:

$$E_{\text{smooth}}(c_i, c_j) = -\log[(P(c_i|c_j) + P(c_j|c_i))/2] \times \delta[c_i \neq c_j]\,, \tag{4}$$

where $P(c|c')$ is the conditional probability of one superpixel having label c given that its neighbor has label c', estimated by counts from the training set. We use the two conditionals $P(c|c')$ and $P(c'|c)$ instead of the joint $P(c, c')$ because they have better numerical scaling, and average them to obtain a symmetric quantity. Qualitatively, we have found eq. (4) to produce very reasonable edge penalties. As can be seen from the examples in Figure 4 (d) and (f), it successfully flags improbable boundaries between "sea" and "sun," and "mountain" and "building." Quantitatively, results with eq. (4) tend to be about 1% more accurate than with the constant Potts penalty $\delta[c_i \neq c_j]$. We perform MRF inference using the efficient graph cut optimization code of [17, 18, 22]. On our large datasets, the resulting global labelings improve the accuracy by 3-5% (Table 2).

2.5 Simultaneous Classification of Semantic and Geometric Classes

Following Gould et al. [6], we consider the task of simultaneously labeling regions into two types of classes: semantic and geometric. Like [6], we use three geometric labels – sky, ground, and vertical – although the sets of semantic labels in our datasets are much larger. In this paper, we make the reasonable assumption that each semantic class is associated with a unique geometric class (e.g., "building" is "vertical," "river" is "horizontal," and so on) and specify this mapping by hand. We jointly solve for the fields of semantic labels ($\mathbf{c}$) and geometric labels ($\mathbf{g}$) by minimizing a cost function that is a simple extension of eq. (4):

$$H(\mathbf{c}, \mathbf{g}) = J(\mathbf{c}) + J(\mathbf{g}) + \mu \sum_{s_i \in SP} \varphi(c_i, g_i), \tag{5}$$

where φ is the term that enforces coherence between the geometric and semantic labels. It is 0 when the semantic class c_i is of the geometric class type g_i and

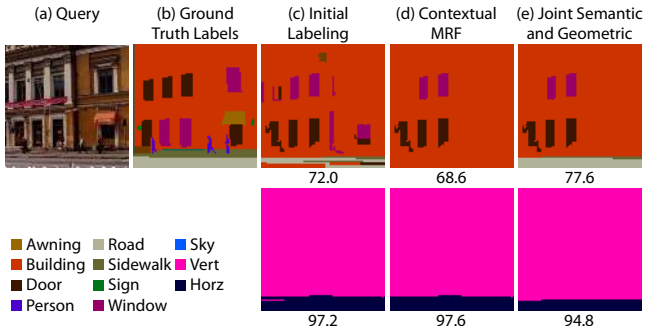


Fig. 2. In the contextual MRF classification, the road gets replaced by "building," while "horizontal" is correctly classified. By jointly solving for the two kinds of labels, we manage to recover some of the "road" and "sidewalk" in the semantic labeling. Note also that in this example, our method correctly classifies some of the windows that are mislabeled as doors in the ground truth, and incorrectly but plausibly classifies the windows on the lower level as doors.

1 otherwise. The constant μ controls how strictly the coherence is enforced (we use $\mu = 8$ in all experiments). Note that we can enforce the semantic/geometric consistency in a hard manner by effectively setting $\mu = \infty$, but we have found that allowing some tradeoff produces better results. Eq. (5) is in a form that can be optimized by the α/β-swap algrithm [17, 18, 22]. The inference takes almost the same amount of time as for the MRF setup of the previous section. Figure 2 shows an example where joint inference over semantic and geometric labels improves the accuracy of the semantic labeling. In many other cases, joint inference improves both labelings.

3 Results

3.1 Large Datasets

The first large-scale dataset in our experiments ("SIFT Flow dataset" in the following) is composed of the 2,688 images that have been throughly labeled by LabelMe users. Liu *et al.*[15] have split this dataset into 2,488 training images and 200 test images and used synonym correction to obtain 33 semantic labels. In our experiments, we use the same training/test split as [15]. Our second dataset ("Barcelona" in the following) is derived from the LabelMe subset used in [7]. It has 14,871 training and 279 test images.[1] The test set consists of street scenes from Barcelona, while the training set ranges in scene type but has no street scenes from Barcelona. We manually consolidated the synonyms in the

[1] Russell et al. [7] use a test set of 560 images, 281 of which are office scenes. However, the entire training set of 14,871 images only contains about 218 office scenes, so we have excluded the office images from the test set.

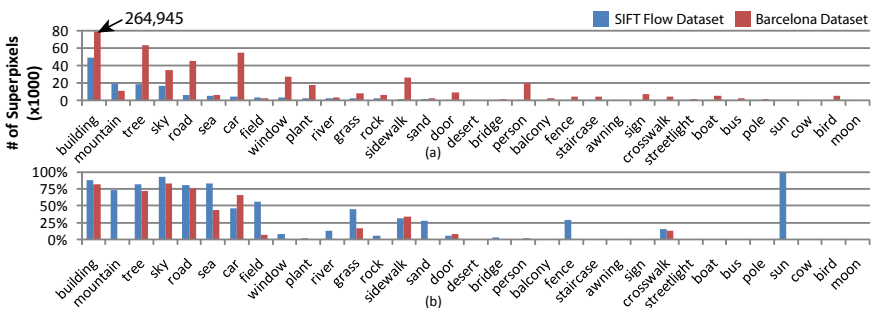


Fig. 3. (a) Label frequencies for the superpixels in the training set. The Barcelona dataset has 170 labels, but we only show the ones that are in common with the SIFT Flow dataset. (b) Per-class classification rates of our system.

label set to 170 unique labels. Note that [7] only gives detection curves for 12 categories on this dataset, so there are no previous baseline results for per-pixel performance. Both datasets have very nonuniform label distributions, as shown in Figure 3(a). Because of this, we report not only the per-pixel classification rate, which mainly reflects performance on the few largest classes, but also the average of per-pixel rates of all the classes.

Our system labels each superpixel of each test image by a *semantic class* (the original 33 and 170 labels, respectively) and a *geometric class* of sky, ground, or vertical (same as [6]). Because the number of geometric classes is small and fixed for all datasets, we have trained a discriminative model for them using a boosted decision tree classifier as in [3]. This classifier outputs a likelihood ratio score that we can directly plug into our MRF framework, and it gives us an improvement of about 1% in the accuracy for geometric classes over the nearest-neighbor scheme of Section 2.3. Apart from this, local and contextual MRF classification for geometric classes proceeds as described in Sections 2.3 and 2.4, and we also put the geometric and semantic likelihood ratios into a joint contextual classification framework as described in Section 2.5.

Table 2 reports per-pixel and average per-class rates for semantic classification of all three setups (local superpixel labeling, contextual MRF, joint MRF). As compared to the local baseline, the contextual MRF improves overall per-pixel rates on the SIFT Flow dataset by about 3% and on the Barcelona dataset by about 4%. Average per-class rates drop slightly due to the MRF "smoothing away" some of the smaller classes. Simultaneous geometric/semantic MRF improves the results for both types of classes on the SIFT Flow dataset, but makes little difference on the Barcelona dataset. Figure 3(b) shows that our per-class rates on both datasets are comparable, with large changes due primarily to differences in label frequency (e.g., there are no mountains in the Barcelona test set). It also shows that, similarly to most other image labeling approaches that do not train object detectors, we get much weaker performance on "things" (people, cars, signs) than on "stuff" (sky, road, trees).

Table 2. Performance of our system on the two large datasets. For semantic classes, we show the per-pixel rate followed by the average per-class rate in parentheses. Because there are only three geometric classes, we report only the per-pixel rate for them.

	SIFT Flow dataset [15]		Barcelona dataset [7]	
	Semantic	Geometric	Semantic	Geometric
Baseline	74.75 [15]	N/A	N/A	N/A
Local labeling (Sec. 2.3)	73.2 (29.1)	89.8	62.5 (8.0)	89.9
Superpixel MRF (Sec. 2.4)	76.3 (28.8)	89.9	66.6 (7.6)	90.2
Simultaneous MRF (Sec. 2.5)	76.9 (29.4)	90.8	66.9 (7.6)	90.7

Our final system on the SIFT Flow dataset achieves a classification rate of 76.9%. Thus, we outperform Liu *et al.*[15], who report a rate of 74.75% on the same test set with a more complex pixel-wise MRF (without the pixel-wise MRF, their rate is 66.24%). Liu *et al.*[15] also cite a rate of 82.72% for the top seven object categories; our corresponding rate is 84.5%. Sample output of our system on several SIFT Flow test images can be seen in Figure 4.

Next, we examine the effects of various components of our system. For each of these tests, we only show the local labeling rates on the SIFT Flow dataset. Table 3(a) shows the effect of different combinations of global features for computing the retrieval set (Section 2.1). Similarly to [12], we find that combining global features of unequal descriptive power gives better scene matches. Table 3(b) shows classification rates of the system with ten superpixel features added consecutively in decreasing order of their contribution to performance. Notice that SIFT histograms constitute four of the top ten features selected. The dilated SIFT histogram, which already incorporates some context from the superpixel neighborhood, is our single strongest feature, and it effectively makes the non-dilated SIFT histogram redundant. Also notice that SIFT and texton histograms are complementary (despite SIFT being stronger), and that all six feature categories from Table 1(b) are represented in the top ten.

Table 4(a) examines the effect of retrieval set size on classification rate. Interestingly, matching test superpixels against the entire dataset (last row of the table) drastically reduces performance. Thus, we quantitatively confirm the intuition that the retrieval set is not just a way to limit the computational compexity of sub-image matching; it acts as a global image-level context by restricting the superpixel matches to come from a small subset of related scenes. Table 4(b) shows the effect of restricting the list of possible labels in a test image to different "shortlists." Effectively, the shortlist used by our system for each test image is composed of all the classes present in the retrieval set (first row). To demonstrate the effect of long-tail class frequencies, the second row of the table shows the performance we get by classifying every superpixel in every test image to the ten most common classes in dataset. This does not change the overall per-pixel rate, but lowers the average per-class rate dramatically, thus underscoring the importance of looking at both numbers. The third row of Table 4(b) shows the results produced by restricting our shortlist to the ground truth labels in the

Table 3. Feature evaluation on the SIFT Flow dataset. (a) Results for local superpixel labeling with different retrieval set feature combinations. (b) Performance with the best retrieval set from (a) and top ten superpixel features added in succession.

(a)

Global Descriptor	Rate
Gist (G)	70.8 (28.7)
Spatial Pyramid (SP)	70.0 (22.4)
Color Hist. (CH)	65.9 (22.1)
Tiny Image (TI)	65.4 (25.5)
G + SP	72.4 (27.6)
G + SP + CH	73.3 (28.8)
G + SP + CH + TI	73.3 (29.1)

(b)

Superpixel Feature	Rate
Dilated SIFT hist.	44.8 (20.8)
+ Texton hist.	54.3 (21.1)
+ Top height	60.2 (23.2)
+ Color thumbnail	63.6 (25.0)
+ Dilated color hist.	66.4 (26.1)
+ Left boundary SIFT hist.	68.1 (26.8)
+ Right boundary SIFT hist.	69.4 (26.3)
+ SP mask over bounding box	69.8 (27.3)
+ Top boundary SIFT hist.	70.5 (27.9)
+ Color hist.	71.0 (27.9)
+ All remaining features	73.3 (29.1)

query image, giving us an upper bound for the performance of superpixel matching. We can see that a perfect shortlist "oracle" would give us a boost of almost 8%. This suggests that to further improve system performance, we may get a bigger payoff from more accurate scene-level label prediction, rather than from more sophisticated edge potentials in the MRF. In fact, we have observed that in many of our unsuccessfully labeled images, incompatible scene classes with strong local support over large regions vie for the interpretation of the image, and neighborhood context, though it may detect the conflict, has no plausible path towards resolving it (Figure 4(f) is one example of this).

Finally, we analyze the computational requirements of our system. Our current implementation is mostly in unoptimized and un-parallelized MATLAB (with some outside C code for feature extraction and MRF optimization), and all our tests are run on a single PC with dual Xeon 2.33 GHz quad core processors and 24 GB RAM. Table 3.1 shows a breakdown of the main stages of the computation. On the SIFT Flow dataset, we are able to extract features and label images in less than 10 seconds. In comparison, as reported in [15], to classify

Table 4. (a) Effect of retrieval set size on performance for the SIFT Flow dataset. (b) Effect of restricting the set of possible classes in the test image to different "shortlists."

(a)

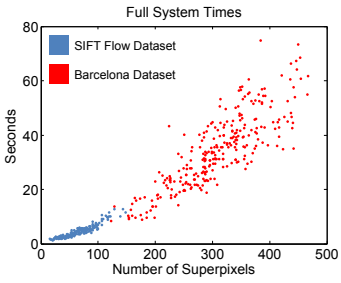
Retrieval Set Size	Rate
50	71.1 (30.1)
100	72.4 (29.7)
200	73.3 (29.1)
400	72.1 (27.2)
2,488	68.6 (19.1)

(b)

Shortlist	Rate
Classes in retrieval set	73.3 (29.1)
10 most common classes	73.2 (20.4)
Perfect shortlist	81.0 (34.0)

Table 5. Left: The average timing in seconds of the different stages in our system (excluding file I/O). While the runtime is significantly longer for the Barcelona dataset, this is primarily due to the change in image size and not the number of images. Right: query time vs. number of superpixels in the query image.

	SIFT Flow	Barcelona
Training set size	2,488	14,871
Image size	256×256	640×480
Ave. # superpixels	63.9	307.9
Feature extraction	$\sim$ 4 sec	$\sim$ 5 min
Retrieval set search	0.04 ± 0.0	0.21 ± 0.0
Superpixel search	4.4 ± 2.3	34.2 ± 13.4
MRF solver	0.005 ± 0.003	0.03 ± 0.02
Total (excluding features)	4.4 ± 2.3	34.4 ± 13.4



a single query image, the SIFT Flow system required 50 alignment operations that took 30 seconds each, or 25 minutes total without parallelization.

At present, our running time is actually dominated by our (wildly inefficient) feature extraction code that can be easily sped up by an order of magnitude. Our algorithm complexity is approximately quadratic in the average number of superpixels per image in the dataset due to the need to exhaustively match every test superpixel to every retrieval set superpixel. On the other hand, this time is independent of the overall number of training images. Moreover, as our dataset gets larger, we expect that target retrieval set size will stay the same or decrease, as the top scene matches will become closer to the test image. For larger datasets, the main bottleneck of our system will not be superpixel search, but retrieval set search and file I/O for loading retrieval set superpixel descriptors from disk. However, we expect to be able to overcome these challenges with appropriate hardware, parallelization, and/or data structures for fast search.

3.2 Small Datasets

To further validate our superpixel-based feature representation, we tested it on two small datasets: that of Gould et al. [6], which has 715 images with eight semantic and three geometric classes, and the geometric context dataset of Hoiem et al. [3], which has 300 images and seven surface layout classes (sky, ground, and five vertical sub-classes). For the latter, we treat these seven classes as the "semantic" classes, and the three geometric classes correspond to the main classes of [3]. Because nearest neighbor search requires a large set of training images to perform well, and because the competing approaches use heavily trained discriminative models, we train boosted decision tree classifiers similar to those of [3] on all the semantic and geometric classes. To obtain initial labelings of test images in these datasets, we do not use a retrieval set, but apply the boosted tree classifier for each class to each superpixel and use its likelihood ratio score in

the same way as eq. (1). Table 6 shows the resulting performance, which is comparable to the results of [3, 6]. Moreover, our system is much simpler than the competing approaches. Unlike [6], we do not need to learn classifiers over pairs of geometric and semantic classes or optimize image regions in a complex CRF framework. Unlike [3], we do not need to search over multiple segmentations with two tiers of features and a discriminative model of region homogeneity. In fact, when restricted to just a single superpixel segmentation, Hoiem et al. [3] report a sub-class rate of 53.5%, which we beat by 7.5% on the same superpixels.

4 Discussion

This paper has presented a superpixel-based approach to image parsing that can take advantage of datasets consisting of tens of thousands of images annotated with hundreds of labels. Our system does not need training, except for basic computation of dataset statistics such as label co-occurrence probabilities, and it relies on just a few constants that are kept fixed for all datasets. Our experimental evaluation carefully justifies every implementation choice. Despite its simplicity, our system outperforms state-of-the-art methods such as SIFT Flow [15]. Like [15], our method is nonparametric and makes use of a retrieval set of similar scenes, but unlike [15], it does not rely on an intricate optical flow-like scene alignment model. Our underlying feature representation, based on multiple appearance descriptors computed over segmentation regions, is similar to that of [3, 5]. However, unlike [3], we do not search over multiple segmentations, and unlike [5], we successfully combine features without learning class-specific or exemplar-specific distance functions. That one can achieve good performance without these costly steps is very encouraging for the prospect of successfully scaling up image parsing algorithms.

There still remain areas to improve our system further. Because our representation makes it easy to "plug in" new features, any advances in feature extraction are likely to give gains in performance. Also, while we have achieved promising results with one bottom-up segmentation algorithm [21], it remains important to examine the effect of segmentation quality on image parsing and to address the problem of finding the right spatial support for objects.

Table 6. A comparison of our system to [6, 3] using five-fold cross-validation and the same evaluation protocols as [6, 3].

	Gould *et al.* dataset [6]		Geometric Context dataset [3]	
	Semantic	Geometric	Sub-classes	Main classes
Gould *et al.* [6]	76.4	91.0	N/A	86.9
Hoiem *et al.* [3]	N/A	N/A	61.5	88.1
Local labeling	76.9	90.5	57.6	87.8
Superpixel MRF	77.5	90.6	61.0	88.2
Simultaneous	77.5	90.6	61.0	88.1

Acknowledgments

This research was supported in part by NSF CAREER award IIS-0845629, Microsoft Research Faculty Fellowship, and Xerox.

References

1. He, X., Zemel, R., Carreira-Perpinan, M.: Multiscale CRFs for image labeling. In: CVPR. (2004)
2. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR. (2008)
3. Hoiem, D., Efros, A., Hebert, M.: Recovering surface layout from an image. IJCV **75** (2007) 151–172
4. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: ICCV, Rio de Janeiro (2007)
5. Malisiewicz, T., Efros, A.A.: Recognition by association via learning per-exemplar distances. In: CVPR. (2008)
6. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV. (2009)
7. Russell, B.C., Torralba, A., Liu, C., Fergus, R., Freeman, W.T.: Object recognition by scene alignment. In: NIPS. (2007)
8. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR. (2008)
9. Heitz, G., Koller, D.: Learning spatial context: Using stuff to find things. In: ECCV. (2008)
10. Divvala, S.K., Hoiem, D., Hays, J.H., Efros, A.A., Hebert, M.: An empirical study of context in object detection. In: CVPR. (2009)
11. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: a database and web-based tool for image annotation. IJCV **77** (2008) 157–173
12. Hays, J., Efros, A.A.: Im2gps: Estimating geographic information from a single image. In: CVPR. (2008)
13. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: a large dataset for non-parametric object and scene recognition. PAMI **30** (2008) 1958–1970
14. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: SIFT flow: dense correspondence across difference scenes. In: ECCV. (2008)
15. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing: label transfer via dense scene alignment. In: CVPR. (2009)
16. Gu, C., Lim, J.J., Arbelaez, P., Malik, J.: Recognition using regions. In: CVPR. (2009)
17. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? PAMI **26** (2004) 147–159
18. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. PAMI **26** (2004) 1124–1137
19. Lazebnik, S., Schmid, C., , Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR. (2006)
20. Oliva, A., Torralba, A.: Building the gist of a scene: The role of global image features in recognition. Visual Perception, Progress in Brain Research, **155** (2006)
21. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International Journal of Computer Vision **2** (2004)
22. Bagon, S.: Graph cut matlab wrapper (2006) www.wisdom.weizmann.ac.il/~bagon.

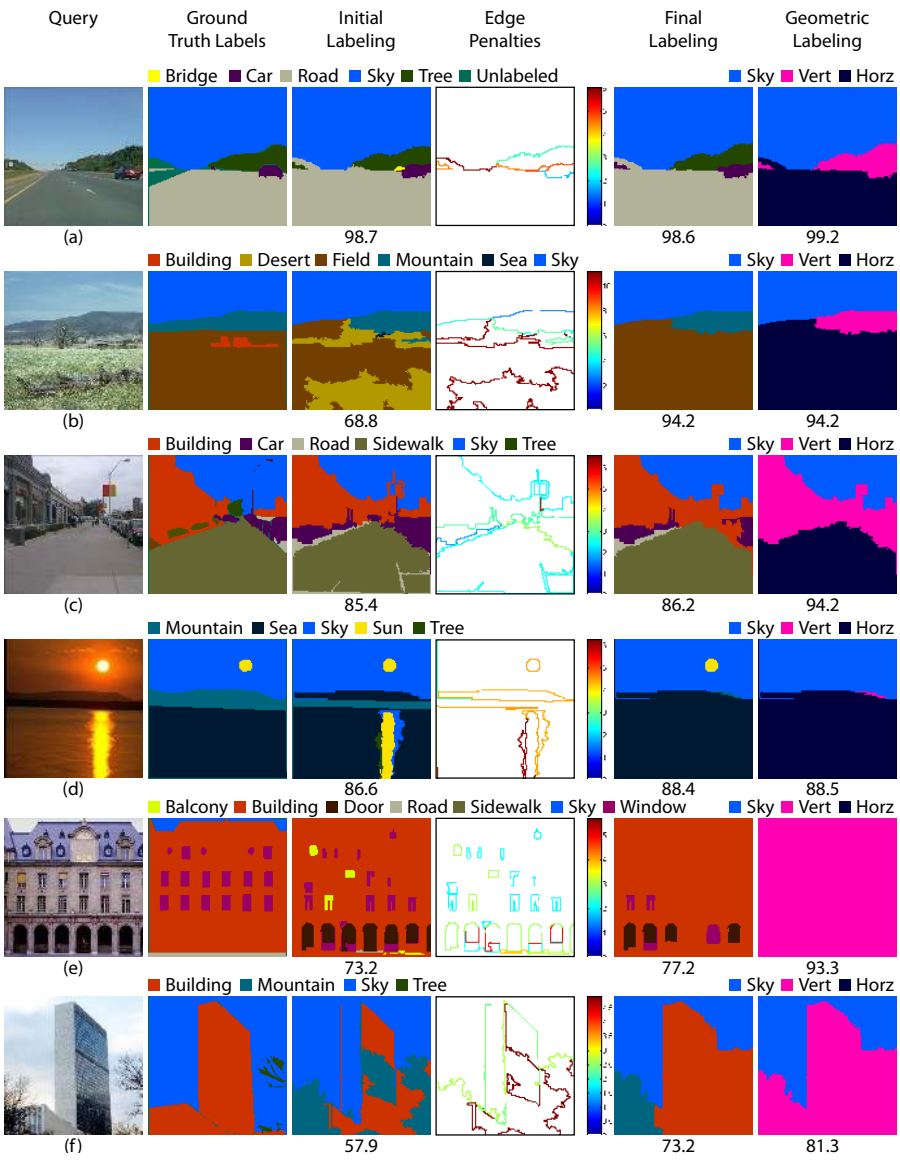


Fig. 4. Example results from the SIFT Flow test set (best viewed in color). In (c), sidewalk is successfully recovered. In (d), the co-occurrence MRF and joint geometric/semantic classification remove the spurious classification of the sun's reflection in the water as "sun." In (e), we find some windows (some of which are smoothed away by the MRF) and plausibly classify the arches at the bottom of the building as doors. In (f), parts of the building and the bare tree get initially classified as "mountain," and while the co-occurrence MRF does not like the boundaries between "building" and "mountain," it is not completely successful in eliminating the errors. For complete results, see `http://www.cs.unc.edu/SuperParsing`.