Noname manuscript No.
(will be inserted by the editor)

Superparsing

Scalable Nonparametric Image Parsing with Superpixels

Joseph Tighe · Svetlana Lazebnik

Received: date / Accepted: date

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 is based on lazy learning, and it can easily scale to datasets with tens of thousands of images and hundreds of labels. Given a test image, it first performs global scene-level matching against the training set, followed by superpixel-level matching 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 45,676 images and 232 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. Finally, we present an extension of our method to video sequences and report results on a video dataset with frames densely labeled at 1 Hz.

Keywords scene understanding, image parsing, image segmentation

Joseph Tighe
University of North Carolina
Computer Science Department
Chapel Hill, NC Tel.: 415.359.3535
E-mail: jtighe@cs.unc.edu

Svetlana Lazebnik
E-mail: lazebnik@cs.unc.edu

1 Introduction

This paper addresses the problem of image parsing, or segmenting all the objects in an image and identifying their categories. Many approaches to this problem have been proposed recently, including ones that estimate labels pixel by pixel [16,23,37,36], ones that aggregate features over segmentation regions [10,11,20,28,32,39], and ones that predict object bounding boxes [4,7,19, 34]. 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 [28, 29]). 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 [36]), processing a test image is also quite slow, as it involves steps like running multiple object detectors over the image, performing graphical model inference, or searching over multiple segmentations.

While most existing methods are tailored for "closed universe" datasets, a new generation of "open universe" datasets is beginning to take over. An example open-universe dataset is LabelMe [35], which consists of complex, real-world scene images that have been segmented and labeled by multiple users (sometimes incompletely or noisily). 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, a few researchers have begun advocating nonparametric, data-driven approaches suitable for open-universe datasets [15,43,27,26]. Such approaches do not

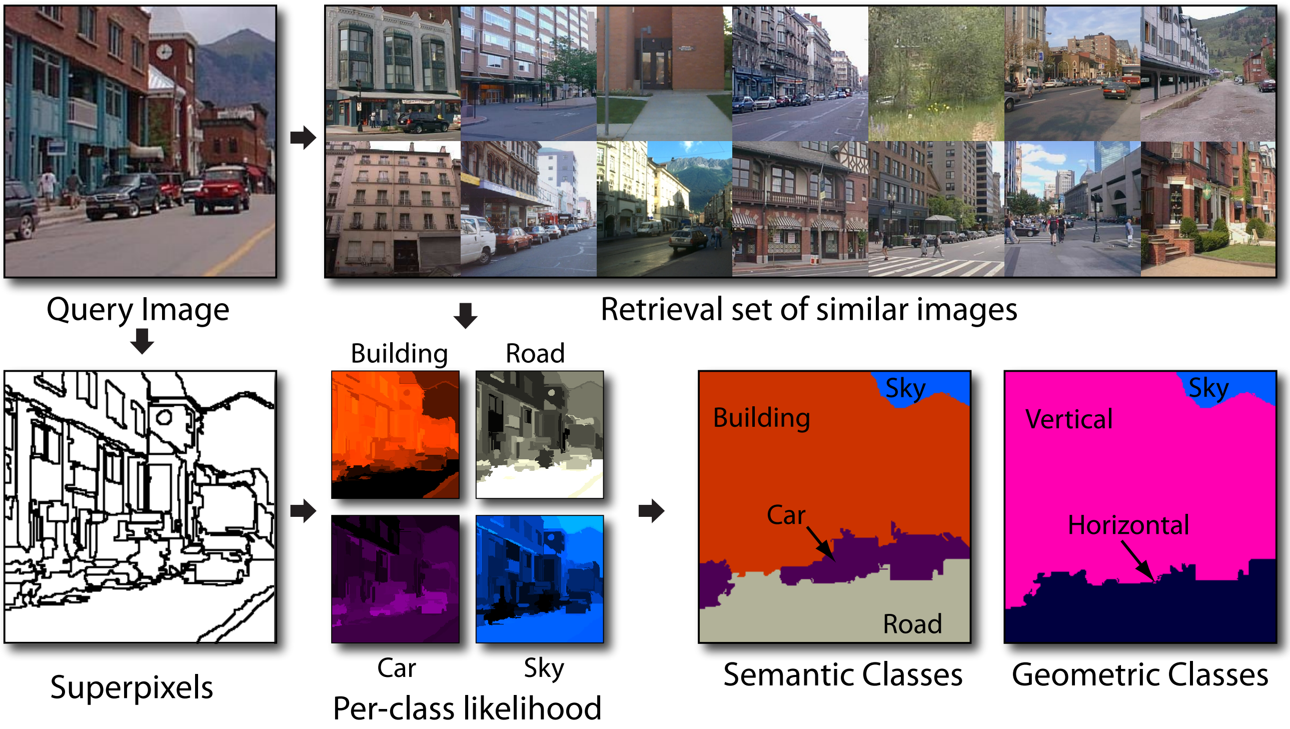

Fig. 1 System overview. Given a query image we retrieve similar images from our dataset using several global features. Next, we divide the query into superpixels and compute a per-superpixel likelihood ratio score for each class based on nearest-neighbor superpixel matches from the retrieval set. These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

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. [26] have proposed a nonparametric label transfer method based on estimating "SIFT flow," or a dense deformation field between images. The biggest drawback of this method is that the optimization problem for finding the SIFT flow is fairly complex and expensive to solve. Moreover, the formulation of scene matching in terms of estimating a dense per-pixel flow field is not necessarily in accord with our intuitive understanding of scenes as collections of discrete objects defined by their spatial support and class identity.

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 even larger image collections and sets of labels. Figure 1 gives an overview of our system. Similarly to [26], 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 [26], 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* [33], 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 [13].

The prevailing consensus in the recognition community is that image parsing requires *context* [4,9,19,20, 32]. However, learning and inference with most existing contextual models are slow and non-exact. Therefore, in keeping with 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 [1,2,22]. Our in-depth analysis presented in Section 3 demonstrates that such simple context is sufficient for good performance provided the local feature representation is powerful enough. We also investigate geometric/semantic context in the manner of Gould et al. [11]. 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 making sure the two types of labels assigned to the same region are consistent (e.g., a building has to be vertical, road horizontal, and so on). Our experiments show that enforcing this coherence improves the performance of both labeling tasks.

Our system exceeds the results reported in [26] 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 from the LabelMe and SUN [44] datasets totaling 45,676 images and 232 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 our work is to establish a new benchmark for large-scale image parsing. Note that unlike other popular benchmarks for image parsing (e.g., [11,20,26,37]), our LabelMe+SUN dataset contains both outdoor and indoor images. As will be discussed in Section 3.3, indoor imagery currently appears to be much more challenging for general-purpose image parsing systems than outdoor imagery, due in part to the greater diversity of indoor scenes, as well as to the smaller amount of training data available for them.

As another contribution, we extend our parsing approach to video and show that we can take advantage of motion cues and temporal consistency to improve performance. Existing video parsing approaches [3,47] use structure from motion to obtain either sparse point clouds or dense depth maps, and extract geometry-based features that can be combined with appearance-based features or used on their own to achieve greater accuracy. We take a simpler approach and only use motion cues to segment the video into temporally consistent regions [12], or *supervoxels*. This helps to better separate moving objects from one another especially when there is no high-contrast edge between them. Our results in Section 4 show that the incorporation of motion cues from video can significantly help parsing performance even without the explicit reconstruction of scene geometry.

A previous version of this work has been published in [41]. The main advances over [41] are: an in-depth analysis of the various parameters of our system, an evaluation on the new LabelMe+SUN dataset containing both outdoor and indoor images, and an extension of our system to video parsing. Our code and data can be found at `http://www.cs.unc.edu/SuperParsing`.

2 System Description

This section presents the details of all the components of our system. It is based on a *lazy learning* philosophy, meaning that (almost) no training takes place offline; given a test image to be interpreted, our system dynamically selects the training exemplars that appear to be the most relevant and proceeds to transfer labels from them to the query. The following is a summary of the steps taken by the system for every query image.

1. Find a retrieval set of images similar to the query image (Section 2.1).
2. Segment the query image into superpixels and compute feature vectors for each superpixel (Section 2.2).
3. For each superpixel and each feature type, find the nearest-neighbor superpixels in the retrieval set according to that feature. Compute a likelihood score for each class based on the superpixel matches (Section 2.3).
4. Use the computed likelihoods together with pairwise co-occurrence energies in an Markov Random Field (MRF) framework to compute a global labeling of the image (Section 2.4). Alternatively, with modifications, the MRF framework can simultaneously solve for both semantic and geometric class labels (Section 2.5).

2.1 Retrieval Set

Similarly to several other data-driven methods [15,26,27,34], 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 matches. 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 will contain images that have similar scene types, objects, and spatial layouts to the query image. In the attempt to indirectly capture this kind of similarity, we use three types of global image features (Table 1(a)): spatial pyramid [25], gist [31], 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 image, and use the top-ranking K images as the retrieval set (a typical value of K in our experiments is 200). Empirically, this method gives us an improvement of 1-2% over other schemes, such as simply 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 (Table 1b).

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
	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 by 10 pix texton histogram	100 × 2
	Quantized SIFT histogram, dilated by 10 pix quantized SIFT histogram	100 × 2
	Left/right/top/bottom boundary quantized SIFT histogram	100 × 4
Color	RGB color mean and std. dev.	3 × 2
	Color histogram (RGB, 11 bins per channel), dilated by 10 pix color histogram	33 × 2
Appearance	Color thumbnail (8 × 8)	192
	Masked color thumbnail	192
	Grayscale gist over superpixel bounding box	320

We examine the contributions of different global features and the effect of changing the retrieval set size K in the experiments of section 3.3.

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 [16, 26, 27] tends to be too inefficient. Instead, like [20, 28, 32], 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 windows centered on every pixel in the image. We obtain superpixels using the fast graph-based segmentation algorithm of Felzenszwalb and Huttenlocher [8] [1] and describe their appearance using 20 different features similar to those of Malisiewcz and Efros [28], with some modifications and additions. A complete list of the features is given in Table 1(b). In particular, we compute histograms of textons[2] and dense SIFT descriptors over the superpixel region, as well as a version of 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 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 a ground truth segment mask with that label.

2.3 Local Superpixel Labeling

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

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

where $\bar{c}$ is the set of all classes excluding c. 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

[1] We set $K = 200$ and $\sigma = .8$

[2] Code: http://www.robots.ox.ac.uk/˜vgg/research/texclass/filters.html

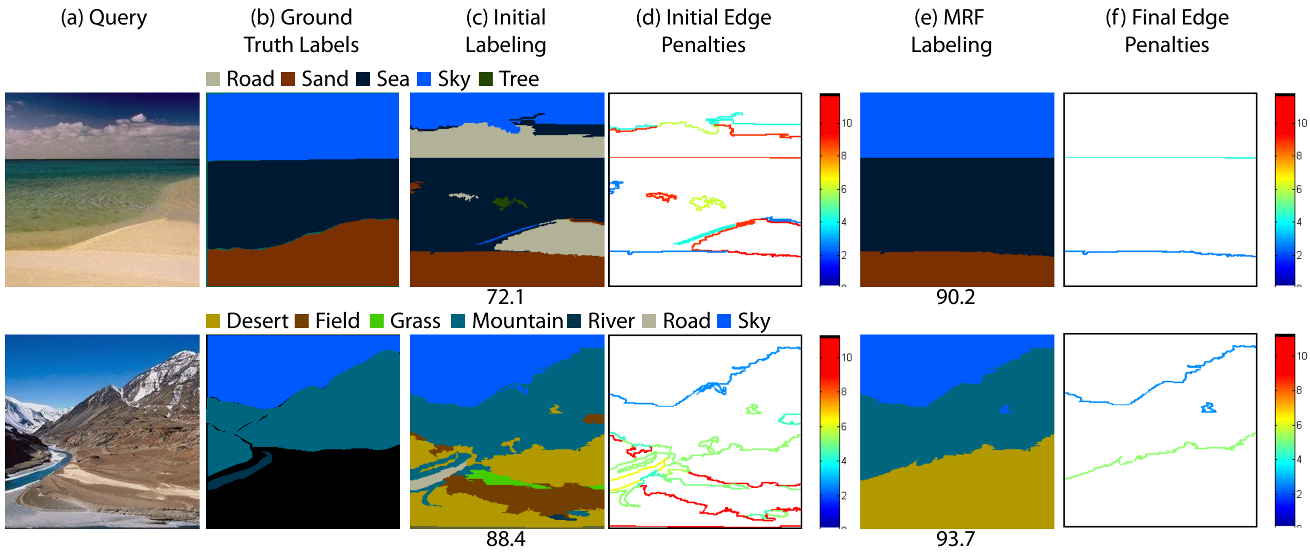

Fig. 2 Our contextual edge penalty before and after we run our MRF optimization. The top row shows our contextual model successfully flags improbable boundaries between "sea" and "road" and the second row shows it flags "Desert" and "Field".

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

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) and ϵ is a constant added to prevent zero likelihoods and smooth the counts. In our implementation, we use the ℓ_2 distance for all features, and set each threshold t_k to the median distance to the Tth nearest neighbor for the kth feature type over the dataset. Interestingly, the radius threshold t_k does not seem to have a large influence on the performance of our system, though using a radius instead of taking a fixed number of nearest neighbors was very important to achieve high performance. We use a target number of near neighbors $T = 80$ for all experiments in this paper. We examine the effect of changing the smoothing constant (ϵ) in section 3.3. The superpixel neighbors $\mathcal{N}_i^k$ are found by linear search through the retrieval set. While approximate nearest neighbor techniques could be used to speed up this search, at our current scale this is not the computational bottleneck of our system as will be discussed in Section 3.3.

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 2.5% of those of [26].

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 [9,11,16,30,32]. 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:

$$E_{\text{data}}(s_i, c_i) = -w_i \sigma(L(s_i, c_i)), \tag{4}$$

where $L(s_i, c_i)$ is the likelihood ratio score from eq. (1), $\sigma(t) = \exp(\gamma t)/(1 + \exp(\gamma t))$ is the sigmoid function[3] and w_i is the superpixel weight (the size of s_i in pixels

[3] Note that our original system [41] did not use the sigmoid nonlinearity, but in our subsequent work [42] we found it necessary to successfully perform more complex multi-level

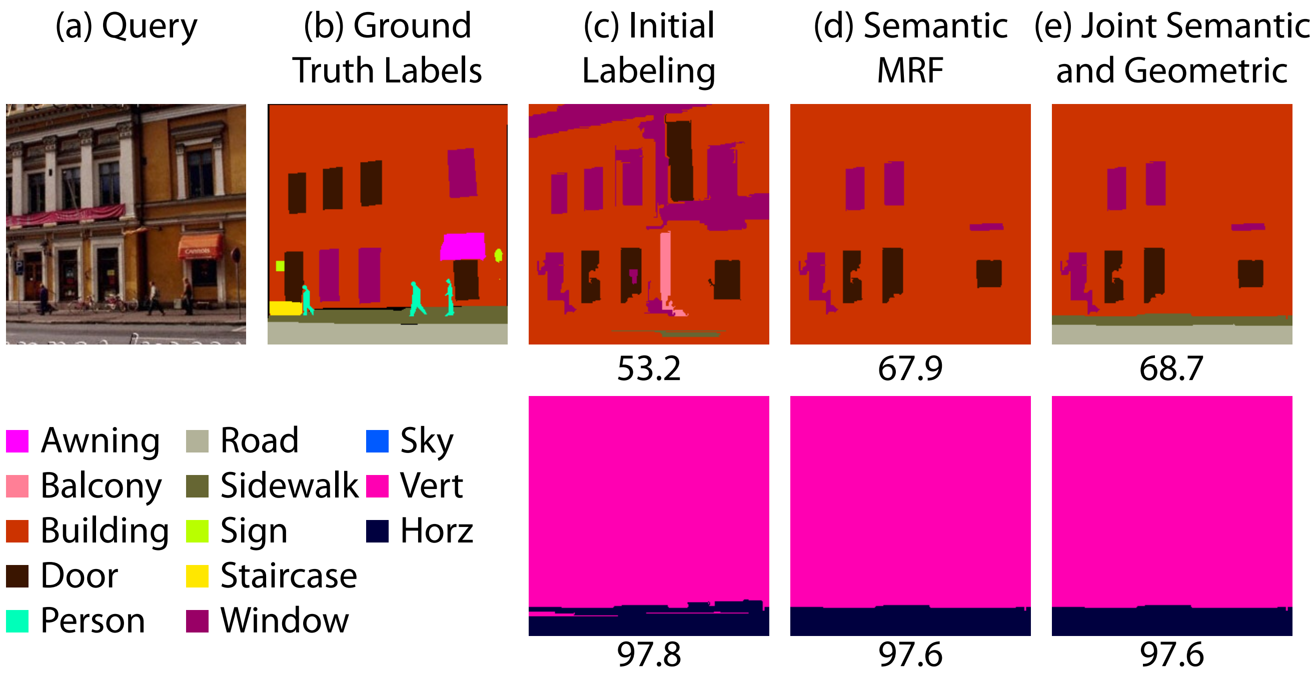

Fig. 3 In the contextual MRF classification, the road gets replaced by "building," while "horizontal" is correctly classified. By jointly solving for the two kinds of labels, we manage to recover some of the "road" and "sidewalk" in the semantic labeling. Note also that in this example, our method correctly 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.

divided by the mean superpixel size). The smoothing term E_{smooth} is defined based on probabilities of label co-occurrence:

$$E_{\text{smooth}}(c_i, c_j) = -\log[(P(c_i|c_j) + P(c_j|c_i))/2] \times \delta[c_i \neq c_j]\,, \tag{5}$$

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. Multiplication by $\delta[c_i \neq c_j]$ is necessary to ensure that this energy term is semi-metric as required by graph cut inference [2]. Qualitatively, we have found eq. (5) to produce very intuitive edge penalties. As can be seen from the examples in Figure 2, it successfully flags improbable boundaries. Quantitatively, results with eq. (5) 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 [1,2,22]. On our large datasets, the resulting labelings improve the accuracy by 2-4% (Table 2 and 3).

We have also experimented with a contrast-sensitive per-pixel MRF similar to that of [26], but have found that our per-superpixel formulation is faster, and achieves the same per-pixel and per-class performance. One reason for this may be that the per-superpixel MRF makes it easier to converge to a better minimum by flipping labels over larger areas of the image. A per-pixel MRF does however produce more visually pleasing labelings, but we chose to use the superpixel-based MRF due to its superior speed.

2.5 Simultaneous Classification of Semantic and Geometric Classes

To achieve more comprehensive image understanding and to explore a higher-level form of context, we consider the task of simultaneously labeling regions into two types of classes: semantic and geometric [11]. Like Gould et al. [11], we use three geometric labels – sky, horizontal, and vertical – although the sets of semantic labels in our datasets are much larger. In this paper, we make the 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. This is a bit restrictive for a few classes (e.g., we force "rock" and "mountain" to be vertical), but for the vast majority of semantic classes,

inference. We have also found that the sigmoid is a good way of making the output of the nonparametric classifier comparable to that of other classifiers, for example, boosted decision trees (see Section 3.1).

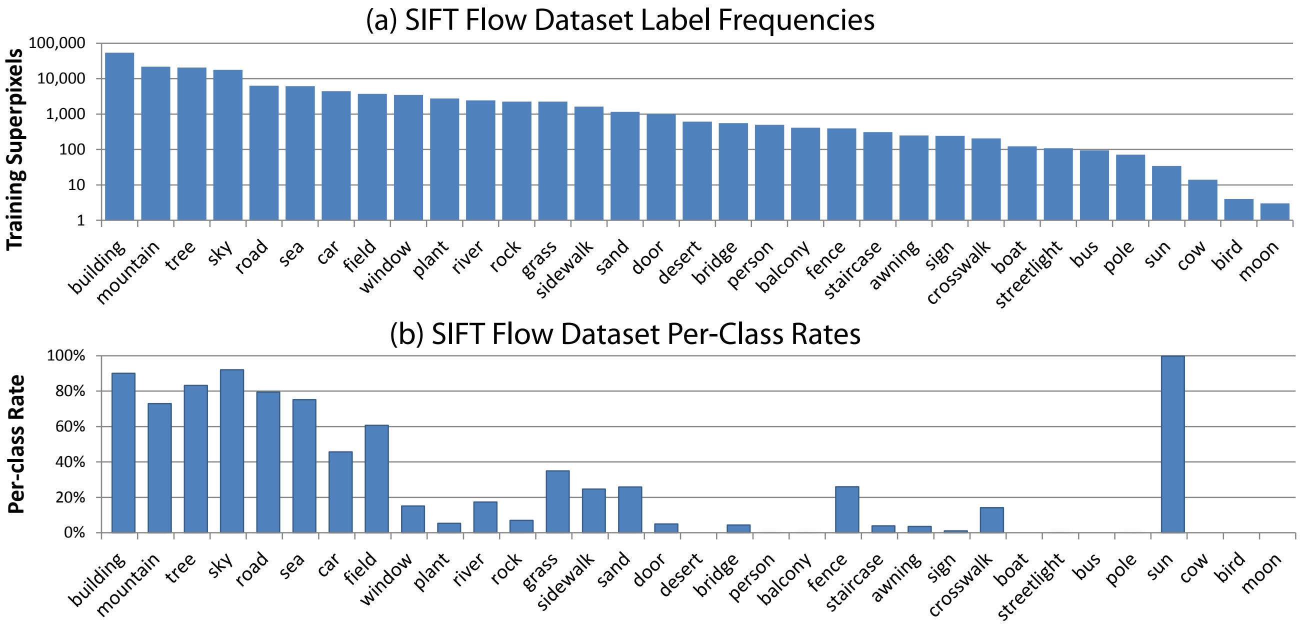

Fig. 4 Label frequencies for the superpixels in the training set and the classification rate broken down by class for our full system on the SIFT Flow dataset.

a unique geometric label makes sense. We jointly solve for the fields of semantic labels (**c**) and geometric labels (**g**) by minimizing a cost function that is a simple extension of eq. (5):

$$H(\mathbf{c},\mathbf{g}) = J(\mathbf{c}) + J(\mathbf{g}) + \mu \sum_{s_i \in SP} \varphi(c_i, g_i), \qquad (6)$$

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 1 otherwise. The constant μ controls how strictly the coherence is enforced (we use $\mu = \lambda = 1$ in all experiments). Note that it is possible to 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. (6) is in a form that can be optimized by the α/β-swap algorithm [1,2,22]. The inference takes almost the same amount of time as for the MRF setup of the previous section. Figure 3 shows an example where joint inference over semantic and geometric labels improves the accuracy of the semantic labeling. More generally, as will be shown by the quantitative results of Section 3, joint inference tends to improve both labelings simultaneously.

3 Image Parsing Results

In Sections 3.1 and 3.2, we give an overview of results on our two large datasets, SIFT Flow and Labelme+SUN. Section 3.3 gives a thorough evaluation of all the major components of the system.

3.1 SIFT Flow Dataset

The first dataset in our experiments, referred to as "SIFT Flow dataset" in the following, comes from [26]. It is composed of the 2,688 images that have been thoroughly labeled by LabelMe users. Liu et al. [26] have split this dataset into 2,488 training images and 200 test images and used synonym correction to obtain 33 semantic labels. We use the same training/test split as [26].

The frequencies of different labels on this dataset are shown in Figure 4(a). It is clear that they are very non-uniform: a few classes like building, mountain, tree, and sky are very common, but there is also a "long tail" of relatively rare classes like person, sign, boat, and bus. To give a fair idea of our system's performance on such unbalanced data, we evaluate accuracy using not only the per-pixel classification rate, which is mainly determined by how well we can label the few largest classes, but also the average of the per-pixel rates over all the classes.

As explained in Section 2.5, our system labels each superpixel by a semantic class (the original 33 labels)

Table 3 Performance on the LMSun dataset broken down by outdoor and indoor test images. Per-pixel classification rate is listed first, followed by the average per-class rate in parentheses.

	All		Outdoor		Indoor	
	Semantic	Geometric	Semantic	Geometric	Semantic	Geometric
Local labeling	50.6 (7.1)	79.8 (85.6)	56.7 (7.7)	83.0 (87.5)	27.0 (4.9)	67.8 (74.6)
MRF	54.4 (6.8)	82.6 (86.8)	60.4 (7.6)	85.2 (88.6)	31.2 (4.5)	72.6 (76.1)
Joint	54.9 (7.1)	85.9 (86.8)	60.8 (7.7)	88.3 (89.3)	32.1 (4.8)	76.6 (74.0)

Table 2 Performance on the SIFTFlow dataset for our system and three state-of-the-art approaches. Per-pixel classification rate is listed first, followed by the average per-class rate in parentheses.

	Semantic	Geometric
Local labeling	74.1 (30.2)	90.2 (88.7)
MRF	76.2 (29.1)	90.6 (88.9)
Joint	77.0 (30.1)	90.8 (89.2)
Liu et al. (2011) [26]	76.7	
Farabet et al. (2012) [6]	78.5 (29.6)	
Eigen and Fergus (2012) [5]	77.1 (32.5)	

and a geometric class of sky, horizontal, or vertical. Because the number of geometric classes is small and fixed, we have trained a boosted decision tree (BDT) classifier as in [20] to distinguish between them. We use a tree depth of 8 and train 100 trees for each class. This classifier outputs a likelihood ratio score that is comparable to the one produced by our nonparametric scheme (eq. 1), but that gets about 2% higher accuracy for geometric classification (Section 3.3 will present a detailed comparison of nearest-neighbor classifiers and BDT). Apart from this, local and MRF classification for geometric classes proceeds as described in Sections 2.3 and 2.4, and we also put the semantic and geometric 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 and geometric classification of local superpixel labeling, separate semantic and geometric MRF, and joint semantic/geometric MRF. As compared to the local baseline, the contextual MRF improves overall per-pixel rates on the SIFT Flow dataset by about 2%. The average per-class rate for the MRF drops due to "smoothing away" some of the smaller classes, while joint semantic/geometric MRF improves the results for both types of classes. Figure 4(b) shows classification rates for the 33 individual classes. Similarly to most other image labeling approaches that do not rely on object detectors, we get much weaker performance on "things" (people, cars, signs) than on "stuff" (sky, road, trees).

Our final system on the SIFT Flow dataset achieves a classification rate of 77.0%. Thus, we outperform Liu et al. [26], who report a rate of 76.7% 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. [26] also cite a rate of 82.72% for the top seven object categories; our corresponding rate is 84.7%. Table 2 also reports results of two new approaches [5, 6] that build on and compare to the earlier version of our system [41]. Eigen and Fergus [5] are able to improve on our average per-class rate, while Farabet et al. [6] are able to improve on the overall rate through the use of more sophisticated learning techniques.

Sample output of our system on several SIFT Flow test images can be seen in Figure 13.

3.2 LM+SUN Dataset

Our second dataset ("LM+SUN" in the following) is derived by combining the SUN dataset [44] with a complete download of LabelMe [35] as of July 2011. We cull from this dataset any duplicate images and any images from video surveillance (about 10,000), and use manual synonym correction to obtain 232 labels. This results in 45,676 images of which 21,182 are indoor and 24,494 are outdoor. We split the dataset into 45,176 training images and 500 test images by selecting test images at random that have at least 90% of their pixels labeled and at least 3 unique labels (a total of 13,839 images in the dataset meet this criteria). Apart from its bigger size, the inclusion of indoor images makes this dataset very challenging.

As shown in Figure 5(a), the LM+SUN dataset has unbalanced label frequencies just like SIFT Flow. Table 3 shows the performance for the three versions of our system (local maximum likelihood labeling, separate semantic and geometric MRF, and joint semantic/geometric MRF) on the entire dataset, as well as on outdoor and indoor images separately. The overall trend is the same as for SIFT Flow: separate MRF inference always increases the overall accuracy over the local baseline though it can sometimes over-smooth, decreasing the average per-class rate. As for joint semantic/geometric inference, it not only gives the highest overall accuracy in all cases, but is also much less prone to over-smoothing.

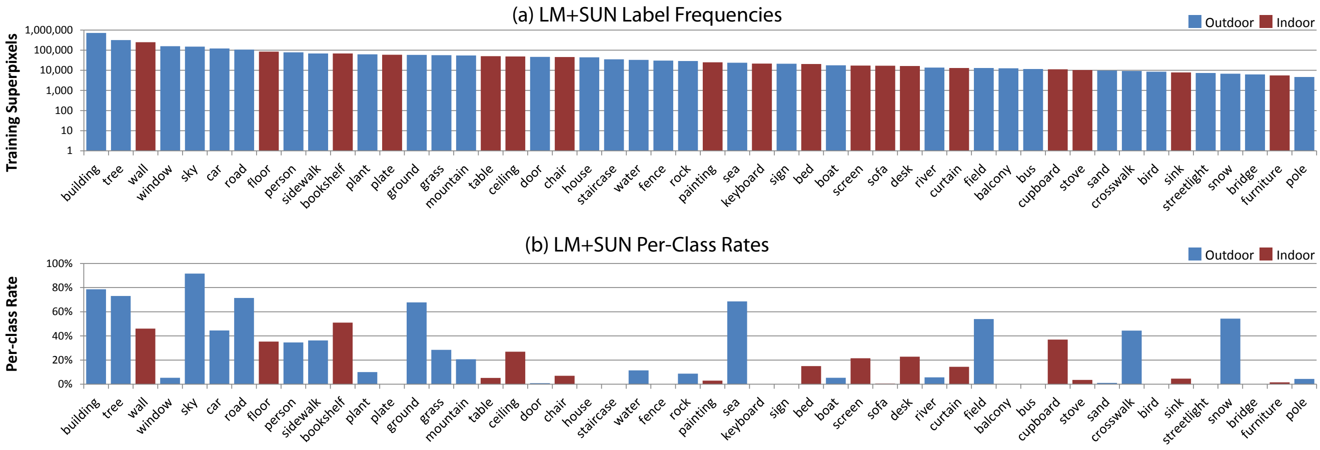

Fig. 5 Label frequencies for the superpixels in the training set and the classification rate broken down by class for our full system on the LM+SUN dataset. Only the 50 most common classes are shown.

Our final system achieves a classification rate of 54.9% across all scene types (as compared to 77% for SIFT Flow); the respective rates for outdoor and indoor images are 60.8% and 32.1%. Figure 5(b) gives a breakdown of rates for the 50 most common classes, and Figure 14 shows the output of our system on a few example images. It is clear that indoor scenes are currently much more challenging than outdoor ones, due at least in part to their greater diversity and sparser training data. In fact, we are not aware of any system that can produce accurate dense labelings of indoor scenes; most existing work dealing with indoor scenes tries to leverage specialized geometric information and focuses on only a few target classes of interest. For example, Hedau et al. [18] infer the "box" of a room and then leverage the geometry of that box to align object detectors to the scene [17]. Gupta et al. infer the possible use of a space [14] rather than directly labeling the objects. There has also been a recent interest in indoor parsing with the help of structured light sensors [38, 21, 24] as a way to combat the ambiguity present in cluttered indoor scenes. It is clear that our system, which relies on generic appearance-based matching, cannot currently achieve very high levels of performance on indoor imagery. However, our reported numbers can serve as a useful baseline for more advanced future approaches. The challenges of indoor classification will be further studied in Section 3.3.

Table 4 Evaluation of global image features for retrieval set generation (retrieval set size 200). "Maximum Label Overlap" is the upper bound that we get by selecting retrieval set images that are the most semantically consistent with the query (see text).

Global Descriptor	SIFT Flow	LMSun
Gist (G)	70.8 (29.7)	45.6 (7.0)
Spatial Pyramid (SP)	69.6 (23.1)	47.9 (6.3)
Color Hist. (CH)	66.9 (24.6)	43.5 (5.7)
G + SP	72.3 (27.9)	50.6 (6.9)
G + SP + CH	74.1 (30.2)	50.6 (7.1)
Maximum Label Overlap	80.2 (33.6) [4]	66.0 (13.2)

3.3 Detailed System Evaluation

This section presents a detailed evaluation of various components of our system. Unless otherwise noted, the evaluation is conducted on both the SIFT Flow dataset and LM+SUN, no MRF smoothing is done, and only semantic classification rates are reported.

3.3.1 Retrieval set selection

The initial step of our system is retrieval set selection. Table 4 shows the performance of different global features used for this step. Similarly to [15], we find that combining global features of complementary descriptive power gives better scene matches. The last line in this table, "Maximum Label Overlap," is meant to be an upper bound on the performance of the retrieval set. Here the retrieval set is found by ranking training images in terms of the number of pixels their ground truth label maps share with the label map of the query. The big gap in accuracy between this "ideal" retrieval set and the one obtained by global appearance-based matching underscores the shortcomings of global image features

[4] The published version of this paper reported an erroneously high result for this test. The result shown here is correct.

Table 5 Effect of retrieval set size on local superpixel labeling. Note that the entire LM+SUN training set is too large for our hardware to store in memory.

Retrieval Set Size	SIFT Flow	LM+SUN
50	73.0 (32.2)	47.3 (8.1)
100	73.7 (30.1)	48.9 (7.4)
200	74.1 (30.2)	50.6 (7.1)
400	73.0 (28.7)	51.0 (7.6)
800	72.1 (28.1)	51.5 (7.5)
1,600	69.9 (26.2)	51.2 (8.1)
Entire training set	68.4 (23.2)	N/A

Table 6 Accuracy of local superpixel labeling obtained by restricting the set of possible classes in the test image to different "shortlists" (see text).

Shortlist	SIFTFlow	LM+SUN
Classes in retrieval set	74.1 (30.2)	50.6 (7.1)
10 most common classes	74.9 (21.4)	51.0 (3.1)
Perfect shortlist	81.4 (35.4)	61.7 (11.1)

Table 7 Effect of indoor/outdoor separation on the accuracy of local superpixel labeling on LM+SUN. "Local labeling" corresponds to our default system with no separation between outdoor and indoor training images (the numbers are the same as in line 1 of Table 3). "Ground truth" uses the ground truth label for the query image to determine if the retrieval set should consist of indoor or outdoor images, while "Classifier" uses a trained indoor/outdoor SVM classifier (see text).

	All	Outdoor	Indoor
Local labeling	50.6 (7.1)	56.7 (7.7)	27.0 (4.9)
Ground truth	54.4 (7.8)	59.1 (8.3)	36.6 (5.6)
Classifier	52.7 (7.3)	57.4 (8.2)	34.4 (5.5)

in terms of finding scenes *semantically* consistent with the query.

Table 5 examines the effect of retrieval set size. Interestingly, using all of the SIFT Flow training set as the retrieval set (last row of Table 5) drastically reduces performance. Thus, we quantitatively confirm the intuition that the retrieval set is not just a way to limit the computational complexity of superpixel matching, but is also a form of scene-level context. By restricting the superpixel matches to come from a small subset of related scenes, we can get a better interpretation of the image. Also note that while on the SIFT Flow dataset the performance degrades once the retrieval set reaches a size greater than 400, the performance on the LM+SUN dataset continues to rise even with a retrieval set size of 1,600: with more than 45,000 images in that dataset, there are usually still 1,600 images that are of a sufficiently similar scene type. Thus, it appears that the right retrieval set size depends in a complex way on the size of the dataset and on the distribution of scene types contained in it. Despite this, in all our other experiments we use a retrieval set size of 200 for both datasets, primarily for efficiency: we must read the descriptor data from disk for each query image on the LM+SUN dataset, which becomes prohibitively slow with larger retrieval set sizes.

While the total number of labels in our datasets is quite high, any single image only contains a small subset of all possible labels. The retrieval set defines not only the set of training images that can be used to interpret the test image, but also the "shortlist" of all possible labels that can appear in the test image. By default, this shortlist in our system is composed of all the classes present in the retrieval set. Table 6 examine the effect of restricting these shortlists in various ways. The first row corresponds to the default shortlist (the same one that is used in the experiments of the previous two sections). To demonstrate the effect of long-tail class frequencies, the second row shows the performance we get by classifying every superpixel in every test image to the ten most common classes in the dataset. This slightly increases the overall per-pixel rate, but lowers the average per-class rate dramatically. On the other hand, it is worth observing that the average per-class rate can be inflated upwards by good performance on a few very rare classes (e.g., there are only two "suns" in the SIFT Flow test set, and we get both of them). Finally, the third row of Table 6 shows the results produced by restricting the shortlist to the ground truth labels in the query image, giving us an upper bound for the performance of superpixel matching. Just like the "maximum label overlap" retrieval set of Table 4, a perfect shortlist "oracle" would give us significant boosts in overall per-pixel rate and average per-class rate on both datasets. This suggests that to further improve system performance, it is important to work on more accurate scene-level label prediction and better scene-level matching for generating the retrieval sets.

The LM+SUN dataset has two obvious sub-classes: indoor and outdoor. However, retrieval set selection based on low-level features does not do a very good job of separating them: for an indoor query image there are often outdoor images in the retrieval set and vice versa (see Figure 6 for an illustration). To get an idea of how much this confusion hurts performance, we can use ground truth knowledge to force the retrieval set to have only images of the correct scene type – that is, an indoor query image would only be matched against indoor images and likewise for an outdoor one (this is equivalent to splitting the LM+SUN dataset into two separate indoor and outdoor datasets). Table 7 (line 2) shows the resulting improvement in overall perfor-

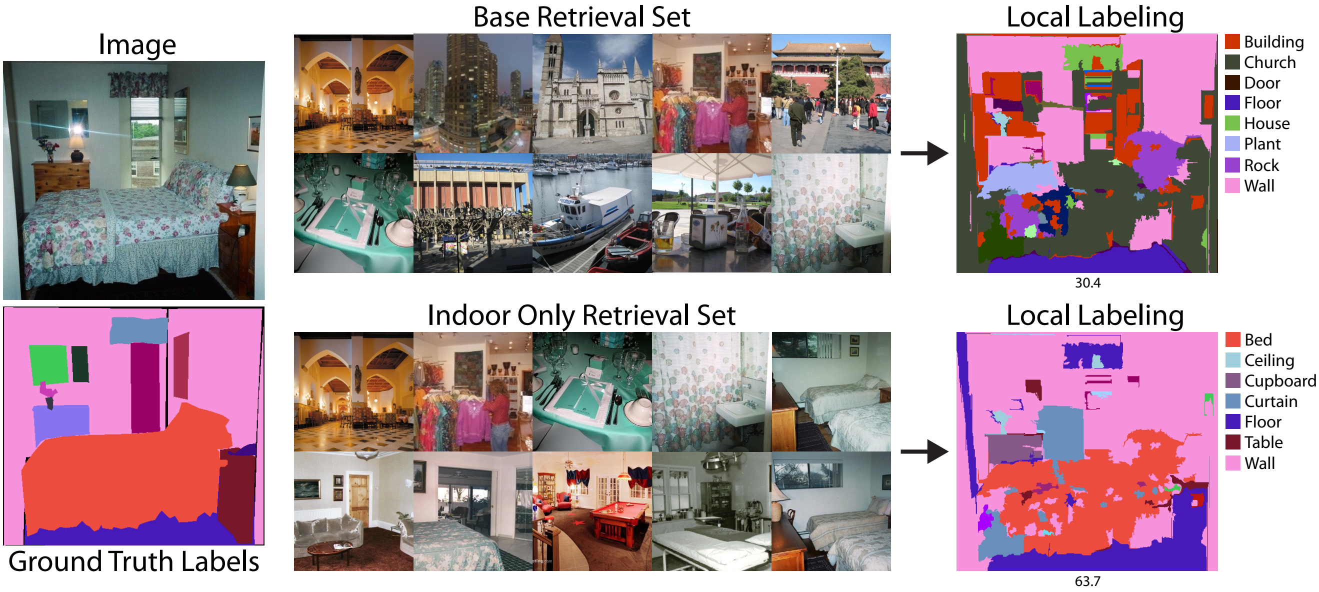

Fig. 6 Effect of drawing the retrieval set from the entire training set ("base retrieval set") vs. drawing it from the correct scene type ("indoor only retrieval set"). With the base retrieval set, the local labeling result has a mix of indoor (floor, wall) and outdoor (building, rock) classes. On the other hand, if we restrict the retrieval set to consist only of indoor images, we get a much cleaner result – in particular, we manage to label most of the bed.

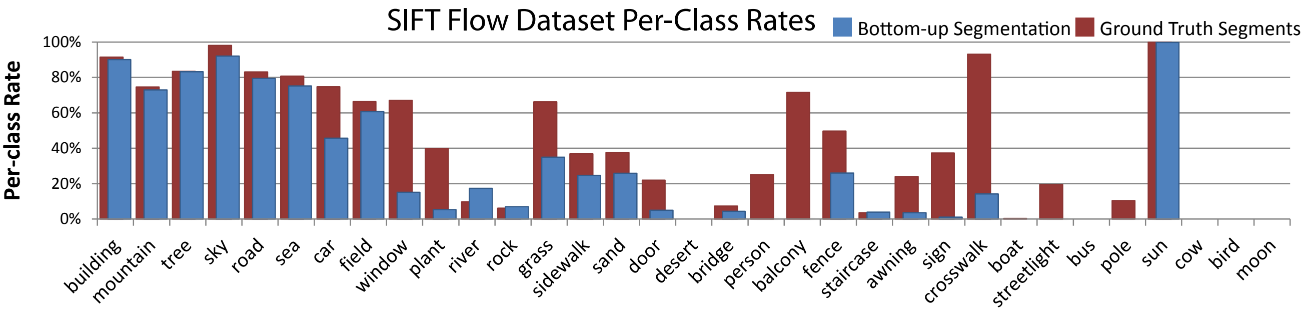

Fig. 7 Per-class classification rates on the SIFT Flow dataset for superpixels (blue) versus ground truth object polygons (maroon). With the ground truth segmentation, we get an overall per-pixel rate of 82.3 and average class rate of 46.0, showing that even with correct knowledge of object shape, we still have a hard time classifying many of the classes.

mance. To get most of this gain without "cheating," we train an indoor/outdoor classifier using a linear SVM on all our global image features concatenated and normalized by the standard deviation along each dimension. This classifier achieves a rate of 92% on our test set. The last row of Table 7 shows the performance of our system when we use this classifier to determine which set of training images to draw our retrieval set from. As expected, the accuracy is somewhere in between that of "perfect" indoor-outdoor classification and no classification altogether.

Note that performing automatic indoor/outdoor image classification and then using the inferred scene type to constrain the interpretation of the image is conceptually similar to performing a geometric labeling of the image and using the inferred geometric classes of regions to constrain the semantic classes. In both cases we are taking advantage of the high accuracy that can be achieved on relatively easier two- and three-class problems to improve the accuracy on a harder many-class problem.

3.3.2 Superpixel classification

After retrieval set selection, the next stage of our system is superpixel classification. One of the most important factors affecting the success of this stage is the quality of the bottom-up segmentation, or how well the spatial support of the superpixels reflects true object boundaries. To see what would happen if we had perfect segmentation, we use the ground truth object polygons to create segments for the SIFT Flow dataset and then

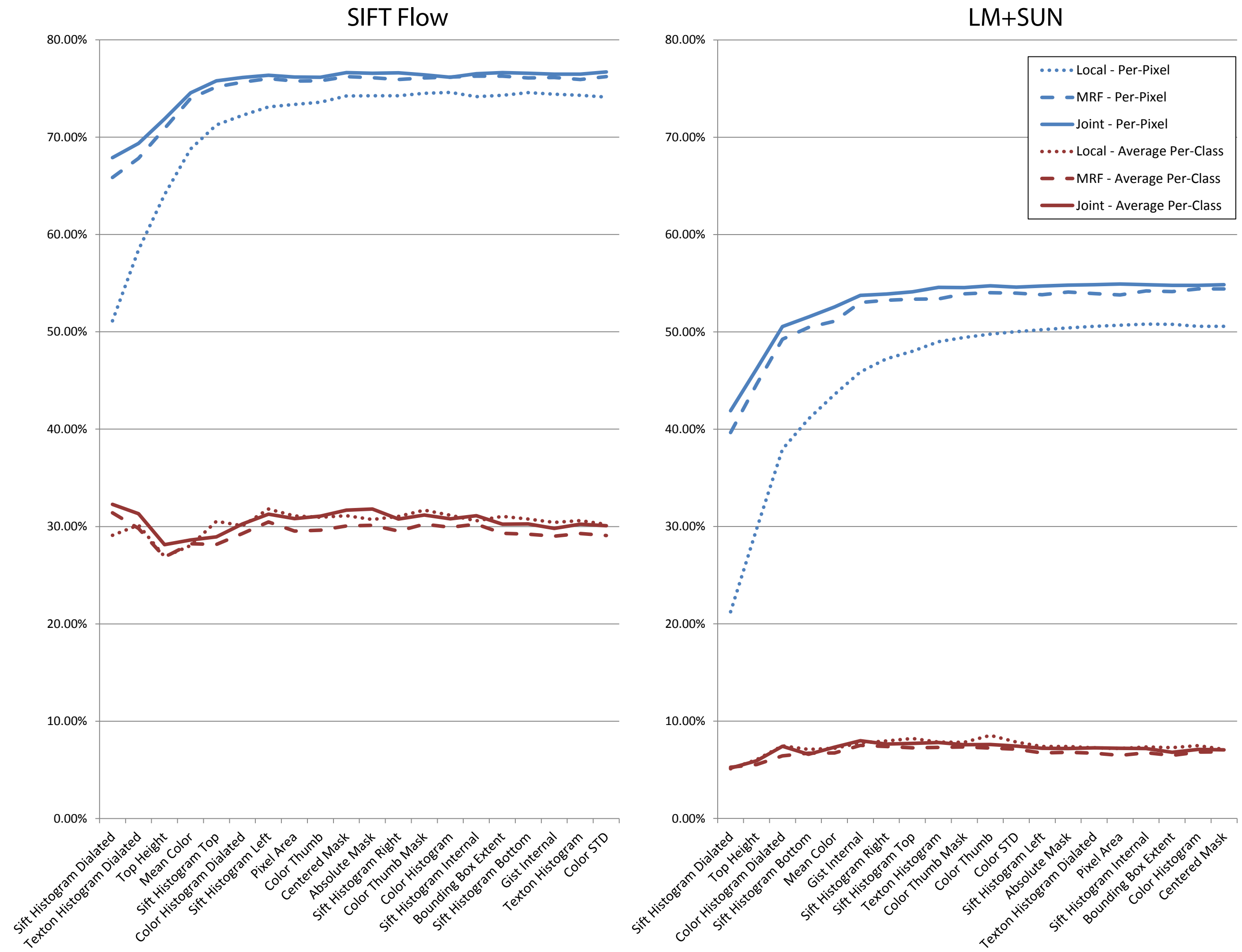

Fig. 8 The classification rate of our system computed by consecutively adding superpixel features in decreasing order of their contribution. Interestingly, the best features for one dataset are not necessarily the best for the other one. We also plot the rates for the MRF and joint solvers after each feature is added. Notice that the MRF and joint solver are more effective when the classifier is weaker, and that the joint solver consistently outperforms the MRF for the average per-class rate, correcting for the over-smoothing that occurs due to the MRF.

apply our local classification scheme to these segments. Figure 7 compares the per-class rates of our system with bottom-up segmentation and to those of ground truth segments. We can see that ground truth segmentation significantly helps with many classes such as car, window, balcony, and crosswalk. However, we still do quite poorly on many of the "thing" classes such as person, boat, and streetlight. Thus, even if we could get perfect object outlines, the task of classifying many of the rarer classes would remain quite challenging.

Next we look at the contributions of our multiple superpixel-level features (refer back to Table 1(b) for a list of these features). Figure 8 plots the classification rate of the system on both datasets with superpixel features added consecutively in decreasing order of their contribution to performance. Note that this evaluation is carried out on the test set itself (as opposed to a separate validation set), as our goal is simply to understand the behavior of our representation, not to tune performance or to perform feature selection. We start with the single superpixel feature that has the highest per-pixel classification rate, and then add one feature at a time, always choosing the one that gives the largest boost in per-pixel classification rate. At each step we show the overall and average per-class rates for local, MRF and joint geometric/semantic labeling. One observation is that SIFT histograms constitute three or 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 for both datasets, and it effectively makes the non-dilated SIFT histogram re-

Table 8 Comparison of our nearest neighbor classifier to boosted decision trees. While the boosted decision trees constantly perform better on the relatively balanced geometric labels, they have worse per-class rates on semantic labels with heavily skewed label counts.

	Nearest Neighbor		Boosted Decision Trees	
	Semantic	Geometric	Semantic	Geometric
Local labeling	74.1 (30.2)	88.4 (86.1)	75.4 (26.7)	90.2 (88.7)
MRF	76.2 (29.1)	89.0 (86.2)	77.0 (26.4)	90.6 (88.9)
Joint	76.5 (29.3)	89.1 (86.9)	76.9 (26.4)	90.7 (88.9)

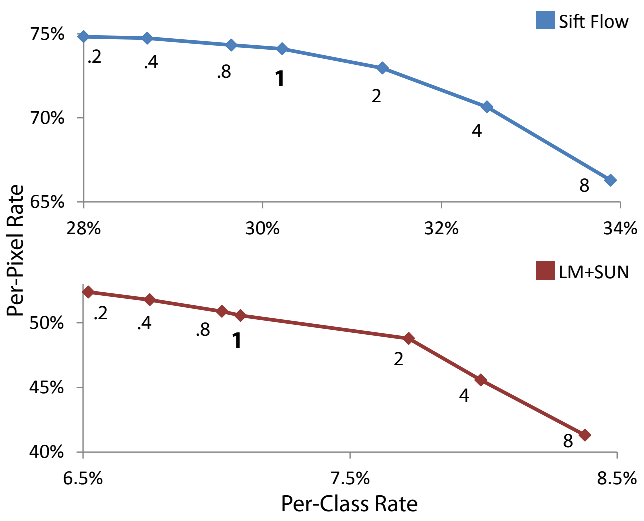

Fig. 9 Results on SIFT Flow and LMSun dataset for local superpixel labeling with different values for the likelihood smoothing constant ϵ (section 2.2, eq. 2). The constant adjusts the tradeoff between average class rate and per-pixel rate.

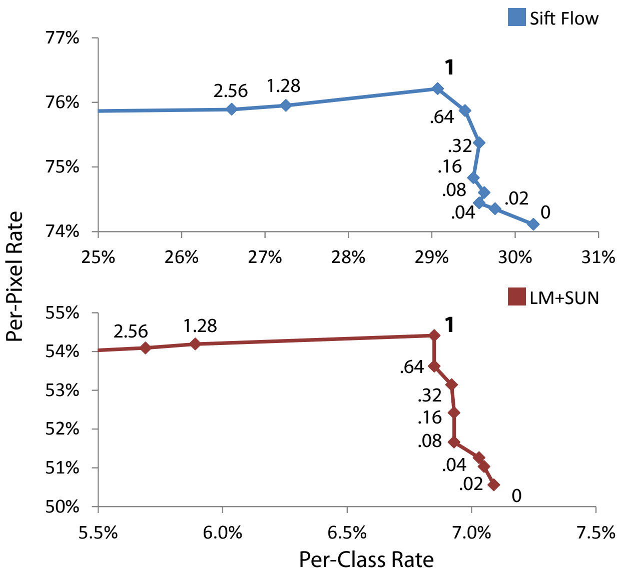

Fig. 10 The effect of our MRF smoothing parameter λ from equation 3.

dundant. After SIFT Histogram Dilated, the order of the features selected for both datasets is quite different, though Top Height and Mean Color show up in the top five in both cases, confirming that location and color cues add important complementary information to texture. Another observation is that while adding features does sometimes hurt performance, it does so minimally. One could combat this effect by learning feature weights as in [5] but this would make our system more prone to overfitting and introduce an offline learning component that we would like to avoid.

It is also interesting to compare the curves for three versions of our system: local superpixel labeling, separate semantic and geometric MRF, and joint semantic/geometric MRF. Consistent with the results reported in Tables 2 and 3, the separate MRF tends to lower the average per-class rate due to over-smoothing and then the joint MRF brings it back up. More surprisingly, Figure 8 reveals that both types of our contextual models have a much greater impact when they are applied on top of a relatively weak local model, i.e., one with fewer features. As we add more local features, the improvements afforded by non-local inference gradually diminish. The important message here is that “local features” and “context” are, to a great degree, interchangeable. For one, many of our features are not truly “local” since they include information from outside the spatial support of the superpixel. But also, it seems that contextual inference can fix relatively few labeling mistakes that cannot just as easily be fixed by a more powerful local model. This is important to keep in mind when critically evaluating contextual models proposed in the literature: a big improvement over a local baseline does not necessarily prove that the proposed form of context is especially powerful – the local features may simply not be strong enough.

Next, Figure 9 show the effect of the smoothing constant ϵ in our likelihood ratio equation (eq. 2). As noted in [5], increasing ϵ biases the classifier toward the rarer classes. In turn, this tends to decrease the overall per-pixel rate and increase the average per-class accuracy. We found the value of $\epsilon = 1$ to achieve a good tradeoff and use it in all other experiments.

Further, we wish to examine how well our nonparametric scheme is doing compared to offline discriminative learning techniques. To this end, we train boosted decision trees (BDT) for all 33 labels in the SIFT Flow dataset the same way we train them for the geometric classes (Section 2.5). We use 100 trees with a depth of 8 for each one-versus-all classifier. Table 8 compares our nearest neighbor (NN) scheme and BDT on both semantic and geometric labels (note that a retrieval set is not used with BDT). On the semantic classes which have a very unbalanced class distribution, the per-pixel rate is higher for BDT but the average per-class rate is lower. On the other hand, BDT easily outperforms the NN classifier for the geometric classes, which have a much more even class distribution. This validates the implementation choice discussed in Section 3.1, namely, using NN for semantic classes and BDT for geometric ones. Indeed, comparing the joint NN/NN and BDT/BDT results in the last line of Table 8 to the hybrid NN/BDT result in Table 2, we can see that the latter one offers the best performance.

Similarly to our likelihood ratio smoothing constant ϵ, the MRF smoothing constant λ (eq. 3) gives us a tradeoff between per-pixel and per-class accuracy, as shown in Figure 10. After $\lambda = 1$ both drop off, so we use that value for both datasets.

3.3.3 Running time

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 Xeon 3.33 GHz six-core processors and 48 GB RAM. Table 9 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 [26], to classify a single query image, the SIFT Flow system required 50 alignment operations that took 31 seconds each, or 25 minutes total without parallelization.

As shown in Figure 11, 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, given a fixed retrieval set size, this time is independent of the overall number of training images. For LM+SUN, the main bottleneck of our system is not superpixel search, but file I/O for loading retrieval set superpixel descriptors from disk. However, it should be possible to overcome this bottleneck with appropriate hardware, parallelization, and/or data structures for fast search. On a more fundamental research level, the dependence of running time on image resolution deserves some attention. The problem of efficiently parsing megapixel images while deriving additional recognition cues from the higher resolution is currently wide open and extremely challenging. Curiously, even as the sizes of datasets used in recognition research have increased dramatically in recent years, the resolution of individual images has not.

Table 9 The average timing in seconds of the different stages in our system (excluding file I/O). While the runtime is significantly longer for the LM+SUN dataset, this is primarily due to the change in image size and not the number of images.

	SIFT Flow	LM+SUN
Training set size	2,488	45,176
Image size	256 × 256	800 × 600
Avg. # superpixels	63.9	178.2
Feature extraction	1.5 ± 0.5	5.2 ± 1.8
Retrieval set search	0.04 ± 0.0	3.5 ± 0.51
Superpixel search	3.75 ± 1.8	13.1 ± 11.2
MRF solver	0.005 ± 0.003	.009 ± .006
Total (excluding features)	4.4 ± 2.3	16.6 ± 11.7

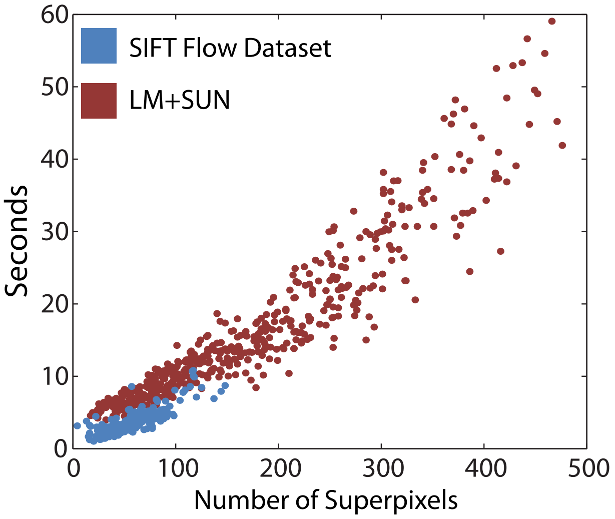

Fig. 11 Query time vs. number of superpixels in the query image. Notice that for small images in the LM+SUN dataset, the retrieval set query time is the only part of the system that takes longer than on the SIFT Flow dataset and thus the total processing time remains similar.

4 Video Parsing

This section presents the extension of our system to video. Video sequences provide richer information which should be useful for better understanding scenes. Intuitively, motion cues can improve object segmentation, and being able to observe the same objects in multiple

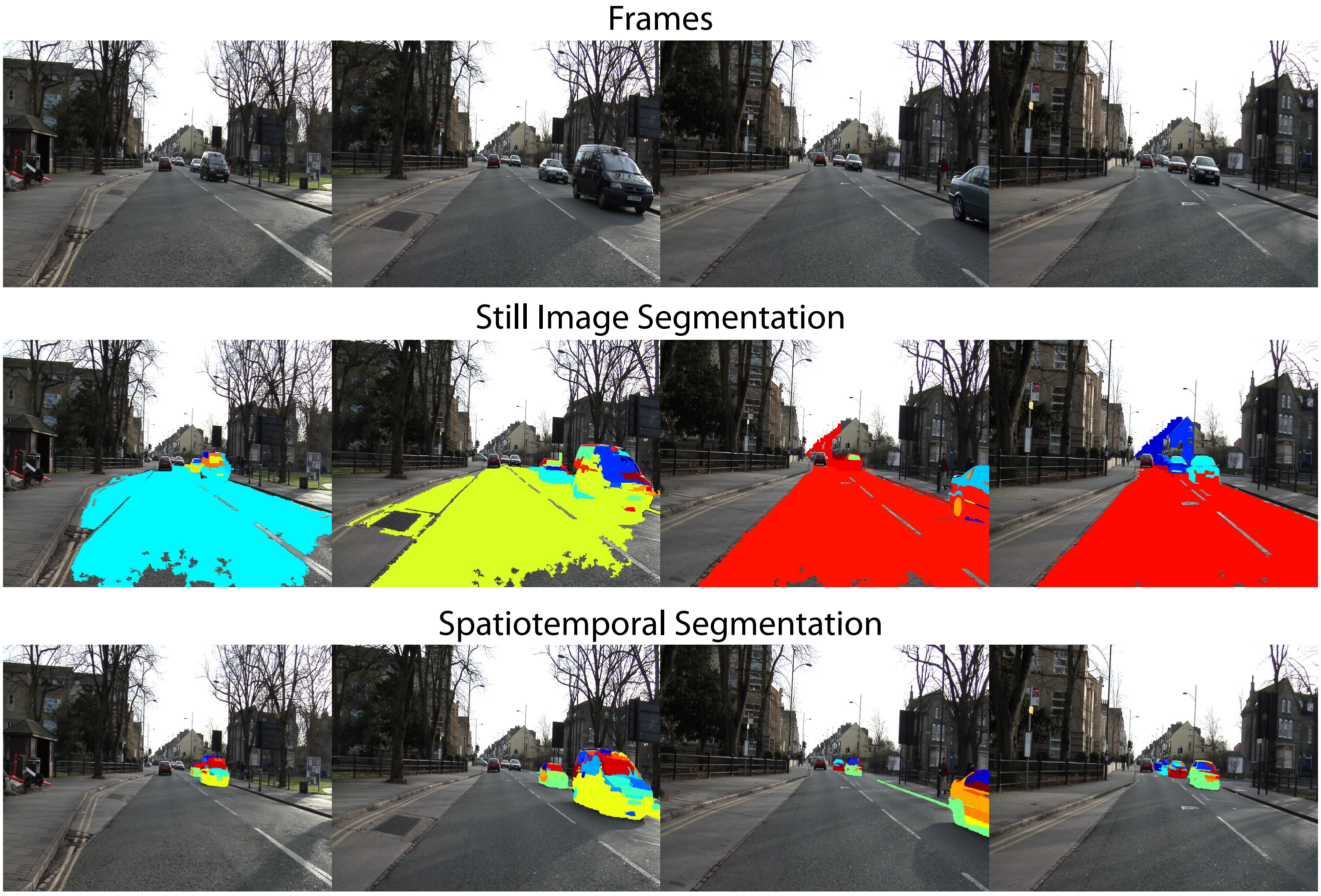

Fig. 12 A comparison of still image segmentation of Felzenszwalb et al. [8] (second row) to the spatiotemporal segmentation of Grundmann et al. [12] (third row). Shown are only the segments required to cover the foreground cars in each frame. The still image segmentation is not able to separate the lower parts of the cars from the road, while the spatiotemporal segmentation does not suffer from the same problem.

frames, possibly at different angles or scales, can help us build a better model of the objects' shape and appearance. On the other hand, the large volume of video data makes parsing very challenging computationally.

Previous approaches have tried a variety of strategies for exploiting the cues contained in video data. Brostow et al. [3], Sturgess et al. [40], and Zhang et al. [47] extract 3D structure (sparse point clouds or dense depth maps) from the video sequences and then use the 3D information as a source of additional features for parsing individual frames. Xiao and Quan [45] run a region-based parsing system on each frame and enforce temporal coherence between regions in adjacent frames as a post-processing step.

We pre-process the video using a spatiotemporal segmentation method [12] that gives 3D regions or *supervoxels* that are spatially coherent within each frame (i.e., have roughly uniform color and optical flow) as well as temporally coherent between frames. The hope is that these regions will contain the same object from frame to frame. We then compute local likelihood scores for possible object labels over each supervoxel, and finally, construct a single graph for each video sequence where each node is a supervoxel and edges connect adjacent supervoxels. We perform inference on this graph using the same MRF formulation as in Section 2.5. Section 4.1 will give details of our video parsing approach, and Section 4.2 will show that this approach significantly improves the performance compared to parsing each frame independently.

4.1 System Description

We wish to take advantage of the motion cues in video without explicitly adding motion or geometric features to our system. We do this by using the hierarchical video segmentation method of Grundmann et al. [12], which Xu and Corso [46] show to be quite effective at capturing the boundaries of objects in video. We run all videos through the segmentation website of [12][5] with default parameters to obtain a hierarchy of segmentation volumes and use the lowest level of the hierarchy as supervoxels.[6] Figure 12 contrasts the outputs of still image and video segmentation, showing that the supervoxel boundaries tend to better adhere to boundaries of objects such as cars.

Once we obtain supervoxels for a video sequence, we need to compute a data term $E_{\text{data}}(v_i, c)$ for each supervoxel v_i and each class label c. In principle, this could be done directly by extracting spatiotemporal features from each v_i, but to simplify the extension of our system from still images, we have chosen to combine scores computed over 2D time slices of v_i. Specifically, given a class c and the slice of supervoxel v_i in the jth frame, denoted s_i^j, we compute a log likelihood ratio score $L(s_i^j, c)$. This can be done with either our NN scheme (eq. (1)) or with BDTs (Section 3.1), though in the experiments of the next section we use only BDTs. For combining the per-frame scores, we have tried a number of approaches and found the following heuristic to give the best performance:

$$E_{\text{data}}(v_i, c) = -\max_j [w_i^j \sigma(L(s_i^j, c))], \tag{7}$$

where w_i^j is the relative size of s_i^j and $\sigma(\cdot)$ is a normalizing sigmoid function as in eq. (4). In other words, we weight each per-frame score by the size of the region in that frame (the idea being that frames in which the supervoxel is larger give better evidence about its class identity) and take the maximum of the weighted scores over all the frames in which the supervoxel appears (intuitively, the frame in which the weighted score is highest is the one in which we got the best "look" at the object and were the most confident about its identity).

Finally, we construct an MRF for the entire video sequence where nodes represent supervoxels and edges connect pairs of supervoxels that are spatially adjacent in at least one frame. We define the edge energy term in the same way as in the 2D case, using eq. (5). We do this for both semantic and geometric classes and solve for them simultaneously using the same joint formulation as in eq. (6). For the video sequences in our experiments, which range from 1,500 to 4,000 frames, we typically obtain graphs of 10,000 to 30,000 nodes, which are very tractable.

Table 10 CamVid dataset results. (a) Still image segmentation baseline. (b) Results with spatiotemporal segmentation (see text). (c) Competing state-of-the-art approaches. As before, per-pixel classification rate is followed by the average per-class rate in parentheses.

		Semantic	Geometric
(a)	Still Image Parsing		
	Local Labeling	76.9 (44.3)	91.6 (92.0)
	MRF	77.4 (43.5)	91.6 (91.9)
	Joint	77.6 (43.8)	91.7 (92.1)
(b)	Spatiotemporal Parsing		
	Temporally Incoherent	82.6 (51.2)	94.6 (94.8)
	Temporally Coherent	82.6 (51.3)	94.2 (94.8)
	MRF	83.0 (51.0)	94.2 (94.4)
	Joint	83.3 (51.2)	94.2 (94.7)
(c)	Brostow et al. [3]	69.1 (53.0)	
	Sturgess et al. [40]	83.8 (59.2)	
	Zhang et al. [47]	82.1 (55.4)	
	Ladicky et al. [23]	83.8 (62.5)	

4.2 Results

We test our video segmentation on the standard CamVid dataset [3], which consists of daytime and dusk videos taken from a car driving through Cambridge, England. There are a total of five video sequences. We follow the training/test split of [3], with two daytime and one dusk sequence used for training, and one daytime and one dusk sequence used for testing. The sequences are densely labeled at one frame per second with 11 class labels: Building, Tree, Sky, Car, Sign-Symbol, Road, Pedestrian, Fence, Column-Pole, Sidewalk, and Bicyclist. There are a total of 701 labeled frames in the dataset with 468 used for training and 233 for testing. Note that while we evaluate the accuracy of our output on only the labeled testing frames, we do obtain dense labels for all frames in the test video.

Table 10(a) shows baseline performance using our still image parsing approach that segments and labels each frame independently. Table 10(b) shows results with spatiotemporal segmentation used in two different ways. The first variant, "temporally incoherent," just uses the segmentation to generate the regions in each frame; each frame is still parsed independently, and regions belonging to the same supervoxel are not required to have the same label from frame to frame. The second variant, "temporally coherent," combines the per-frame likelihood scores as described in Section 4.1 to assign a single label to each supervoxel. Both methods give a significant improvement over still image parsing. Note that even though the temporally coherent method has a similar accuracy to the incoherent one, the output video is much more visually pleasing in the former case, since the labeling "flickers" much less over time (see Figure 15 for examples).

[5] http://videosegmentation.com/

[6] Since the videos were taken from a forward-moving camera, we have found the segmentation results to be better if we run the videos through the system backwards.

Table 11 Per-class performance on the CamVid [3] dataset.

	Building	Tree	Sky	Car	Sign-Symbol	Road	Pedestrian	Fence	Column-Pole	Sidewalk	Bicyclist	Average	Global
Still image parsing (joint sem./geom.)	84.8	65.1	94.7	47.5	24.6	96.2	8.3	9.1	3.4	43.7	3.9	43.8	78.6
Spatiotemporal parsing (joint sem./geom.)	**87.0**	67.1	96.9	62.7	30.1	95.9	14.7	17.9	1.7	70.0	19.4	51.2	83.3
Brostow et al. [3]	46.2	61.9	89.7	68.6	42.9	89.5	**53.6**	46.6	0.7	60.5	22.5	53.0	69.1
Sturgess et al. [40]	84.5	72.6	**97.5**	72.7	34.1	95.3	34.2	45.7	8.1	77.6	28.5	59.2	**83.8**
Zhang et al. [47]	85.3	57.3	95.4	69.2	**46.5**	**98.5**	23.8	44.3	**22.0**	38.1	28.7	55.4	82.1
Ladicky et al. [23]	81.5	**76.6**	96.2	**78.7**	40.2	93.9	43.0	**47.6**	14.3	**81.5**	**33.9**	**62.5**	**83.8**

The sixth and seventh rows of Table 10(b) show the performance of the temporally coherent setup following contextual MRF smoothing and joint semantic/geometric inference. Somewhat disappointingly, both versions of the MRF give a very minimal improvement. This is likely due to a number of factors. First, MRF energy minimization on the spatiotemporal graph appears to be a harder problem, and the solutions tend to show a much greater tendency to oversmooth. Second, we gain a big improvement in object boundaries by incorporating motion cues into the segmentation, and this is likely diminishing the subsequent power of the MRF. Recall that in Section 3.3 we have seen a similar effect: as we made the local appearance model more powerful by adding features, the improvement afforded by the MRF diminished (Figure 8). Finally, joint semantic/geometric inference introduces very few new constraints, since the CamVid dataset has only three non-vertical classes (sky, road, and sidewalk).

For reference, Table 10(c) shows the performance of recent state-of-the-art methods on the CamVid dataset. Our system beats [3] and comes close to [23,40,47]. Table 11 gives a more detailed class-by-class comparison. By comparing the first two lines of the table, we can see that spatiotemporal segmentation gives us the biggest improvements on the smaller moving object classes such as car, pedestrian, and bicyclist. In absolute terms, however, we do not do well on these classes, just as we did not do well on them in our still image datasets. Interestingly, spatiotemporal segmentation also gives us a significant boost on "sidewalk," which happens to be similar to the effect we got by using ground truth segmentation on the SIFT Flow dataset (Figure 7). Thus, it is plausible that the video segmentation gets us closer to the true object boundaries.

While we do not outperform current state-of-the-art methods on the CamVid dataset, our results are encouraging as our system is the most simple and scalable. Note that we use the motion information in video only to improve our segmentation, not to change our features. By contrast, [3,40,47] use features derived from 3D point clouds or depth maps, while [23] incorporate sliding window object detectors. Overall, our experiments on video confirm the flexibility and broad applicability of our image parsing framework, and give us additional insights into its strengths and weaknesses that complement our findings on still image datasets.

5 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 underlying feature representation, based on multiple appearance descriptors computed over segmentation regions, is simple and allows new features to be easily incorporated. We also use efficient MRF optimization to capture label co-occurrence context, and to jointly label regions with semantic and geometric classes.

We have demonstrated state-of-the-art results on the SIFT Flow and LM+SUN datasets with a nonparametric version of our system based on a two-stage approach (global retrieval set matching followed by superpixel matching). This framework does not need any training, except for computation of basic statistics such as label co-occurrence probabilities, and it relies on just a few constants that are kept fixed for all datasets. In principle, it is applicable to "open universe" datasets where the set of training examples and target classes may evolve over time. In particular, our results on the LM+SUN dataset, which has 45,676 images and 232 labels, constitute an important baseline for future approaches. To our knowledge, it is currently the largest dense per-pixel image parsing dataset and, unlike most other general-purpose image parsing benchmarks, it includes both outdoor and indoor images. As we have shown, the latter pose severe recognition challenges, and deserve more study in the future.

Besides the nonparametric "open universe" regime, our system has the flexibility to operate with offline pre-trained classifiers, such as boosted decision trees. The use of these may be preferable for static datasets with smaller numbers of classes and a more balanced class distribution (see the conference version of this paper [41] for additional results on the small-scale Geometric Context [20] and Stanford Background [11] datasets).

Finally, we have demonstrated an extension of our system to video. This extension segments the video into spatiotemporal "supervoxels" and uses a simple heuristic to combine local appearance cues across frames. The resulting approach does not exploit all the motion information that is potentially available in video (in particular, it does not attempt to extract 3D geometry), but it still affords a big improvement over incoherent frame-by-frame parsing.

Through the extensive analysis of Section 3.3, we have identified two major limitations of our system. First, the scene matching step for obtaining the retrieval set suffers from an inability of low-level global features such as GIST to retrieve semantically similar scenes, resulting in incoherent interpretations (e.g., indoor and outdoor class labels mixed together). We plan to investigate supervised feature learning methods for improving the semantic consistency of retrieval sets. Second, our reliance on bottom-up segmentation really hurts our performance on "thing" classes. Traditionally, such classes are handled using sliding window detectors, and there exists work (e.g., [23]) attempting to incorporate such detectors into region-based parsing. We are interested in exploring the idea of per-exemplar detectors [29] to complement our superpixel-based approach in a manner that still allows for lazy learning in the "open universe" mode.

Acknowledgements This research was supported in part by NSF grants IIS-0845629 and IIS-0916829, DARPA Computer Science Study Group, Microsoft Research Faculty Fellowship, and Xerox.

References

1. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(9), 1124–37 (2004)
2. Boykov, Y., Veksler, O., Zabih, R.: Efficient Approximate Energy Minimization via Graph Cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(12), 1222–1239 (2001)
3. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and Recognition using Structure from Motion Point Clouds. In: Proceedings European Conference Computer Vision, pp. 1–15 (2008)
4. Divvala, S., Hoiem, D., Hays, J., Efros, A., Hebert, M.: An empirical study of context in object detection. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition, pp. 1271–1278 (2009)
5. Eigen, D., Fergus, R.: Nonparametric Image Parsing using Adaptive Neighbor Sets. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition (2012)
6. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Scene Parsing with Multiscale Feature Learning, Purity Trees, and Optimal Covers. Arxiv preprint arXiv: (2012)
7. Felzenszwalb, P., Mcallester, D., Ramanan, D.: A Discriminatively Trained , Multiscale , Deformable Part Model. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition (2008)
8. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient Graph-Based Image Segmentation. International Journal of Computer Vision **2**(2), 1–26 (2004)
9. Galleguillos, C., Belongie, S.: Context based object categorization: A critical survey. Computer Vision and Image Understanding **114**(6), 712–722 (2010)
10. Galleguillos, C., Mcfee, B., Belongie, S., Lanckriet, G.: Multi-Class Object Localization by Combining Local Contextual Interactions. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition (2010)
11. Gould, S., Fulton, R., Koller, D.: Decomposing a Scene into Geometric and Semantically Consistent Regions. In: Proceedings IEEE International Conference Computer Vision (2009)
12. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient Hierarchical Graph-Based Video Segmentation. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition (2010)
13. Gu, C., Lim, J.J., Arbel, P., Malik, J.: Recognition using Regions. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition (2009)
14. Gupta, A., Satkin, S., Efros, A.A., Hebert, M.: From 3D Scene Geometry to Human Workspace. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition (2011)
15. Hays, J., Efros, A.A.: IM 2 GPS : estimating geographic information from a single image. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition, vol. 05 (2008)
16. He, X., Zemel, R.S., Carreira-Perpinan, M.A.: Multiscale Conditional Random Fields for Image Labeling. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition (2004)
17. Hedau, V., Hoiem, D.: Thinking inside the box: Using appearance models and context based on room geometry. In: Proceedings European Conference Computer Vision, pp. 1–14 (2010)
18. Hedau, V., Hoiem, D., Forsyth, D.: Recovering the Spatial Layout of Cluttered Rooms. In: Proceedings IEEE International Conference Computer Vision (2009)
19. Heitz, G., Koller, D.: Learning Spatial Context : Using Stuff to Find Things. In: Proceedings European Conference Computer Vision, pp. 1–14 (2008)
20. Hoiem, D., Efros, A.A., Hebert, M.: Recovering Surface Layout from an Image. International Journal of Computer Vision **75**(1) (2007)
21. Janoch, A., Karayev, S., Jia, Y., Barron, J.T., Fritz, M., Saenko, K., Darrell, T.: A Category-Level 3-D Object Dataset : Putting the Kinect to Work. In: ICCV Workshop (2011)

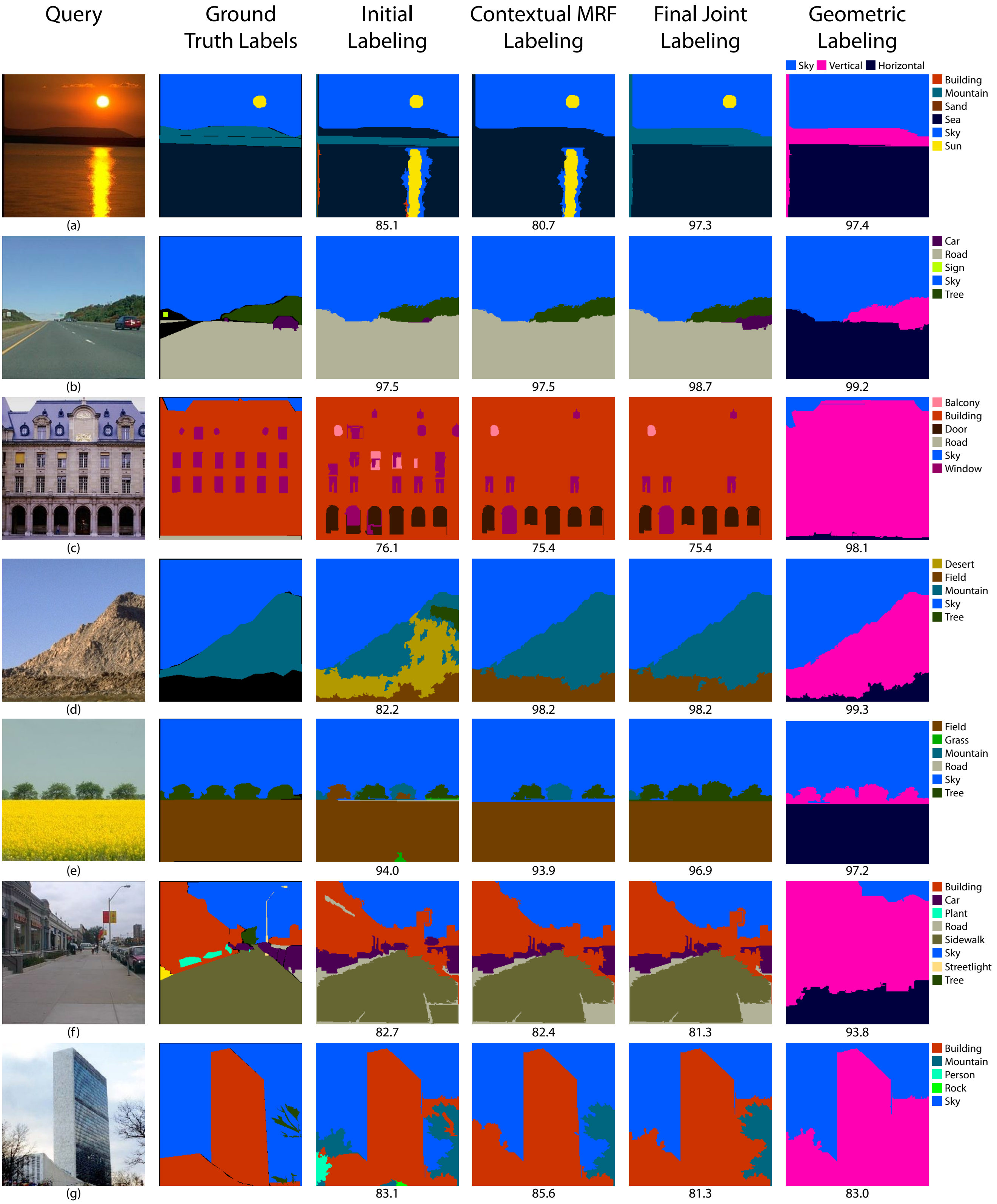

Fig. 13 Example results from the SIFT Flow test set (best viewed in color). The number under each result image is the percentage of pixels labelled correctly. In (a), joint geometric/semantic inference removes the spurious classification of the sun's reflection in the water. In (c), we find some windows (some of which are smoothed away by the MRF) and plausibly classify the arches at the bottom of the building as doors. In (d), "field" and "desert" never co-occur so "field" wins and "desert" is removed. In (f), sidewalk is successfully recovered. For complete output on this dataset, see `http://www.cs.unc.edu/SuperParsing`.

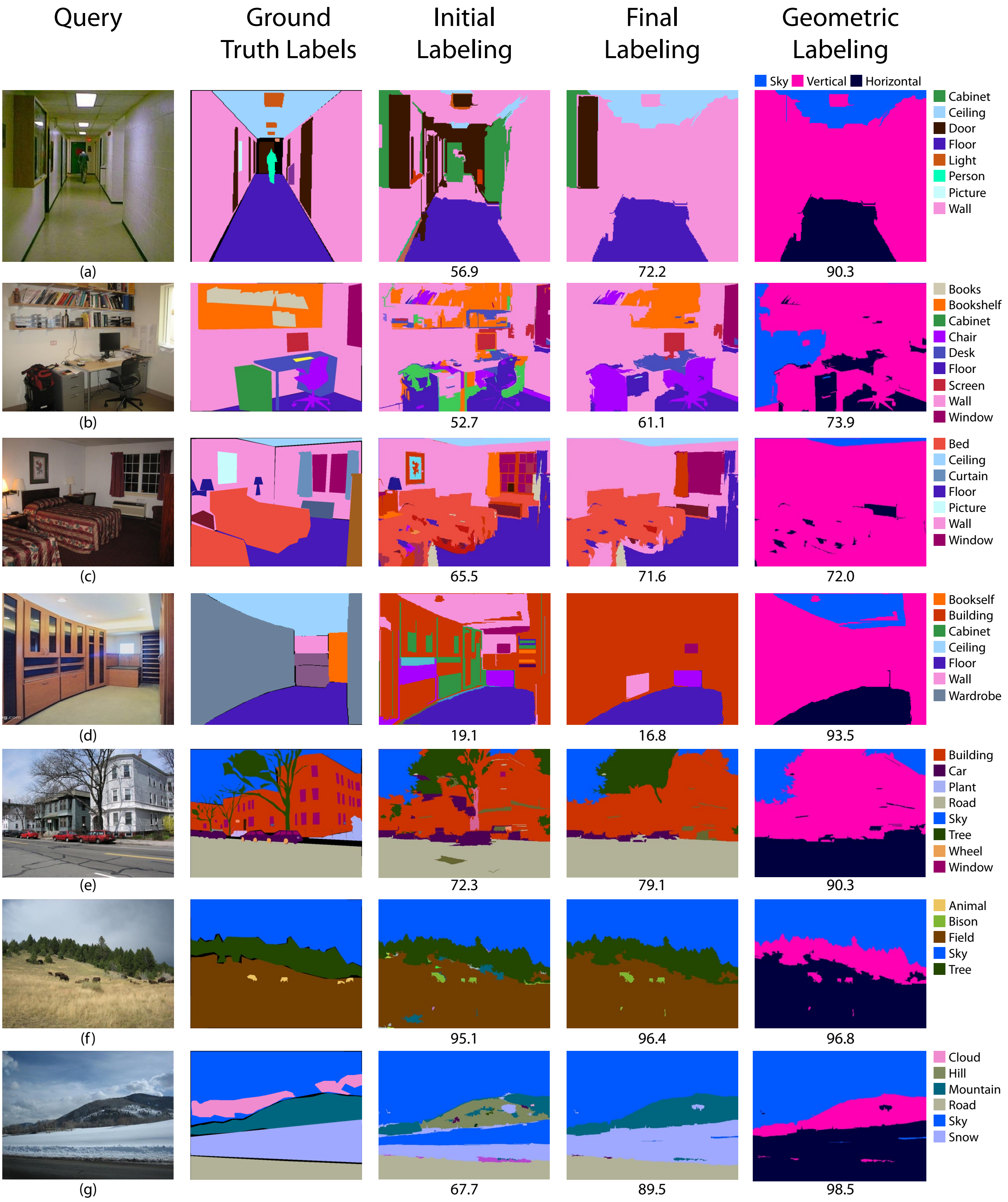

Fig. 14 Example results from the LM+SUN test set (best viewed in color). For indoor images the geometric label of "sky" corresponds to the semantic label of "ceiling." Examples (a-c) show the best performance we can achieve on indoor scenes: we are able to get wall, ceiling, floor and even get some chairs, bookshelves, beds and desks. Example (d) shows how a retrieval set of mixed indoor and outdoor images can produce incorrect labels. Examples (e-g) show the varity of images our system can work on. In (f), we even correct the ground truth label "animal" to the more specific class "bison."

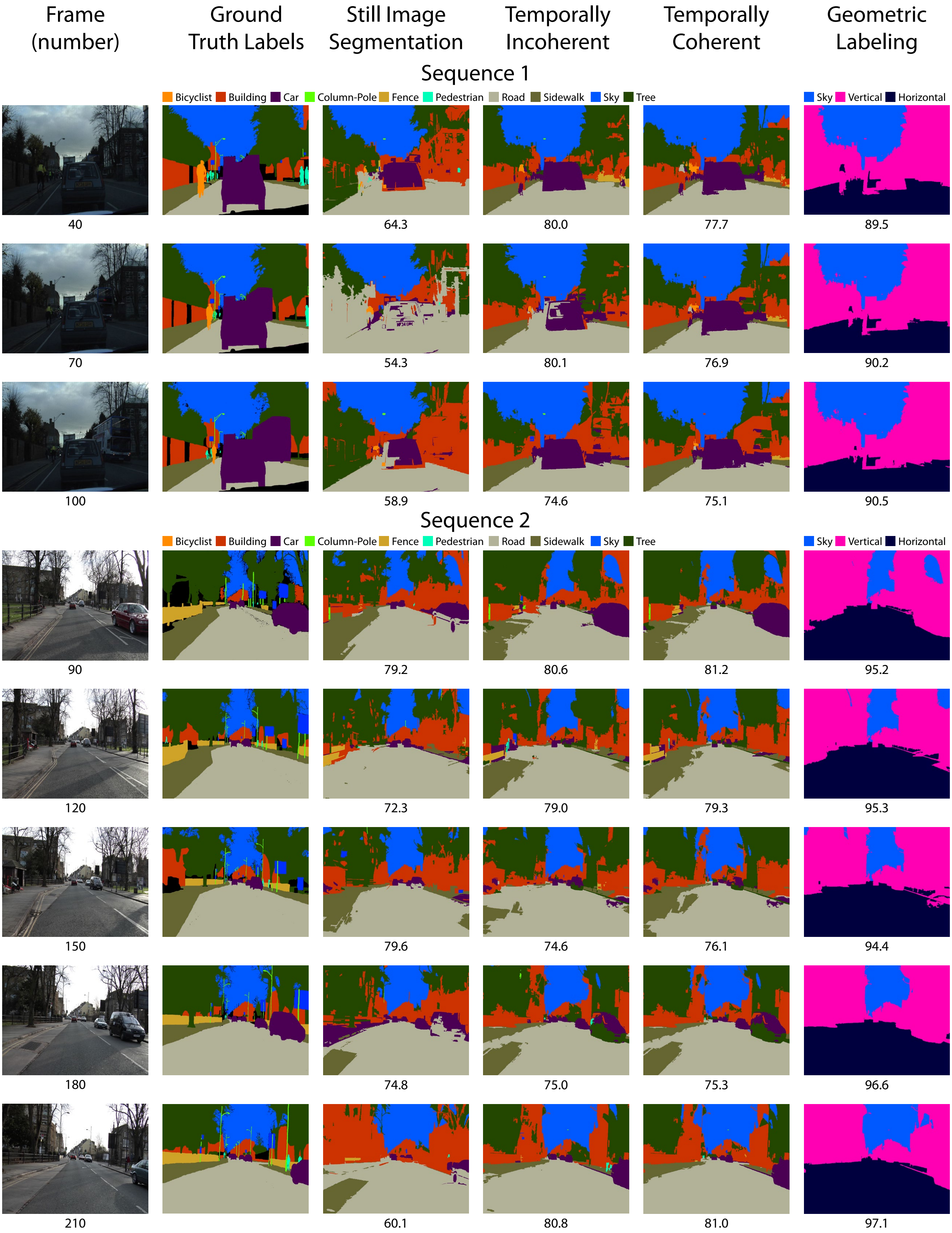

Fig. 15 Example results from the CamVid test set (best viewed in color). Still image parsing (third column) is very unstable, with both the superpixels and their inferred labels changing incoherently from one frame to the next. Temporally incoherent parsing (fourth column) uses spatiotemporal segmentation, which corrects most of these issues but can still be inconsistent in time as shown in frames 70 and 100. The final system (fifth column) has temporally consistent segments and labels.

22. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(2), 147–59 (2004)
23. Ladicky, L., Sturgess, P., Alahari, K., Russell, C., Torr, P.H.S.: What , Where and How Many ? Combining Object Detectors and CRFs. In: Proceedings European Conference Computer Vision, pp. 424–437 (2010)
24. Lai, K., Bo, L., Ren, X., Fox, D.: A Scalable Tree-based Approach for Joint Object and Pose Recognition. In: Artificial Intelligence (2011)
25. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition, vol. 2 (2006)
26. Liu, C., Yuen, J., Torralba, A.: Nonparametric Scene Parsing via Label Transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(12), 2368–2382 (2011)
27. Liu, C., Yuen, J., Torralba, A.: SIFT flow: dense correspondence across scenes and its applications. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(5), 978–94 (2011)
28. Malisiewicz, T., Efros, A.a.: Recognition by association via learning per-exemplar distances. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition, pp. 1–8 (2008)
29. Malisiewicz, T., Efros, A.A.: Ensemble of Exemplar-SVMs for Object Detection and Beyond. In: Proceedings IEEE International Conference Computer Vision, pp. 89–96 (2011)
30. Nowozin, S., Carsten Rother, Bagon, S., Sharp, T., Yao, B., Kohli, P.: Decision Tree Fields. In: Proceedings IEEE International Conference Computer Vision, pp. 1668–1675 (2011)
31. 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**, 23–36 (2006)
32. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in Context. In: Proceedings IEEE International Conference Computer Vision, pp. 1–8 (2007)
33. Ren, X., Malik, J.: Learning a classification model for segmentation. In: Proceedings IEEE International Conference Computer Vision (2003)
34. Russell, B.C., Torralba, A., Liu, C., Fergus, R., Freeman, W.T.: Object Recognition by Scene Alignment. In: Neural Information Processing Systems Foundation (2007)
35. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe : a database and web-based tool for image annotation. International Journal of Computer Vision **77**(1-3), 157–173 (2008)
36. Shotton, J., Johnson, M., Cipolla, R.: Semantic Texton Forests for Image Categorization and Segmentation. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition (2008)
37. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Texton-Boost : Joint Appearance , Shape and Context Modeling for Multi-Class Object Recognition and Segmentation. In: Proceedings European Conference Computer Vision, pp. 1–14 (2006)
38. Silberman, N., Fergus, R.: Indoor Scene Segmentation using a Structured Light Sensor. In: Proceedings IEEE International Conference Computer Vision Workshop (2011)
39. Socher, R., Lin, C.C.Y., Ng, A.Y., Manning, C.D.: Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In: Proceedings of the International Conference on Machine Learning (2011)
40. Sturgess, P., Alahari, K., Ladicky, L., Torr, P.H.S.: Combining Appearance and Structure from Motion Features for Road Scene Understanding. British Machine Vision Conference pp. 1–11 (2009)
41. Tighe, J., Lazebnik, S.: SuperParsing: Scalable Nonparametric Image Parsing with Superpixels. In: Proceedings European Conference Computer Vision (2010)
42. Tighe, J., Lazebnik, S.: Understanding Scenes on Many Levels. In: Proceedings IEEE International Conference Computer Vision, pp. 335–342 (2011)
43. Torralba, A., Fergus, R., Freeman, W.T.: 80 Million Tiny Images: a Large Data Set for Nonparametric Object and Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(11), 1958–1970 (2008)
44. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: SUN database: Large-scale scene recognition from abbey to zoo. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition, pp. 3485–3492 (2010)
45. Xiao, J., Quan, L.: Multiple View Semantic Segmentation for Street View Images. In: Proceedings IEEE International Conference Computer Vision (2009)
46. Xu, C., Corso, J.J.: Evaluation of Super-Voxel Methods for Early Video Processing. Proceedings IEEE Conference Computer Vision and Pattern Recognition (2012)
47. Zhang, C., Wang, L., Yang, R.: Semantic Segmentation of Urban Scenes Using Dense Depth Maps. In: Proceedings European Conference Computer Vision, pp. 708–721 (2010)