Nonparametric Scene Parsing via Label Transfer

Ce Liu, *Member*, *IEEE*, Jenny Yuen, *Student Member*, *IEEE*, and Antonio Torralba, *Member*, *IEEE*

Abstract—While there has been a lot of recent work on object recognition and image understanding, the focus has been on carefully establishing mathematical models for images, scenes, and objects. In this paper, we propose a novel, nonparametric approach for object recognition and scene parsing using a new technology we name *label transfer*. For an input image, our system first retrieves its nearest neighbors from a large database containing fully annotated images. Then, the system establishes dense correspondences between the input image and each of the nearest neighbors using the dense SIFT flow algorithm [28], which aligns two images based on local image structures. Finally, based on the dense scene correspondences obtained from SIFT flow, our system warps the existing annotations and integrates multiple cues in a Markov random field framework to segment and recognize the query image. Promising experimental results have been achieved by our nonparametric scene parsing system on challenging databases. Compared to existing object recognition approaches that require training classifiers or appearance models for each object category, our system is easy to implement, has few parameters, and embeds contextual information naturally in the retrieval/alignment procedure.

Index Terms—Object recognition, scene parsing, label transfer, SIFT flow, Markov random fields.

1 Introduction

Scene parsing, or recognizing and segmenting objects in an image, is one of the core problems of computer vision. Traditional approaches to object recognition begin by specifying an object model, such as template matching [8], [49], constellations [13], [15], bags of features [19], [24], [44], [45], or shape models [2], [3], [14], etc. These approaches typically work with a fixed number of object categories and require training generative or discriminative models for each category from training data. In the parsing stage, these systems try to align the learned models to the input image and associate object category labels with pixels, windows, edges, or other image representations. Recently, context information has also been carefully modeled to capture the relationship between objects at the semantic level [20], [22]. Encouraging progress has been made by these models on a variety of object recognition and scene parsing tasks.

However, these learning-based methods do not, in general, scale well with the number of object categories. For example, to include more object categories in an existing system, we need to train new models for the new categories and, typically, adjust system parameters. Training can be a tedious job if we want to include thousands of object categories in a scene parsing system. In addition, the complexity of contextual relationships among objects also increases rapidly as the quantity of object categories expands.

Recently, the emergence of large databases of images has opened the door to a new family of methods in computer vision. Large database-driven approaches have shown the potential for nonparametric methods in several applications. Instead of training sophisticated parametric models, these methods try to reduce the inference problem for an unknown image to that of matching to an existing set of annotated images. In [41], the authors estimate the pose of a human, relying on 0.5 million training examples. In [21], the proposed algorithm can fill holes on an input image by introducing elements that are likely to be semantically correct through searching a large image database. In [38], a system is designed to infer the possible object categories that may appear in an image by retrieving similar images in a large database [39]. Moreover, the authors in [47] showed that with a database of 80 million images, even simple SSD match can give semantically meaningful parsing for 32×32 images.

In this paper, we propose a novel, nonparametric scene parsing system to transfer the labels from existing samples in a large database to annotate an image, as illustrated in Fig. 1. For a query image (Fig. 1a), our system first retrieves the top matches in a large, annotated image database using a combination of GIST matching [34] and SIFT flow [29]. Since these top matches are labeled, we transfer the annotation (Fig. 1c) of the top matches to the query image and obtain the scene parsing result in (Fig. 1d). For comparison, the ground-truth user annotation of the query is displayed in (Fig. 1e). Our system is able to generate promising scene parsing results if images from the same scene type as the query are retrieved in the annotated database.

However, it is nontrivial to build an efficient and reliable scene parsing system using dense scene alignment. To account for the multiple annotation suggestions from the top matches, a Markov random field model is used to merge multiple cues (e.g., likelihood, prior, and spatial

- *C. Liu is with Microsoft Research New England, One Memorial Drive, Cambridge, MA 02142 and also with the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, Cambridge, MA 02139. E-mail: celiu@microsoft.com.*
- *J. Yuen and A. Torralba are with the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, Cambridge, MA 02139. E-mail: {jenny, torralba}@csail.mit.edu.*

Manuscript received 28 Aug. 2010; revised 4 Apr. 2011; accepted 11 May 2011; published online 22 June 2011.
Recommended for acceptance by S.B. Kang and I. Essa.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMISI-2010-08-0663.
Digital Object Identifier no. 10.1109/TPAMI.2011.131.

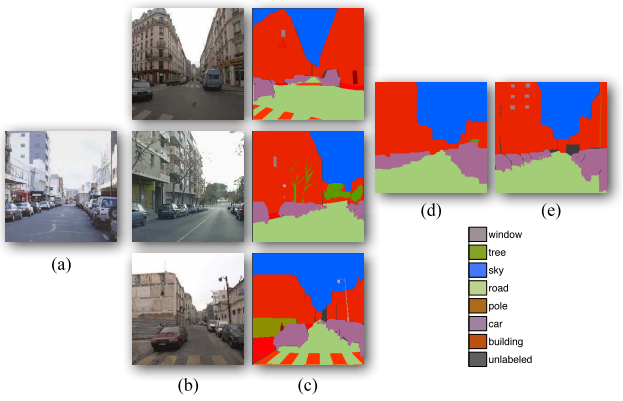


Fig. 1. For a query image (a), our system finds the top matches (b) (three are shown here) using scene retrieval and a SIFT flow matching algorithm [28], [29]. The annotations of the top matches (c) are transferred and integrated to parse the input image, as shown in (d). For comparison, the ground-truth user annotation of (a) is shown in (e).

smoothness) into a robust annotation. Promising experimental results are achieved on images from the LabelMe database [39].

Our goal is to explore the performance of scene parsing through the transfer of labels from existing annotated images, rather than building a comprehensive object recognition system. We show, however, that the performance of our system outperforms existing approaches [8], [43] on our databases. Our code and databases can be downloaded at http://people.csail.mit.edu/celiu/LabelTransfer/.

This paper is organized as follows: In Section 2, we briefly survey the object recognition and detection literature. After giving a system overview in Section 3, we describe, in detail, each component of our system in Section 4. Thorough experiments are conducted in Section 5 for evaluation, and in-depth discussion is provided in Section 6. We conclude our paper in Section 7.

2 Related Work

Object recognition is an area of research that has greatly evolved over the last decade. Many works focusing on single-class modeling, such as faces [11], [48], [49], digits, characters, and pedestrians [2], [8], [25], have been proven successful and, in some cases, the problems have been mostly deemed as solved. Recent efforts have turned to mainly focusing in the area of multiclass object recognition. In creating an object detection system, there are many basic building blocks to take into account; feature description and extraction is the first stepping stone. Examples of descriptors include gradient-based features such as SIFT [30] and HOG [8], shape context [2], and patch statistics [42]. Consequently, selected feature descriptors can be further applied to images in either a sparse [2], [16], [19] manner by selecting the top key points containing the highest response from the feature descriptor, or densely by observing feature statistics across the image [40], [51].

Sparse key point representations are often matched among pairs of images. Since the generic problem of matching two sets of key points is NP-hard, approximation algorithms have been developed to efficiently compute key point matches minimizing error rates (e.g., the pyramid match kernel [19] and vocabulary trees [32], [33]). On the other hand, dense representations have been handled by modeling distributions of the visual features over neighborhoods in the image or in the image as a whole [24], [40], [51]. We chose the dense representation in the paper due to recent advances in dense image matching [28], [29].

At a higher level, we can also distinguish two types of object recognition approaches: *parametric* approaches that consist of learning generative/discriminative models, and *nonparametric* approaches that rely on image retrieval and matching. In the parametric family we can find numerous template-matching methods, where classifiers are trained to discriminate between an image window containing an object or a background [8]. However, these methods assume that objects are mostly rigid and are susceptible to little or no deformation. To account for articulated objects, constellation models have been designed to model objects as ensembles of parts [13], [14], [15], [50], considering spatial information [7], depth ordering information [53], and multiresolution modes [35]. Recently, a new idea of integrating humans in the loop via crowd sourcing for visual recognition of specialized classes such as plants and animal species has emerged [5]; this method integrates the description of an object in less than 20 discriminative questions that humans can answer after visually inspecting the image.

In the realm of nonparametric methods we find systems such as Video Google [44], a system that allows users to specify a visual query of an object in a video and subsequently retrieve instances of the same object across the movie. Another nonparametric system is the one in [38], where a previously unknown query image is matched against a densely labeled image database; the nearest neighbors are used to build a label probability map for the query, which is further used to prune out object detectors of classes that are unlikely to take place in the image. Nonparametric methods have also been widely used in web data to retrieve similar images. For example, in [17], a customized distance function is used at a retrieval stage to compute the distance between a query image and images in the training set, which subsequently cast votes to infer the object class of the query. In the same spirit, our nonparametric label transfer system avoids modeling object appearances explicitly as our system parses a query image using the annotation of similar images in a training database and dense image correspondences.

Recently, several works have also considered contextual information in object detections to clean and reinforce individual results. Among contextual cues that have been used are object-level co-occurrences, spatial relationships [6], [9], [18], [31], [36], and 3D scene layout [23]. For a more detailed and comprehensive study and benchmark of contextual works, we refer to [10]. Instead of explicitly modeling context, our model incorporates context implicitly as object co-occurrences and spatial relationships are retained in label transfer.

An earlier version of our work appeared at [27]; in this paper, we will explore the label-transfer framework in-depth with more thorough experiments and insights. Other recent papers have also introduced similar ideas. For instance, in [46], oversegmentation is performed to the query image and segment-based classifiers trained on the nearest neighbors are applied to recognize each segment. In [37], scene boundaries are discovered by the common edges shared by nearest neighbors.

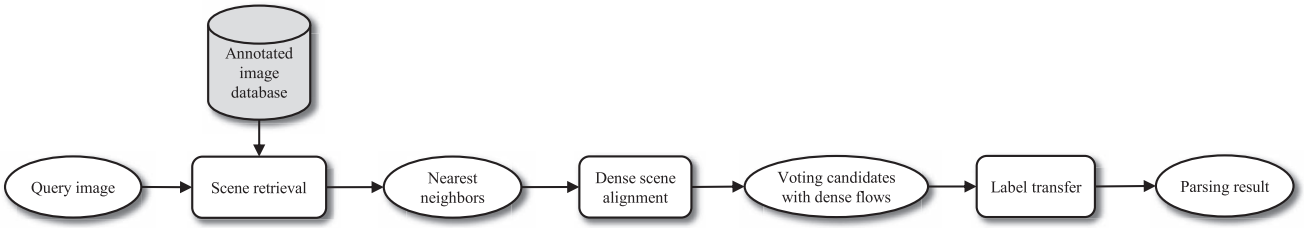


Fig. 2. System pipeline. There are three key algorithmic components (rectangles) in our system: *scene retrieval*, *dense scene alignment*, and *label transfer*. The ovals denote data representations.

3 System Overview

The core idea of our nonparametric scene parsing system is recognition-by-matching. To parse an input image, we match the visual objects in the input image to the images in a database. If images in the database are annotated with object category labels and if the matching is semantically meaningful, i.e., *building* corresponds to *building*, *window* to *window*, *person* to *person*, then we can simply transfer the labels of the images in the database to parse the input. Nevertheless, we need to deal with many practical issues in order to build a reliable system.

Fig. 2 shows the pipeline of our system, which consists of the following three algorithmic modules:

- **Scene retrieval**: Given a query image, use *scene retrieval* techniques to find a set of *nearest neighbors* that share similar scene configuration (including objects and their relationships) with the query.
- **Dense scene alignment**: Establish *dense scene correspondence* between the query image and each of the retrieved nearest neighbors. Choose the nearest neighbors with the top matching scores as *voting candidates*.
- **Label transfer**: Warp the annotations from the voting candidates to the query image according to estimated dense correspondence. Reconcile multiple labeling and impose spatial smoothness under a Markov random field (MRF) model.

Although we are going to choose concrete algorithms for each module in this paper, any algorithm that fits to the module can be plugged into our nonparametric scene parsing system. For example, we use SIFT flow for dense scene alignment, but it would also suffice to use sparse feature matching and then propagate sparse correspondences to produce dense counterparts.

A key component of our system is a large, dense, and annotated image database.[1] In this paper, we use two sets of databases, both annotated using the LabelMe online annotation tool [39], to build and evaluate our system. The first is the LabelMe Outdoor (LMO) database [27], containing 2,688 fully annotated images, most of which are outdoor scenes including street, beach, mountains, fields, and buildings. The second is the SUN database [52], containing 9,566 fully annotated images, covering both indoor and outdoor scenes; in fact, LMO is a subset of SUN.

1. Other scene parsing and image understanding systems also require such a database. We do not require more than others.

We use the LMO database to explore our system in-depth, and also report the results on the SUN database.

Before jumping into the details of our system, it is helpful to look at the statistics of the LMO database. The 2,688 images in LMO are randomly split into 2,488 for training and 200 for testing. We chose the top 33 object categories with the most labeled pixels. The pixels that are not labeled, or labeled as other object categories, are treated as the 34th category: "unlabeled." The per pixel frequency count of these object categories in the training set is shown at the top of Fig. 3. The color of each bar is the average RGB value of the corresponding object category from the training data with saturation and brightness boosted for visualization purposes. The top 10 object categories are *sky*, *building*,

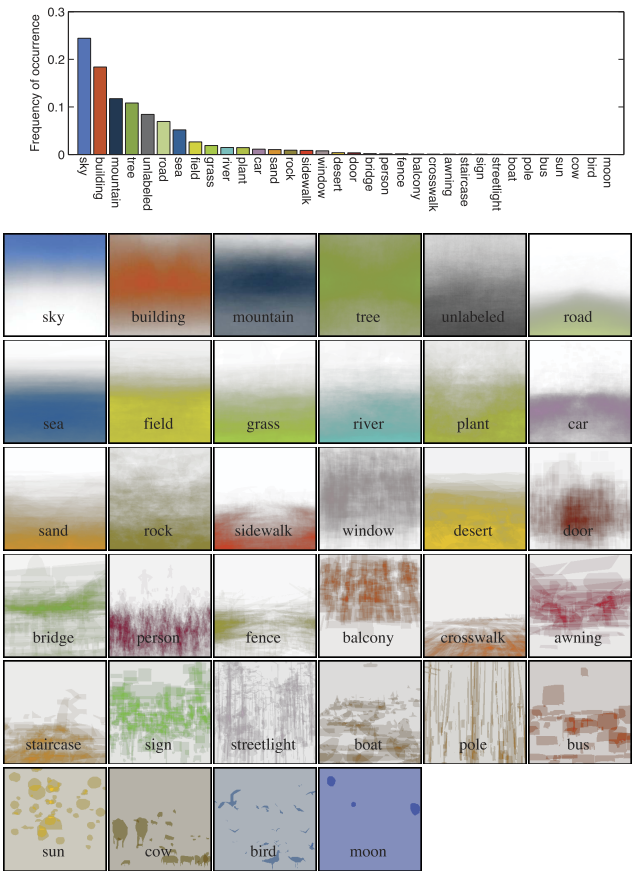


Fig. 3. Top: The per-pixel frequency counts of the object categories in our data set (sorted in descending order). The color of each bar is the average RGB value of each object category from the training data with saturation and brightness boosted for visualization. Bottom: The spatial priors of the object categories in the database. White means zero and the saturated color means high probability.

mountain, *tree*, *unlabeled*, *road*, *sea*, *field*, *grass*, and *river*. The spatial priors of these object categories are displayed at the bottom of Fig. 3, where white denotes zero probability and the saturation of color is directly proportional to its probability. Note that, consistent with common knowledge, *sky* occupies the upper part of the image grid and *field* occupies the lower part. Furthermore, there are only limited samples for the *sun*, *cow*, *bird*, and *moon* classes.

4 System Design

In this section, we will describe each module of our nonparametric scene parsing system.

4.1 Scene Retrieval

The objective of scene retrieval is to retrieve a set of nearest neighbors in the database for a given query image. There exist several ways for defining a nearest neighbor set. The most common definition consists of taking the K closest points to the query (K-NN). Another model, ϵ-NN, widely used in texture synthesis [12], [26], considers all of the neighbors within $(1+\epsilon)$ times the minimum distance from the query. We generalize these two types to $\langle K, \epsilon\rangle$-NN, and define it as

$$\begin{aligned}\mathcal{N}(x) = \{y_i \mid \mathrm{dist}(x, y_i) \le (1+\epsilon)\mathrm{dist}(x, y_1),\\ y_1 = \arg\min\ \mathrm{dist}(x, y_i), i \le K\}.\end{aligned} \tag{1}$$

As $\epsilon \to \infty$, $\langle K, \infty\rangle$-NN is reduced to K-NN. As $K \to \infty$, $\langle \infty, \epsilon\rangle$-NN is reduced to ϵ-NN. However, $\langle K, \epsilon\rangle$-NN representation gives us the flexibility to deal with the density variation of the graph, as shown in Fig. 5. We will show how K affects the performance in the experimental section. In practice, we found that $\epsilon = 5$ is a good parameter and we will use it through our experiments. Nevertheless, dramatic improvement of $\langle K, \epsilon\rangle$-NN over K-NN is not expected as sparse samples are few in our databases.

We have not yet defined the distance function $\mathrm{dist}(\cdot, \cdot)$ between two images. Measuring image similarities/distances is still an active research area; a systematic study of image features for scene recognition can be found in [52]. In this paper, three distances are used: euclidean distance of GIST [34], spatial pyramid histogram intersection of HOG visual words [24], and spatial pyramid histogram intersection of the ground-truth annotation. For the HOG distance, we use the standard pipeline of computing HOG features on a dense grid and quantizing features to visual words over a set of images using k-means clustering. The ground truth-based distance metric is used to estimate an upper bound of our system for evaluation purposes. Both the HOG and the ground truth distances are computed in the same manner. The ground truth distance is computed by building histograms of pixel-wise labels. To include spatial information, the histograms are computed by dividing an image into 2×2 windows and concatenating the four histograms into a single vector. Histogram intersection is used to compute the ground truth distance. We obtain the HOG distance by replacing pixel-wise labels with HOG visual words.

In Fig. 4, we show the importance of the distance metric as it defines the neighborhood structure of the large image database. We randomly selected 200 images from the LMO database and computed pair-wise image distances using GIST (top) and the ground-truth annotation (bottom). Then, we use multidimensional scaling (MDS) [4] to map these images to points on a 2D grid for visualization. Although the GIST descriptor is able to form a reasonably meaningful image space where semantically similar images are clustered, the image space defined by the ground-truth annotation truly reveals the underlying structures of the image database. This will be further examined in the experimental section.

4.2 SIFT Flow for Dense Scene Alignment

As our goal is to transfer the labels of existing samples to parse an input image, it is essential to find the dense correspondence for images across scenes. In our previous work [29], we have demonstrated that SIFT flow is capable of establishing semantically meaningful correspondences among two images by matching local SIFT descriptors. We further extended SIFT flow into a hierarchical computational framework to improve the performance [27]. In this section, we will provide a brief explanation of the algorithm; for a detailed description, we refer to [28].

Similarly to optical flow, the task of SIFT flow is to find dense correspondence between two images. Let $\mathbf{p} = (x, y)$ contain the spatial coordinate of a pixel, and $\mathbf{w}(\mathbf{p}) = (u(\mathbf{p}), v(\mathbf{p}))$ be the flow vector at $\mathbf{p}$. Denote s_1 and s_2 as the per-pixel SIFT descriptor [30] for two images,[2] and ε contains all the spatial neighborhood (a four-neighbor system is used). The energy function for SIFT flow is defined as:

$$E(\mathbf{w}) = \sum_{\mathbf{p}} \min(\|s_1(\mathbf{p}) - s_2(\mathbf{p} + \mathbf{w}(\mathbf{p}))\|_1, t) + \tag{2}$$

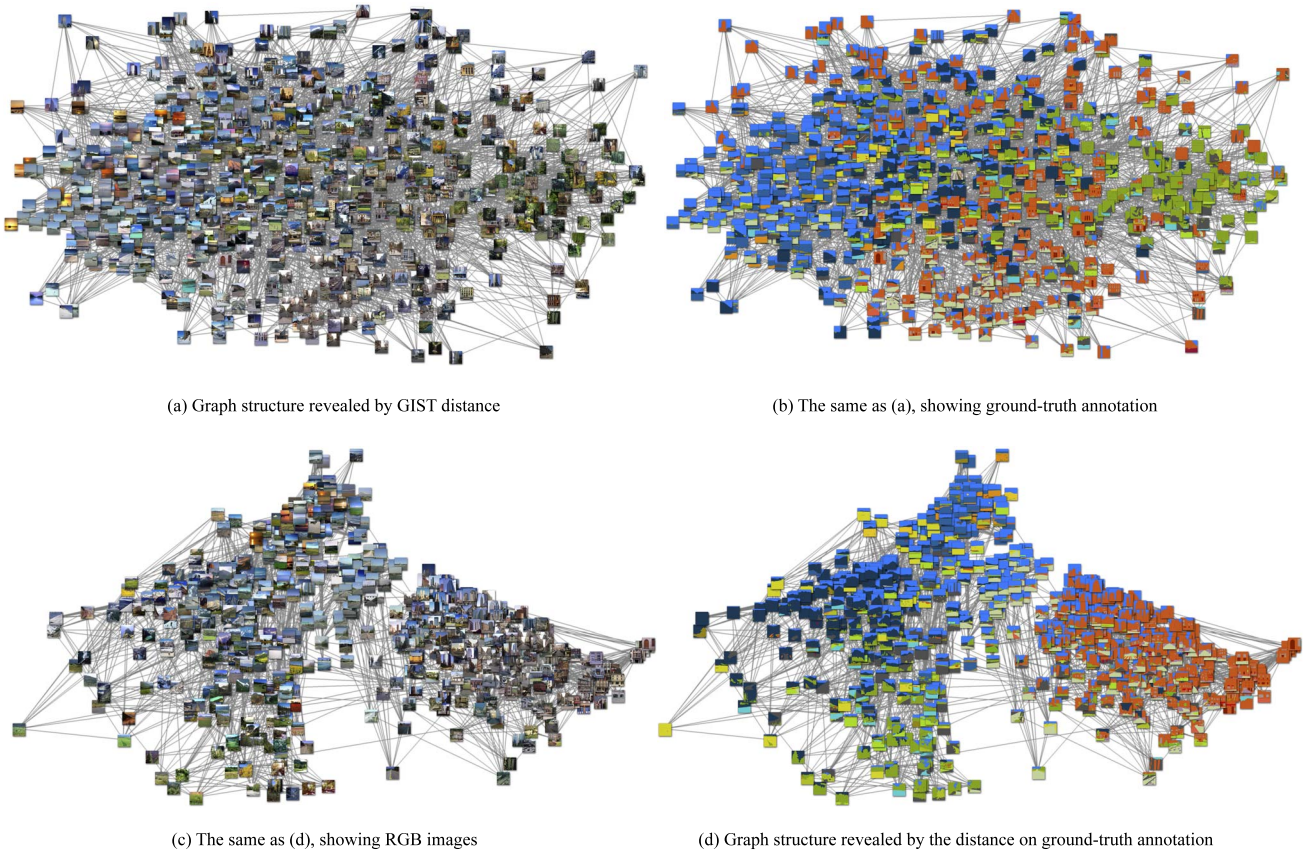
$$\sum_{\mathbf{p}} \eta(|u(\mathbf{p})| + |v(\mathbf{p})|) + \tag{3}$$

$$\begin{aligned}\sum_{(\mathbf{p},\mathbf{q})\in\varepsilon} \min(\lambda|u(\mathbf{p}) - u(\mathbf{q})|, d) +\\ \min(\lambda|v(\mathbf{p}) - v(\mathbf{q})|, d),\end{aligned} \tag{4}$$

which contains a *data term*, *small displacement term*, and *smoothness term* (a.k.a. spatial regularization). The *data term* in (2) constrains the SIFT descriptors to be matched along with the flow vector $\mathbf{w}(\mathbf{p})$. The *small displacement term* in (3) constrains the flow vectors to be as small as possible when no other information is available. The *smoothness term* in (4) constrains the flow vectors of adjacent pixels to be similar. In this objective function, truncated L1 norms are used in both the data term and the smoothness term to account for matching outliers and flow discontinuities, with t and d as the threshold, respectively.

While SIFT flow has demonstrated the potential for aligning images across scenes [29], the original implementation scales poorly with respect to the image size. In SIFT flow, a pixel in one image can literally match to any other pixel in another image. Suppose the image has h^2 pixels, then the time and space complexity of the belief propagation algorithm to estimate the SIFT flow is $O(h^4)$. As reported

2. SIFT descriptors are computed at each pixel using a 16×16 window. The window is divided into 4×4 cells, and image gradients within each cell are quantized into a 8-bin histogram. Therefore, the pixel-wise SIFT feature is a 128D vector.



(a) Graph structure revealed by GIST distance

(b) The same as (a), showing ground-truth annotation

(c) The same as (d), showing RGB images

(d) Graph structure revealed by the distance on ground-truth annotation

Fig. 4. The structure of a database depends on image distance metric. Top: The $\langle K, \epsilon \rangle$-NN graph of the LabelMe Outdoor database visualized by scaled MDS using GIST feature as distance. Bottom: The $\langle K, \epsilon \rangle$-NN graph of the same database visualized using the pyramid histogram intersection of ground-truth annotation as distance. Left: RGB images; right: annotation images. Notice how the ground-truth annotation emphasizes the underlying structure of the database. In (c) and (d), we see that the image content changes from urban, streets (right), to highways (middle), and to nature scenes (left) as we pan from right to left. Eight hundred images are randomly selected from LMO for this visualization.

in [29], the computation time for 145×105 images with an 80×80 searching neighborhood is 50 seconds. The original implementation of SIFT flow would require more than 2 hours to process a pair of 256×256 images in our database with a memory usage of 16 GB to store the data term. To address the performance drawback, a coarse-to-fine SIFT flow matching scheme was designed to significantly improve the performance. As illustrated in Fig. 6, the basic idea consists of estimating the flow at a coarse level of image grid, and then gradually propagating and refining the flow from coarse to fine; please refer to [28] for details. As a result, the complexity of this coarse-to-fine algorithm is $O(h^2 \log h)$, a significant speed up compared to $O(h^4)$. The matching between two 256×256 images take 31 seconds on a workstation with two quad-core 2.67 GHz Intel Xeon CPUs and 32 GB memory, in a C++ implementation. We also discovered that the coarse-to-fine scheme not only runs

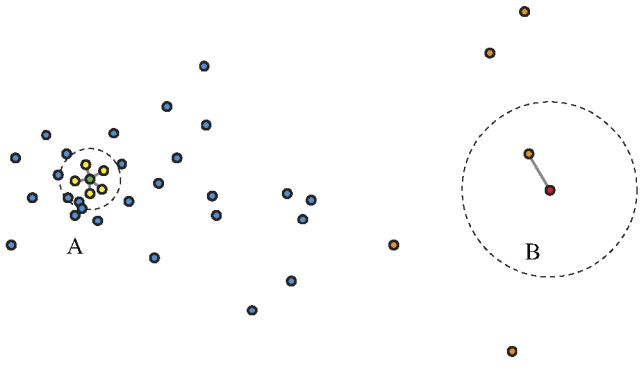


Fig. 5. An image database can be nonuniform as illustrated by some random 2D points. The green node (A) is surrounded densely by neighbors, whereas the red node (B) resides in a sparse area. If we use K-NN ($K = 5$), then some samples (orange nodes) far away from the query (B) can be chosen as neighbors. If, instead, we use ϵ-NN and choose the radius as shown in the picture, then there can be too many neighbors for a sample such as (A). The combination, $\langle K, \epsilon \rangle$-NN, shown as gray-edges, provides a good balance for these two criteria.

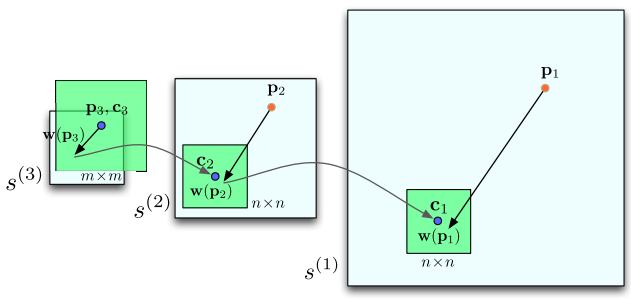


Fig. 6. An illustration of our coarse-to-fine pyramid SIFT flow matching. The green square denotes the searching window for $\mathbf{p}_k$ at each pyramid level k. For simplicity, only one image is shown here, where $\mathbf{p}_k$ is on image s_1 and $\mathbf{c}_k$ and $\mathbf{w}(\mathbf{p}_k)$ are on image s_2. The details of the algorithm can be found in [28].

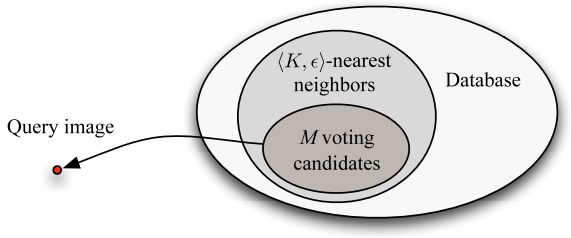


Fig. 7. For a query image, we first find a $\langle K, \epsilon \rangle$-nearest neighbor set in the database using GIST matching [34]. The nearest neighbors are reranked using SIFT flow matching scores, and form a top M-voting candidate set. The annotations are transferred from the voting candidates to parse the query image.

significantly faster, but also achieves lower energies most of the time compared to the ordinary matching algorithm.

Some SIFT flow examples are shown in Fig. 8, where dense SIFT flow fields (Fig. 8f) are obtained between the query images (Fig. 8a) and the nearest neighbors (Fig. 8c). It is trial to verify that the warped SIFT images (Fig. 8h) based on the SIFT flows (Fig. 8f) look very similar to the SIFT images (Fig. 8b) of the inputs (Fig. 8a), and that the SIFT flow fields (Fig. 8f) are piecewise smooth. The essence of SIFT flow is manifested in Fig. 8g, where the same flow field is applied to warp the RGB image of the nearest neighbor to the query. SIFT flow is trying to hallucinate the structure of the query image by smoothly shuffling the pixels of the nearest neighbors. Because of the intrinsic similarities within each object categories, it is not surprising that, through aligning image structures, objects of the same categories are often matched. In addition, it is worth noting that one object in the nearest neighbor can correspond to multiple objects in the query since the flow is asymmetric. This allows reuse of labels to parse multiple object instances.

4.3 Scene Parsing through Label Transfer

Now that we have a large database of annotated images and a technique for establishing dense correspondences across scenes, we can transfer the existing annotations to parse a query image through dense scene alignment. For a given query image, we retrieve a set of $\langle K, \epsilon \rangle$-*nearest neighbors* in our database using GIST matching [34]. We then compute the SIFT flow from the query to each nearest neighbor, and use the achieved minimum energy (defined in (4)) to rerank the $\langle K, \epsilon \rangle$-nearest neighbors. We further select the top M reranked retrievals ($M \leq K$) to create our voting candidate set. This voting set will be used to transfer its contained annotations into the query image. This procedure is illustrated in Fig. 7.

Under this setup, scene parsing can be formulated as the following label transfer problem: For a query image I with its corresponding SIFT image s, we have a set of voting candidates $\{s_i, c_i, \mathbf{w}_i\}_{i=1:M}$, where s_i, c_i, and w_i are the SIFT image, annotation, and SIFT flow field (from s to s_i) of the ith voting candidate, respectively. c_i is an integer image where $c_i(\mathbf{p}) \in \{1, \ldots, L\}$ is the index of object category for pixel $\mathbf{p}$. We want to obtain the annotation c for the query image by transferring c_i to the query image according to the dense correspondence $\mathbf{w}_i$.

We build a probabilistic Markov random field model to integrate multiple labels, prior information of object category, and spatial smoothness of the annotation to parse image I. Similarly to that of [43], the posterior probability is defined as:

$$\begin{aligned} -\log P\big(c|I, s, \{s_i, c_i, \mathbf{w}_i\}\big) &= \sum_{\mathbf{p}} \psi\big(c(\mathbf{p}); s, \{s_i'\}\big) \\ &+ \alpha \sum_{\mathbf{p}} \lambda\big(c(\mathbf{p})\big) + \beta \sum_{\{\mathbf{p},\mathbf{q}\} \in \varepsilon} \phi\big(c(\mathbf{p}), c(\mathbf{q}); I\big) + \log Z, \end{aligned} \tag{5}$$

where Z is the normalization constant of the probability. This posterior contains three components, i.e., likelihood, prior, and spatial smoothness.

The *likelihood* term is defined as

$$\psi\big(c(\mathbf{p}) = l\big) = \begin{cases} \min\limits_{i \in \Omega_{\mathbf{p},l}} \|s(\mathbf{p}) - s_i(\mathbf{p} + \mathbf{w}(\mathbf{p}))\|, & \Omega_{\mathbf{p},l} \neq \emptyset, \\ \tau, & \Omega_{\mathbf{p},l} = \emptyset, \end{cases} \tag{6}$$

where $\Omega_{\mathbf{p},l} = \{i; c_i(\mathbf{p} + \mathbf{w}(\mathbf{p})) = l\}$, $l = 1, \ldots, L$, is the index set of the voting candidates whose label is l after being warped to pixel $\mathbf{p}$. τ is set to be the value of the maximum difference of SIFT feature: $\tau = \max_{s_1, s_2, \mathbf{p}} \|s_1(\mathbf{p}) - s_2(\mathbf{p})\|$.

The *prior* term $\lambda(c(\mathbf{p}) = l)$ indicates the prior probability that object category l appears at pixel $\mathbf{p}$. This is obtained from counting the occurrence of each object category at each location in the training set:

$$\lambda\big(c(\mathbf{p}) = l\big) = -\log \mathrm{hist}_l(\mathbf{p}), \tag{7}$$

where $\mathrm{hist}_l(\mathbf{p})$ is the spatial histogram of object category l.

The *smoothness* term is defined to bias the neighboring pixels into having the same label in the event that no other information is available, and the probability depends on the edge of the image: The stronger luminance edge, the more likely it is that the neighboring pixels may have different labels:

$$\phi\big(c(\mathbf{p}), c(\mathbf{q})\big) = \delta[c(\mathbf{p}) \neq c(\mathbf{q})] \left(\frac{\xi + e^{-\gamma \|I(\mathbf{p}) - I(\mathbf{q})\|^2}}{\xi + 1} \right), \tag{8}$$

where $\gamma = (2 < \|I(\mathbf{p}) - I(\mathbf{q})\|^2 >)^{-1}$ [43].

Notice that the energy function is controlled by four parameters, K and M that decide the mode of the model and α and β that control the influence of spatial prior and smoothness. Once the parameters are fixed, we again use the BP-S algorithm to minimize the energy. The algorithm converges in two seconds on a workstation with two quad-core 2.67 GHz Intel Xeon CPUs.

A significant difference between our model and that in [43] is that we have fewer parameters because of the nonparametric nature of our approach, whereas classifiers were trained in [43]. In addition, color information is not included in our model at the present as the color distribution for each object category is diverse in our databases.

5 Experiments

Extensive experiments were conducted to evaluate our system. We shall first report the results on a small scale database which we will refer to as the LabelMe Outdoor (LMD) database in Section 5.1; this database will aid us for an in-depth exploration of our model. Furthermore, we will

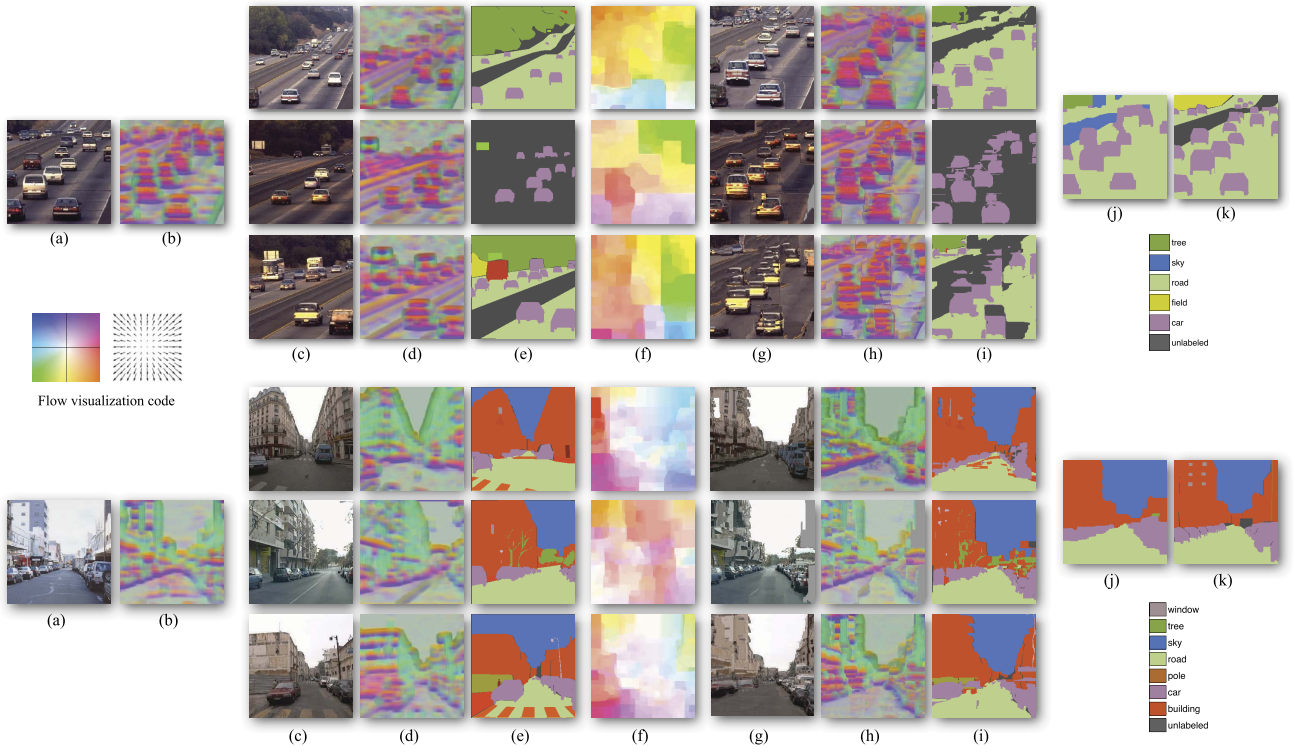


Fig. 8. System overview. For a query image, our system uses scene retrieval techniques such as [34] to find $\langle K, \epsilon \rangle$-nearest neighbors in our database. We apply coarse-to-fine SIFT flow to align the query image to the nearest neighbors, and obtain top M as voting candidates ($M = 3$ here). (c), (d), (e): The RGB image, SIFT image and user annotation of the voting candidates. (f): The inferred SIFT flow field, visualized using the color scheme shown on the left (hue: orientation; saturation: magnitude). (g), (h), and (i) are the warped version of (c), (d), (e) with respect to the SIFT flow in (f). Notice the similarity between (a) and (g), (b) and (h). Our system combines the voting from multiple candidates and generates scene parsing in (j) by optimizing the posterior. (k): The ground-truth annotation of (a).

report results on the SUN database, a larger and more challenging data set in Section 5.2.

5.1 LabelMe Outdoor Database

As mentioned in Section 3, the LMO database consists of 2,688 outdoor images, which have been randomly split into 2,466 training and 200 test images. The images are densely labeled with 33 object categories using the LabelMe online annotation tool. Our scene parsing system is illustrated in Fig. 8. The system retrieves a $\langle K, \epsilon \rangle$-nearest neighbor set for the query image (Fig. 8a), and further selects M voting candidates containing minimum SIFT matching scores. For illustration purposes, we set $M = 3$ here. The original RGB image, SIFT image, and annotation of the voting candidates are shown in Figs. 8c, 8d, and 8e, respectively. The SIFT flow field is visualized in Fig. 8f using the same visualization scheme as in [29], where hue indicates orientation and saturation indicates magnitude. After we warp the voting candidates into the query with respect to the flow field, the warped RGB (Fig. 8g) and SIFT image (Fig. 8h) are very close to the query Fig. 8a and Fig. 8b, respectively. Combining the warped annotations in Fig. 8i, the system outputs the parsing of the query in Fig. 8j, which is close to the ground-truth annotation in Fig. 8k.

5.1.1 Evaluation Criterion

We use average pixel-wise recognition rate $\bar{r}$ (similar to precision or true positive) to evaluate the performance of our system, computed as

$$\bar{r} = \frac{1}{\sum_i m_i} \sum_i \sum_{\mathbf{p} \in \Lambda_i} \mathbf{1}(o(\mathbf{p}) = a(\mathbf{p}),\ a(\mathbf{p}) > 0), \qquad (9)$$

where, for pixel $\mathbf{p}$ in image i, the ground-truth annotation is $a(\mathbf{p})$ and system output is $o(\mathbf{p})$; for unlabeled pixels, $a(\mathbf{p}) = 0$. Notation Λ_i is the image lattice for test image i, and $m_i = \sum_{\mathbf{p} \in \Lambda_i} \mathbf{1}(a(\mathbf{p}) > 0)$ is the number of labeled pixels for image i (some pixels are unlabeled). We also compute the per-class average rate r_l as

$$r_l = \frac{\sum_i \sum_{\mathbf{p} \in \Lambda_i} \mathbf{1}(o(\mathbf{p}) = a(\mathbf{p}), a(\mathbf{p}) = l)}{\sum_i \sum_{\mathbf{p} \in \Lambda_i} \mathbf{1}(a(\mathbf{p}) = l)}, \quad l = 1, \ldots, L. \qquad (10)$$

5.1.2 Results and Comparisons

Some label transfer results are shown in Fig. 10. The input image from the test set is displayed in Fig. 10a. We show the best match, its corresponding annotation, and the warped best match in Figs. 10b, 10c, and 10d, respectively. While the final labeling constitutes the integration of the top M matches, the best match can provide the reader an intuition of the process and final result. Notice how the warped image (Fig. 10d) looks similar to the input (Fig. 10a), indicating that SIFT flow successfully matches image structures. The scene parsing results output by our system are listed in Fig. 10e with parameter setting $K = 85, M = 9, \alpha = 0.06, \beta = 20$. The ground-truth user annotation is listed in Fig. 10f. Notice that the gray pixels in Fig. 10f are "unlabeled," but our system does not generate "unlabeled" output. For samples 1, 5, 6, 8, and 9,

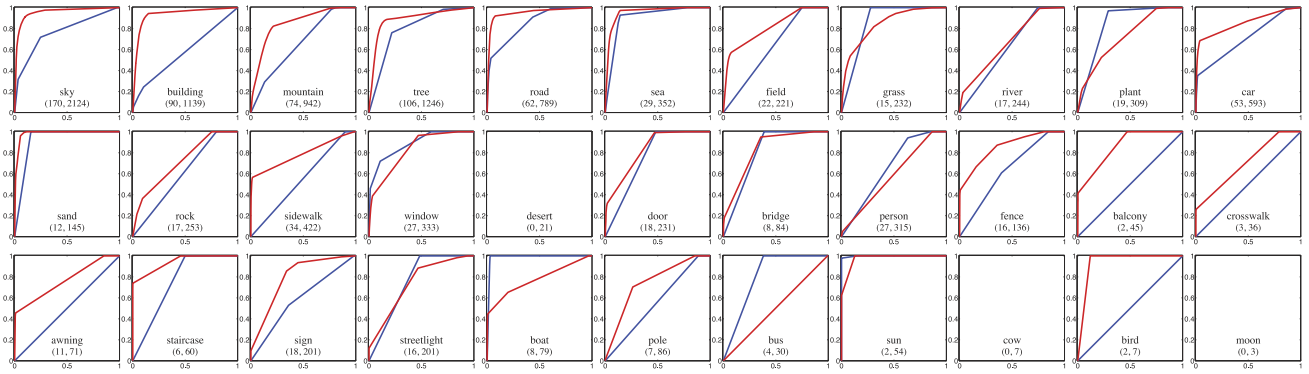


Fig. 9. The ROC curve of each individual pixel-wise binary classifier. Red curve: Our system after being converted to binary classifiers; blue curve: the system in [8]. We used the convex hull to make the ROC curves strictly concave. The number (n, m) underneath the name of each plot is the quantity of the object instances in the test and training set, respectively. For example, (170, 2,124) under "sky" means that there are 170 test images containing sky, and 2,124 training images containing sky (there are in total 2,488 training images and 200 test images). Our system obtains reasonable performance for objects with sufficient samples in both training and test sets, e.g., *sky*, *building*, *mountain*, and *tree*. We observe truncation in the ROC curves where there are not enough test samples, e.g., *field*, *sea*, *river*, *grass*, *plant*, *car*, and *sand*. The performance is poor for objects without enough training samples, e.g., *crosswalk*, *sign*, *boat*, *pole*, *sun*, and *bird*. The ROC does not exist for objects without any test samples, e.g., *desert*, *cow*, and *moon*. In comparison, our system outperforms or equals [8] for all object categories except for *grass*, *plant*, *boat*, *person*, and *bus*. The performance of [8] on our database is low because the objects have drastically different poses and appearances.

our system generates reasonable predictions for the pixels annotated as "unlabeled." The average pixel-wise recognition rate of our system is **76.67 percent** by excluding the "unlabeled" class [43]. Some failure examples from our system are shown in Fig. 11 when the system fails to retrieve images with similar object categories to the query, or when the annotation is ambiguous.

Overall, our system is able to predict the right object categories in the input image with a segmentation fit to image boundary, even though the best match may look different from the input, e.g., 2, 11, 12, and 17. If we divide the object categories into *stuff* (e.g., *sky*, *mountains*, *tree*, *sea*, and *field*) and *things* (e.g., *cars*, *sign*, *boat*, and *bus*) [1], [22], our system generates much better results for stuff than for things. The recognition rate for the top seven object categories (all are "stuff") is **82.72 percent**. This is because in our current system, we only allow one labeling for each pixel, and smaller objects tend to be overwhelmed by the labeling of larger objects. We plan to build a recursive system in our future work to further retrieve things based on the inferred stuff.

For comparison purposes, we downloaded and executed the texton-boost code from [43] using the same training and test data with the Markov random field turned off. The overall pixel-wise recognition rate of their system on our data set is **51.67 percent**, and the per-class rates are displayed in Fig. 12c. For fairness we also turned off the Markov random field model as well as spatial priors in our framework by setting $\alpha = \beta = 0$, and plotted the corresponding results in Fig. 12f. Clearly, our system outperforms [43] in terms of both overall and per-class recognition rate. Similar performance to texton-boost is achieved by matching color instead of matching dense SIFT descriptors in our system, as shown in Fig. 12b. The recognition rate of the class *grass* and *sand* dramatically increases through matching color because color is the salient feature for these categories. However, the performance drops for other color-variant categories. This result supports the importance of matching appearance-invariant features in our label transfer system.

We also compared the performance of our system with a classifier-based system [8]. We downloaded their code and trained a classifier for each object category using the same training data. We converted our system into a binary object detector for each class by only using the per-class likelihood term. The per-class ROC curves of our system (red) and theirs (blue) are plotted in Fig. 9. Except for five object categories, *grass*, *plant*, *boat*, *person*, *streetlight*, and *bus*, our system outperforms or equals theirs.

5.1.3 Parameter Selection

Since the SIFT flow module is essential to our system, we first test spatial smoothness coefficient λ in (4), which determines matching results. We compute the average pixel-wise recognition rate as a function of λ, shown in Fig. 13a. We first turn off the MRF model in the label transfer module by setting $\alpha = \beta = 0$, and find that when $\lambda = 0.7$, the maximal recognition rate is achieved. Then, we turn on the MRF model by setting $\alpha = 0.1, \beta = 60$, and find that $\lambda = 0.7$ leads to a good performance as well. Therefore, we fix $\lambda = 0.7$ throughout our experiments.

We investigated the performance of our system by varying the parameters K, M, α, and β. We have found that the influence of ϵ is smaller than that of K when ϵ is set such that most samples have K nearest neighbors. We vary $M = 1, 3, 5, 7, 9$ and $K = 1, 5, 10, \ldots, 100$. For each combination of K and M ($M \leq K$), coordinate descend is used to find the optimal parameter of α and β by maximizing the recognition rate. We plot the recognition rate as a function of K for a variety of Ms in Fig. 13b. Overall, the recognition rate increases as more nearest neighbors are retrieved ($K \uparrow$) and more voting candidates are used ($M \uparrow$) since, obviously, multiple candidates are needed to transfer labels to the query. However, the recognition rate drops as K and M continue to increase as more candidates may introduce noise to the label transfer process. In particular, the recognition rate drops when K increases, suggesting that scene retrieval does not only serve as a way to obtain neighbors for SIFT flow, but also rule out some bad images that SIFT flow would otherwise choose. The maximum performance is obtained when $K = 85$ and $M = 9$.

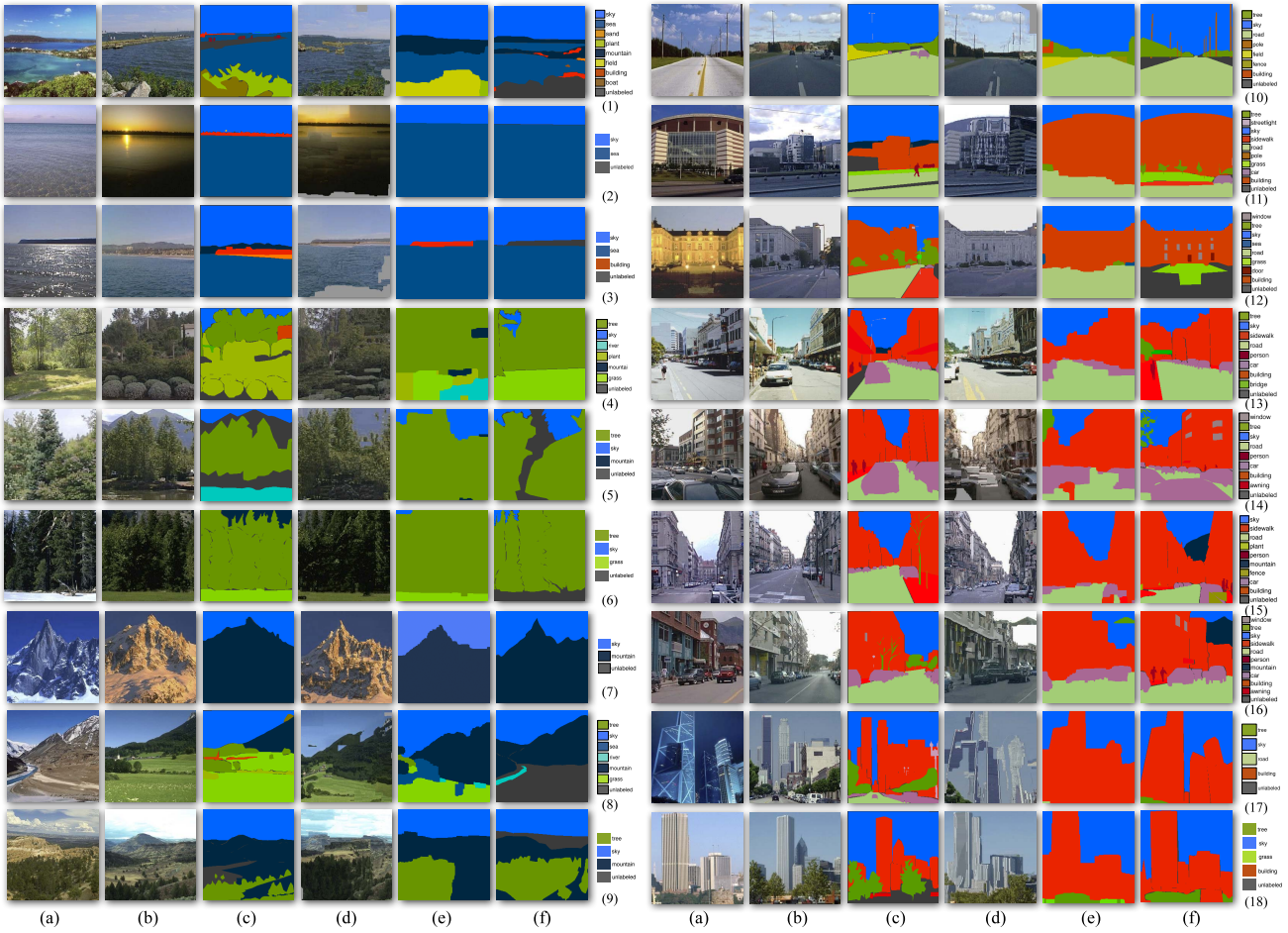


Fig. 10. Some scene parsing results output from our system. (a): Query image, (b): the best match from nearest neighbors, (c): the annotation of the best match, (d): the warped version of (b) according to the SIFT flow field, (e): the inferred per-pixel parsing after combining multiple voting candidates, (f): the ground truth annotation of (a). The dark gray pixels in (f) are "unlabeled." Notice how our system generates a reasonable parsing even for these "unlabeled" pixels.

Because the regularity of the database is the key to the success, we remove the SIFT flow matching, i.e., set the flow vector to be zero for every pixel, and obtain an average recognition rate of 61.23 percent without MRF and **67.96 percent** with MRF, shown in Figs. 12d and 12f, respectively. This result is significant because SIFT flow is the bottleneck of the system in terms of speed. A fast implementation of our system consists of removing the dense scene alignment module, and simply performing a grid-to-grid label transfer (the likelihood term in the label transfer module still comes from SIFT descriptor distance).

How would different scene retrieval techniques affect our system? Other than the GIST distance used for retrieving nearest neighbors for the results in Fig. 12, we also use the spatial pyramid histogram intersection of HOG visual words and of the ground-truth annotation, with the corresponding per-class recognition rate displayed in Figs. 12g and 12h, respectively. For this database, GIST performs slightly better than HOG visual words. We also explore an upper bound of the label transfer framework in the ideal scenario of having access to perfect scene matching. In particular, we retrieve the nearest neighbors for each image using their ground

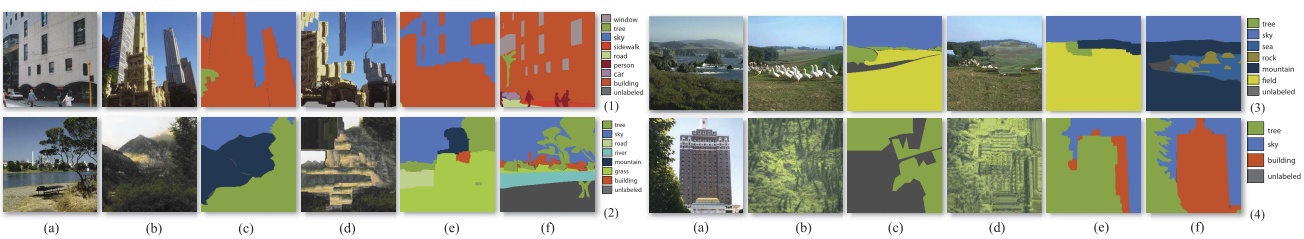


Fig. 11. Some typical failures. Our system fails when no good matches can be retrieved in the database. In (2), for example, since the best matches do not contain river, the input image is mistakenly parsed as a scene of grass, tree, and mountain in (e). The ground-truth annotation is in (f). The failure may also come from ambiguous annotations, for instance in (3), where the system outputs field for the bottom part, whereas the ground-truth annotation is mountain.

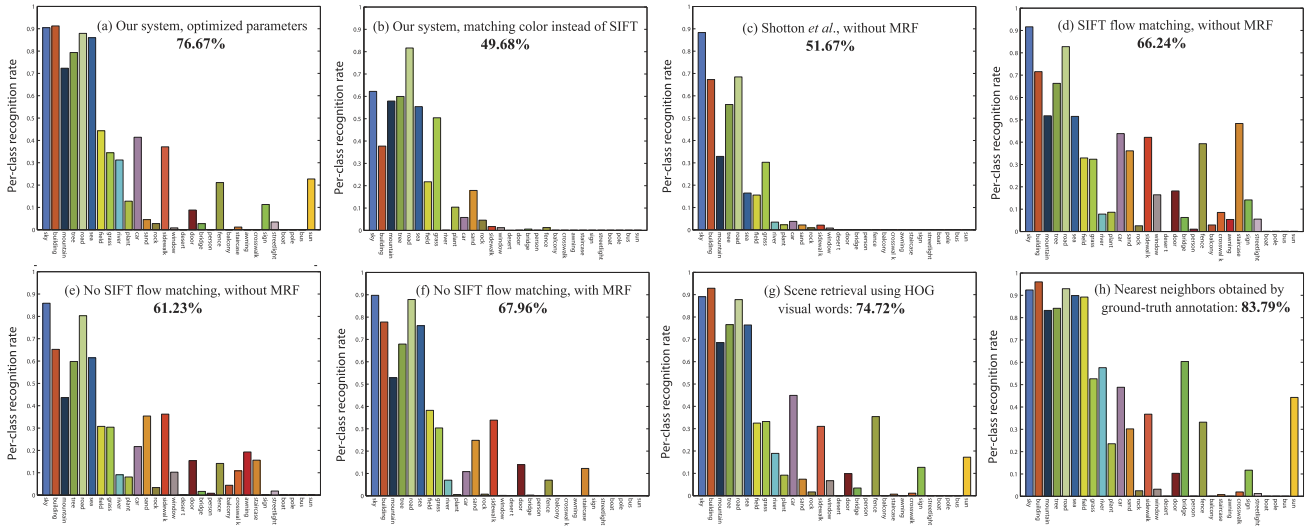


Fig. 12. We study the performance of our system in depth. (a) Our system with the parameters optimized for pixel-wise recognition rate. (b) Our system, matching RGB instead of matching dense SIFT descriptors. (c) The performance of [43] with the Markov random field component turned off, trained and tested on the same data sets as (a). In (d), (e), (f), we show the importance of SIFT flow matching and the MRF for label transfer by turning them on and off. In (g) and (h), we show the system performance affected by other scene retrieval methods. The performance in (h) shows the upper limit of our system, by adopting ideal scene retrieval using ground-truth annotation (of course, the ground-truth annotation is not available in practice). See text for more details.

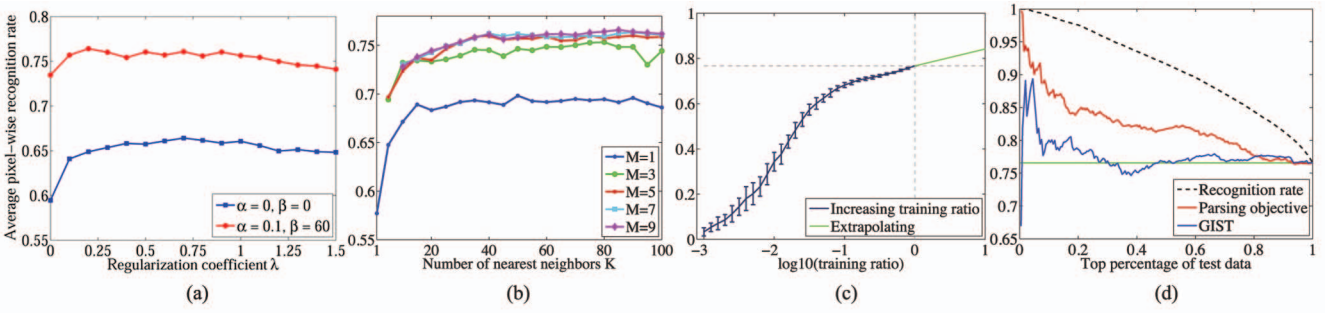


Fig. 13. (a): Recognition rate as a function of the spatial smoothness coefficient λ under two settings for α and β. (b): Recognition rate as a function of number of nearest neighbors K and the number of voting candidates M. Clearly, prior and spatial smoothness help improve the recognition rate. The fact that the curve drops down as we further increase K indicates that SIFT flow matching cannot replace scene retrieval. (c): Recognition rate as a function of the log training ratio while the test set is fixed. A subset of training samples are randomly drawn from the entire training set according to the training ratio to test how the performance depends on the size of the database. (d): Recognition rate as a function of the proportion of the top ranked test images according to metrics including GIST, the parsing objective in (5), and the recognition rate (with ground-truth annotation). The black, dashed curve with recognition rate as sorting metric is the ideal case, and the parsing objective is better than GIST. These curves suggests the system is somewhat capable of distinguishing good parsing results from bad ones.

truth annotations; please refer to Section 4.1 for the details. This upper bound is an **83.79 percent** recognition rate.

To further understand our data-driven system, we evaluated the performance of the system as a function of the ratio of training samples while fixing the test set. For each fixed ratio, we formed a small training database randomly drawing samples from the original database and evaluated the performance of the system under this database. This experiment was performed 15 times for each ratio to obtain a mean and standard deviation of its performance, shown in Fig. 13c. Clearly, the recognition rate depends on the size of the training database. Using the last 10 data points for extrapolation, we found that if we increase the training data by 10 times (corresponding to 1 on the horizontal axis), the recognition rate may increase to **84.16 percent**.[3] Note, however, that this linear extrapolation does not consider potential saturation issues as it can be observed when more than 10 percent of training samples were used. This indicates that the training quantity is reasonable for this database.

Another aspect we evaluated is the capacity to detect *good* and *bad* parsing results. For this purpose, we rerank the test image using three metrics: recognition rate (the ideal metric; evaluated with respect to ground truth), the parsing objective in (5) after energy minimization, and the average GIST-based distance to the nearest neighbors. After ranking, we computed the accumulated average recognition rate as a function of the ratio of testing samples, as shown in Fig. 13b. If we use the parsing objective as a metric, for example, then the average recognition rate can be greater than **80 percent** when only the top 75 percent parsing results are picked. The system can reject the rest 25 percent with low scores.

5.2 SUN Database

We further evaluated the performance of our system on the SUN database [52], which contains 9,556 images of both

3. This extrapolation is different from moving to a larger database in Section 5.2, where indoor scenes are included. This number is anticipated only when images similar to the LMO database are added.

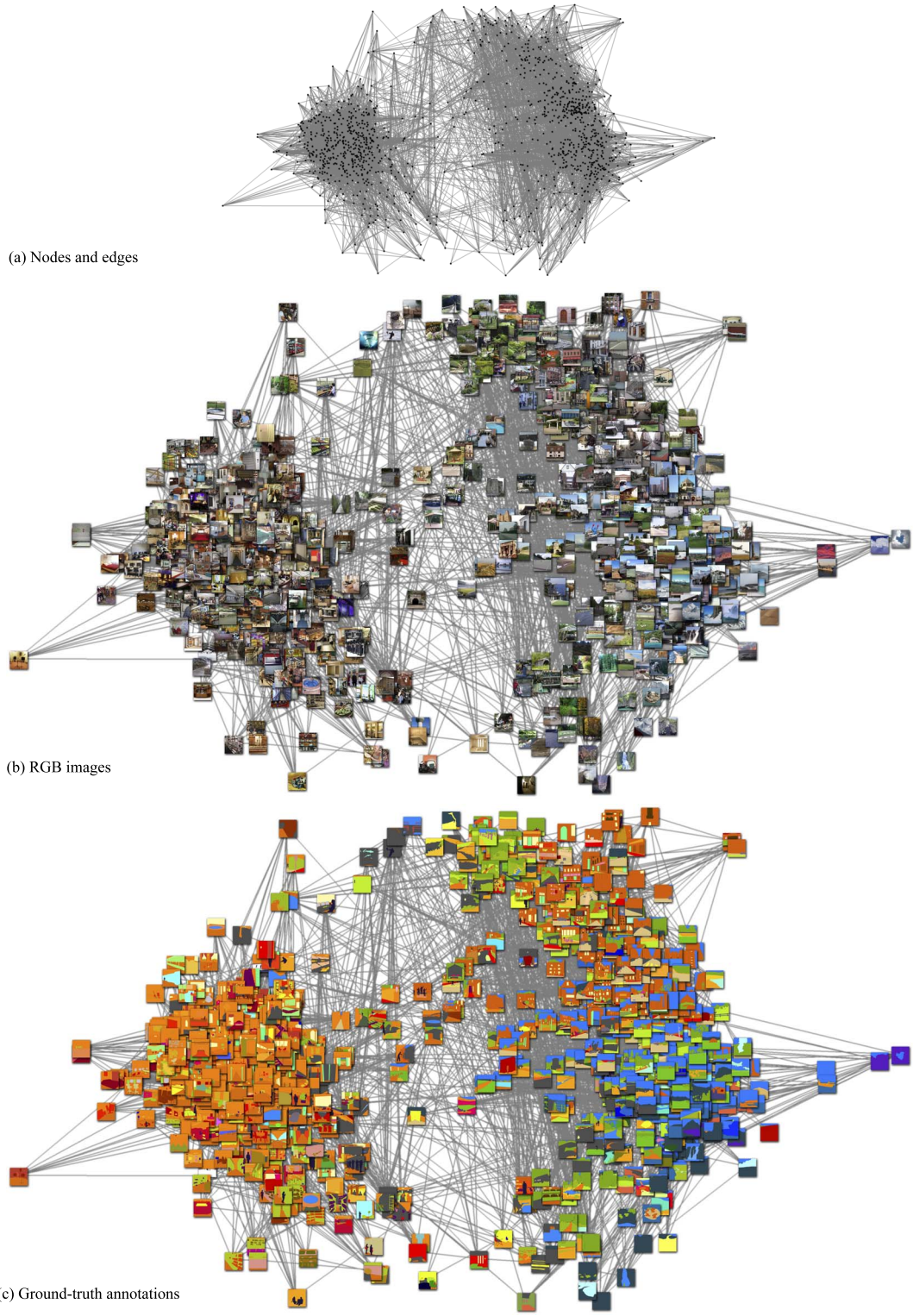


Fig. 14. Visualization of the SUN database [52] using 1,200 random images. Notice that the SUN database is larger but not necessarily denser than the LMO database. We use the spatial pyramid histogram intersection distance of the ground-truth annotation to measure the distance between the images and project them to a 2D space using scaled multidimensional scaling. Clearly, the images are clustered to indoor (left) and outdoor (right) scenes, and there is smooth transition in between. In the outdoor cluster, we observe the change from *garden*, *street*, to *mountain* and *valley* as we move from top to bottom. Please visit http://people.csail.mit.edu/celiu/LabelTransfer/ to see the full resolution of the graphs.

indoor and outdoor scenes. This database contains a total of 515 object categories; the pixel frequency counts for the top 100 categories are displayed in Fig. 15a. The data corpus is randomly split into 8,556 images for training and 1,000 for testing. The structure of the database is visualized in Fig. 14 using the same technique to plot Fig. 5, where the image distance is measured by the ground-truth annotation. Notice the clear separation of indoor (left) and outdoor (right) scenes in this database. Moreover, the images are not evenly distributed in the space; they tend to reside around a

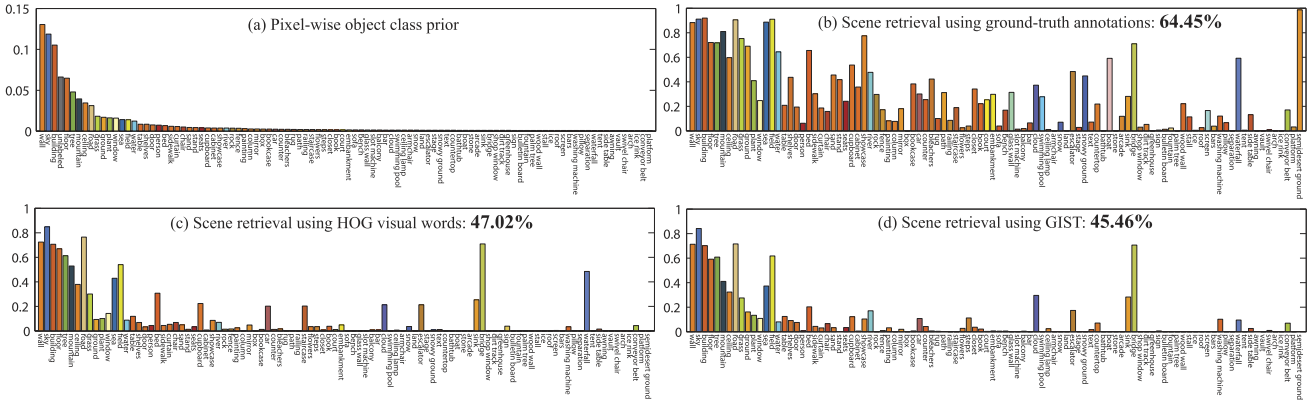


Fig. 15. The per-class recognition rate of running our system on the SUN database. (a) Pixel-wise frequency histogram of the top 100 object categories. (b): The per-class recognition rate when the ground-truth annotation is used for scene retrieval. This is the upper limit performance that can be achieved by our system. (c) and (d): Per-class recognition rate using HOG and GIST for scene retrieval, respectively. The HOG visual words features generate better results for this larger database.

few clusters. This phenomenon is consistent with human perception, drawing a clear separation between outdoor and indoor images.

Some scene parsing results are displayed in Fig. 16 in the same format as Fig. 10. Since the SUN database is a superset of the LabelMe Outdoor database, the selection of results is slightly biased toward indoor and activity scenes. Overall, our system performs reasonably well in parsing these challenging images.

We also plot the per-class performance in Fig. 15. In Fig. 15a, we show the pixel-wise frequency count of the top 100 object categories. Similarly to LMO, this prior distribution is heavily biased toward stuff-like classes, e.g., *wall*, *sky*, *building*, *floor*, and *tree*. In Fig. 15b, the performance is achieved when the ground-truth annotation is used for scene retrieval. Again, the average **64.45 percent** recognition rate reveals the upper limit and the potential of our system in the idealized case of perfect nearest neighbor retrieval. In Figs. 15c and 15d, the performance using HOG and GIST features for scene retrieval is plotted, suggesting that the HOG visual words features outperform GIST for this larger database. This is consistent with the discovery that the HOG feature is the best among a set of features, including GIST, in scene recognition in the SUN database [52].

Overall, the recognition rate on the SUN database is lower than that on the LMO database. A possible explanation for this phenomenon is that indoor scenes contain less regularity compared to outdoor ones, and there are 515 object categories in SUN, whereas there are only 33 categories in LMO.

6 Discussion

6.1 Label Transfer: An Open, Database-Driven Framework for Image Understanding

A unique characteristics of our nonparametric scene parsing system is its openness: To support more object categories, one can simply add more images of the new categories into the database without requiring additional training. This is an advantage over classical learning-based systems where all of the classifiers have to be retrained when new object categories are inserted into the database.

Although there is no parametric model (probabilistic distributions or classifiers) of object appearances in our system, the ability to recognize objects depends on reliable image matching across different scenes. When good matches are established between objects in the query and objects in the nearest neighbors in the annotated database, the known annotation naturally explains the query image as well. We chose SIFT flow [28] to establish a semantically meaningful correspondence between two different images. Nonetheless, this module can easily be substituted by other or better scene correspondence methods.

Although context is not explicitly modeled in our system, our label transfer-based scene parsing system naturally embeds contextually-coherent label sets. The nearest neighbors retrieved in the database and reranked by SIFT flow scores mostly belong to the same type of scene category, implicitly ensuring contextual coherence. Using Fig. 16 (9) as an example, we can see that even though the reflection of the mountain has been misclassified to *field*, *ground*, *tree*, and *plant*, the parsing result is context-coherent.

6.2 The Role of Scene Retrieval

Our nonparametric scene parsing system largely depends on the scene retrieval technique through which the nearest neighbors of the query image in the large database are obtained. We have tried two popular techniques, GIST and HOG visual words, and have found that GIST-based retrieval yields higher performance in the LMO database, whereas HOG visual words tend to retrieve better neighbors in the SUN database. We also show the upper bound performance of our system by using the ground-truth annotation for scene retrieval. This upper bound provides an intuition of the efficacy of our system given an ideal scene retrieval system. The recent advances in this area by combining multiple kernels [52] point out promising directions for scene retrieval.

6.3 Better Evaluation Criterion

Presently, we use a simple criterion, pixel-wise recognition rate to measure the performance of our scene parsing system. A pixel is correctly recognized only when the parsing result is exactly the same as the ground-truth annotation. However, human annotation can be ambiguous. For instance, in the parsing example depicted in Fig. 16 (9), the pixel-wise recognition rate is low because the *mountain*

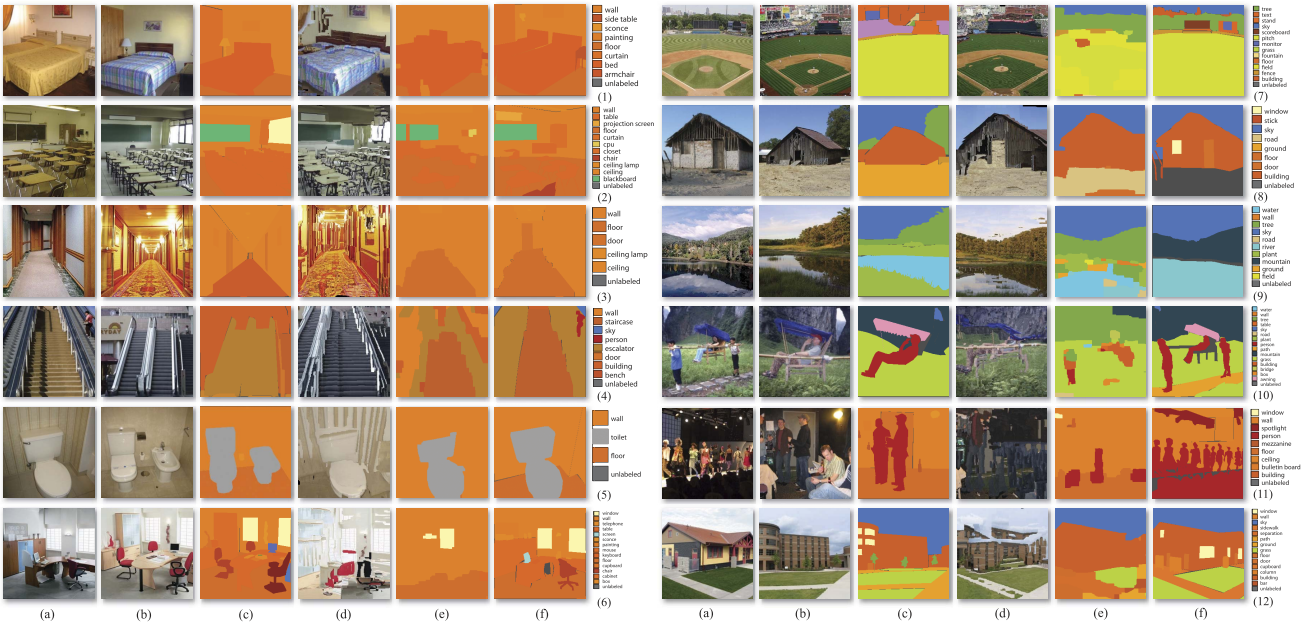


Fig. 16. Some scene parsing results on the SUN database following the same notation as in Fig. 10. Note that the LabelMe Outdoor database is a subset of the SUN database, but the latter contains a larger proportion of indoor and activity-based scenes. As it is in all cases, the success of the final parsing depends on the semantic similarity of the query in (a) to the warped support image (d). Examples (10) and (11) are two failure examples where many people are not correctly labeled.

is recognized as *tree* and the *water* is recognized as *river*. While, in our current evaluation framework, these pixels are considered misclassified, this parsing would be considered accurate when evaluated by a human. A more precise evaluation criterion would take synonyms into account. Another example is shown in Fig. 16 (12), where the windows are not present in the parsing result. Therefore, the *window* pixels are labeled wrong because they are classified as *building*, which is a more favorable label than, for example, *car*, as windows tend to appear on top of buildings. A superior evaluation criterion should also consider co-occurrence and occlusion relationships. We leave these items as future work.

6.4 Nonparametric versus Parametric Approaches

In this paper, we have demonstrated promising results of our nonparametric scene parsing system using label transfer by showing how it outperforms existing recognition-based approaches. However, we do not believe that our system alone is the ultimate answer to image understanding since it does not work well for small objects such as *person*, *window*, *bus*, etc., which can be better handled using detectors. Moreover, pixel-wise classifiers such as textonboost can also be useful when good matching cannot be established or good nearest neighbors can hardly be retrieved. Therefore, a natural future step is to combine these methods for scene parsing and image understanding.

7 Conclusion

We have presented a novel, nonparametric scene parsing system to integrate and transfer the annotations from a large database to an input image via dense scene alignment. A coarse-to-fine SIFT flow matching scheme is proposed to reliably and efficiently establish dense correspondences between images across scenes. Using the dense scene correspondences, we warp the pixel labels of the existing samples to the query. Furthermore, we integrate multiple cues to segment and recognize the query image into the object categories in the database. Promising results have been achieved by our scene alignment and parsing system on challenging databases. Compared to existing approaches that require training for each object category, our nonparametric scene parsing system is easy to implement, has only a few parameters, and embeds contextual information naturally in the retrieval/alignment procedure.

Acknowledgments

Funding for this research was provided by the Royal Dutch/Shell Group, NGA NEGI-1582-04-0004, MURI Grant N00014-06-1-0734, US National Science Foundation (NSF) Career award (IIS 0747120), and a National Defense Science and Engineering Graduate Fellowship. Ce Liu wishes to thank Professor William T. Freeman and Professor Edward H. Adelson for insightful discussions.

References

[1] E.H. Adelson, "On Seeing Stuff: The Perception of Materials by Humans and Machines," *Proc. SPIE*, vol. 4299, pp. 1-12, 2001.

[2] S. Belongie, J. Malik, and J. Puzicha, "Shape Context: A New Descriptor for Shape Matching and Object Recognition," *Proc. Advances in Neural Information Processing Systems*, 2000.

[3] A. Berg, T. Berg, and J. Malik, "Shape Matching and Object Recognition Using Low Distortion Correspondence," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005.

[4] I. Borg and P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, second ed. Springer-Verlag, 2005.

[5] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie, "Visual Recognition with Humans in the Loop," *Proc. European Conf. Computer Vision*, 2010.

[6] M.J. Choi, J.J. Lim, A. Torralba, and A. Willsky, "Exploiting Hierarchical Context on a Large Database of Object Categories," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[7] D. Crandall, P. Felzenszwalb, and D. Huttenlocher, "Spatial Priors for Part-Based Recognition Using Statistical Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2005.

[8] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2005.

[9] C. Desai, D. Ramanan, and C. Fowlkes, "Discriminative Models for Multi-Class Object Layout," *Proc. IEEE Int'l Conf. Computer Vision,* 2009.

[10] S.K. Divvala, D. Hoiem, J.H. Hays, A.A. Efros, and M. Hebert, "An Empirical Study of Context in Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[11] G. Edwards, T. Cootes, and C. Taylor, "Face Recognition Using Active Appearance Models," *Proc. European Conf. Computer Vision,* 1998.

[12] A.A. Efros and T. Leung, "Texture Synthesis by Non-Parametric Sampling," *Proc. IEEE Int'l Conf. Computer Vision,* 1999.

[13] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A Discriminatively Trained, Multiscale, Deformable Part Model," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[14] P. Felzenszwalb and D. Huttenlocher, "Pictorial Structures for Object Recognition," *Int'l J. Computer Vision,* vol. 61, no. 1, pp. 55-79, 2005.

[15] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2003.

[16] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2003.

[17] A. Frome, Y. Singer, and J. Malik, "Image Retrieval and Classification Using Local Distance Functions," *Proc. Advances in Neural Information Processing Systems,* 2006.

[18] C. Galleguillos, B. McFee, S. Belongie, and G.R.G. Lanckriet, "Multi-Class Object Localization by Combining Local Contextual Interactions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[19] K. Grauman and T. Darrell, "Pyramid Match Kernels: Discriminative Classification with Sets of Image Features," *Proc. IEEE Int'l Conf. Computer Vision,* 2005.

[20] A. Gupta and L.S. Davis, "Beyond Nouns: Exploiting Prepositions and Comparative Adjectives for Learning Visual Classifiers," *Proc. European Conf. Computer Vision,* 2008.

[21] J. Hays and A.A. Efros, "Scene Completion Using Millions of Photographs," *ACM Trans. Graphics,* vol. 26, no. 3, 2007.

[22] G. Heitz and D. Koller, "Learning Spatial Context: Using Stuff to Find Things," *Proc. European Conf. Computer Vision,* 2008.

[23] D. Hoiem, A. Efros, and M. Hebert, "Putting Objects in Perspective," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2006.

[24] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 2169-2178, 2006.

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proc. IEEE,* vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[26] L. Liang, C. Liu, Y.Q. Xu, B.N. Guo, and H.Y. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling," *ACM Trans. Graphics,* vol. 20, no. 3, pp. 127-150, July 2001.

[27] C. Liu, J. Yuen, and A. Torralba, "Nonparametric Scene Parsing: Label Transfer via Dense Scene Alignment," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[28] C. Liu, J. Yuen, and A. Torralba, "SIFT Flow: Dense Correspondence across Different Scenes and Its Applications," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 5, pp. 978-994, May 2011.

[29] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W.T. Freeman, "SIFT Flow: Dense Correspondence across Different Scenes," *Proc. European Conf. Computer Vision,* 2008.

[30] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[31] K.P. Murphy, A. Torralba, and W.T. Freeman, "Using the Forest to See the Trees: A Graphical Model Relating Features, Objects, and Scenes," *Proc. Advances in Neural Information Processing Systems,* 2003.

[32] D. Nister and H. Stewenius, "Scalable Recognition with a Vocabulary Tree," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2006.

[33] S. Obdrzalek and J. Matas, "Sub-Linear Indexing for Large Scale Object Recognition," *Proc. British Machine Vision Conf.,* 2005.

[34] A. Oliva and A. Torralba, "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope," *Int'l J. Computer Vision,* vol. 42, no. 3, pp. 145-175, 2001.

[35] D. Park, D. Ramanan, and C. Fowlkes, "Multiresolution Models for Object Detection," *Proc. European Conf. Computer Vision,* 2010.

[36] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in Context," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[37] B.C. Russell, A.A. Efros, J. Sivic, W.T. Freeman, and A. Zisserman, "Segmenting Scenes by Matching Image Composites," *Proc. Advances in Neural Information Processing Systems,* 2009.

[38] B.C. Russell, A. Torralba, C. Liu, R. Fergus, and W.T. Freeman, "Object Recognition by Scene Alignment," *Proc. Advances in Neural Information Processing Systems,* 2007.

[39] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman, "LabelMe: A Database and Web-Based Tool for Image Annotation," *Int'l J. Computer Vision,* vol. 77, nos. 1-3, pp. 157-173, 2008.

[40] S. Savarese, J. Winn, and A. Criminisi, "Discriminative Object Class Models of Appearance and Shape by Correlatons," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2006.

[41] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast Pose Estimation with Parameter Sensitive Hashing," *Proc. IEEE Int'l Conf. Computer Vision,* 2003.

[42] E. Shechtman and M. Irani, "Matching Local Self-Similarities across Images and Videos," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[43] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context," *Int'l J. Computer Vision,* vol. 81, no. 1, pp. 2-23, 2009.

[44] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," *Proc. IEEE Int'l Conf. Computer Vision,* 2003.

[45] E. Sudderth, A. Torralba, W.T. Freeman, and W. Willsky, "Describing Visual Scenes Using Transformed Dirichlet Processes," *Proc. Advances in Neural Information Processing Systems,* 2005.

[46] J. Tighe and S. Lazebnik, "Superparsing: Scalable Nonparametric Image Parsing with Superpixels," *Proc. European Conf. Computer Vision,* 2010.

[47] A. Torralba, R. Fergus, and W.T. Freeman, "80 Million Tiny Images: A Large Dataset for Non-Parametric Object and Scene Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 11, pp. 1958-1970, Nov. 2008.

[48] M. Turk and A. Pentland, "Face Recognition Using Eigenfaces," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1991.

[49] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2001.

[50] M. Weber, M. Welling, and P. Perona, "Unsupervised Learning of Models for Recognition," *Proc. European Conf. Computer Vision,* 2000.

[51] J. Winn, A. Criminisi, and T. Minka, "Object Categorization by Learned Universal Visual Dictionary," *Proc. IEEE Int'l Conf. Computer Vision,* 2005.

[52] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba, "SUN Database: Large-Scale Scene Recognition from Abbey to Zoo," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[53] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes, "Layered Object Detection for Multi-Class Segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

Ce Liu received the BS degree in automation and the ME degree in pattern recognition from the Department of Automation, Tsinghua University in 1999 and 2002, respectively. After receiving the PhD degree from the Massachusetts Institute of Technology in 2009, he now holds a researcher position at Microsoft Research New England. From 2002 to 2003, he worked at Microsoft Research Asia as an assistant researcher. His research interests include computer vision, computer graphics, and machine learning. He has published more than 20 papers in the top conferences and journals in these fields. He received a Microsoft Fellowship in 2005, the Outstanding Student Paper award at the Advances in Neural Information Processing Systems (NIPS) in 2006, a Xerox Fellowship in 2007, and the Best Student Paper award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009. He is a member of the IEEE.

Jenny Yuen received the BS degree in computer engineering from the University of Washington in 2006, followed by the MS degree in computer science from the Massachusetts Institute of Technology in 2008. She is currently working toward the PhD degree in computer science at the Massachusetts Institute of Technology. She was awarded a National Defense Science and Engineering fellowship as well as a US National Science Foundation Fellowship in 2007. She received the Best Student Paper award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009. She is a student member of the IEEE.

Antonio Torralba received the degree in telecommunications engineering from the Universidad Politécnica de Cataluña, Spain; he received the PhD degree in signal, image, and speech processing from the Institut National Polytechnique de Grenoble, France. Thereafter, he spent postdoctoral training at the Brain and Cognitive Science Department and the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology (MIT). He is an associate professor of electrical engineering and computer science in the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT. He is a member of the IEEE.