Nonparametric Scene Parsing: Label Transfer via Dense Scene Alignment

Ce Liu Jenny Yuen Antonio Torralba
Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology
{celiu,jenny,torralba}@csail.mit.edu

Abstract

In this paper we propose a novel nonparametric approach for object recognition and scene parsing using dense scene alignment. Given an input image, we retrieve its best matches from a large database with annotated images using our modified, coarse-to-fine SIFT flow algorithm that aligns the structures within two images. Based on the dense scene correspondence obtained from the 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 a challenging database. Compared to existing object recognition approaches that require training for each object category, our system is easy to implement, has few parameters, and embeds contextual information naturally in the retrieval/alignment procedure.

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 [28, 5], constellations [9, 7], bags of features [24, 14, 10, 25], or shape models [2, 3, 6], etc. These approaches typically work with a fixed-number of object categories and require training generative or discriminative models for each category given 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 [11, 13]. 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 expand an existing system to include more object categories, we need to train new models for these categories and, typically adjust system parameters. Training can be a tedious job if we want to include thousands of object categories for a scene parsing system. In addition, the complexity of contextual relationships amongst objects also increases rapidly as the quantity of object categories expands.

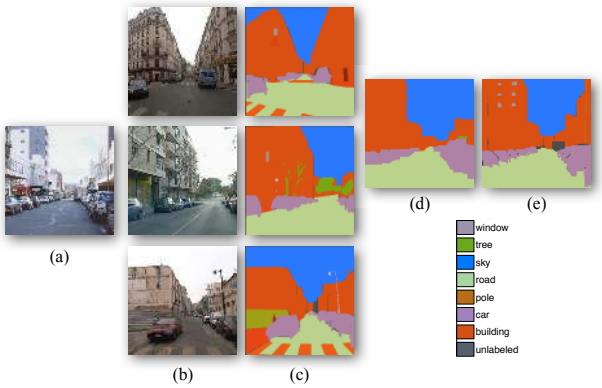


Figure 1. For a query image (a), our system finds the top matches (b) (three are shown here) using a modified, corse-to-fine SIFT flow matching algorithm. The annotations of the top matches (c) are transferred and integrated to parse the input image as shown in (d). For comparison, the ground-truth user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

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 [21], the authors estimate the pose of a human relying on 0.5 million training examples. In [12], 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 [19], a system is designed to infer the possible object categories that may appear in an image by retrieving similar images in a large database [20]. Moreover, the authors in [27] showed that with a database of 80 million images, even simple SSD match can give semantically meaningful parsing for 32×32 images.

Motivated by the recent advances in large database-driven approaches, we designed a nonparametric scene parsing system to transfer the labels from existing samples to annotate an image through dense scene alignment, as illustrated in Figure 1. For a query image (a), our system first retrieves the top matches in the LabelMe database [20] using a combination of GIST matching [18] and SIFT flow

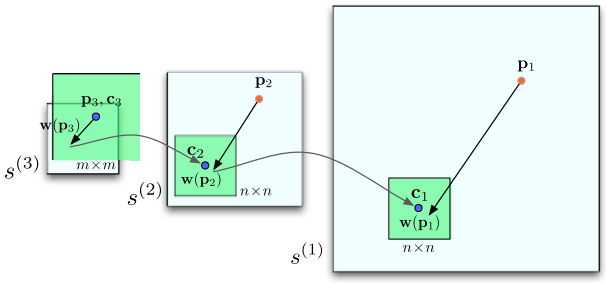


Figure 2. An illustration of our coarse-to-fine pyramid SIFT flow matching. The green square is the searching window for $\mathbf{p}_k$ at each pyramid level k. For simplicity only one image is shown here, where $\mathbf{p}_k$ is on image s_1, and $\mathbf{c}_k$ and $\mathbf{w}(\mathbf{p}_k)$ are on image s_2.

[16]. Since these top matches are labeled, we transfer the annotation (c) of the top matches to the query image and obtain the scene parsing result in (d). For comparison, the ground-truth user annotation of the query is displayed in (e). Our system is able to generate promising scene parsing results if images from the same scene category are retrieved in the annotated database.

However, it is nontrivial to build an efficient and reliable scene parsing system using dense scene alignment. The SIFT flow algorithm proposed in [16] does not scale well with image dimensions. Therefore, we propose a flexible, coarse-to-fine matching scheme to find dense correspondences between two images. 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 smoothness) into reliable annotation. Promising experimental results are achieved on images from the LabelMe database [20].

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 [5, 23] on our dataset.

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. Liu *et al*. [16] have demonstrated that SIFT flow is able to establish semantically meaningful correspondences between two images through matching local SIFT structures. In this section we extend the SIFT flow algorithm [16] to be more robust to matching outliers by modifying the objective function for matching, and more efficient for aligning large-scale images using a coarse-to-fine approach.

2.1. Modified matching objective

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 feature [17] for two images [1], and ε contains all the spatial neighborhood (a four-neighbor system is used). Our modified energy function is defined as:

$$E(\mathbf{w}) = \sum_{\mathbf{p}} \min\Big(\big\|s_1(\mathbf{p}) - s_2(\mathbf{p}+\mathbf{w}(\mathbf{p}))\big\|_1, t\Big) + \sum_{\mathbf{p}} \eta\Big(|u(\mathbf{p})| + |v(\mathbf{p})|\Big) + \sum_{(\mathbf{p},\mathbf{q})\in\varepsilon} \min\Big(\alpha|u(\mathbf{p}) - u(\mathbf{q})|, d\Big) + \sum_{(\mathbf{p},\mathbf{q})\in\varepsilon} \min\Big(\alpha|v(\mathbf{p}) - v(\mathbf{q})|, d\Big). \quad (1)$$

In this objective function, truncated L1 norms are used in both the data and the smoothness terms to account for matching outliers and flow discontinuities, with t and d as the threshold, respectively. An L1 norm is also imposed on the magnitude of the flow vector as a bias towards smaller displacement when no other information is available. Notice that in [16] only an L1 norm is used for the data term and the small displacement biased is formulated as an L2 norm. This energy function is optimized by running sequential Belief Propagation (BP-S) [26] on a dual plane setup [22].

2.2. Coarse-to-fine matching scheme

While SIFT flow has demonstrated the potential for aligning images across scenes [16], its performance 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 BP algorithm to estimate the SIFT flow is $O(h^4)$. As reported in [16], 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 two hours to process a pair of 256×256 images in our database with a memory usage of 16GB to store the data term.

To address the performance drawback, we designed a coarse-to-fine SIFT flow matching scheme that significantly improves the performance. The procedure is illustrated in Figure 2. For simplicity, we use s to represent both s_1 and s_2. A SIFT pyramid $\{s^{(k)}\}$ is established, where $s^{(1)} = s$ and $s^{(k+1)}$ is smoothed and downsampled from $s^{(k)}$. At each pyramid level k, let $\mathbf{p}_k$ be the coordinate of the pixel to match, $\mathbf{c}_k$ be the offset or centroid of the searching window, and $\mathbf{w}(\mathbf{p}_k)$ be the best match from BP. At the top pyramid level $s^{(3)}$, the searching window is centered at $\mathbf{p}_3$ ($\mathbf{c}_3 = \mathbf{p}_3$) with size $m \times m$, where m is the width (height) of $s^{(3)}$. The complexity of BP at this level is $O(m^4)$. After BP

[1] 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 128-D vector.

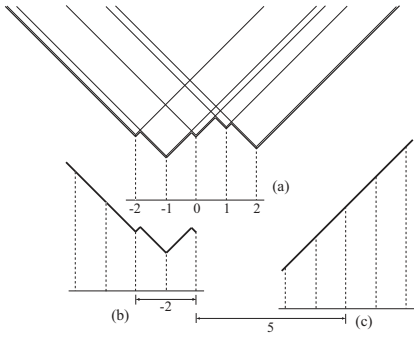


Figure 3. We generalized distance transform function for truncated L1 norm [8] to pass message between neighboring nodes that have different offsets (centroids) of the searching window.

converges, the system propagates the optimized flow vector $\mathbf{w}(\mathbf{p}_3)$ to the next (finer) level to be $\mathbf{c}_2$ where the searching window of $\mathbf{p}_2$ is centered. The size of this searching window is fixed to be $n \times n$ with $n = 11$. This procedure iterates from $s^{(3)}$ to $s^{(1)}$ until the flow vector $\mathbf{w}(\mathbf{p}_1)$ is estimated. Since n is fixed at all levels except for the top, the complexity of this coarse-to-fine algorithm is $O(h^2 \log h)$, a significant speed up compared to $O(h^4)$.

When the matching is propagated from an coarser level to the finer level, the searching windows for two neighboring pixels may have different offsets (centroids). We modify the the distance transform function developed for truncated L1 norm [8] to cope with this situation, with the idea illustrated in Figure 3. To compute the message passing from pixel $\mathbf{p}$ to its neighbor $\mathbf{q}$, we first gather all other messages and data term, and apply the routine in [8] to compute the message from $\mathbf{p}$ to $\mathbf{q}$ assuming that $\mathbf{q}$ and $\mathbf{p}$ have the same offset and range. The function is then extended to be outside the range by increasing α per step, as shown in Figure 3 (a). We take the function in the range that $\mathbf{q}$ is relative to $\mathbf{p}$ as the message. For example, if the offset of the searching window for $\mathbf{p}$ is 0, and the offset for $\mathbf{q}$ is 5, then the message from $\mathbf{p}$ to $\mathbf{q}$ is plotted in Figure 3 (c). If the offset of the searching window for $\mathbf{q}$ is -2 otherwise, the message is shown in Figure 3 (b).

Using the proposed coarse-to-fine matching scheme and modified distance transform function, the matching between two 256×256 images takes 31 seconds on a workstation with two quad-core 2.67 GHz Intel Xeon CPUs and 32 GB memory, in a C++ implementation. Further speedup (up to 50x) can be achieved through GPU implementation [4] of the BP-S algorithm since this algorithm can be parallelized. We leave this as future work.

A natural question is whether the coarse-to-fine matching scheme can achieve the same minimum energy as the ordinary matching scheme (only one level without coarse-to-fine) [16]. An experiment is conducted to compare these two algorithms (refer to Section 4.1 for more details). In general, we found that the coarse-to-fine matching outperforms the ordinary matching in terms of obtaining lower energy. This is consistent with what has been discovered in the optical flow community: coarse-to-fine search not only speeds up computation but also leads to lower energy. This can be caused by the inherent self-similarity nature of SIFT features across scales: the correspondence at a coarser level is a good prediction for the correspondence at a finer level.

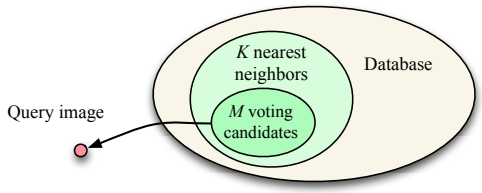


Figure 4. For a query image, we first find a K-nearest neighbor set in the database using GIST matching [18]. The nearest neighbors are re-ranked using SIFT flow matching scores, and form a top M-voting candidate set. The annotations are transferred from the voting candidates to the query image.

3. Scene Parsing through Label Transfer

Now that we have a large database of annotated images and a technique of establishing dense correspondences across scenes, we can transfer the existing annotations to a query image through dense scene alignment. For a given query image, we retrieve a set of *K-nearest neighbors* in our database using GIST matching [18]. We then compute the SIFT flow from the query to each nearest neighbor, and use the achieved minimum energy (defined in Eqn. 1) to re-rank the K-nearest neighbors. We further select the top M re-ranked retrievals 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 Figure 4.

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. c_i is an integer image where $c_i(\mathbf{p}) \in \{1, \cdots, L\}$ 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. Similar to that of [23], the posterior probability is defined as:

$$-\log P\big(c|I, s, \{s_i, c_i, \mathbf{w}_i\}\big) = \sum_{\mathbf{p}} \psi\big(c(\mathbf{p}); s, \{s'_i\}\big) + \alpha \sum_{\mathbf{p}} \lambda\big(c(\mathbf{p})\big) + \beta \sum_{\{\mathbf{p},\mathbf{q}\}\in\varepsilon} \phi\big(c(\mathbf{p}), c(\mathbf{q}); I\big) + \log Z, \quad (2)$$

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

$$\psi\big(c(\mathbf{p})=l\big)=\begin{cases}\min\limits_{i\in\Omega_{\mathbf{p},l}} \|s(\mathbf{p})-s_i(\mathbf{p}+\mathbf{w}(\mathbf{p}))\|, & \Omega_{\mathbf{p},l}\neq\emptyset\\ \tau, & \Omega_{\mathbf{p},l}=\emptyset\end{cases} \tag{3}$$

where $\Omega_{\mathbf{p},l}=\{i; c_i(\mathbf{p}+\mathbf{w}(\mathbf{p}))=l\}$ 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 is $\lambda(c(\mathbf{p})=l)$ 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.

$$\lambda\big(c(\mathbf{p})=l\big)=-\log \mathrm{hist}_l(\mathbf{p}) \tag{4}$$

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 if no other information is available, and the probability depends on the edge of the image: the stronger luminance edge, the more likely that the neighboring pixels may have different labels.

$$\phi\big(c(\mathbf{p}),c(\mathbf{q})\big)=\delta[c(\mathbf{p})\neq c(\mathbf{q})]\left(\frac{\epsilon+e^{-\gamma\|I(\mathbf{p})-I(\mathbf{q})\|^2}}{\epsilon+1}\right) \tag{5}$$

where $\gamma=(2<\|I(\mathbf{p})-I(\mathbf{q})\|^2>)^{-1}$ [23].

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 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 [23] is that we have fewer parameters because of the nonparametric nature of our approach, whereas classifiers where trained in [23]. 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 database.

4. Experiments

We used a subset of the LabelMe database [20] to test our system. This dataset contains 2688 fully annotated images, most of which are outdoor scenes including street, beach, mountains, fields and buildings. From these images we randomly selected 2488 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 Figure 5. 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. The top 10 object categories are *sky*, *building*, *mountain*, *tree*, *unlabeled*, *road*, *sea*, *field*, *grass*, and *river*. The spatial priors of these object categories are displayed at the bottom of Figure 5. White means zero probability and saturated color means the highest probability. We observe that sky occupies the upper part of image grid and field occupies the lower part. Notice that there are only limited numbers of samples for the objects such as *sun*, *cow*, *bird*, and *moon*.

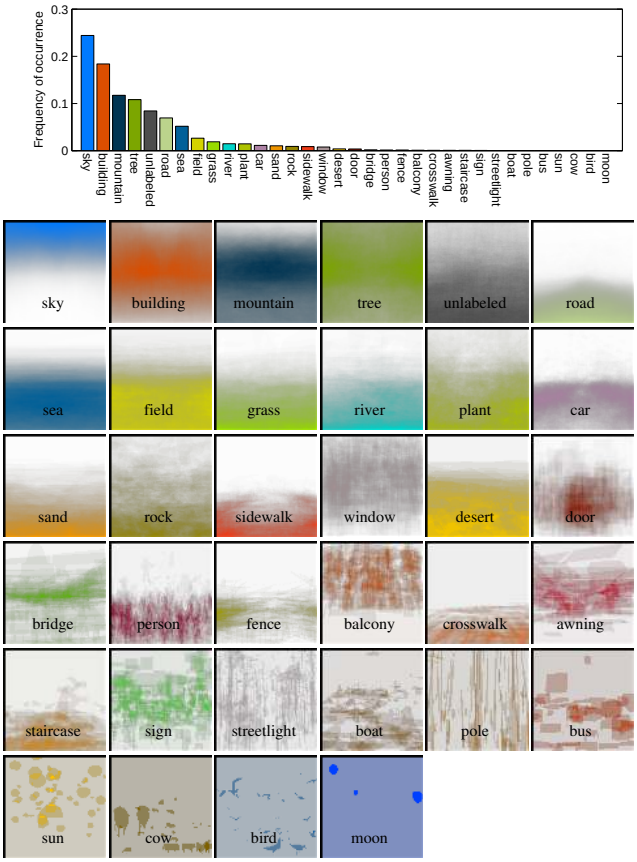


Figure 5. Above: the per-pixel frequency counts of the object categories in our dataset (sorted in descending order). The color of each bar is the average RGB value of each object category from the training data with saturation and brightness boosted for visualization. Bottom: the spatial priors of the object categories in the database. White means zero and the saturated color means high probability.

4.1. Evaluation of the dense scene alignment

We first evaluate our coarse-to-fine SIFT flow matching for dense scene alignment. We randomly selected 10 images from the test set as the query, and check the minimum energy obtained between the query and the best SIFT flow match using coarse-to-fine scheme and ordinary scheme (non coarse-to-fine), respectively. For these 256×256 images, the average running time coarse-to-fine SIFT flow is 31 seconds, whereas it takes 127 minutes in average for the ordinary matching. The coarse-to-fine scheme not only runs

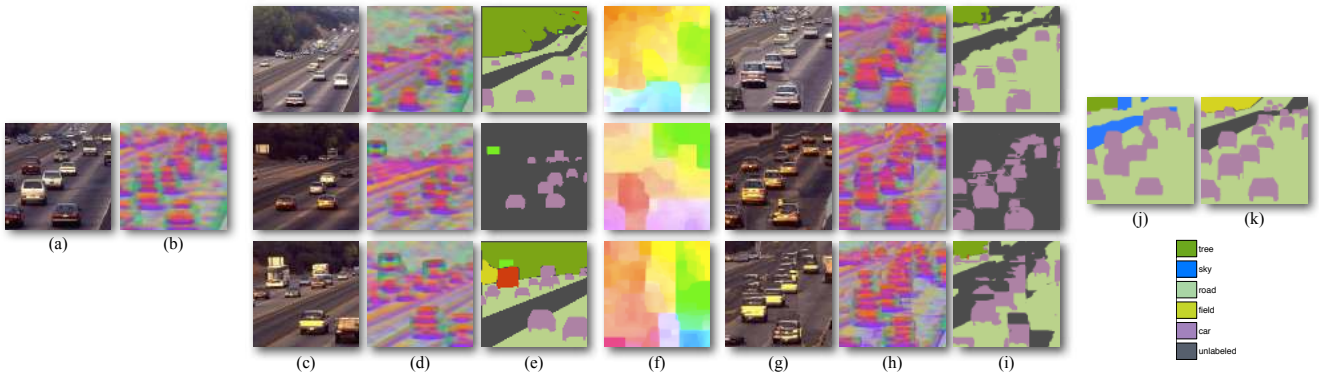


Figure 6. System overview. Our algorithm computes the SIFT image (b) of an query image (a), and uses GIST [18] to find its K nearest neighbors in our database. We apply coarse-to-fine SIFT flow to align the query image to the nearest neighbors, and obtain top M as voting candidates ($M = 3$ here). (c) to (e): the RGB image, SIFT image and user annotation of the voting candidates. (f): the inferred SIFT flow. From (g) to (i) are the warped version of (c) to (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).

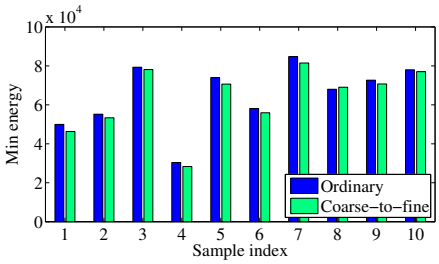


Figure 7. Coarse-to-fine SIFT flow not only runs significantly faster, but also achieves lower energies most of the time. In this experiment, we randomly selected 10 samples in the test set and computed the lowest energy of the best match with the nearest neighbors. We tested both the coarse-to-fine algorithm proposed in this paper and the ordinary matching scheme in [16]. Except for sample #8, coarse-to-fine matching achieves lower energy than the ordinary matching algorithm.

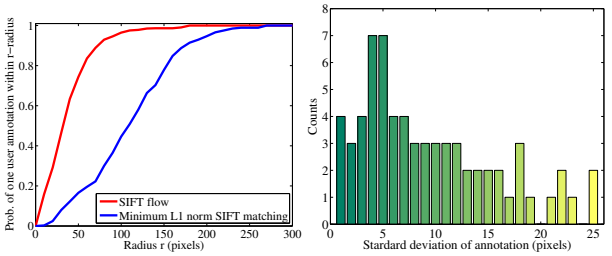


Figure 8. The evaluation of SIFT flow using human annotation. Left: the probability of one human annotated flow lies within r distance to the SIFT flow as a function of r (red curve). For comparison, we plot the same probability for direct minimum L1-norm matching (blue curve). Clearly SIFT flow matches human perception better. Right: the histogram of the standard deviation of human annotation. Human perception of scene correspondence varies from subject to subject.

significantly faster, but also achieves lower energies most of the time compared to the ordinary matching algorithm [16] as shown in Figure 7.

Before evaluating the performance our system on object recognition, we want to evaluate how well SIFT flow performs in matching structures across different images and how it compares with human selected matches. Traditional optical flow is a well-defined problem and it is straightforward for humans to annotate motion for evaluation [15]. In the case of SIFT flow, however, there may not be obvious or unique best pixel-to-pixel matching as the two images may contain different objects, or the same object categories with very different instances.

To evaluate the matching obtained by SIFT flow, we performed a user study where we showed 11 users image pairs with preselected sparse points in the first image and asked the users to select the corresponding points in the second image. As shown on the right of Figure 8, user annotation can be ambiguous. Therefore, we use the following metric to evaluate SIFT flow: for a pixel $\mathbf{p}$, we have several human annotations $\mathbf{z}_i$ as its flow vector, and $\mathbf{w}(\mathbf{p})$ as the estimated SIFT flow vector. We compute $\Pr\big(\exists \mathbf{z}_i, \|\mathbf{z}_i - \mathbf{w}(\mathbf{p})\| \leq r|r\big)$, namely the probability of one human annotated flow is within distance r to SIFT flow $\mathbf{w}(\mathbf{p})$. This function (or r is plotted on the left of Figure 8 (red curve). For comparison, we plot the same probability function (blue curve) for minimum L1-norm SIFT matching, *i.e.* SIFT flow matching without spatial terms. Clearly SIFT flow matches better to human annotation than minimum L1-norm SIFT matching.

4.2. Results of scene parsing

Our scene parsing system is illustrated in Figure 6. The system retrieves a K-nearest neighbor set for the query image (a), and further selects M voting candidates with the minimum SIFT matching score. For the purpose of illustration we set $M = 3$ here. The RGB image, SIFT image, and annotation of the voting candidates are shown in (c) to (e), respectively. The SIFT flow field is visualized in (f) using

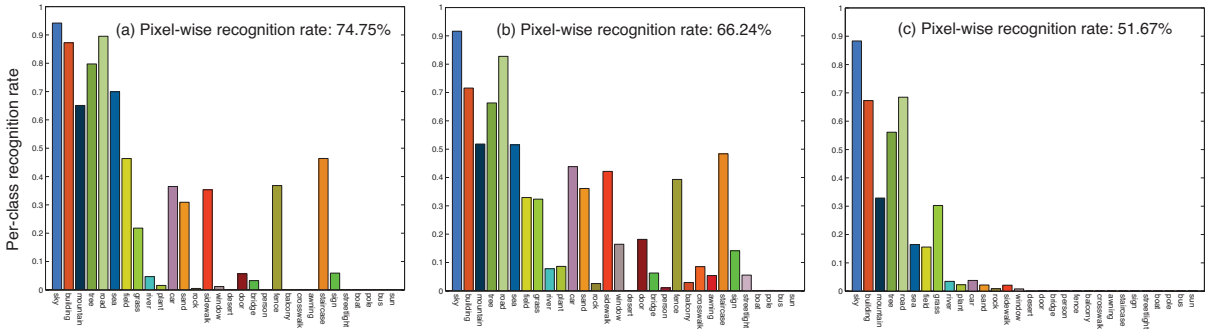


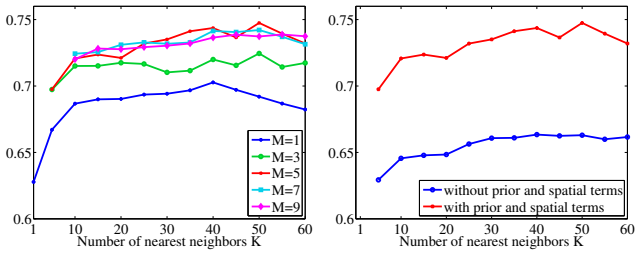
Figure 9. The per-class recognition rate of our system and the one in [23]. (a) Our system with the the parameters optimized for pixel-wise recognition rate. (b) Our system with $\alpha = \beta = 0$, namely, with the Markov random field model turned off. (c) The performance of the system in [23] also with the conditional random field turned off, trained and tested on the same data sets as (a) and (b).

the same visualization scheme as in [16]. After we warp the voting candidates into the query with respect to the flow field, the warped RGB (g) and SIFT image (h) are very close to the query (a) and (b). Combining the warped annotations in (i), the system outputs the parsing of the query in (j), which is close to the ground-truth annotation in (k).

Some label transferring results are displayed in Figure 12. The input image from the test set is displayed in column (a). We show the best match, its corresponding annotation, and the warped best match in (b), (c) and (d), respectively, to hint the annotation for the query, even though our system takes the top M matches as voting candidates. Again, the warped image (d) looks similar to the input, indicating that SIFT flow successfully matches image structures. The scene parsing results output from our system are listed in column (e) with parameter setting $K = 50,\ M = 5,\ \alpha = 0.1, \beta = 70$. The ground-truth user annotation is listed in (f). Notice that the gray pixels in (f) are "unlabeled", but our system does not generate "unlabeled" output. For sample 1, 5, 6, 8 and 9, our system generates reasonable predictions for the pixels annotated as "unlabeled". The pixel-wise recognition rate of our system is **74.75%** by excluding the "unlabeled" class [23]. A failure example for our system is shown in Figure 13 when the system fails to retrieve images with similar object categories to the query.

For comparison, we downloaded and ran the code from [23] using the same training and test data with the conditional random field turned off. The overall pixel-wise recognition rate of their system on our data set is **51.67%**, and the per-class rates are displayed in Figure 9 (c). For fairness we also turned off the Markov random field model in our framework by setting $\alpha = \beta = 0$, and plotted the corresponding results in Figure 9 (b). Clearly, our system outperforms [23] in terms of both overall and per-class recognition rate.

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, 13], our system generates much better results for stuff than for things.



(a) (b)

Figure 10. (a): Recognition rate as a function of the number of nearest neighbors K and the number of voting candidates M. (b): recognition rate as a function of the number of nearest neighbors K. Clearly, prior and spatial smoothness help improve the recognition rate.

The recognition rate for the top 7 object categories (all are "stuff") is 82.72%. 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.

We investigate the performance of our system by varying the parameters K, M, α and β. First, we fix $\alpha = 0.1,\ \beta = 70$ and plot the recognition rate as a function of K in Figure 10 (a) with different M. 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 drops as K and M continue to increase as more candidates may include noise to label transfer. The maximum performance is obtained when $K = 50$ and $M = 5$. Second, we fix $M = 5$, and plot the recognition rate as a function of K by turning on prior and spatial terms ($\alpha = 0.1,\ \beta = 70$) and turning them off ($\alpha = \beta = 0$) in Figure 10 (b). Prior and spatial smoothness increase the performance of our system by about 7 percentage.

Lastly, we compared the performance of our system with a classifier-based system [5]. 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

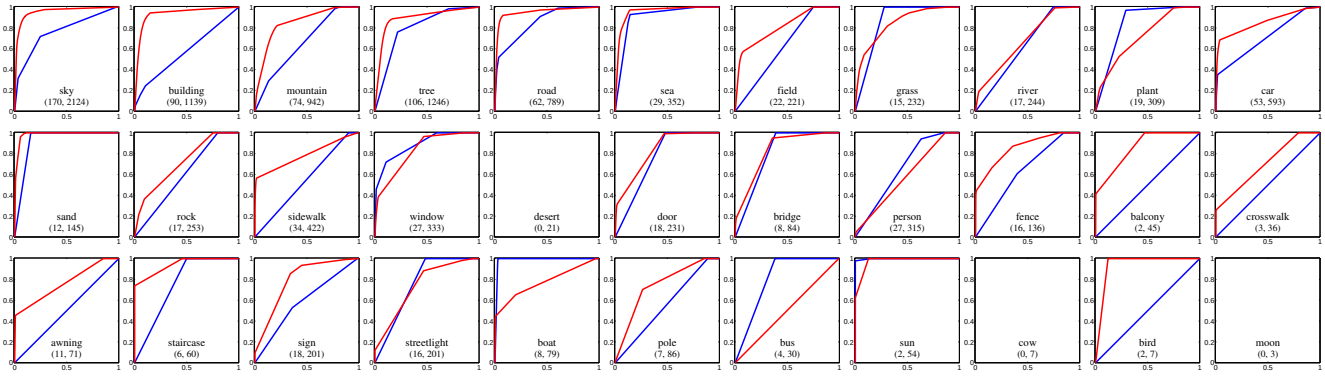


Figure 11. The ROC curve of each individual pixel-wise binary classifier. Red curve: our system after being converted to binary classifiers; blue curve: the system in [5]. We used convex hull to make the ROC curves strictly concave. The number (n, m) underneath the name of each plot is the quantity of the object instances in the test and training set, respectively. For example, (170, 2124) under "sky" means that there are 170 test images containing sky, and 2124 training images containing sky. 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 [5] for all object categories except for grass, plant, boat, person and bus. The performance of [5] on our database is low because the objects have drastically different poses and appearances.

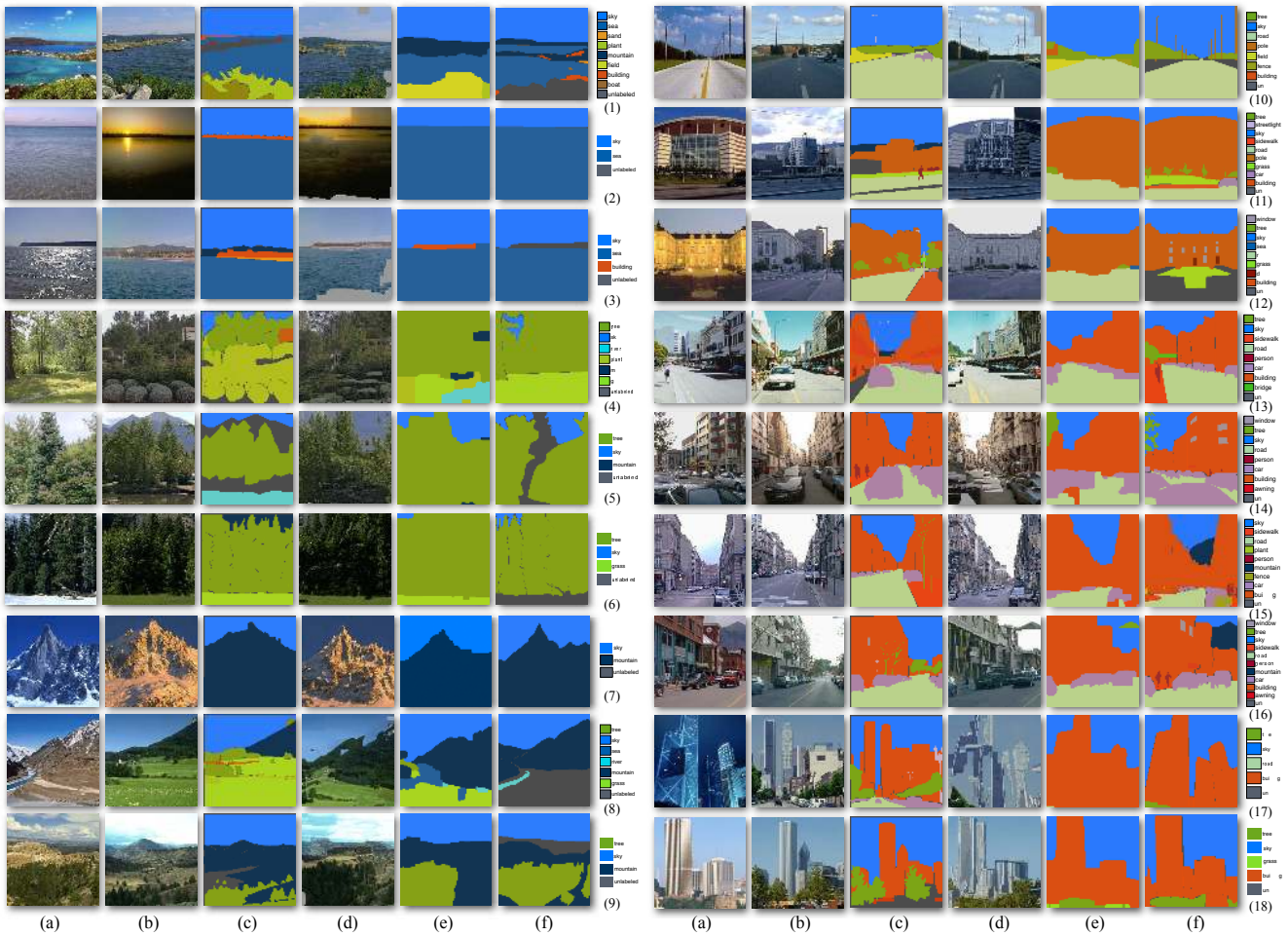
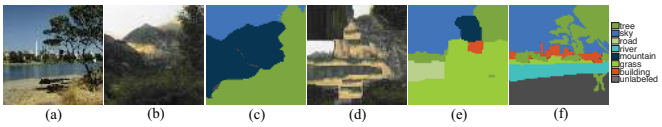


Figure 12. Some scene parsing results output from our system. (a): query image; (b): the best match from nearest neighbors; (c): the annotation of the best match; (d): the warped version of (b) according to the SIFT flow field; (e): the inferred per-pixel parsing after combining multiple voting candidates; (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.



(a) (b) (c) (d) (e) (f)

Figure 13. Our system fails when no good matches can be retrieved in the database. 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).

(red) and theirs (blue) are plotted in Figure 11. Except for five object categories, *grass*, *plant*, *boat*, *person* and *bus*, our system outperforms or equals theirs.

5. Conclusion

We presented a novel, nonparametric scene parsing system to transfer the annotations from a large database to an input image using 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 a challenging database. 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.

6. Acknowledgements

Funding for this research was provided by the Royal Dutch/Shell Group, NGA NEGI-1582-04-0004, MURI Grant N00014-06-1-0734, NSF Career award (IIS 0747120), and a National Defense Science and Engineering Graduate Fellowship.

References

[1] E. H. Adelson. On seeing stuff: the perception of materials by humans and machines. In *SPIE, Human Vision and Electronic Imaging VI*, pages 1–12, 2001.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, 2000.

[3] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR*, 2005.

[4] N. Cornelis and L. V. Gool. Real-time connectivity constrained depth map computation using programmable graphics hardware. In *CVPR*, pages 1099–1104, 2005.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[6] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1), 2005.

[7] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006.

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

[10] K. Grauman and T. Darrell. Pyramid match kernels: Discriminative classification with sets of image features. In *ICCV*, 2005.

[11] A. Gupta and L. S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*, 2008.

[12] J. Hays and A. A. Efros. Scene completion using millions of photographs. *ACM SIGGRAPH*, 26(3), 2007.

[13] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV*, 2008.

[14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, volume II, pages 2169–2178, 2006.

[15] C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss. Human-assisted motion annotation. In *CVPR*, 2008.

[16] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. SIFT flow: dense correspondence across different scenes. In *ECCV*, 2008.

[17] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, Kerkyra, Greece, 1999.

[18] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[19] B. C. Russell, A. Torralba, C. Liu, R. Fergus, and W. T. Freeman. Object recognition by scene alignment. In *NIPS*, 2007.

[20] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: a database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008.

[21] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *ICCV*, 2003.

[22] A. Shekhovtsov, I. Kovtun, and V. Hlavac. Efficient MRF deformation model for non-rigid image matching. In *CVPR*, 2007.

[23] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[24] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *ICCV*, 2003.

[25] E. Sudderth, A. Torralba, W. T. Freeman, and W. Willsky. Describing visual scenes using transformed dirichlet processes. In *NIPS*, 2005.

[26] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. *TPAMI*, 30(6):1068–1080, 2008.

[27] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *TPAMI*, 2008.

[28] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.