Recognizing Scene Viewpoint using Panoramic Place Representation

Jianxiong Xiao jxiao@mit.edu
Krista A. Ehinger kehinger@mit.edu
Aude Oliva oliva@mit.edu
Antonio Torralba torralba@mit.edu
Massachusetts Institute of Technology

Abstract

We introduce the problem of scene viewpoint recognition, the goal of which is to classify the type of place shown in a photo, and also recognize the observer's viewpoint within that category of place. We construct a database of 360° panoramic images organized into 26 place categories. For each category, our algorithm automatically aligns the panoramas to build a full-view representation of the surrounding place. We also study the symmetry properties and canonical viewpoint of each place category. At test time, given a photo of a scene, the model can recognize the place category, produce a compass-like indication of the observer's most likely viewpoint within that place, and use this information to extrapolate beyond the available view, filling in the probable visual layout that would appear beyond the boundary of the photo.

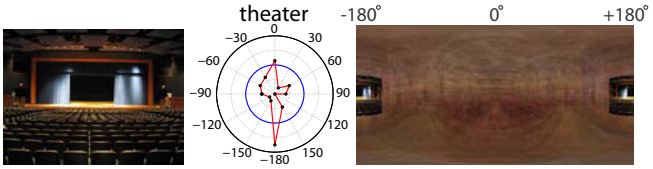
1. Introduction

The pose of an object carries crucial semantic meaning for object manipulation and usage (e.g., grabbing a mug, watching a television). Just as pose estimation is part of object recognition, viewpoint recognition is a necessary and fundamental component of scene recognition. For instance, as shown in Figure 1, a theater has a clear distinct distributions of objects – a stage on one side and seats on the other – that defines unique views in different orientations. Just as observers will choose a view of a television that allows them to see the screen, observers in a theater will sit facing the stage when watching a show.

Although the viewpoint recognition problem has been well studied in objects [12], most research in scene recognition has focused exclusively on the classification of views. There are many works that emphasize different aspects of scene understanding [18, 3, 4, 5, 8], but none of them make a clear distinction between views and places. In fact, current scene recognition benchmarks [7, 17] define categories that are only relevant to the classification of single views. However, recent evidence [10, 6] suggests that humans have a world-centered representation of the surrounding space, which is used to integrate views into a larger spatial context and make predictions about what exists beyond the available field of view.

(a) Despite belonging to the same place category (e.g., theater), the photos taken by an observer inside a place look very different from different viewpoints. This is because typical photos have a limited visual field of view and only capture a single scene viewpoint (e.g., the stage) at a time.

(b) We use panoramas for training place categorization **and** viewpoint recognition models, because they densely cover all possible views within a place.



(c) Given a limited-field-of-view photo as testing input (left), our model recognizes the place category, and produces a compass-like prediction (center) of the observer's viewpoint. Superimposing the photo on an averaged panorama of many theaters (right), we can automatically predict the possible layout that extends beyond the available field of view.

Figure 1. Scene viewpoint recognition within a place.

The goal of this paper is to study the viewpoint recognition problem in scenes. We aim to design a model which, given a photo, can classify the place category to which it belongs (e.g., a theater), and predict the direction in which the observer is facing within that place (e.g., towards the stage). Our model learns the typical arrangement of visual features in a 360° panoramic representation of a place, and learns to map individual views of a place to that represen-

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

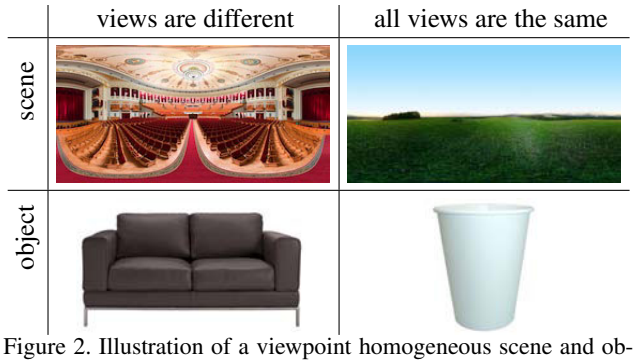


Figure 2. Illustration of a viewpoint homogeneous scene and object.

tation. Now, given an input photo, we will be able to place that photo within a larger panoramic image. This allows us to extrapolate the layout beyond the available view, as if we were to rotate the camera all around the observer.

We also study the symmetry property of places. Like objects, certain places exhibit rotational symmetry. For example, an observer standing in an open field can turn and see a nearly-identical view in every direction (isotropic symmetry). Similarly, a cup presents similar views when it is rotated. Other objects (e.g., sofa) and places (e.g., theaters) do not have rotational symmetry, but present some views that are mirror images of each other (left and right sides of the sofa, left and right views of the theater). Still other objects and places have no similar views (asymmetry). We design our algorithm to automatically discover the symmetry type of each place category and incorporate this information into its panoramic representation.

2. Overview

We aim to train a model to predict place category and viewpoint for a photo. We use 360-degree full-view panoramas for training, because they unbiasedly cover all views in a place. The training data are 26 place categories (Figure 6), each of which contains many panoramas without annotation. We design a two-stage algorithm to train and predict the place category, and then the viewpoint. We first generate a set of limited-field-of-view photos from panoramas to train a 26-way classifier to predict the place category (Section 3). Then, for each place category, we automatically align the panoramas and learn a model to predict the scene viewpoint (Section 4). We model the viewpoint prediction as a m-way classification problem (m=12 in our experiments) to assign a given photo into one of the m viewpoints, uniformly sampled from different orientations on the 0°-pitch line. Simultaneously with the alignment and classification, we automatically discover the symmetry type of each place category (Section 5).

Given a view of a place, our model can infer the semantic category of the place and identify the observer's viewpoint within that place. We can even extrapolate the possible layout that extends beyond the available field of view, as illustrated in Figure 1(c) and 8. We represent the extrapolated layout using either the average panorama of the place category, or the nearest neighbor from the training examples. We can clearly see how correct view alignment allows us to predict visual information, such as edges, that extend beyond the boundaries of the test photo. If the objects in the training panoramas are annotated, the extrapolated layout generated by our algorithm can serve as a semantic context priming model for object detections, even when those objects are outside the available field of view.

Terminology We will use the term *photo* to refer to a standard limited-field-of-view image as taken with a normal camera (Figure 1(a)) and the term *panorama* to denote a 360-degree full-view panoramic image (Figure 1(b)). We use the term *place* to refer to panoramic image categories, and use the term *scene category* to refer to semantic categories of photos. We make this distinction because a single panoramic place category can contain views corresponding to many different scene categories (for example, *street* and *shopfront* are treated as two different scene categories in [17], but they are just different views of a single place category, *street*).

3. Place category classifier

We use a non-linear kernel Support Vector Machine (SVM) to train a 26-way classifier to predict the place category of a photo. The training data are 12 limited-field-of-view photos sampled from each panorama in each category, uniformly across different view angles in the horizontal orientation. Although we regard all different viewpoints from the same category as the same class in this stage, the kernel SVM with non-linear decision boundary nevertheless gives very good partition of positives and negatives. Refer to Section 7 for details on the dataset construction, features, kernels and classifiers that we used.

4. Panorama alignment & viewpoint classifier

For each place category, we align the panoramas for training, and train the viewpoint recognition model. For each category, if we know the alignment of panoramas, we can train a m-way classifier for m viewpoints, using the limited-field-of-view photos for each viewpoint sampled from the panoramas. However, the alignment is unknown and we need to simultaneously align the panoramas and train the classifier. Here, we propose a very simple but powerful algorithm, which starts by training the viewpoint classifier using only one panorama, and then incrementally adds a new panorama into the training set in each iteration.

At the first iteration, training with only one panorama requires no alignment at all, and we can learn a meaningful viewpoint classifier easily. In each subsequent iteration, as illustrated in Figure 3, we use the current classifier to predict the viewpoint for the rest of the photos sampled from

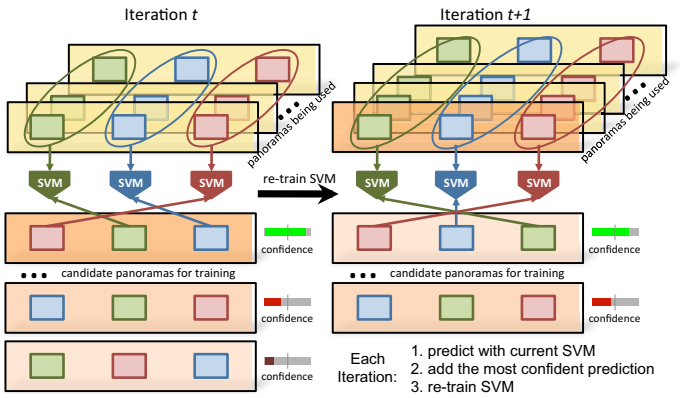


Figure 3. Incremental algorithm for simultaneous panorama alignment and viewpoint classifier training. Each long rectangle denotes a panorama, and each smaller rectangle inside the panorama denotes a limited-field-of-view photo generated from the panorama.

unused panoramas. We pick the panorama with the highest overall confidence in the classifier prediction, add all its generated photos into the training set of the classifier, and retrain the classifier. The same process continues until all panoramas have been used for training. This produces a viewpoint classifier and an alignment for all panoramas at the same time.

This process exploits the fact that the most confident prediction of a nonlinear kernel SVM classifier is usually correct, and therefore we maintain a high-quality alignment for re-training a good viewpoint classifier at each iteration. Of course, starting with a different panorama results in a different alignment and model. Therefore, we exhaustively try each panorama as the starting point and use cross validation to pick the best model.

In each iteration, the viewpoint classifier predicts results for all m limited-field-of-view photos generated from the same panorama, which have known relative viewpoints. Hence, we can use this viewpoint relation as a hard constraint to help choose the most confident prediction and assign new photos into different viewpoints for re-training. Furthermore, in each iteration, we also adjust all the previously used training panoramas according to the current viewpoint classifier's predictions. In this way, the algorithm is able to correct mistakes made in earlier iterations during the incremental assignment.

When selecting the most confident prediction, we use the viewpoint classifier to predict on both the original panorama and a horizontally-flipped version of the same panorama, and pick the one with higher confidence[1]. This allows the algorithm to handle places which have two types of layouts which are 3D mirror reflection of each other. (For example, in a hotel room, the bed may be located to the left of the doorway or to the right of the doorway – the spatial layout of the room is the same, only flipped.) Because we give the algorithm the freedom to horizontally flip the panorama, these two types of layout can be considered as just one layout to train a better model with better alignment.

[1]In order to avoid adding an artificial symmetry structure to the data, only one of the original panorama and the flipped panorama is allowed to participate in the alignment.

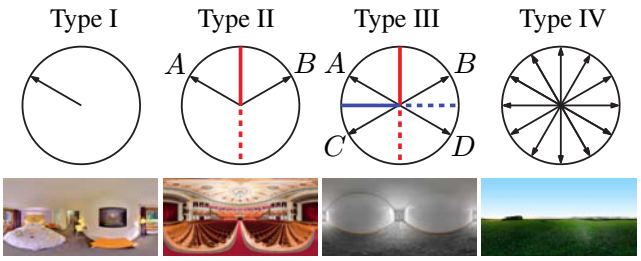


Figure 4. Four types of symmetry structure found in place categories: Type I (asymmetry), Type II (bilateral symmetry with one axis); Type III (bilateral symmetry with two axis); Type IV (isotropic symmetry). Each circle represents a panorama as seen from above, arrows represent camera viewpoints that are similar, and **red** and **blue** lines denote symmetry axes. The second row shows example panoramas for each type of symmetry. See Section 5 for a detailed description.

5. Symmetry discovery and view sharing

Many places have a natural symmetry structure, and we would like to design the model to automatically discover the symmetry structure of the place categories in the data set. A description of place symmetry is a useful contribution to scene understanding in general, and it also allows us to borrow views across the training set, increasing the effective training set size. This does not solve the ambiguity inherent in recognizing views in places with underlying symmetry, but it will reduce the model's errors to other, irrelevant views. This is illustrated in Figure 9(c): the model trained with no sharing of training examples (top figure) has more errors around 90°, and hence less frequent detections on 0° and 180°. With sharing of training examples (bottom figure), the errors at 90° are reduced, and the frequency of detections on 0° and 180° are increased, which means that the accuracy is increased. Finally, understanding the symmetry structure of places allows us to train a simpler model with fewer parameters, and a simpler model is generally preferred to explain data with similar goodness of fit. Here, we consider four common types of symmetry structure found in place categories, shown in Figure 4:

Type I: Asymmetry. There is no symmetry in the place. Every view is unique.

Type II: Bilateral Symmetry with One Axis. Each view matches a horizontally-flipped view that is mirrored with respect to an axis.

Type III: Bilateral Symmetry with Two Axes. Same as Type III but with two axes that are perpendicular to each other. This type of symmetry also implies 180° rotational symmetry.

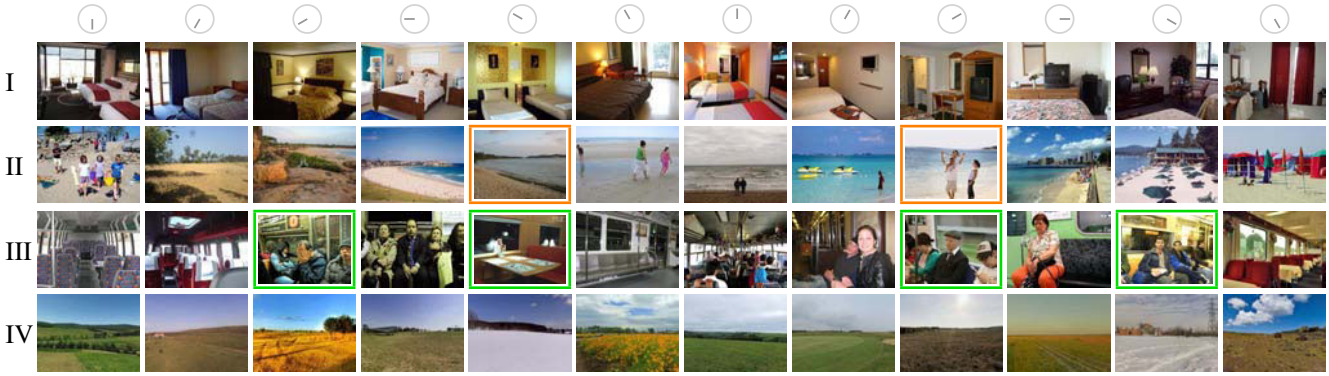


Figure 5. Example photos from various viewpoints of 4 place categories. Each row shows typical photos from 12 different viewpoints in a category: hotel room (Type I), beach (Type II), street (Type III), field (Type IV). The **grey** compasses on the top indicate the viewpoint depicted in each column. We can see how different views can share similar visual layouts due to symmetry. For example, the two photos with **orange** frames are a pair with type II symmetry, and the four photos with **green** frames are a group with type III symmetry.

Type IV: Isotropic Symmetry. Every view looks the same.

To allow view sharing with symmetry, we need to modify the alignment and viewpoint classifier training process proposed in Section 4 for each type of symmetry as follows:

Type I: The algorithm proposed in the previous section can be applied without any modification in this case.

Type II: Assuming we know the axis of symmetry, we need to train a m-way classifier as with Type I. But each pair of symmetric views can share the same set of training photos for training, with the appropriate horizontal flips to align mirror-image views. As illustrated in Figure 4, denote A and B as symmetric views under a particular axis; the photos $\{I_A\}$ and $\{I_B\}$ are their respective training examples. Also denote $h(I)$ as a function to horizontally flip a photo I. Then, we can train a model of viewpoint A using not only $\{I_A\}$, but also $\{h(I_B)\}$. Same for B, which we will train using photos $\{h(I_A), I_B\}$. If the viewpoint happens to be on the symmetric axis, i.e. $A = B$, we can train the model using photos $\{I_A, h(I_A)\}$. But all this assumes that the symmetric axis is known. To discover the location of the symmetry axis, we use exhaustive approach to learn this one extra degree of freedom for the place category. In each iteration, for each of the $\frac{m}{2}$ possible symmetric axes (axes are assumed to align with one of the m viewpoints), we re-train the model based on the new training set and the symmetric axes. We use the classifier to predict on the same training set to obtain the confidence score, and choose the axis that produces the highest confidence score from the trained viewpoint classifier in each iteration.

Type III: This type of place symmetry indicates that each view matches the opposite view that is 180° away. Therefore, instead of training a m-way viewpoint classifier, we only need to train a $(\frac{m}{2})$-way viewpoint classifier. As illustrated in Figure 4, denote that views A and B are symmetric under one axis, A and C are symmetric under the other perpendicular axis, A and D are opposite views that are 180° away, and $\{I_A\}, \{I_B\}, \{I_C\}, \{I_D\}$ are their respective training examples. Then, we will train a viewpoint model for A using examples $\{I_A, h(I_B), h(I_C), I_D\}$. If the view is on one of the symmetric axes, i.e. $A = B$, we have $C = D$, and we will train the model for A using examples $\{I_A, h(I_A), h(I_D), I_D\}$. Because we know that the two axes of symmetry are perpendicular, there is actually only one degree of freedom to learn for the axes during training, which is the same as Type II and the same exhaustive approach for symmetric axis identification is applied. But instead of $\frac{m}{2}$ possible symmetric axes, there are only $\frac{m}{4}$ possibilities due to symmetry.

Type IV: The viewpoint cannot be identified under isotropic symmetry, so there is no need for alignment or a viewpoint classification model. The optimal prediction is just random guessing.

To automatically decide which symmetry model to select for each place category, i.e. discover the symmetry structure, we use cross validation on the training set to pick the best symmetry structure. If there is a tie, we choose the model with higher symmetry because it is simpler. We then train a model with that type of symmetry using all training data. Discovered symmetry axes for our place categories are shown in Figure 6.

6. Canonical view of scenes

It is well known that objects usually have a certain "canonical view" [12], and recent work [1] suggests that people show preferences for particular views of places which depend upon the shape of the space. Here we study the canonical views of different place categories. More specifically, we are interested in which viewpoint people choose when they want to take a photo of a particular type of place. To study the preferred viewpoints of scenes, we use the popular SUN dataset for scene classification from [17], manually selecting scene categories that corresponded to our panoramic place categories. We obtain the viewpoint information for each photo by running a view-matching task

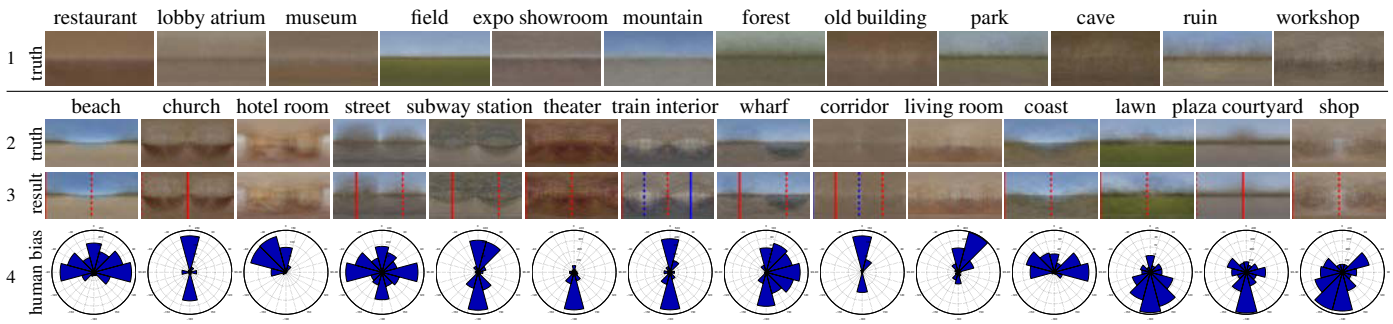


Figure 6. The first three rows show the average panoramas for the 26 place categories in our dataset. At the top are the Type IV categories (all views are similar, so no alignment is needed). Below are the 14 categories with Symmetry Types I, II and III, aligned using manual (the 2nd row) or automatic (the 3rd row) alignment. Discovered symmetry axes are drawn in red and blue lines, corresponding to the red and blue lines of symmetry shown in Figure 4. The 4th row shows the view distribution in the SUN dataset for each place category; the top direction of each polar histogram plot corresponds to the center of the averaged panorama.

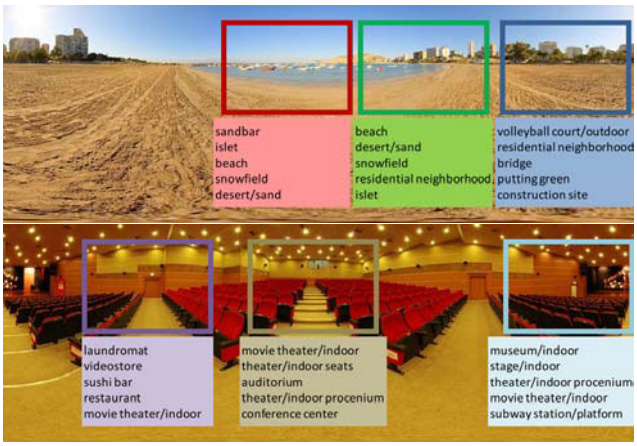


Figure 7. Prediction of SUN categories on different viewpoints. Each rectangle denotes a view in the panorama and the text below gives the top five predictions of the SUN category classifier, using [17] as training data. (The rectangles are only for illustration – proper image warping was used in the algorithm.)

on Amazon Mechanical Turk. Workers were shown a SUN photo and a representative panorama from the same general place category. The panorama was projected in an Adobe Flash viewer with 65.5° field of view (as it might appear through a standard camera), and workers were asked to rotate the view within this Flash viewer to match the view shown in the SUN photo. For each photo, we obtained votes from several different workers with quality control. Figure 5 shows examples of SUN database photos which workers mapped to each viewpoint of a corresponding panoramic place category. In each category, some views were more common in the SUN database than others, and we visualize this view bias in the last row of Figure 6. There are clearly biases specific to the place category, such as a preference for views which show the bed in the hotel room category.

To further illustrate how each view is correlated with specific scene categories, we train a scene classifier using the SUN database from [17], which covers 397 common scene categories. We use this 397-way classifier to predict on each viewpoint and get an average response from all photos generated from our panorama dataset. We show some examples in Figure 7. For instance, the beach views which show the sea are predicted to be beach-related categories (sandbar, islet, beach, etc.), while the opposite views are predicted to be non-beach categories (volleyball court and residential neighborhood).

7. Implementation details

Dataset construction We downloaded panoramas from the Internet [16], and manually labeled the place categories for these scenes. We first manually separated all panoramas into indoor verse outdoor, and defined a list of popular categories among the panoramas. We asked Amazon Mechanical Turk workers to pick panoramas for each category, and manually cleaned up the results by ourselves. The final dataset contains 80 categories and 67,583 panoramas, all of which have a resolution of 9104 $\times$ 4552 pixels and cover a full 360° $\times$ 180° visual angle using equirectangular projection. We refer to this dataset as "SUN360" (Scene UNderstanding 360° panorama) database[2].

In the following experiments, we selected 26 place categories for which there were many different exemplars available, for a total of 6161 panoramas. To obtain viewpoint ground truth, panoramas from the 14 place category of symmetry types I, II, and III were manually aligned by the authors, by selecting a consistent key object or region for all the panoramas of the category. The 2nd row of Figure 6 shows the resulting average panorama for each place category after manual alignment. No alignment was needed for places with type IV symmetry (the 1st row of Figure 6). The averaged panoramas reveal some common key structures for each category, such as the aligned ocean in the beach panoramas and the corridor structure in the subway station.

We generated m=12 photos from each panorama, for a total of 73,932 photos. Each photo covers a horizontal an-

[2]All the images are available at http://sun360.mit.edu.

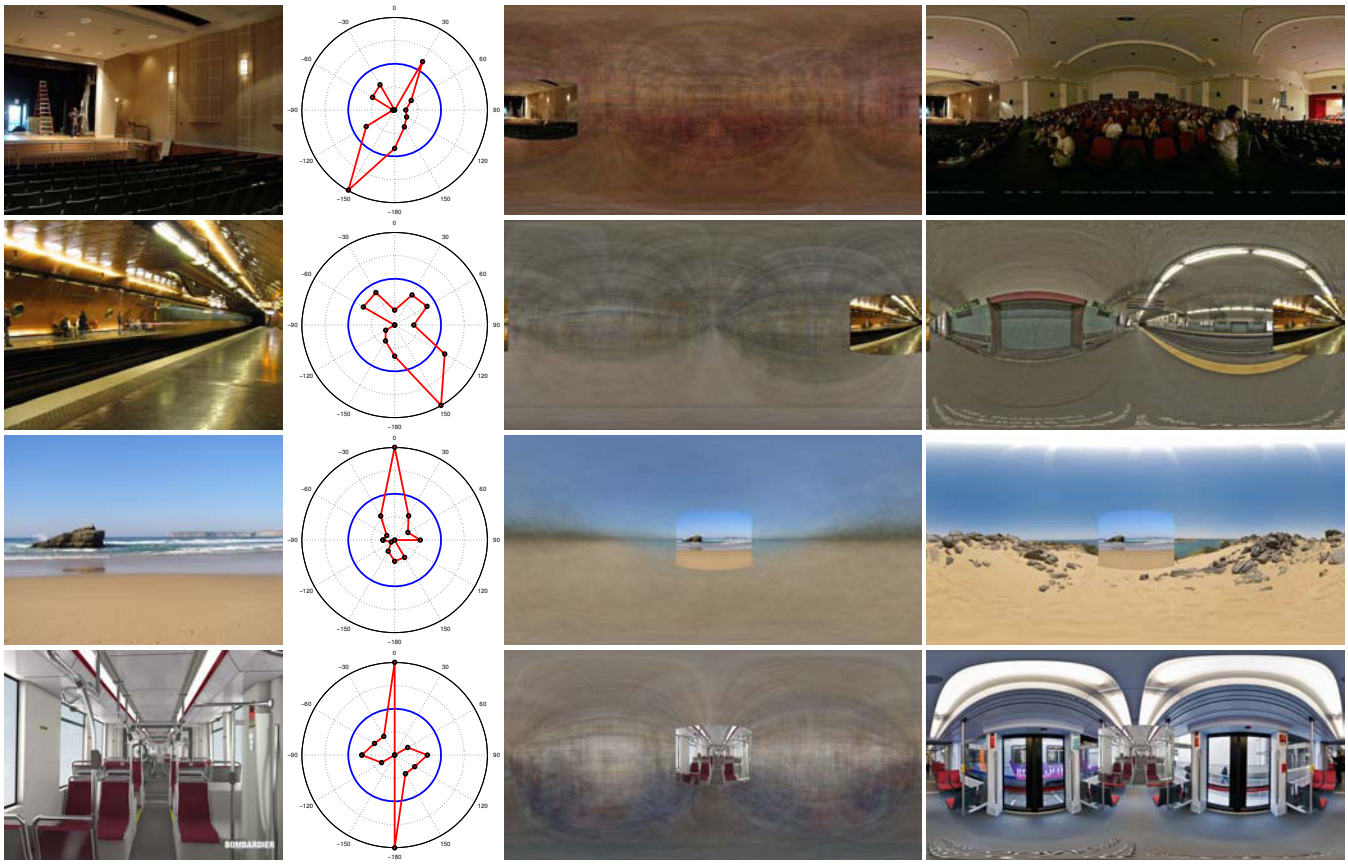
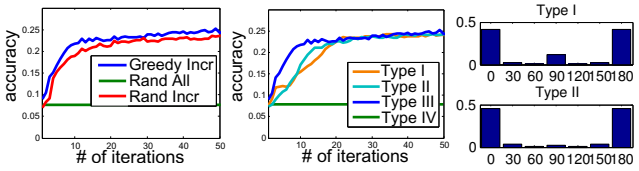


Figure 8. Visualization of testing results. In each row, the first column is the input photo used to test the model. The second figure visualizes of the scores from SVM indicating the most likely orientation of the photo. The blue circle is the zero crossing point, and the red polyline is the predicted score for each orientation. The last two columns illustrate the predicted camera orientation and the extrapolated scene layout beyond the input view (the left image extrapolates the layout using the average panorama from the place category; the right image uses the nearest neighbor from the training set.) Please refer to http://sun360.mit.edu for more results.

gle of 65.5°, which corresponds to 28mm focal length for a full-frame Single-Lens Reflex Camera (typical parameters for digital compact cameras). We tested the algorithm on two sets of limited-field-of-view photos. For the first set, we randomly split the panoramas to use one half for training and the other half for testing. Using the same training dataset, we also constructed a second testing set using the photos from the SUN dataset (Section 6).

Features and classifiers We select several popular state-of-art features used in scene classification tasks, including GIST [11], SIFT [7], HOG2×2 [17], texton, geometric map (GeoMap) and texton statistics (GeoTexton) [17], self-similarity (SSIM) [13], local binary pattern (LBP) [9], and tiny image [14] as baseline. For SIFT, HOG2×2, texton and SSIM, we construct a visual words vocabulary of 300 centroids using k-means to have a spatial pyramid histogram [7] as the descriptor of a photo. We use a linear weighted sum of the kernel matrices as the final combined kernel (COM), using the weights from [17]. The same features and kernels are also used by the viewpoint prediction classifier. We use One-versus-Rest SVM as our multi-class classifier for both place category classification and viewpoint classification, because it outperforms all other popular classifiers and regressors in our experiments.

Algorithm behavior analysis Figure 9 shows an example run on the street place category which characterizes the behavior of our algorithm. Figure 9(a) shows how different strategies of iterative training affect the convergence results: incremental addition of training examples gives much better results than starting with all examples (Rand All), and greedily adding the most confident examples (Greedy Incr) gives better performance than adding examples in a random order (Rand Incr). Figure 9(b) compares the four types of symmetry structure with respect to the testing accuracy. We can see that view sharing from symmetry structure is most beneficial when the number of training examples is small (in early iterations), but as more training examples are introduced in later iterations, all models converge to similar accuracy. Figure 9(c) shows the histogram of predicted angle deviation from the truth. We can clearly see the ambiguity due to the symmetry of the street place category. A model assuming Type I symmetry (top figure) performs slightly



(a) Training strategies. (b) Different symmetries. (c) Angle deviation.

Figure 9. Illustration for the algorithm behavior for the place category *street*. (a) compares different training strategies: our proposed greedy incremental algorithm, a random incremental algorithm that adds a random panorama, and a totally random algorithm that adds everything at once. (b) illustrates the performance of algorithms with different symmetry assumptions. (c) shows that although incorporating symmetry does not resolve ambiguity (0° and 180°), it does reduce random mistakes (e.g., 90°).

worse than a model assuming Type II (bottom figure), by making more errors on angles between 0° and 180°. Our algorithm is related to curriculum learning, k-means and Expectation-Maximization (EM), and can be interpreted as Latent Structural SVM with Concave-Convex Procedure. Refer to the project website for further analysis.

8. Evaluation

8.1. Evaluation methods

Place categorization performance is evaluated by determining the accuracy per category. Viewpoint prediction can also be evaluated by classification accuracy if the panoramas are manually aligned. However, because we use unsupervised alignment of the panoramas to train the viewpoint predictor from the aligned result, we cannot know which viewpoints in the alignment result correspond to which viewpoints in the labeled truth for evaluation. Similar to Unsupervised Object Discovery [15], we use an *oracle* to assign each aligned viewpoint to one of the m view directions from the truth, and then evaluate the accuracy based on the resultant viewpoint-to-viewpoint mapping. Due to circular constraints, the total number of all possible solutions is only m, and we try all of them to search for the best viewpoint-to-viewpoint mapping for evaluation.

Besides viewpoint prediction, another natural evaluation metric is the average viewpoint angle deviation from the truth to the prediction. However, due to the symmetry structure of certain types of places, the viewpoint may be ambiguous, and the average angle deviation is not always a good metric. For example, in a street scene, there are usually two reasonable interpretations for a given view, 180° apart, so the viewpoint deviations cluster at 0° and 180°(as shown in Figure 9(c)). This means the average viewpoint deviation is about 90°, the expected value for chance performance.

8.2. Evaluation on training

We evaluate the automatic alignment results obtained during training and summarize the result in Table 1, using the accuracy with the oracle, and average angle deviation. For comparison purpose, we design a baseline algorithm of panorama alignment: For each pair of panoramas, we try all m alignment possibilities. For each possibility, we extract HOG features [2], and use the sum of squared differences of the two feature maps to look for the best alignment between these two panoramas. Then we use the panorama with the minimum difference from all other panoramas as the centroid, and align the other panoramas to this common panorama. We tried two different cell sizes for HOG (8 pixels and 40 pixels, labeled HOG-S and HOG-L in the table).

Evaluation	HOG-S	HOG-L	Tiny	HOG2×2	COM	**Final**	Chance
Accuracy	41.8	45.0	25.9	56.4	62.2	**69.3**	8.3
Deviation	65.6°	62.5°	73.4°	48.0°	41.4°	**34.9°**	90.0°

Table 1. Performance of automatic panorama alignment. "HOG-S" and "HOG-L" are two baseline algorithms designed for comparison purpose only (Section 8.2). "Tiny", "HOG2×2" and "COM" are the results that do NOT make use of symmetry and view sharing, using various features presented in Section 7. "Final" is our complete model with symmetry and view sharing.

We also study the performance of different image features using our algorithm, including TinyImage and HOG2×2, and COM (the combined kernel from all features). Furthermore, we compare the performance between no sharing of training views (i.e., always assuming Type I symmetry), and sharing using the symmetry type discovered by the algorithm. Our complete model "Final" using all features and automatic discovered symmetry performs the best. Visually, the average panoramas from the automatic alignment algorithm, shown in the third row ("result") of Figure 6, look very similar to the averages produced by manual alignment in the second row ("truth").

8.3. Evaluation on testing

As mentioned in Section 7, we evaluate our testing result on two test sets: limited-field-of-view photos generated from our SUN360 panorama dataset, and real-world, camera-view photos from SUN dataset. The performance is reported in Table 2, which shows accuracy and average viewpoint deviation. In the table, we report the place categorization accuracy without taking viewpoint into account as "Place". "Both" is the accuracy requiring both correct place category prediction and correct viewpoint prediction, and it is the final result. Note that we expect lower performance when testing on the SUN photos [17], since the dataset statistics of the SUN database may differ from the statistics of our panoramic dataset, and there is no exact correspondence between the SUN database categories and our panoramic place categories.

To see how unsupervised automatic alignment affects the prediction result, we compare its performance to the models trained using manual alignment. We designed two new algorithms for comparison purposes only. The first is a one-stage algorithm (denoted as "Accuracy 1" in the table), in

Test Set	Manual Alignment				Automatic Alignment						
	Accuracy 1		Accuracy 2		Accuracy			Angle Deviation			
	Place	Both	Place	Both	Place	View	Both	I	II	III	IV
SUN360	48.4	27.3	51.9	23.8	51.9	50.2	24.2	55°	51°	86°	90°
SUN	22.2	14.5	24.1	13.0	24.1	55.7	13.9	29°	30°	38°	90°
Chance	3.8	0.3	3.8	0.3	3.8	8.3	0.3	90°	90°	90°	90°

Table 2. Testing accuracy and average angle deviation. We compare the performance of our automatic alignment algorithm with manual alignments using two algorithms (Section 8.3): a 1-step algorithm (Accuracy 1) and a 2-step algorithm (Accuracy 2). For each algorithm, "Place" is the accuracy of place classification, "View" is the accuracy of viewpoint prediction, and "Both" is the accuracy requiring both correct place category prediction and correct viewpoint prediction, and is the final result.

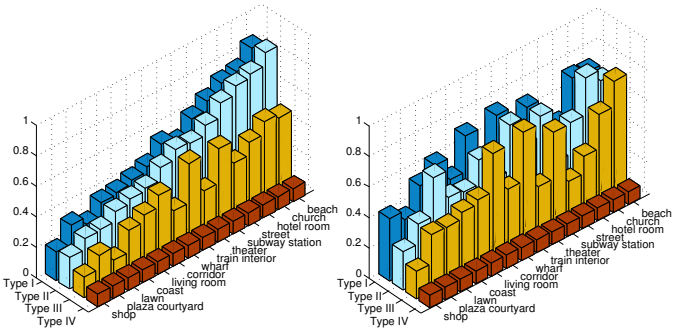
which we train a (26×12)-way SVM to predict the place category and viewpoint at the same time. The second is a two-stage algorithm "Accuracy 2", which first trains a 26-way SVM to predict the place category, and then, for each category, trains a 12-way SVM to predict the viewpoint. Compared to the models trained from manual alignment, our automatic alignment performs very well, producing results comparable to manual alignment (see Table 2). Furthermore, we can evaluate the view prediction accuracy assuming that the place categorization is correct. We compare the four types of symmetry structure for each place category in Figure 10. We can see that imposing correct symmetry structure is usually helpful, even when the imposed symmetry is incomplete, e.g., using Type II instead of Type III. Imposing incorrect types of symmetry always hurts performance.

9. Conclusion and future work

We study a new problem of recognizing place category and viewpoint in scenes. We construct a new panorama dataset, and design a new algorithm for automatic panorama alignment. We introduce the concept of scene symmetry and also study the canonical view biases exhibited by people taking photos of places. Since this is a first attempt at a new problem, we are simplifying the question by considering only viewpoint and ignoring the observer's position within a place. However, the principle ideas of this algorithm can be extended to address observer location and intra-category variations for alignment, by modifying the algorithm from one aligned model per category to a mixture of several models. All data and source code are publicly available at http://sun360.mit.edu.

Acknowledgements

We thank Tomasz Malisiewicz, Andrew Owens, Aditya Khosla, Dahua Lin and reviewers for helpful discussions. This work is funded by NSF grant (1016862) to A.O, Google research awards to A.O and A.T., ONR MURI N000141010933 and NSF Career Award No. 0747120 to A.T., and a NSF Graduate Research fellowship to K.A.E.



(a) Test on SUN360 Panorama Dataset. (b) Test on SUN Dataset.

Figure 10. Viewpoint prediction accuracy.

References

[1] K. A. Ehinger and A. Oliva. Canonical views of scenes depend on the shape of the space. In *CogSci*, 2011. 4

[2] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010. 7

[3] A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010. 1

[4] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. 1

[5] D. Hoiem. *Seeing the World Behind the Image: Spatial Layout for 3D Scene Understanding*. PhD thesis, 2007. 1

[6] H. Intraub. *Rethinking Scene Perception: A Multisource Model*. 2010. 1

[7] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1, 6

[8] K. Ni, A. Kannan, A. Criminisi, and J. Winn. Epitomic location recognition. *PAMI*, 2009. 1

[9] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 2002. 6

[10] A. Oliva, S. Park, and T. Konkle. *Representing, perceiving, and remembering the shape of visual space*. 2011. 1

[11] A. Oliva and A. Torralba. Modeling the shape of the scene:a holistic representation of the spatial envelope. *IJCV*, 2001. 6

[12] S. Palmer, E. Rosch, and P. Chase. Canonical perspective and the perception of objects. *Attention and Performance IX*, 1981. 1, 4

[13] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007. 6

[14] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large database for non-parametric object and scene recognition. *PAMI*, 2008. 6

[15] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *IJCV*, 2010. 7

[16] Website. www.360cities.net. 5

[17] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 1, 2, 4, 5, 6, 7

[18] J. Xiao and L. Quan. Multiple view semantic segmentation for street view images. In *ICCV*, 2009. 1