PLISS: Detecting and Labeling Places Using Online Change-Point Detection

Ananth Ranganathan
Honda Research Institute, USA
aranganathan@honda-ri.com

***Abstract*— We present PLISS (Place Labeling through Image Sequence Segmentation), a novel technique for place recognition and categorization from visual cues. PLISS operates on video or image streams and works by segmenting it into pieces corresponding to distinct places in the environment. An online Bayesian change-point detection framework that detects changes to model parameters is used to segment the image stream. Unlike current place recognition methods, in addition to using previously learned place models for labeling, PLISS can also detect and learn a previously unknown place or place category in an online manner. Moreover, since both the inferred boundaries of places (change-points) and the place labels are fully probabilistic, they can indicate when the inference is uncertain. New places and categories are detected using a systematic statistical hypothesis testing framework. We present extensive experiments on a large and difficult image dataset. We validate our claims by comparing results obtained using different types of features and by comparing results from PLISS against the state of the art.**

I. Introduction

Place recognition is the task of consistently labeling a particular place every time it is visited, while place categorization is the corresponding problem for a category of places. Such labels may range from "Place No. 1" and "Kitchen on 2nd floor with microwave and coffee machine", which are labels for particular places, to "Kitchen" and "corridor", which are category labels. Place recognition and categorization are essential in order for a robot or an intelligent agent to recognize places in a manner similar to that done by people. Place recognition facilitates human-robot communication [28] and is an integral part of semantic mapping procedures [35]. For instance, the label of a place strongly affects, among other things, the types of objects found there[29].

Most existing place recognition systems assume a finite set of place labels, which are learned offline from supervised training data. This is done using classifiers, which then categorize places based on input measurements during runtime. Classifier-based systems have the advantage of simplicity but also have many corresponding disadvantages

1) If a label category has large variation in measurements (for e.g., offices are vastly dissimilar in all aspects), a huge training set is necessary for adequate performance
2) The system is constrained to classify the input into the specified category set and cannot recognize the existence of a new label
3) Temporal consistency in the output labeling has to be externally enforced since the classifier simply labels each measurement individually

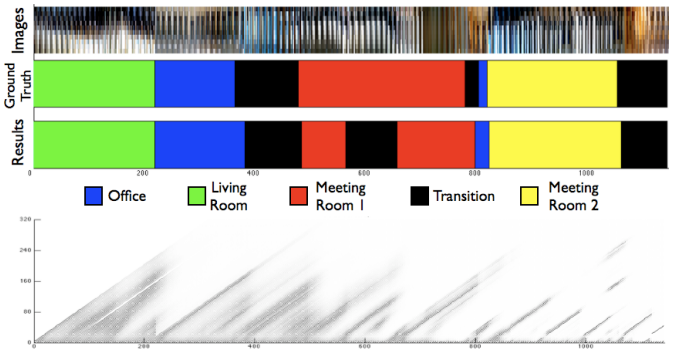

Fig. 1. Place labeling using PLISS: Output is for an image sequence of 1043 frames containing 5 labels. Thumbnails of the images are shown on top, followed by groundtruth and maximum-likelihood output labeling. Many, but not all, changes in labels correspond to visible changes in appearance of the image thumbnails. The change-point posterior on segment lengths is shown at the bottom with darker regions denoting higher probability.

In this paper, we present a new place recognition method called PLISS, for Place Labeling through Image Sequence Segmentation, which tackles these problems. As its name suggests, PLISS works with video or image streams, thus intrinsically considering the temporal component of the problem. We take the novel approach of using change-point detection to segment the image streams into portions corresponding to places. Change-point detection is the problem of detecting abrupt changes to the parameters of a statistical model. The locations of these abrupt changes within the image stream are also the place boundaries, where a place is entered or exited.

We use an online Bayesian change-point detection algorithm that computes the probability of a change-point occurring at each timestep. The algorithm keeps track of all possibilities and does not make an irrevocable decision at any step. The probability of a change-point at any given timestep is obtained by combining a prior on the occurrence of change-points with the likelihood of the current measurement given all the possible scenarios in which change-points could have occurred in the past. We demonstrate that this computation can be done exactly for certain classes of prior and likelihood functions. Since the memory requirement for the exact algorithm increases linearly with the number of measurements processed thus far, we also provide an approximation using Rao-Blackwellized particle filters that is fast and requires only constant memory.

While change-point detection provides boundaries, the place

label is assigned probabilistically based on the measurement, the most recent label, and the change-point distribution. Statistical hypothesis testing is used to determine if the current measurement could have been generated by any of the pre-learned place models. If this is not the case, the measurement is declared to have come from a previously unknown place. In this manner, PLISS can systematically recognize a previously unknown place type and assign it a new label if required. Further, PLISS can learn and update place models online, which is helpful in cases where measurements from the same category show variation (albeit only if the variation is gradual). Hence, PLISS can operate with labeled data, which is used to learn place models at training time, or without it, when it recognizes new place categories as they arise and learns place models for them online.

We model places using the Multivariate Polya distribution, also known as the Dirichlet Compound Multinomial (DCM) model [19]. The DCM is a bag-of-words model and captures burstiness of the data, i.e. it models the observation that if a word occurs once in a document, it usually occurs repeatedly. We use the DCM model to model histogram measurements obtained by quantizing dense features computed on the images.

A sample result from PLISS on a sequence with 5 labels is shown in Figure 1. To test our claims, we present experiments on the large and difficult Visual Place Categorization (VPC) dataset [32]. We also provide comparisons using different types of features, as well as comparisons of the performance of PLISS against the VPC system [33].

II. Related Work

Work on place recognition in robotics has ranged over matching SIFT features across images [37] or other derived measures of distinctiveness for places such as Fourier signatures [18], subspace representations [12], and color histograms [31]. These methods have the disadvantage of not being able to generalize and also are invariant to perspective mainly through the use of omnidirectional images. Many approaches that generalize well, learn classifiers, such as SVMs, for each place from labeled data [25], [27]. A very recent system by Pronobis et al [24] focusses on merging cues from different sensing modalities, the output for each of which is obtained from individually trained SVMs. These, and similar classifier-based approaches cannot detect or learn previously unknown places and place categories, in contrast to PLISS. Place classification methods based on laser and sonar range scans also exist [14], [20], though we do not review them here.

In computer vision, place classification is also known as scene classification and image categorization. Two broad methods can be discerned - those that model local features [3] and distinctive parts of images[26], and those that extract global representations of images and learn from them[22][11][4]. The latter is closer to this work and includes the CENTRIST-based VPC system [33], and Spatial Pyramid Matching [16]. These methods also use classifiers to learn place categories and cannot detect new categories online. Further, they do not deal with image streams and videos except when filtering procedures are added on top to get temporal consistency [33]. Other approaches use keypoint matching for location recognition based on image retrieval [21][7] but cannot generalize.

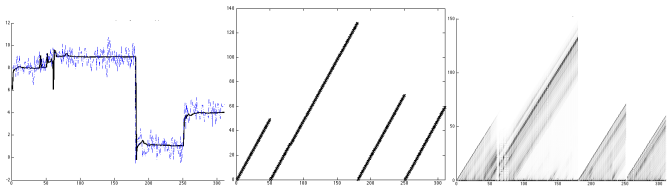

Fig. 2. Change-point detection: (a) Univariate Gaussian input data (blue/dashed) with 4 segments (b) Groundtruth for change-points shown as length of segments, which is the variable c_t used in our inference. (c) Output of change-point detection which is probability distribution on c_t. The means of the segments can also be inferred and is shown in (a) (black)

Change-point detection has a long history in statistics and the best-known technique is the CUSUM detector [23] which involves piece-wise segments of Gaussian mean with noise. A more realistic method is segmented regression [8]. In our exposition, we closely follow the more general methods of [2], [10] which is applicable to conjugate-exponential models. Many applications in computer vision have used change-point detection previously[36][30][6]. However, to our knowledge, this is the first application of change-point detection to place recognition.

III. Place Recognition using Change-Points

We formulate the place recognition problem as follows. We are given a measurement stream that produces measurements at (possibly changing) intervals, which we will also call as timesteps. For each measurement, we are required to come up with a label corresponding to the type of the place. The place types are given in the form of L place models $M_1, M_2, \ldots, M_L$. We are also required to say if the measurement does not correspond to any of these labels.

We approach the problem by noting that the place label remains the same for the period of time when a robot is moving inside the particular type of place. It only changes sporadically when the robot travels into the next place. Thus, the measurement stream can be segmented into pieces corresponding to places, i.e. measurements in each segment are assumed to have been generated by the corresponding place model. The start and end of segments where the generating model changes, hence called change-points, provide a very strong indication regarding the place label.

We assume that a sequence of data $y_1, y_2, \ldots, y_t$ can be segmented into non-overlapping but adjacent segments. The boundaries between these segments are the change-points. We deal with the model-based change-point scenario here, wherein the form of the probability distribution in each segment remains the same and only the parameter value changes. We assume that the data are i.i.d within each segment and denote by c_t the length of the segment at time t. Note that c_t is also the time since the last change-point. If the current timestep is a change-point, then $c_t = 0$, and if no change-points have occurred yet, then $c_t = t$. A sample problem setup and output

for change-point detection in the univariate Gaussian case in shown in Figure 2.

We denote the place label at time t as x_t^c. The place label is indexed by the current segment since the whole segment has a single label. However, this label is updated with each measurement, and hence the time index t is also used. The probability distribution over x_t^c is taken to be a discrete distribution of size L, one for each of the place models. The case where the place label takes none of these values is detected using statistical hypothesis testing.

We need to compute the joint posterior on c_t and x_t^c given the data, $p(c_t, x_t^c|y_{1:t})$, where $y_{1:t}$ denotes all the data from time 1 to time t. The posterior can be factored as

$$p(c_t, x_t^c|y_{1:t}) \quad = \quad p(c_t|y_{1:t})p(x_t^c|c_t, y_{1:t}) \tag{1}$$

The first term is the posterior over the segment length while the second term is the conditional posterior on the place label given the segment length. Note that the posterior over segment length is equivalent to inferring the change-points since $c_t = 0$ implies a change-point. Further, though the place label posterior is conditioned on the segment length, this does not imply that the place label can only change at a change-point. Since the place label is updated at every timestep, the algorithm can "change it's mind" regarding the place label at any point in time, given enough evidence.

We address change-point detection, i.e. computing $p(c_t|y_{1:t})$ in the next section. The place label posterior $p(x_t^c|c_t, y_{1:t})$ is addressed subsequently in Section V.

IV. Model-based Change-Point Detection

In the following exposition, we closely follow [2] and [10], both of which state essentially similar algorithms but with different state representations. We represent the likelihood of the data in segment c_t as $p(y_t|\xi_t^c)$ where ξ_t^c is a parameter set. The data inside each segment are assumed to be i.i.d and the parameters are assumed i.i.d according to a prior parameter distribution.

The change-point posterior from (1) can be expanded using Bayes law as

$$p(c_t|y_{1:t}) \quad \propto \quad p(y_t|c_t, y_{1:t-1})p(c_t|y_{1:t-1}) \tag{2}$$

The first term is the likelihood while the second term can be further expanded by marginalizing over the segment length at the previous timestep to yield a recursive formulation for c_t

$$p(c_t|y_{1:t-1}) \quad = \quad \sum_{c_{t-1}} p(c_t|c_{t-1})p(c_{t-1}|y_{1:t-1}) \tag{3}$$

where $p(c_t|c_{t-1})$ is the transition probability, $p(c_{t-1}|y_{1:t-1})$ is the posterior from the previous step, and we have made use of the fact that $c_1, c_2, \ldots, c_t$ form a Markov chain.

A. The Transition Probability

For characterizing the transition probability $p(c_t|c_{t-1})$ in (3), we note that the only two possible outcomes are $c_t = c_{t-1} + 1$ when there is no change-point at time t, and $c_t = 0$ otherwise. Hence, this is a prior probability on the "lifetime" of this particular segment where the segment ends if a change-point occurs. In survival analysis, such situations are routinely modeled using a hazard function, which represents the probability of failure in a unit time interval conditional on the fact that failure has not already occurred. If $H(.)$ is a hazard function, the transition probability can be modeled as

$$p(c_t|c_{t-1}) \quad = \quad \begin{cases} H(c_{t-1}+1) & \text{if } c_t = 0 \\ 1 - H(c_{t-1}+1) & \text{if } c_t = c_{t-1}+1 \end{cases} \tag{4}$$

In the simplest case where the probability of a change-point at every step is assumed constant, the length of a segment has to be modeled using an exponential distribution with time scale λ [10], so that $H(t) = 1/\lambda$.

B. The Data Likelihood

The data likelihood from (2) can be calculated only if we know the distribution parameter to use. Hence, we integrate over the parameter value using the parameter prior

$$p(y_t|c_t, y_{1:t-1}) \quad = \quad \int_{\xi^c} p(y_t|\xi^c)p(\xi^c|c_t, y_{t-1}^c) \tag{5}$$

where ξ^c is the model parameter for segment c_t, and y_{t-1}^c is the data from the current segment. The above integral can be computed in closed form if the two distributions inside the integral are in the conjugate-exponential family of distributions. Expensive numerical integrations have to be employed otherwise. In the following exposition, we assume the conjugate case.

Further, let the integrated function be denoted as $p(y_t|c_t, \eta_t^c)$ where η_t^c parametrizes the integrated data likelihood. Even though this function is usually not in the exponential family, it may be possible to update it directly using the sufficient statistics of the data corresponding to the current segment $\{y_{t-1}^c, y_t\}$, i.e. the integration need not be performed at every step. In the case where t is a change-point, the function is computed with prior values for $\eta_t^{(0)}$ (since $c_t = 0$) instead of any sufficient statistics.

C. Computational Cost

The algorithm described so far is exact. After n measurements, the possible segment lengths range from 0 to n, and the posterior contains the probability of all these cases. Further, if the optimal change-point locations are also needed, the posteriors from all timesteps have to be kept. Hence, the runtime and memory costs per timestep are $O(n)$ while the total memory cost is $O(n^2)$. These requirements are incompatible with long term online operation. Hence, we next provide a simple and intuitive particle filtering approximation that exhibits constant runtime and $O(n)$ total memory cost.

D. Online operation using particle filtering

We need the locations of change-points and the exact algorithm above accomplishes this by maintaining the posterior over segment lengths c_t for all t. We now approximate the posterior using N weighted particles, thereby obtaining a constant runtime algorithm.

Combining (2), (3), and (4) we get the segment length posterior as

$$p(c_t|y_{1:t}) \propto \begin{cases} w_t^{(0)} \sum_{c_{t-1}} H(c_{t-1}+1)\rho_{t-1} & \text{if } c_t = 0 \\ w_t^{(c)} \sum_{c_{t-1}} \{1 - H(c_{t-1}+1)\}\, \rho_{t-1} & \text{if } c_t = c_{t-1}+1 \end{cases} \tag{6}$$

where $w_t^{(c)} = p(y_t|c_t, y_{t-1}^c)$ and, for the case where t is a change-point and y_{t-1}^c is the empty set, $w_t^{(0)} = p(y_t|c_t)$. $\rho_{t-1} = p(c_{t-1}|y_{1:t-1})$ is the posterior from the previous timestep.

Clearly, the posterior (6) is amenable to straight-forward use in a particle filter with w_t as the particle weights. We use the optimal stratified resampling method of [9] that ensures runtime and memory usage proportional to the number of particles, i.e. $O(1)$ with regard to number of measurements seen so far. The particle weights are given by (5).

Note that since the likelihood parameters ξ^c in (5) are integrated out, this particle filter is Rao-Blackwellized [5] and has lower variance than a standard particle filter. This also makes the convergence of the algorithm more efficient.

V. Inferring the Place Label

The conditional posterior on the place label from (1) can be expanded using Bayes law as

$$p(x_t^c|c_t, y_{1:t}) \quad \propto \quad p(y_t^c|x_t^c, c_t)p(x_t^c|c_t) \tag{7}$$

where y_t^c is the data in the current segment, i.e. $y_t^c = \{y_{t-c_t}, \ldots, y_t\}$, since the place label only depends on these measurements. If we have L place models $M_1, M_2, \ldots, M_L$, the probabilities of each of these cases can be updated using (7). We simply use the label probability for the segment computed in the previous timestep as the prior, i.e. $p(x_t^c|c_t) = x_{t-1}^c$. For a new segment with $c_t = 0$, we set the prior to be a uniform distribution over the known labels.

Detection of an unknown place requires more involved calculation. In this case, we are required to show that the data does not arise from any of the models corresponding to known places. A systematic way of arriving at this conclusion involves statistical hypothesis testing. Hence, at each timestep, we perform L hypothesis tests to determine if the data arises from a known place. If these tests are expensive, they may also be performed once every T timesteps.

We now consider hypothesis testing for model M_i with parameter vector η so that the exact probability under this model is $p_0 = p(y_t^c|\eta)$. The significance of the observed data is the probability of all the data that is equally or less probable (more extreme) under η, $p_\sigma = \sum_{p(y|\eta)<p_0} p(y|\eta)$. This expression is exact but intractable for almost all but the most trivial models. We approximate using the likelihood ratio where the ratio of the model under the maximum likelihood parameter value and the true parameter value is computed

$$R \quad = \quad \frac{p_0}{p(y_t^c|\eta_{ml})} \tag{8}$$

where $\eta_{ml} = \text{argmax}_{\ \eta}\, p(y_t^c|\eta)$.

The statistic used in the hypothesis test is $-2\ln R$, where R is the likelihood ratio (8). This statistic can be shown to converge to the Chi-squared distribution with $k-1$ degrees of freedom [15] where k is the dimensionality of the parameter vector θ. The model M_i can be rejected if the Chi-squared probability is less than some threshold, usually set at 5% (0.05) or 1% (0.01). The test statistic converges to the Chi-squared distribution at the rate of $O(N^{-1})$, where N is the number of measurements used to compute the maximum likelihood parameter value η_{ml} [13]. In our case, each image feature is a measurement, and since the number of features per image is in the hundreds, convergence is not an issue.

We carry out the above test for each known place model and declare the place to be previously unknown if the tests reject all of them. Since the Chi-squared probabilities from the test do not say anything about the probability of the new label, the place distribution is set to a prior value for unknown places $p(x|\text{new label})$, which in our case is set so that the new label has probability 0.5 and all the other labels are equally probable. The new place label can be either stored for future reference along with the maximum likelihood model parameters η_{ml}, or be discarded if new places are of no interest. Thus, PLISS can detect new places and learn models for them online in contrast to most existing place categorization methods.

In terms of implementation, we augment the change-point algorithm of Section IV so that the discrete distribution on places is stored with each segment c_t. Similarly, in the particle filter, each particle maintains a place distribution. The place distribution becomes increasingly confident as the length of the segment c_t increases and is robust to noisy measurements and outliers. However, since it is also recomputed with each measurement, the algorithm does not make any irrevocable decision with regard to the place label and can "change its mind" given enough evidence. The cost of updating the place distribution is linear in the number of labels and hence, does not affect the runtime of the change-point algorithm.

VI. Measurements and Place Models

We model images using a "bag of words" model wherein a histogram is used to represent the image. These histograms are also used as input measurements to the PLISS algorithm. First, in an offline phase, SIFT features are computed on a dense grid on each of a set of images. The features are vector quantized using K-means to create a codebook of pre-specified size. Note that it is not necessary in this step for the image set used to be similar to the test images, though better results are obtained if this is true.

We compute a spatial pyramid [16] from the quantized SIFT features which is used as input to the PLISS system. The spatial pyramid is obtained by computing histograms, at various spatial resolutions, of feature frequencies in each of the codebook clusters. Following [16], we obtain successive spatial resolutions by dividing the image into a grid as shown in Figure 3. Note that only the histograms at the finest resolution need to be computed since the coarser resolution histograms can be obtained by simply adding the appropriate histograms at an immediately finer level. All the histograms

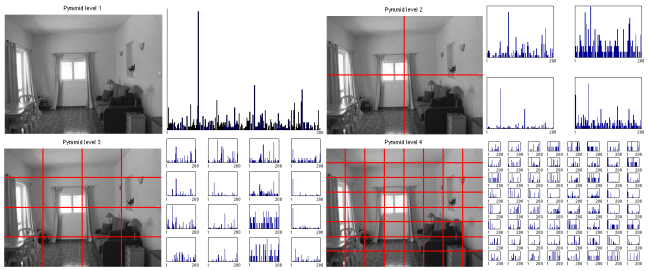

Fig. 3. The Spatial Pyramid histogram: Histograms of clustered SIFT features, computed on image regions at different spatial resolutions, are concatenated to yield the representation of the image. The image regions are obtained by dividing it into successively finer grids.

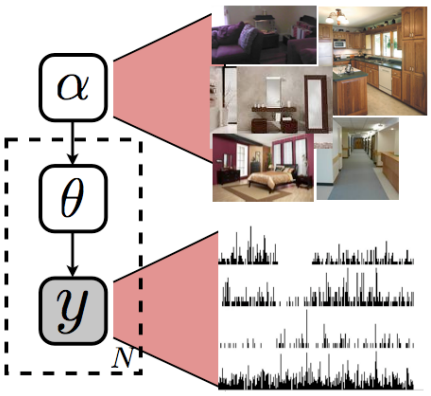

Fig. 4. Graphical model illustrating the Multivariate Polya distribution. To obtain a measurement z, which is a quantized feature histogram, we first sample from a Dirichlet distribution with parameter α to obtain a Multinomial vector θ . This Multinomial distribution is, in turn, sampled to obtain the measurement histogram y. Note that a different θ has to be sampled for each y. Each α corresponds to a place or place category.

from the different grids are then concatenated to yield the spatial pyramid representation. The two parameters for computing the spatial pyramid are thus, the number of levels in the pyramid V and the number of clusters computed in SIFT space (size of the dictionary) K. SIFT features only have local information about an image patch while an image histogram only has global information. By combining both of these at different scales, the spatial pyramid obtains more fine-grained discriminative power.

In addition to SIFT features, we also compute spatial pyramids using two other features, CENTRIST [32] and texture, which we now describe. CENTRIST is based on the census transform [34], which is a local feature computed densely for every pixel of an image, and encodes the value of a pixel's intensity relative to that of its neighbors. It was originally introduced for identifying correspondence between local patches. The census transform is computed by considering a patch centered at every pixel. The transform value is a positive integer that takes a range of values depending on the size of the patch. For instance, a patch size of 3, where there are 8 pixels in the patch apart from the central pixel, yields transform values between 0 and 255 (2^8 values). This is equivalent to having a dictionary size of 256 without the need for the clustering step. Computing a histogram of these 256 values yields the CENTRIST (census transform histogram) descriptor. Additionally, the census transform itself is extremely efficient to compute. Hence, a spatial pyramid can be calculated much faster using CENTRIST than using SIFT features.

We also compute a spatial pyramid using texture. Texture is extracted using the 17-dimensional Leung-Malik filter bank [17]. Similar to the case of SIFT features, an image set is used on which texture is extracted and clustered using K-means to obtain a set of textons. Histograms containing the proportions of each of the textons are computed at different resolutions as before to obtain the texture-based spatial pyramid.

Place models using Spatial Pyramids

The place model is used to compute the measurement likelihood in (5). Since the measurements are histograms of word counts, we model them using a multinomial distribution having dimensions equal to the dictionary size. Further the prior over the multinomial parameter is the conjugate Dirichlet distribution to aid in ease of computation. Given a histogram measurement y, its likelihood according to (5) is

$$P(y|\alpha) = \int_{\theta} P(y|\theta)P(\theta|\alpha) \quad (9)$$

where $\theta = [\theta_1, \theta_2, \ldots, \theta_W]$ and $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_W]$ are the multinomial parameter and Dirichlet prior respectively, and W is the dictionary size. Assuming that the histogram y has bin counts given by $[n_1, n_2, \ldots, n_W]$, the distributions in the integrand above can be written as

$$p(y|\theta) = \frac{n!}{n_1!n_2!\ldots n_W!}\theta_1^{n_1}\theta_2^{n_2}\ldots\theta_W^{n_W} \quad (10)$$

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{w=1}^{W}\alpha_w)}{\sum_{w=1}^{W}\Gamma(\alpha_w)}\theta_1^{\alpha_1-1}\theta_2^{\alpha_2-1}\ldots\theta_W^{\alpha_W-1} \quad (11)$$

The likelihood model in (9), where $P(y|\theta)$ is a multinomial distribution (10) and $P(\theta|\alpha)$ is a Dirichlet distribution (11), is called the Multivariate Polya model [19], or equivalently in document modeling, the Dirichlet Compound Multinomial (DCM) model [1]. The DCM distribution models burstiness in the data, i.e. given that a quantized feature appears once in a document, it is much more likely for it to occur multiple times rather than just once or twice. This is an intuitive observation in all realistic image data, particularly since we are dealing with densely computed features.

Performing the integration in (9), we get the final form of the likelihood, which is also our place model, as

$$P(y|\alpha) \propto \frac{n!}{\prod_{w=1}^{W} n_w}\frac{\Gamma(|\alpha|)}{\Gamma(n+|\alpha|)}\prod_{w=1}^{W}\frac{\Gamma(n_w+\alpha_w)}{\Gamma(\alpha_w)} \quad (12)$$

where $n = \sum_w n_w$, $|\alpha| = \sum_w \alpha_w$, and $\Gamma(.)$ denotes the Gamma function. Graphical intuition for the DCM model is provided by Figure 4. If the likelihood of a set of measurements is to be computed, then n is taken to be the total counts across all measurements, while n_w is the total count for a particular word.

Given a set of D images with features detected on them, the maximum likelihood value for α can be found by optimizing using gradient descent. It can be shown that this leads to the following fixed point update for the α parameter[19]

$$\alpha_w^{new} = \alpha_w \frac{\sum_{d=1}^{D} \psi(n_{dw} + \alpha_w) - \psi(\alpha_w)}{\sum_{d=1}^{D} \psi(n_{dw} + \alpha) - \psi(\alpha)} \quad (13)$$

where $\alpha = \sum_w \alpha_w$ as before, and $\psi(.)$ is the Digamma function, the derivative of the Gamma function. Faster but more complicated updates using Gauss-Newton iterations also exist [19].

We use DCM distributions as place models in PLISS. The α parameter for each place is learned from labeled images in an offline training phase, if training data is provided. During runtime, the distribution is used to compute the likelihood (5), and the α parameter is also updated after each measurement using the iterative rule (13) or the slightly faster Gauss-Newton updates. This facilitates online learning but if online learning is not required, the updated parameter can be discarded at the end of the segment.

Note that, since we are using spatial pyramids as input, it is only required to model the histograms at the finest level using the Multivariate Polya model since the coarser level histograms are simply summations of these. Thus, for a pyramid with V levels and level $V = 0$ denoting the whole image, the dimensionality of α is $4^V W$. However, a value of $V > 3$ has not been required in our experience. The expression for the hypothesis testing statistic, which is $-2 \ln R$ where R is the likelihood ratio, can also be easily obtained by substituting the distribution expression (12) into the likelihood ratio (8).

A recap of the overall PLISS algorithm using the DCM model is given in Algorithm 1.

Algorithm 1 Particle filtering algorithm for PLISS with online learning using the Multivariate Polya model

1) **Initialize:** Set prior parameter α_0. For all particles, $c_0 = 0$ and $x_0^0 = unif$ (unif. dist. over known labels).
2) **Update particle set:**
   For every timestep t do -
   For every particle $\{w_{t-1}, c_{t-1}, x_{t-1}^c, \alpha_{t-1}^c\}$ do
   - Create two new particles (1) No change-point case $l_1 = \{w_{t-1}, c_{t-1} + 1, x_{t-1}^c, \alpha_{t-1}^c\}$ (2) change-point $l_2 = \{w_{t-1}, 0, unif, \alpha_{t-1}^c\}$
   - Compute the prior for the (4) and update weights of l_1 and l_2
   - Using y_t, learn new parameter α_t^c for l_1 using (13) and set parameter α_t^0 for l_2 to α_0
   - Compute incremental weights for l_1 and l_2 from likelihood function (12) and multiply with particle weights
3) **Resample** from weights to get new set of particles
4) **Update place distributions:**
   For every particle $\{w_t, c_t, x_{t-1}^c, \alpha_t^c\}$ do
   - Perform a statistical hypothesis test using the likelihood ratio test (8) for each known place model
   - If a test indicates y_t^c to be arising from an existing place model, update x_{t-1}^c using (7) to get x_t^c
   - If all existing place models are rejected, create new place label and set x_t^c to the prior distribution $p(x|\text{new label})$

	Correct	False Positive	False Negative	% Correct
SIFT	62	6	14	81.6
CENTRIST (3x3)	55	11	21	72.3
CENTRIST (5x5)	59	6	17	77.6
Texture	51	3	25	67.1

(a)

	Labeling Accuracy (%)
SIFT	61.2
CENTRIST (3x3)	53.9
CENTRIST (5x5)	58.3
Texture	52.0

(b)

Fig. 5. Results with no training data: (a)Change-point detection performance for PLISS using various feature types (b)Place labeling accuracy

VII. Experiments

We tested the PLISS system extensively on actual image data and describe the methodology and results here. The PLISS system was implemented in Matlab with no attempt made at any optimization. All parts of the algorithms, including feature detection, were performed in Matlab.

We use the Visual Place Categorization (VPC) dataset [32] for our experiments. The dataset consists of image sequences from six different homes, each containing multiple floors. The image set from each home consists of between 6000 and 10000 frames. In our experiments, we consider sequences from each floor to be a different image sequence. The dataset has been manually labeled into 5 categories to provide ground truth for the place categorization problem. In addition, a "transition" category is used to mark segments that do not correspond clearly to any class. The VPC dataset is significantly difficult since no effort has been made to keep all the images in the sequence informative. Thus, a number of images contain only a wall, which is something that could also be expected when a robot is moving around.

We computed SIFT features on a grid having width and height 8 pixels per cell. Features were computed on 16x16 image patches. The features were clustered to obtain a dictionary of size 256. For the CENTRIST features, we used patch sizes of 3x3 and 5x5. While the natural dictionary size of 256 was used for the 3x3 case, we clustered the census transform obtained from the 5x5 case to obtain 256 clusters. The number of textons generated for the texture features was also 256. We used a two-level pyramid to compute the spatial pyramid histogram in all cases. In experiments without training data, the dictionary was computed using K-means clustering of features extracted from a set of 500 images drawn randomly from the VPC data. Prior values for the place model parameter α to be used in the likelihood computation (5) at a change-point (when no data is available for the segment) were learned from this same 500 image set. The significance level for the hypothesis tests was set at 1%. The prior distribution for a new label $p(x|\text{new label})$ was set to 0.75 for the new label and equal probability for the others. All results were obtained using the particle filtering algorithm with 100 particles, as this gave similar results to the exact algorithm.

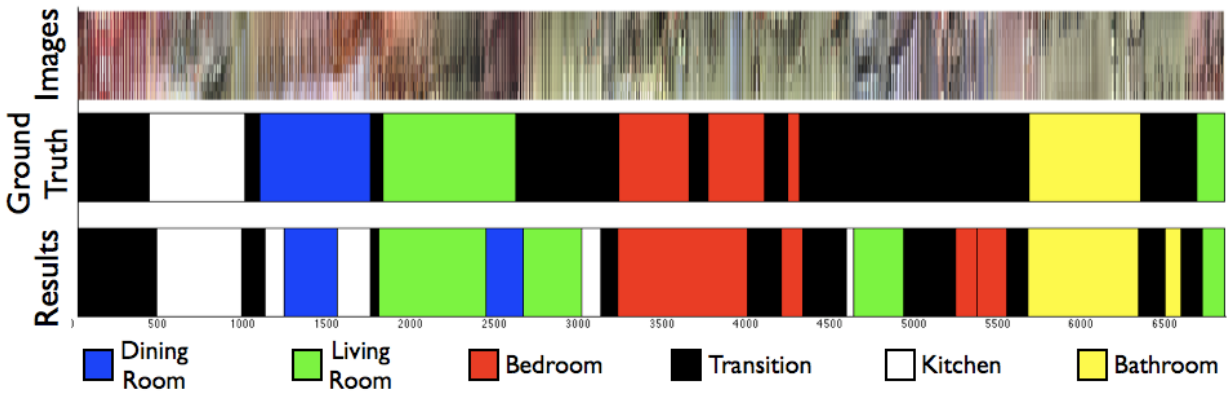

Fig. 6. Maximum likelihood output labeling for the 1st floor image sequence from Home 3 of the VPC dataset, which is one of the more difficult ones and contains 6839 images. Thumbnails of original images, groundtruth labeling and the PLISS result are shown. Note that it is possible to spot many of the change-points from just the high-level image characteristics visible here but not all of them.

A. *Experiments without offline training*

Almost all existing place recognition systems would be unusable if no training data were available. However, PLISS is still able to segment the data into pieces corresponding to different places and learn their models online. Hence, this experiment also tests the online learning capability of our system. Output was obtained for six image sequences from the VPC dataset containing a total of 14,346 frames. The total number of changepoints in these sequences, taken to be frames where the place label changes, was 76. The change-point detection performance using various features is shown in Figure 5(a). SIFT features perform best while texture features perform the worst. The system was declared to have found the change-point if it fired within 20 frames of the groundtruth. Note that many of the change-points, especially involving "transition" regions, are as such difficult even for a person to recognize.

Place labels for the sequences were also learnt online, and it was expected that the system would give the same place label when returning to a specific place but not when encountering another place with the same category label. The labeling was modified manually to reflect this for the six sequences and 39 distinct places were labeled. Results for place labeling are given in Figure 5(b). Considering that no training data has been used, the results are very promising when compared with the VPC system (Figure 8).

B. *Place Recognition and Categorization*

We next present experiments on specific place recognition which differs from the task of place categorization. The same six sequences were used, with 39 distinct labeled places, as in the previous experiment. We trained the system on every 3rd image and its corresponding label from these sequences to obtain the place models for each place. Note that k-means clustering and prior learning were done on the whole training set. Results obtained on the 9564 test image frames are given in Figure 7. The recognition rates are much higher since the training data and test data are similar, due to which all three features also yield similar results. While a more realistic test would also account for variation in lighting in the test data, the current results provide indication of the soundness of our approach even for a relatively large number of places.

We evaluated PLISS on the place categorization problem using the leave one out strategy followed by [32] to facilitate comparison. The system was trained on labeled data from 5 houses and tested on the 6th one. Average place categorization results from all 6 houses are reported. In practice, we only used every 3rd frame of the training set. We also omitted frames from the "transition" category during training and a model for this category was not learned. During runtime, any frame not belonging to one of 5 known labels was labeled as a "transition" frame. The maximum likelihood place labeling for a sequence is shown in Figure 6 along with groundtruth.

	Recognition Accuracy (%)
SIFT	84.2
CENTRIST (3x3)	81.6
CENTRIST (5x5)	83.0
Texture	81.2

Fig. 7. Results for place recognition

Figure 8 shows the result of place categorization across all sequences for each of the six categories in the dataset. Results for the VPC system are taken from [32], which does not give performance for the "transition" class. PLISS using SIFT features performs similar to the VPC system on average and is better on a few individual classes. However, this is a very strong result considering that we have used a simple generative model while VPC uses SVM classifiers. Note that we also include results of the "transition" class, which PLISS can recognize reasonably well even without any training data provided in that category. The SVM-based VPC system would not be expected to perform well in this class since it is a catch-all "other" category with widely varying visual characteristics. PLISS with CENTRIST is a good compromise if a faster system is required.

VIII. Discussion

We have presented PLISS, a system for place recognition and categorization based on change-point detection. PLISS has significant advantages over existing methods of being able to detect and learn previously unknown places and place

	Bedroom	Kitchen	Living Room	Bath	Dining	Transition	Average
SIFT	61.72	51.11	29.08	69.61	14.94	42.82	44.88
CENTRIST (3x3)	60.26	43.36	17.49	63.31	13.32	36.38	39.02
CENTRIST (5x5)	61.81	48.27	24.07	65.68	16.18	37.43	42.24
Texture	56.17	39.41	18.92	60.15	8.84	31.91	35.9
VPC	64.89	48.24	20.59	74.77	19.61	-	45.62

Fig. 8. Results for place categorization (% correctness): VPC average is over 5 categories while PLISS results include the "transition" category. VPC results have been taken from [32].

categories, of being able to learn online, and of being able to operate without any training data. Experiments on a difficult dataset show that, along with these advantages, PLISS also matches the state of the art in performance.

The basic assumption underlying PLISS is that places are sufficiently distinct to be identified visually. If this is not the case, i.e. the environment is severely perceptually aliased, performance will degrade as with any vision-based system. However, PLISS can be easily extended to incorporate multiple sensors to overcome such scenarios.

We envision significant performance improvements with more sophisticated place models. Such models may even incorporate object and context information from places, which cannot be done easily with SVM classifiers. Currently, PLISS does not distinguish between place labels and category labels and relies on the place models for making this distinction. It is future work to overcome this deficiency.

REFERENCES

[1] Modeling word burstiness using the dirichlet distribution. In *Intl. Conf. on Machine Learning (ICML)*, pages 545–552, 2005.

[2] R.P. Adams and D.J.C. MacKay. Bayesian online changepoint detection. Technical report, University of Cambridge, Cambridge, UK, 2007. arXiv:0710.3742v1 [stat.ML].

[3] H. Andreasson, A. Treptow, and T. Duckett. Localization for mobile robots using panoramic vision, local features and particle filter. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2005.

[4] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1–8, 2007.

[5] G. Casella and C.P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, March 1996.

[6] A. Taylan Cemgil, W. Zajdel, and B. Krose. A hybrid graphical model for robust feature extraction from video. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[7] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[8] S. R. Esterby and A. H. El-Shaarawi. Inference about the point of change in a regression model. *Applied Statistics*, 30(3):277–285, 1981.

[9] P. Fearnhead and P. Clifford. Online inference for hidden markov models. 2003.

[10] P. Fearnhead and Z. Liu. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B*, 69(4):589–605, 2007.

[11] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[12] J. Gaspar, N. Winters, and J. Santos-Victor. Vision-based navigation and environmental representations with an omnidirectional camera. *IEEE Trans. Robot. Automat.*, 16(6):890–898, Dec 2000.

[13] W. Hoeffding. Asymptotically optimal tests for multinomial distributions. *Ann. Math. Statist.*, 36:369–408, 1965.

[14] B. Kuipers and P. Beeson. Bootstrap learning for place recognition. In *Proc. 19^{th} AAAI National Conference on AI*, pages 174–180, Edmonton, Alberta, Canada, 2002.

[15] D. N. Lawley. A general method of approximating to the distribution of likelihood ratio criteria. *Biometrika*, 43:295–303, 1956.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[17] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *Intl. J. of Computer Vision*, 43:7–27, June 2001.

[18] E. Menegatti, T. Maeda, and H. Ishiguro. Image-based memory for robot navigation using properties of the omnidirectional images. *Journal of Robotics and Autonomous Systems*, 47(4):251–267, 2004.

[19] T.P. Minka. Estimating a dirichlet distribution. 2003.

[20] O. Martínez Mozos, A. Rottmann, R. Triebel, P. Jensfelt, and W. Burgard. Semantic labeling of places using information extracted from laser and vision sensor data. In *In Proc. of the IEEE/RSJ IROS 2006 Workshop: From Sensors to Human Spatial Concepts*, 2006.

[21] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[22] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Visual Perception, Progress in Brain Research*, 155, 2006.

[23] E. S. Page. Continuous inspection scheme. *Biometrika*, 41:100–115, 1954.

[24] A. Pronobis, O. M. Mozos, B. Caputo, and P. Jensfelt. Multi-modal semantic place classification. *Intl. J. of Robotics Research*, 2010. accepted.

[25] A. Rottmann, O. Martinez Mozos, C. Stachniss, and W. Burgard. Semantic place classification of indoor environments with mobile robots using boosting. In *Nat. Conf. on Artificial Intelligence (AAAI)*, 2005.

[26] C. Siagian and L. Itti. Biologically-inspired robotics vision monte-carlo localization in the outdoor environment. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[27] A. Tapus, N. Tomatis, and R. Siegwart. Topological global localization and mapping with fingerprint and uncertainty. In *Proceedings of the International Symposium on Experimental Robotics*, 2004.

[28] E.A. Topp, H. Hüttenrauch, H.I. Christensen, and K.S. Eklundh. Bringing together human and robotic environment representations - a pilot study. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006.

[29] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *Intl. Conf. on Computer Vision (ICCV)*, volume 1, pages 273–280, 2003.

[30] G. Tsechpenakis, D. Metaxas, O. Hadjiliadis, and C. Neidle. Robust online change-point detection in video sequences. In *2nd IEEE Workshop on Vision for Human Computer Interaction (V4HCI), in conjunction with the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[31] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 1023 – 1029, April 2000.

[32] J. Wu, H. Christensen, and J. M. Rehg. Visual place categorization: Problem, dataset, and algorithm. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[33] J. Wu and J. M. Rehg. Where am i: Place instance and category recognition using spatial pact. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[34] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Eur. Conf. on Computer Vision (ECCV)*, volume 2, pages 151–158, 1994.

[35] H. Zender, P. Jensfelt, O. M. Mozos, G.-J. Kruijff, and W. Burgard. An integrated robotic system for spatial understanding and situated interaction in indoor environments. In *Nat. Conf. on Artificial Intelligence (AAAI)*, 2007.

[36] Y. Zhai and M. Shah. A general framework for temporal video scene segmentation. In *Intl. Conf. on Computer Vision (ICCV)*, volume 2, pages 1111–1116, 2005.

[37] Z. Zivkovic, O. Booij, and B. Kröse. From images to rooms. *Journal of Robotics and Autonomous Systems*, 55(5):411–418, 2007.