Classifying Large-Scale Environment Shapes with Linear Optical Flow Templates

Anonymous CVPR submission

Paper ID ****

Abstract

In this paper we deal with classifying coarse, large-scale environment shapes using image motion observed by a moving camera or robot. We apply approximate Bayesian model selection over a set of learned linear optical flow templates to explain the motion between adjacent video frames. Each template is a learned probabilistic model of the flow fields that may be observed in a large-scale environment shape types such as 'left of path', 'center of path', 'right of path', etc. *We perform inference directly from spatial image gradients instead of first computing optical flow. Linear optical flow templates encode a set of* basis optical flow fields, *which are valid under the assumption that the scene depth field remains constant over time, hold for nearly arbitrary optics, and do not require a calibrated camera or known camera motion to learn. Our results show that our method classifies between training and evaluation datasets whose corresponding environment types are similar in large-scale structure but different in appearance and contain outliers like passing objects. We also perform a comparison with a neural network classifier using Gist features.*

1. Introduction

In this paper we deal with classifying coarse, large-scale environment shapes using the image motion observed by a moving camera or robot. For each pair of video frames we perform approximate Bayesian model selection over a set of learned linear optical flow templates, based on how well each explains image motion. The templates are learned probabilistic models of the flow fields that may be observed in environment shape types such as 'left of path', 'center of path', 'right of path', *etc*. Importantly, we perform inference directly from spatial image gradients instead of first computing optical flow.

Each linear optical flow template is a learned probabilistic model of the flow fields that may be observed in a single large-scale environment shape. As shown in Figure 1,

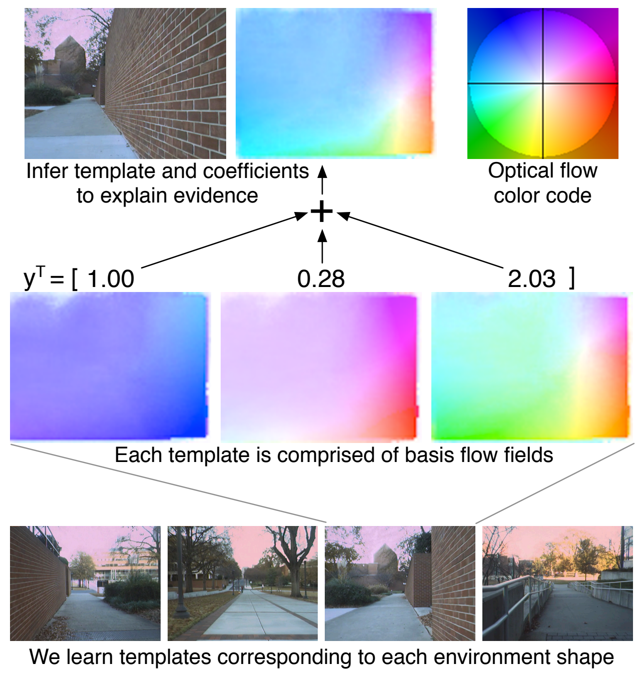


Figure 1: **Bottom**: we classify large-scale environment shape types such as 'left of path', 'center of path', 'right of path' with approximate model selection over a set of *linear optical flow templates*. **Middle**: a single linear optical flow template comprises a set of *basis flows* that span the subspace of possible optical flow fields resulting from ego-motion in the template's environment shape. **Top:** in the illustrated video frame, the image motion is explained by the particular linear combination specified by the latent variable assignment $y = [\ 1.00\ 0.28\ 2.03\ ]^{\mathsf{T}}$, which combines forward motion with some camera rotation caused by uneven ground and turning of the platform. Because we learn the templates with an unsupervised method, the basis flows do not correspond to canonical motions such as pure forward motion or pure pitch, and are instead combinations of such motions.

this model implicitly encodes the camera optics and typical scene depth field as seen by the camera. Explicitly this encoding is a linear mapping from latent variables to flow fields, through a set of basis flows..

We learn the linear optical flow templates from video recorded in each environment type using the method of Roberts *et al.* [25]. Known environment type labels, but not known camera motion, are required for learning.

Determining coarse environment shape is important for high-speed autonomous robot navigation. Modern autonomous mobile robot control methods discretely switch between different controllers, sometimes called "motion primitives", depending on the shape of the desired trajectory [8, 26]. In addition to information about the position and velocity of the robot, these methods need to know the discrete class, or type, of trajectory to follow or control to perform, which in turn depends on the environment shape. In autonomous driving, examples of important discrete environment shapes include 'wall on left', 'right turn'.

Additionally, coarse environment shape is important for high-level vision tasks that reason about 3D structure and object locations. A rough idea of the scene structure permits application of top-down knowledge such as "pedestrians appear on the ground". This idea has been investigated heavily under scenarios like urban driving and indoor scene understanding, with information from monocular cues, stereo, and laser point clouds [12, 3, 28].

Scene classification from image appearance is sensitive to coarse environment structure, though does not explicitly consider the structure. Oliva and Torralba [22] describe the "spatial envelope" and use image frequency and location information to classify *gist*, degrees of "size", "perspective", "openness", "depth", *etc*., and differentiate between mountains, streets, forests, *etc*.. Later work has combined this with other models and cues, including saliency [27] and explicit 3D information [29]. Recent work has even achieved autonomous driving by mapping between Gist and control action [1, 24]. Another approach to scene classification leverages the statistics of *local* image features [6, 16, 23]. Previous work had used similar methods for object recognition.

The current standard for autonomous driving is to compute a 2D traversability map for path planning using information from 3D laser range scans, stereo correspondences, and structure from motion; for examples see [17, 13]. Drawbacks of this scene include large computational resources required to deal with large point clouds, and powerful sensors, including 3D LIDAR and wide-baseline stereo rigs, to collect point clouds dense and complete enough to support planning. Recent methods produce similar traversability maps using image appearance and learning [19, 15]. Becker *et al*. [2] accumulate optical flow information over short spans of time to infer a dense 3D reconstruction of the scene in front of the robot.

Recent work has been towards obtaining 3D information for navigation aided by constraints from top-down models. Though not limited to robotics, Hoiem *et al*. [12] use monocular cues to estimate 3D structure. Brostow *et al*. [3] segment images into relevant regions such as street, sidewalk, car, *etc*. using structure-from-motion cues. Sturgess *et al*. [28] estimate similar segmentations using motion appearance and structure-from-motion information. Geiger *et al*. [9] infer 3D street and traffic patterns from video from a moving platform, combining information from vehicle tracking, vanishing points, and image appearance. Recent work in "Manhattan World" environments produces high-quality estimates of large structures like walls and floors, see for example [7, 30].

Optical flow was used for place recognition by Nourani-Vatani *et al*. [21], who matched flow fields to a database of locations using the flow field spatial statistics. Because they subtract rotation from the flow fields, the statistics are sensitive to scene depth. Mozos *et al*. [20] apply learning to categorize hallways, doorways, and rooms from 2D laser range scans coupled with visual features.

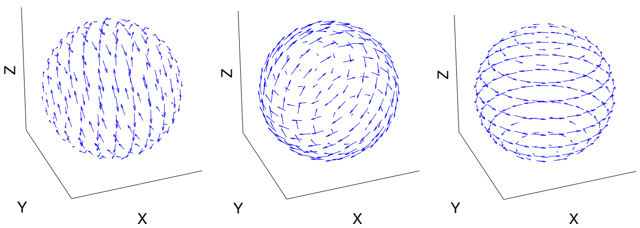
A difference between our method and these "pure machine learning" approaches is that we opt for a constrained optical flow model that leverages assumptions about the physical scene structure and camera motion. Though this can prevent overfitting and reduce sensitivity to noise, it also limits the types of variability that can be captured by our model. Thus for some situations a pure learning approach would be preferred. Our future plans include relaxing some model assumptions to capture more variability.

While our goal is related to that of scene classification, the information of image motion we use is quite different from the image appearance used in scene classification. Image appearance is sensitive to large-scale scene structure, but also to many other possible variations such as texture and lighting. Scene classification work has not been evaluated with respect to the goal of classifying large-scale environment structure as it is pertinent to mobile robot navigation. Thus, in this paper we compare our results to a neural network classifier using Gist features.

In Section 2 we introduce the notion of linear optical flow templates and in Section 3 describe our method of approximate model selection performed at runtime. In Section 4 we present quantitative and qualitative results and a comparison with scene classification with Gist features.

2. Linear Optical Flow Templates

In this section we introduce the notion of *probabilistic linear optical flow templates*. Each template models an environment shape type by encoding a continuous probability density over the possible flow fields a robot may observe as it moves through that environment type with *any velocity*.



(a) Rotation about $+X$, $v = [\,1\,0\,0\,0\,0\,0\,]^{\mathsf{T}}$. (b) Rotation about $+Y$, $v = [\,0\,1\,0\,0\,0\,0\,]^{\mathsf{T}}$. (c) Rotation about $+Z$, $v = [\,0\,0\,1\,0\,0\,0\,]^{\mathsf{T}}$.

Figure 2: For illustrative purposes, three basis flows corresponding to rotational camera motion, for rotation about canonical camera axes for a spherical imaging surface. These basis flows form the first through third columns of the velocity mapping flow matrix V.

Linear optical flow templates assume that the scene depth field observed by the camera is roughly constant over time.

2.1. Linearity of Optical Flow

An optical flow template encodes a linear mapping from low-dimensional set of latent variables $y \in \mathbb{R}^q$ to predicted optical flow

$$u_i = W_i y \tag{1}$$

where $W_i \in \mathbb{R}^{2\times q}$ is the linear mapping to flow corresponding to the i^{th} image location.

Linear optical flow templates take advantage of the linear relationship between camera velocity and optical flow when scene depth at each image location remains constant over time. The optical flow u_i at the i^{th} image location is related to camera velocity $v = [\omega_x\, \omega_y\, \omega_z\, v_x\, v_y\, v_z]^{\mathsf{T}}$, assuming no noise, according to

$$u_i = V_i\,(z_i)\,v \tag{2}$$

where $V_i\,(z_i)$ is an optical flow matrix, which depends on the camera optics and which depends nonlinearly on the scene depth z_i at the i^{th} image location. For a standard perspective camera, the flow matrix is (for example, see [11])

$$V_i\,(f, z) \triangleq \begin{bmatrix} \frac{x_i y_i}{f} & \frac{-f-x_i^2}{f} & y_i & \frac{-f}{z_i} & 0 & \frac{x_i}{z_i} \\ \frac{f+y_i^2}{f} & \frac{-x_i y_i}{f} & -x_i & 0 & \frac{-f}{z_i} & \frac{y_i}{z_i} \end{bmatrix}, \tag{3}$$

where (x_i, y_i) is the image location at the i^{th} pixel. When the focal length f and the scene depth at each pixel z_i remain constant over time, the flow matrices V_i for each pixel are also constant, and thus V also defines a special linear optical flow template where the velocity components are the latent variables.

Remarkably, this linearity holds for more general cameras of nearly arbitrary optics for which a parametric calibration is not possible, including distortion, catadioptrics, and multiple viewpoints, as shown by Roberts *et al.* To illustrate this, Figure 2 shows the first three columns of a velocity-mapping flow matrix for a spherical imaging surface. In these cases flow matrices W in latent variables, or flow matrices V in platform velocity, may be learned from recorded video [25] using unsupervised and supervised methods, respectively.

In our application, the latent variable "version" of linear optical flow templates as in Eq. 1 has advantages over the velocity mapping version in Eq. 2, so we opt to use the former in this paper. First, while calculating the velocity mapping V requires either known robot's velocity while learning, or a known camera calibration and scene structure, the latent variable mapping W may be learned from recorded video with *unknown* camera motion using the method presented in [25].

An additional advantage of the latent variable mapping is that some variations in inverse depth $\frac{1}{z_i}$ are approximately captured by a linear relationship with the latent variables, yet are not linear in the camera velocity. This allows the linear optical flow template to remain valid under small amounts of "nonlinear-$\frac{1}{z_i}$" motions, like side-to-side and pitching motions of a mobile ground robot. Our experiments include such motions.

2.2. Robust Probabilistic Linear Mapping

Instead of a deterministic relationship, a linear optical flow template defines a probability density on optical flow that is robust to outliers,

$$p\,(u_i\,|\,y, \lambda_i) \propto \begin{cases} \mathcal{N}\left(W_i y, \Sigma_u^{\mathrm{v}}\right), & \lambda_i = 1 \\ \mathcal{N}\left(W_i y, \Sigma_u^{\mathrm{f}}\right), & \lambda_i = 0 \end{cases} \tag{4}$$

where $\Sigma_u^{\mathrm{v}} \in \mathbb{R}^{2\times 2}$ is the (small) covariance of an optical flow vector that is an inlier to the template, Σ_u^{f} is the (large) covariance of an outlier to the template, and $\lambda_i \in \{1, 0\}$ indicates a pixel is an inlier or an outlier, respectively, to the template. In this paper we will derive an expectation-maximization algorithm to bound this likelihood using estimated inlier probabilities, but inlier assignments could also be calculated using other methods, such as RANSAC. Thus an optical flow template is $\left(W, \Sigma_u^{\mathrm{v}}, \Sigma_u^{\mathrm{f}}, p\,(\lambda)\right)$, where $p\,(\lambda)$ is a constant Bernoulli prior probability that any pixel is an inlier to the template.

2.3. Learning the Linear Optical Flow Templates

We learn the optical flow templates $\left(W_k, \Sigma_{uk}^{\mathrm{v}}, \Sigma_{uk}^{\mathrm{f}}, p\,(\lambda_k)\right)$ for each k^{th} environment type from videos collected during robot motion using the method presented in [25]. This method uses an expectation-maximization algorithm to optimize for the mapping W treating the latent variables and inlier/outlier indicators as hidden variables. To compute sparse optical flow input to the learning method we use the pyramidal Lucas-Kanade tracker [18] in OpenCV.

We apply the method independently to videos captured separately in each *known* environment type, with *unknown* camera velocity. Learning multiple optical flow templates in an unsupervised manner, from arbitrary video with *unknown* environment type labels, is part of our ongoing work.

3. Inferring the Environment Type

In this section we describe a method for inferring the probability of each environment type directly from the spatial image gradients of two adjacent video frames. This is greatly preferable to *first* extracting optical flow because optical flow is an under-constrained and computationally-intensive problem in the absence of top-down information, in part due to the aperture problem. The optical flow templates provide top-down information, reducing the problem down to optimizing only a handful of latent variables (their dimensionality q ranges from 3 to 6 in our experiments).

We now derive the posterior distribution over the environment type k_t at time t conditioned on measuring the previous and current frames, $I_{t,t-1}$. Ideally, this would be obtained by marginalizing out the unknown latent variables y_{kt} and indicator variables λ_{kt},

$$p\left(k_t \,|\, I_{t,t-1}\right) = \int\limits_{y_{kt}} \sum_{\lambda_{kt}} p\left(k_t, y_{kt}, \lambda_{kt} \,|\, I_{t,t-1}\right)$$
$$\propto \int\limits_{y_{kt}} \sum_{\lambda_{kt}} p\left(I_t \,|\, y_{kt}, \lambda_{kt}, k_t, I_{t-1}\right) p\left(y_{kt}\right) p\left(\lambda_{kt}\right) p\left(k_t\right) \tag{5}$$

where we assume $p\left(k_t\right)$ to be a categorical, i.e. constant-probability, prior over the environment types.

3.1. Expected Log-likelihood Approximation

In practice, we replace the sum over the latent variable assignments with an expected log-likelihood formulation from an expectation-maximization (EM) algorithm. The true sum over λ_{kt} in Eq. 5 is an intractable sum over all possible combinations of inlier assignments for all pixels. Given an expectation $\langle\lambda_{kti}\rangle \in [0, 1]$ of the inlier indication, a lower bound (see [5]) on the image likelihood with λ_{kt} marginalized out is

$$\begin{aligned} p\left(I_t \,|\, y_{kt}, k_t, I_{t-1}\right) &= \sum_{\lambda_{kt}} p\left(I_t \,|\, y_{kt}, \lambda_{kt}, k_t, I_{t-1}\right) \\ &= \sum_{\lambda_{kt}} \prod_i p\left(I_{ti} \,|\, y_{kt}, \lambda_{kti}, k_t, I_{t-1}\right) \\ &\approx \exp \sum_i \left(\langle\lambda_{kti}\rangle \mathcal{L}\left(I_{ti} \,|\, y_{kt}, \lambda_{kti}{=}1, k_t, I_{t-1}\right) + \right. \\ &\qquad \left. \langle 1-\lambda_{kti}\rangle \mathcal{L}\left(I_{ti} \,|\, y_{kt}, \lambda_{kti}{=}0, k_t, I_{t-1}\right) \right), \end{aligned} \tag{6}$$

where $\mathcal{L}\left(\cdot\right) \triangleq \log p\left(\cdot\right) + C$ is a log-likelihood. Using a similar scheme, we replace the prior $p\left(\lambda_{kt}\right)$ in Eq. 5 with

$$p\left(\lambda_{kt}\right) \approx \exp \sum_i \left(\mathcal{L}\left(\lambda_k{=}1\right) \langle\lambda_{kti}\rangle + \mathcal{L}\left(\lambda_k{=}0\right) \langle 1-\lambda_{kti}\rangle \right) \tag{7}$$

In practice, we find these lower bounds to be suitable approximations for the purpose of model selection.

Using EM, the expectation $\langle\lambda_{kti}\rangle$ is evaluated as

$$\begin{aligned} \langle\lambda_{kti}\rangle &\equiv p\left(\lambda_{kti}{=}1 \,|\, y_{kt}, k_t, I_{t,t-1,i}\right) \\ &= \frac{p\left(I_{ti} \,|\, y_{kt}, k_t, I_{t-1,i}\right) p\left(\lambda_{it}\right)\big|_{\lambda_{it}=1}}{\sum\limits_{\lambda_{it}=\{1,0\}} p\left(I_{ti} \,|\, y_{kt}, k_t, I_{t-1,i}\right) p\left(\lambda_{it}\right)}. \end{aligned} \tag{8}$$

3.2. Integrating out Optical Flow

In order to perform inference directly on image gradients without first computing optical flow, and thereby evaluate the likelihood $p\left(I_{ti} \,|\, y_{kt}, \lambda_{kti}, k_t, I_{t-1}\right)$ that appears in Eq. 6, we marginalize out the unknown optical flow,

$$p\left(I_{ti} \,|\, y_{kt}, k_t, I_{t-1}\right) = \int\limits_{u_{ti}} p\left(I_{ti} \,|\, u_{ti}, I_{t-1}\right) p\left(u_{ti} \,|\, y_{kt}, k_t\right). \tag{9}$$

An issue is that the image is nonlinear so Eq. 9 cannot be evaluated exactly in closed-form. Instead, we approximate it with a Gaussian centered at the maximum-likelihood estimate (MLE) of the latent variables. To find the MLE we perform nonlinear Gauss-Newton optimization. We start with an initial guess of the latent variables $\mathring{y}_t$, which induces $\mathring{u}_t \equiv V_k \mathring{y}_{kt}$ signifying the optical flow predicted according to the template given the latent variable estimate. Let $x_i \in \mathbb{R}^2$ be the image location at the i^{th} pixel location. Linearizing the image by computing the spatial gradient ∇I_{ti} at each i^{th} image location, we define the image likelihood $p\left(\mathcal{I}_t \,|\, u_t\right)$ as a probabilistic version of the brightness constancy constraint from classical optical flow estimation,

$$p\left(I_{ti} \,|\, \delta u_{ti}, I_{t-1}\right) \approx \mathcal{N}(I_{t-1}(x_i - \mathring{u}_{ti}) - \nabla I_{ti} \delta u_{ti}, \sigma_{\mathcal{I}}), \tag{10}$$

where $\delta u_{ti} \equiv u_{ti} - \mathring{u}_{ti}$, $\sigma_{\mathcal{I}}$ is the standard deviation of a small amount of Gaussian noise on the image intensity, and where $I\left(x\right)$ is the image intensity at the pixel coordinates x. In practice we evaluate the image intensity by resampling with a Gaussian kernel because in general the pixel locations are non-integral.

Marginalizing out the optical flow in Eq. 9 is then done in closed-form using this Gaussian-approximated image-likelihood in Eq. 10 and the expected log-likelihood approximation from Eq. 6,

$$p\left(I_t \,|\, \delta y_t, k_t\right) \propto \exp \frac{-1}{2} \sum_i J_{ti}^2 \left(\overline{I_{ti}} - \nabla I_{ti} V_{ki} \delta y_{kt}\right)^2, \tag{11}$$

where $\overline{I_{ti}} \triangleq I_{ti} - I_{t-1}\left(x_i - \mathring{u}_{ti}\right)$ and

$$
\begin{aligned}
J_{ti}^2 \triangleq{} & \langle \lambda_{kti} \rangle \left(I_{ti} \Sigma_u^{\mathrm{v}} I_{ti}^\mathsf{T} + \sigma_\mathcal{I}^{\mathrm{v}} \right)^{-1} \\
& + \langle 1 - \lambda_{kti} \rangle \left(I_{ti} \Sigma_u^{\mathrm{f}} I_{ti}^\mathsf{T} + \sigma_\mathcal{I}^{\mathrm{f}} \right)^{-1}
\end{aligned}
$$

is the precision on the spatial image gradient with the flow u_{ti} marginalized out, and $\delta y_t \equiv y_t - \mathring{y}_t$. In the quantity $\nabla I_{ti} V_{ki} \delta y_{kt}$ in Eq. 11, the term $\nabla I_{ti} V_{ki}$ is the image Jacobian w.r.t. the latent variables, analogous to the Jacobian images w.r.t. camera motion described in more detail in [4].

We iteratively update the latent variable estimate $\mathring{y}_t$ with the increment δy_t, until at convergence it becomes the final center of the Gaussian approximation of Eq. 11.

3.3. Computing the Environment Type Marginal

Finally, after approximating the marginal over the inlier indicator variables with the expected log-likelihoods in Eqs. 6 and 9, and approximating the image probability in Eq. 6 as a marginal Gaussian centered around the MLE of the latent variables y_{kt} (with the flow u_t marginalized out) in Eq. 11, we can write the environment type marginal as

$$
\begin{aligned}
p\left(k_t \,|\, I_{t,t-1}\right) \propto & \left(\int_{y_{kt}} p\left(I_t \,|\, y_{kt}, k_t, I_{t-1}\right) p\left(y_{kt}\right) \right) \\
& p\left(\lambda_{kt}\right) p\left(k_t\right)
\end{aligned}
\tag{12}
$$

where each component likelihood is either constant or Gaussian. The integral is over the joint Gaussian $p\left(I_t \,|\, y_{kt}, k_t, I_{t-1}\right) p\left(y_{kt}\right) \equiv p\left(I_t, y_{kt} \,|\, k_t, I_{t-1}\right)$,

$$
\begin{aligned}
& p\left(I_t, y_{kt} \,|\, k_t, I_{t-1}\right) \propto \exp \\
& \frac{-1}{2} \left(\sum_i J_{ti}^2 \left(\overline{I_{ti}} - \nabla I_{ti} V_{ki} \delta y_{kt} \right)^2 + \left\| \mathring{y}_{kt} + \delta y_{kt} \right\|^2 \right)
\end{aligned}
\tag{13}
$$

Interestingly, while integrating out the latent variable increment δy_{ky} using Gaussian elimination would result in the marginal $p\left(I_t \,|\, k_t, I_{t-1}\right)$ having a dense, intractable $I \times I$ information matrix, the structure of the joint Gaussian in Eq. 13 leads to an efficient factorization of the marginal using the Schur complement. To do this, note that the log of Eq. 13 can be written as

$$
\frac{-1}{2} \begin{bmatrix} \delta y_{kt} \\ \overline{I_t} \\ 1 \end{bmatrix}^\mathsf{T} \begin{bmatrix} A_{q \times q} & B_{q \times I}^\mathsf{T} & \mathring{y}_{kt} \\ B_{I \times q} & D_{I \times I} & \mathbf{0}_{I \times 1} \\ \mathring{y}_{kt}^\mathsf{T} & \mathbf{0}_{1 \times I} & \mathring{y}_{kt}^\mathsf{T} \mathring{y}_{kt} \end{bmatrix} \begin{bmatrix} \delta y_{kt} \\ \overline{I_t} \\ 1 \end{bmatrix}
\tag{14}
$$

where A, B, and D are

$$
\begin{aligned}
& \underset{q \times q}{A} \triangleq \mathbf{I}_{q \times q} + \textstyle\sum_i J_{ti}^2 V_{ki}^\mathsf{T} \nabla I_{ti}^\mathsf{T} \nabla I_{ti} V_{ki} \\
& \underset{I \times q}{B} \triangleq \begin{bmatrix} -J_{t1}^2 \nabla I_{t1} V_1 \\ -J_{t2}^2 \nabla I_{t2} V_2 \\ \vdots \end{bmatrix} \quad \underset{I \times I}{D} \triangleq \begin{bmatrix} J_{t1}^2 & & \\ & J_{t2}^2 & \\ & & \ddots \end{bmatrix}
\end{aligned}
\tag{15}
$$

Using the Schur complement, the information matrix $\Lambda_\mathcal{I}$, information vector $\eta_\mathcal{I}$, and constant term $f_\mathcal{I}$ of the marginal $p\left(I_t \,|\, k_t, I_{t-1}\right)$ are

$$
\begin{aligned}
\Lambda_\mathcal{I} = D - B A^{-1} B^\mathsf{T} \quad & \eta_\mathcal{I} = -B A^{-1} \mathring{y}_{kt} \\
f_\mathcal{I} = \mathring{y}_{kt}^\mathsf{T} & \left(\mathbf{I} - A^{-1} \right) \mathring{y}_{kt}
\end{aligned}
\tag{16}
$$

To compute the normalizing constant of the resulting Gaussian, the determinant of the information matrix can be calculated efficiently using the matrix determinant lemma,

$$
\left| \Lambda_\mathcal{I} \right| = \det\left(A - B^\mathsf{T} D^{-1} B \right) \det A^{-1} \det D
\tag{17}
$$

Combining Eqs. 16 and 17, the marginal image likelihood is

$$
\begin{aligned}
& p\left(I_t \,|\, k_t, I_{t-1}\right) = \\
& \quad (2\pi)^{\frac{-I}{2}} \left| \Lambda_\mathcal{I} \right|^{\frac{1}{2}} \exp \frac{-1}{2} \left(\overline{I_t}^\mathsf{T} \Lambda_\mathcal{I} \overline{I_t} + \overline{I_t}^\mathsf{T} \eta_\mathcal{I} + f_\mathcal{I} \right)
\end{aligned}
\tag{18}
$$

where $\overline{I_t}$ is the vector of all $\overline{I_{ti}}$ concatenated together for all image locations i.

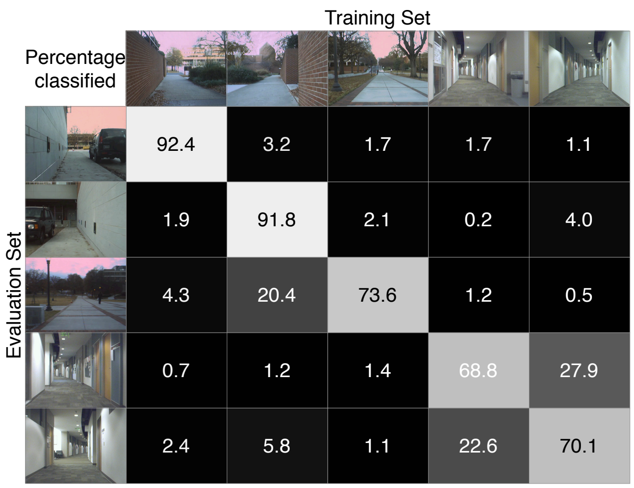
Importantly, with the above factorization evaluating the environment type marginal in Eq. 12 is computationally efficient. This is because in Eq. 18 neither the products with the dense $I \times I$ information matrix $\Lambda_\mathcal{I}$ nor the determinant $|\Lambda_\mathcal{I}|$ written need to be calculated directly. Instead Eqs. 16 and 17 evaluate them efficiently due to the diagonal form of D, the small width q of B, and the small size $q \times q$ of A. Here q is the length of the latent variable vector y_t, which in our experiments is on the order of $q \approx 5$.

The last piece required to evaluate the environment type marginal in Eq. 12 is to normalize it by dividing by the sum of the evaluated likelihoods of Eq. 12 for each k_t.

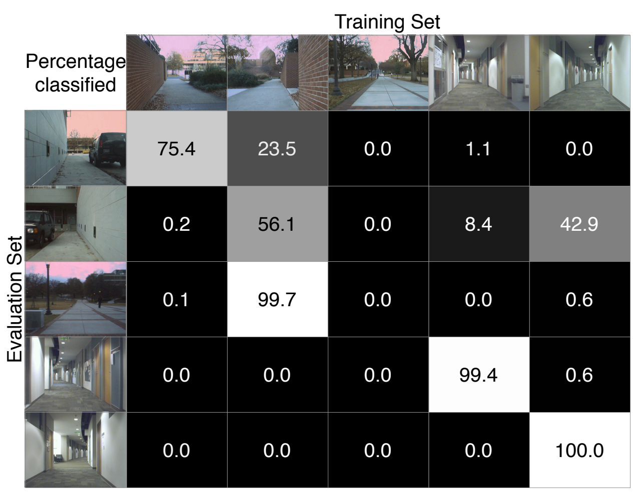
4. Experimental Results

We evaluate our method with qualitative and quantitative accuracy experiments, as well as a quantitative accuracy comparison with a neural network classifier using Gist features. All datasets were collected from a 640×480 30Hz Unibrain Fire-i camera mounted on a wheeled platform.

The free parameters of our method are the standard deviation of the image intensity noise for inlier and outlier pixels, for which we used $\sigma_\mathcal{I}^{\mathrm{v}} = \frac{1}{255}$ and $\sigma_\mathcal{I}^{\mathrm{f}} = \frac{5}{255}$, both in normalized grayscale units, and the per-pixel inlier prior,



(a) Our method, model selection over linear optical flow templates (overall accuracy 74.7%)



(b) Neural network classifier using Gist features (overall accuracy 67.3%)

Figure 3: Confusion matrices showing classification results of our method and a neural network classifier using Gist features. The images are representative of each environment type in the training and testing datasets. Our higher accuracy on 'left wall', 'right wall', and 'walkway' highlight our use of image motion information versus image appearance. Gist's higher accuracy in differentiating between 'left curve' and 'right curve' is due to the appearance similarity between the training and testing sets, which were taken on two different floors of the same building. The image motion information, on the other hand, is subtle in these two environments because the hallway curvature is gentle.

$p(\lambda_{kti}{=}1) = 0.95$. The optical flow covariances Σ_k^{v} and Σ_k^f are learned from the data as part of the templates.

In our implementation, we perform the optimization in Section 3.2 at multiple scales, creating a Gaussian-resampled pyramid both of the images and the basis flows, and initializing the optimization at each level from the next smaller one. The smallest level is initialized with $y_{kt} = \mathbf{0}$. We initialize all indicator expectations with $\langle \lambda_{kti} \rangle = 1$. We perform the optimization using Gauss-Newton optimization.

Additionally, it is not necessary to perform inference up to the largest pyramid level. In our experiments we stop at level 3, corresponding to 80×60 images and basis flows scaled down from the original 640×480. Additionally, for optimization and inference (i.e. in all ranges over i in Section 3), we sample only every other pixel, meaning that for the same image size, 1200 pixels are sampled at the largest pyramid level. With these parameters our single-threaded research implementation operates at approximately 15Hz on a 2.2GHz Intel Core i7 laptop.

Our method is highly parallelizable, in that the optimization and likelihood computation described in Section 3 may be performed independently and in parallel for each template. Also the image filtering operations such as gradient computations, resampling, and differencing may be threaded or even implemented on DSP, FPGA, or GPU hardware [14].

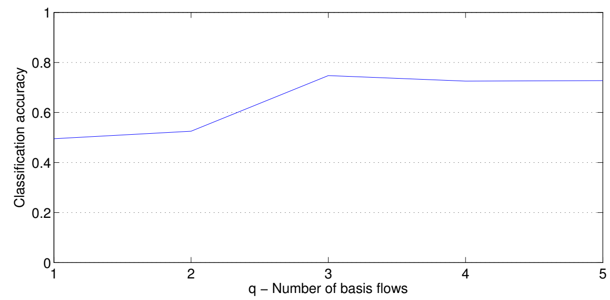


Figure 4: Classification accuracy for linear optical flow templates learned with various numbers of basis flows, i.e. latent variable dimensionalities q, learned and evaluated with the same datasets as in Figure 3.

4.1. Quantitative Evaluation

We learned linear optical flow templates for the environments exemplified by the top row of thumbnails of Figure 3, and performed inference on the environments exemplified by the left column of thumbnails. For some of these environments, between the training and evaluation sets, the large-scale structure is similar but the image appearance is quite different. We empirically selected to learn templates with $q = 3$ basis flows as this provided the highest accuracy.

Figure 3 shows the confusion matrices for our method and a neural network classifier using Gist features. The

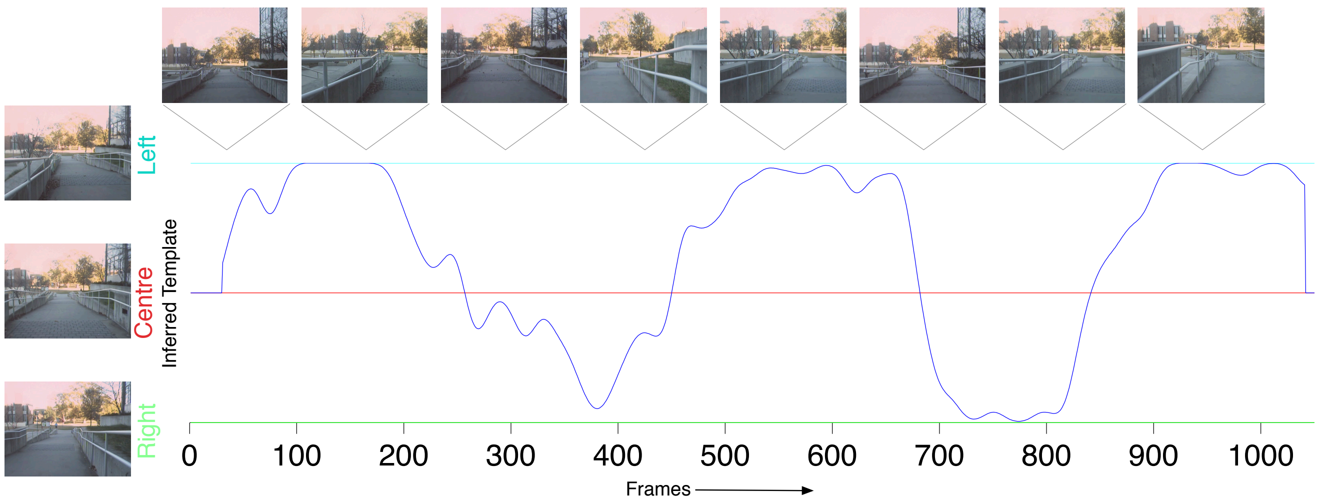


Figure 5: Time-smoothed plot of the inferred environment type at each frame. Camera motion for this evaluation sequence was a rough sinusoidal motion along the walkway.

Gist features were computed by the software[1] accompanying [27]. We selected the subset of Gist features suggested by [27], and trained a neural network with 200 and 100 node hidden layers for 500 epochs (verifying that error on a hold-out set did not increase during training) using Weka [10].

Figure 4 shows the classification accuracy on the same evaluation set for models learned with various numbers of basis flows, i.e. latent variable dimensionality q.

Our accuracy on average is comparable to the Gist feature classifier, but the key point to note is the difference in the accuracy of the Gist feature classifier when the training and evaluation images appear similar versus when they appear different. When appearance differs, the accuracy of Gist decreases, where the accuracy of our method remains the approximately the same.

4.2. Qualitative Evaluation

We learned three linear optical flow templates on video sequences collected from the platform moving down a walkway. One template was learned from the platform moving along the left side of the walkway, one along the center, and one along the right side.

The dataset is far from perfect, as the walkway alternated between sloped and flat causing platform pitch, contained platform vibration, and yawing motion of the platform as it did not move in a perfectly straight line. Additionally large areas of the frames are textureless.

For evaluation, we performed inference using our method amongst the three learned templates. The input video for evaluation was a sequence in which the platform moved in a roughly sinusoidal motion between the left and right sides of the walkway. The results are shown in Figure 5, which a time-smoothed plot of the environment type with the highest likelihood at each frame.

[1] Available from http://ilab.usc.edu/siagian/Research/Gist/Gist.html

5. Summary

In this paper we presented a method for classifying coarse environment shape from image motion. To do this classification, the method performs approximate model selection over a collection of linear optical flow templates. Each template encodes a coarse environment shape, by means of a set of *basis flows* spanning the subspace of optical flow fields that a moving platform may observe in that environment, under the assumption of per-pixel depth constancy over time. The input is a video stream, and the output is a set of likelihoods for each frame that the image change from the previous frame is explained by each linear optical flow template. Inference takes place directly on spatial image gradients, not requiring optical flow to be computed first. Our results show that our method classifies between training and evaluation datasets whose corresponding environment types are similar in large-scale structure but different in appearance and contain outliers like passing objects.

References

[1] C. Ackerman and L. Itti. Robot steering with spectral image information. *IEEE Transactions on Robotics*, 21(2):247–251, Apr 2005. 2

[2] F. Becker, F. Lenzen, J. H. Kappes, and C. Schnörr. Variational recursive joint estimation of dense scene structure and camera motion from monocular high speed traffic sequences. In *International Conference on Computer Vision*, 2011. 2

[3] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point

clouds. In *Proceedings of the European Conference on Computer Vision*, 2008. 2

[4] F. Dellaert, S. Thrun, and C. Thorpe. Jacobian images of super-resolved texture maps for model-based motion estimation and tracking. In *IEEE Workshop on Applications of Computer Vision (WACV)*, 1998. 5

[5] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977. 4

[6] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. *CVPR*, pages 524–531, 2005. 2

[7] A. Flint, D. Murray, and I. Reid. Manhattan scene understanding using monocular, stereo, and 3d features. In *Proceedings of the International Conference on Computer Vision*, 2011. 2

[8] E. Frazzoli, M. Dahleh, and E. Feron. A hybrid control architecture for aggressive maneuvering of autonomous helicopters. In *Proceedings of the IEEE Conference on Decision and Control*, volume 3, pages 2471–2476, 1999. 2

[9] A. Geiger, M. Lauer, and R. Urtasun. A generative model for 3d urban scene understanding from movable platforms. In *Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, USA, June 2011. 2

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009. 7

[11] D. Heeger and A. Jepson. Subspace methods for recovering rigid motion I: Algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117, 1992. 3

[12] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *International Conference of Computer Vision (ICCV)*. IEEE, October 2005. 2

[13] A. Huertas, L. Matthies, and A. Rankin. Stereo-based tree traversability analysis for autonomous off-road navigation. In *IEEE Workshops on Application of Computer Vision, WACV/MOTIONS'05*, volume 1, pages 210–217, 2005. 2

[14] B. Hutchings, B. Nelson, S. West, and R. Curtis. Optical flow on the ambric massively parallel processor array (MPPA). In *IEEE Symposium on Field Programmable Custom Computing Machines (FCCM)*, pages 141–148, 2009. 6

[15] Y. Khan, P. Komma, and A. Zell. High resolution visual terrain classification for outdoor robots. In *IEEE ICCV Workshop on Challenges and Opportunities in Robot Perception*, 2011. 2

[16] C. S. L. Lazebnik 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. 2

[17] J. Lalonde, N. Vandapel, D. Huber, and M. Hebert. Natural terrain classification using three-dimensional ladar data for ground robot mobility. *Journal of Field Robotics*, 23(10):839–861, 2006. 2

[18] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, volume 3, 1981. 4

[19] J. Michels, A. Saxena, and A. Y. Ng. High speed obstacle avoidance using monocular vision and reinforcement learning. In *Intl. Conf. on Machine Learning (ICML)*, 2005. 2

[20] O. M. Mozos and W. Burgard. Supervised learning of topological maps using semantic information extracted from range data. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2772–2777, 2006. 2

[21] N. Nourani-Vatani, P. V. K. Borges, J. M. Roberts, and M. V. Srinivasan. Topological localization using optical flow descriptors. In *Proceedings of the 1st IEEE Workshop on Challenges and Opportunities in Robotic Perception, with ICCV'2011*, 2011. 2

[22] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001. 2

[23] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *Proceedings of the International Conference on Computer Vision*, 2011. 2

[24] N. Pugeault and R. Bowden. Driving me around the bend: Learning to drive from visual gist. In *Proceedings of the 1st IEEE Workshop on Challenges and Opportunities in Robotic Perception, with ICCV'2011*, 2011. 2

[25] R. Roberts, C. Potthast, and F. Dellaert. Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009. 2, 3

[26] T. Schouwenaars, B. Mettler, E. Feron, and J. How. Robust motion planning using a maneuver automation with built-in uncertainties. In *Proceedings of the American Control Conference*, volume 3, pages 2211–2216, 2003. 2

[27] C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):300–312, 2007. 2, 7

[28] P. Sturgess, K. Alahari, L. Ladicky, and P. Torr. Combining appearance and structure from motion features for road scene understanding. In *Proceedings of the British Machine Vision Conference*, 2009. 2

[29] A. Swadzba and S. Wachsmuth. Indoor scene classification using combined 3D and Gist features. In *Asian Conference on Computer Vision*, pages 201–215. Springer, 2010. 2

[30] G. Tsai, C. Xu, J. Liu, and B. Kuipers. Real-time indoor scene understanding using bayesian filtering with motion cues. In *Proceedings of the International Conference on Computer Vision*, 2011. 2