Classifying Environment Shape from Image Motion for Mobile Robot Path Following

Anonymous ECCV submission

Paper ID 1696

Abstract. For a high-speed mobile robot following a road, hallway, or other type of path, modern robot motion controllers must know the *coarse shape* of the path, such as whether it curves left or right, or on which sides it is bounded by dangerous obstacles such as walls. In this paper, towards addressing the perceptual needs of high-speed mobile robot controllers, we classify these coarse shapes directly from the image motion observed by a camera mounted on a robot driving along the paths. We apply approximate Bayesian model selection over a set of learned probabilistic optical flow subspaces to explain the change between adjacent video frames. We perform inference directly from spatial image gradients instead of first computing optical flow. Optical flow subspaces encode a set of *basis optical flow fields*, which are valid under the assumption that the scene depth field remains constant over time, a conditional that holds approximately for a robot following a path-like environment. We do not require a calibrated camera or known camera motion. Our experimental results support the claims that our method efficiently classifies between multiple coarse path shapes, and that to do so it is important to use information from image motion, instead of image appearance.

1 Introduction

For a high-speed mobile robot following a road, hallway, or other type of path, modern robot motion controllers must know the *coarse shape* of the path, such as whether it curves left or right, or on which sides it is bounded by dangerous obstacles such as walls. Coarse path shape is important to know because it usually determines the desired robot trajectory, for example to stay in the middle of the path when going straight, or to execute optimal sliding turns in sharp corners. Coarse shape is in contrast to a detailed 3D reconstruction of the environment, which is often not useful for high-speed, dynamic navigation and control, for reasons we discuss later.

Often, control switches between different types of controllers depending on the *discrete* path shape encountered. For example, in case of sharp curves it is often desirable to execute sliding turn maneuvers common in rally racing, versus during gentle turns the controller does not slide the tires. In modern control methods these switched controllers are often called "motion primitives" [1,2]. When navigating using vision, a visual perception method must classify which discrete path shape the vehicle is encountering.

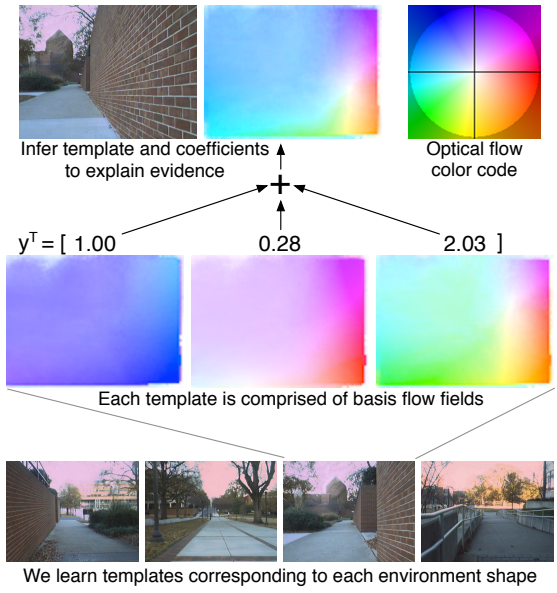


Fig. 1: **Bottom**: we classify large-scale environment shape types such as 'left of path', 'center of path', 'right of path' with approximate model selection over a set of *linear optical flow templates*. **Middle**: a single linear optical flow template comprises a set of *basis flows* that span the subspace of possible optical flow fields resulting from egomotion 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.

Additionally, coarse environment shape is important for high-level vision tasks that detect and track obstacles and objects such as trees, cars, and pedestrians. 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 [3,4,5].

In this paper, towards addressing the perceptual needs of high-speed mobile robot controllers described above, we develop and evaluate a method to classify which discrete coarse path shapes on which a vehicle is traveling, directly from the instantaneous image motion observed by an onboard camera. To do this, we perform approximate Bayesian model selection over learned models of the optical flow fields that can be observed while traveling in each path shape, to

determine which optical flow model best explains the observed image intensity changes.

For the optical flow models for each shape class, we use probabilistic optical flow subspaces, which we learn using the method of Roberts *et al.* [6]. Videos with known environment type labels are required for learning, but neither camera motion nor calibration need to be known. As shown in Figure 1, 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.

After reviewing related work in Section 2, we introduce the notion of linear optical flow templates in Section 3, and in Section 4 describe our method of approximate model selection performed at runtime. In Section 5 we present quantitative and qualitative results.

2 Related Work

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 [7,8]. 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 [9,10]. Becker *et al.* [11] 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.* [3] use monocular cues to estimate 3D structure. Brostow *et al.* [4] segment images into relevant regions such as street, sidewalk, car, *etc.* using structure-from-motion cues. Sturgess *et al.* [5] estimate similar segmentations using motion appearance and structure-from-motion information. Geiger *et al.* [12] 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 [13,14].

Optical flow was used for environment shape recognition with a discriminative learning method by Nourani-Vatani *et al.* [15], 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.* [16] 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.

Our goal of classifying coarse path shape is not addressed by scene classification from image appearance. It has nonetheless been shown that image appearance methods are in fact sensitive to coarse environment structure, as described by Oliva and Torralba [17], who 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 combined Gist with other models and cues, including saliency [18] and explicit 3D information [19]. Recent work has even achieved autonomous driving by mapping between Gist and control action [20,21]. Another approach to scene classification leverages the statistics of *local* image features [22,23,24]. Previous work had used similar methods for object recognition.

In this paper we show that despite appearance-based scene classification being sensitive to coarse environment structure, much better performance is obtained for the goal of path shape classification when we use information from image motion, as it is much more directly sensitive to environment structure. Image appearance based methods can be confounded by pure-appearance variations such as texture and lighting. To show this, in this paper we compare our results to a neural network classifier using Gist features.

3 Linear Optical Flow Templates

In this section we describe the compact and efficient optical flow model that we learn for each coarse path shape. This model encodes a probability density over the optical flow fields that the vehicle could observe while traveling on the path, with any velocity. We will refer to the models of each coarse path shape as *linear optical flow templates*.

The key observation making the model compact and efficient is that for a vehicle driving along a path-like environment, the optical flow the vehicle may observe from an on-board camera lies very close to a subspace. For each optical flow template, we thus choose the probabilistic optical flow subspace model, as described by Roberts *et al.* [6]. This section summarizes the explanation of that paper.

3.1 Linearity of Optical Flow

An optical flow subspace 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.

The key assumption of optical flow subspaces is that scene distance at each image location remains approximately constant *over time*, leading to a linear relationship between camera velocity and optical flow. To show why this linearity holds, 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 [25])

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

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.

In fact though, remarkably, we can leverage the linearity proved by Eq. 3 to learn and use the subspace W in Eq. 1 without knowing the camera calibration nor motion, and even for cameras with nearly-arbitrary, non-pinhole, optics, as shown in [6].

3.2 Robust Probabilistic Linear Mapping

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

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

where $\Sigma_u^{\text{v}} \in \mathbb{R}^{2\times 2}$ 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 use expectation-maximization to perform inference with this model, though other methods, such as RANSAC could be used.

3.3 Learning the Linear Optical Flow Templates

We learn the optical flow templates $\left(W_k, \Sigma_{uk}^{\text{v}}, \Sigma_{uk}^{\text{f}}, p\,(\lambda_k)\right)$ for each k^{th} environment type from videos collected during robot motion using the method of [6], an expectation-maximization algorithm. To compute sparse optical flow input to the learning method we use the pyramidal Lucas-Kanade tracker [26] 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.

4 Inferring the Environment Type

As described in the introduction, we infer which discrete path shape class best explains the image intensity changes and image gradients using approximate Bayesian model selections over the learned linear optical flow templates. Inference directly from image gradients and intensity changes 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).

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

$$\begin{aligned} p\left(k_t \,|\, I_{t,t-1}\right) &= \int_{y_{kt}} \sum_{\lambda_{kt}} p\left(k_t, y_{kt}, \lambda_{kt} \,|\, I_{t,t-1}\right) \\ &\propto \int_{y_{kt}} \sum_{\lambda_{kt}} p\left(I_t \,|\, y_{kt}, \lambda_{kt}, k_t, I_{t-1}\right) p\left(y_{kt}\right) p\left(\lambda_{kt}\right) p\left(k_t\right) \end{aligned} \tag{5}$$

where we assume $p\left(k_t\right)$ to be a categorical, i.e. constant-probability, prior over the environment types. In practice, though, this marginalization is intractable, so we introduce several approximations, as described in the following sub-sections.

4.1 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 [27]. The true sum over λ_{kt} in Eq. 5 is an intractable sum over all possible combinations of inlier assignments for all pixels. We first replace this sum with a lower-bound from EM by approximating Eq. 5 as

$$p\left(k_t \,|\, I_{t,t-1}\right) \approx \int_{y_{kt}} p\left(I_t \,|\, y_{kt}, \left\langle \lambda_{kt} \right\rangle, k_t, I_{t-1}\right) p\left(\left\langle \lambda_{kt} \right\rangle\right) p\left(y_{kt}\right) p\left(k_t\right), \tag{6}$$

where $\left\langle \lambda_{kti} \right\rangle \in [0, 1]$ is the expectation of the inlier indicator. We evaluate the terms involving this expectation using their expected log-likelihoods,

$$\begin{aligned} &p\left(I_t \,|\, y_{kt}, \left\langle \lambda_{kt} \right\rangle, k_t, I_{t-1}\right) = \\ &\exp \sum_i \left(\left\langle \lambda_{kti} \right\rangle \mathcal{L}\left(I_{ti} \,|\, y_{kt}, \lambda_{kti}{=}1, k_t, I_{t-1}\right) + \left\langle 1 - \lambda_{kti} \right\rangle \mathcal{L}\left(I_{ti} \,|\, y_{kt}, \lambda_{kti}{=}0, k_t, I_{t-1}\right)\right), \end{aligned} \tag{7}$$

where $\mathcal{L}(\cdot) \triangleq \log p(\cdot) + C$ is a log-likelihood. Using a similar scheme,

$$p\left(\langle \lambda_{kt} \rangle\right) = \exp \sum_i \left(\mathcal{L}\left(\lambda_k{=}1\right) \langle \lambda_{kti} \rangle + \mathcal{L}\left(\lambda_k{=}0\right) \langle 1 - \lambda_{kti} \rangle \right) \tag{8}$$

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

$$\begin{aligned} \langle \lambda_{kti} \rangle &\equiv p\left(\lambda_{kti}{=}1 \,|\, y_{kt}, k_t, I_{t,t-1,i}\right) \\ &= \frac{p\left(I_{ti} \,|\, y_{kt}, k_t, I_{t-1,i}\right) p\left(\lambda_{it}\right)|_{\lambda_{it}=1}}{\sum\limits_{\lambda_{it}=\{1,0\}} p\left(I_{ti} \,|\, y_{kt}, k_t, I_{t-1,i}\right) p\left(\lambda_{it}\right)}. \end{aligned} \tag{9}$$

4.2 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. 7, we marginalize out the unknown optical flow,

$$p\left(I_{ti} \,|\, y_{kt}, k_t, I_{t-1}\right) = \int_{u_{ti}} p\left(I_{ti} \,|\, u_{ti}, I_{t-1}\right) p\left(u_{ti} \,|\, y_{kt}, k_t\right). \tag{10}$$

An issue is that the image is nonlinear so Eq. 10 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,

$$p\left(I_{ti} \,|\, \delta u_{ti}, I_{t-1}\right) \approx \mathcal{N}(I_{t-1}(x_i - \mathring{u}_{ti}) - \nabla I_{ti} \delta u_{ti}, \sigma_{\mathcal{I}}), \tag{11}$$

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 $I(x)$ 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. 10 is then done in closed-form using this Gaussian-approximated image-likelihood in Eq. 11 and the expected log-likelihood approximation from Eq. 7,

$$p\left(I_t \,|\, y_t, k_t, I_{t-1}\right) \propto \exp \frac{-1}{2} \sum_i J_{ti}^2 \left(\overline{I_{ti}} - \nabla I_{ti} V_{ki} \delta y_{kt} \right)^2, \tag{12}$$

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

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

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. 12, 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 [28].

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. 12.

4.3 Computing the Environment Type Marginal

Finally, after approximating the marginal over the inlier indicator variables with the expected log-likelihoods in Eqs. 7 and 10, and approximating the image probability in Eq. 7 as a marginal Gaussian centered around the MLE of the latent variables y_{kt} (with the flow u_t marginalized out) in Eq. 12, we can write the environment type marginal as

$$p\left(k_t \,|\, I_{t,t-1}\right) \propto \left(\int_{y_{kt}} p\left(I_t \,|\, y_{kt}, k_t, I_{t-1}\right) p\left(y_{kt}\right) \right) p\left(\lambda_{kt}\right) p\left(k_t\right) \tag{13}$$

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

$$p\left(I_t, y_{kt} \,|\, k_t, I_{t-1}\right) \propto \exp \frac{-1}{2} \left(\sum_i J_{ti}^2 \left(\overline{I_{ti}} - \nabla I_{ti} V_{ki} \delta y_{kt} \right)^2 + \left\| \mathring{y}_{kt} + \delta y_{kt} \right\|^2 \right) \tag{14}$$

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. 14 leads to an efficient factorization of the marginal using the Schur complement. To do this, note that the log of Eq. 14 can be written as

$$\frac{-1}{2} \begin{bmatrix} \delta y_{kt} \\ \overline{I_t} \\ 1 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} A_{q \times q} & B_{q \times I}^{\mathsf{T}} & \mathring{y}_{kt} \\ B_{I \times q} & D_{I \times I} & \mathbf{0}_{I \times 1} \\ \mathring{y}_{kt}^{\mathsf{T}} & \mathbf{0}_{1 \times I} & \mathring{y}_{kt}^{\mathsf{T}} \mathring{y}_{kt} \end{bmatrix} \begin{bmatrix} \delta y_{kt} \\ \overline{I_t} \\ 1 \end{bmatrix} \tag{15}$$

where A, B, and D are

$$\begin{aligned} \underset{q \times q}{A} &\triangleq \mathbf{I}_{q \times q} + \textstyle\sum_i J_{ti}^2 V_{ki}^{\mathsf{T}} \nabla I_{ti}^{\mathsf{T}} \nabla I_{ti} V_{ki} \\ \underset{I \times q}{B} &\triangleq \begin{bmatrix} -J_{t1}^2 \nabla I_{t1} V_1 \\ -J_{t2}^2 \nabla I_{t2} V_2 \\ \vdots \end{bmatrix} \quad \underset{I \times I}{D} \triangleq \begin{bmatrix} J_{t1}^2 & & \\ & J_{t2}^2 & \\ & & \ddots \end{bmatrix} \end{aligned} \tag{16}$$

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

$$\Lambda_{\mathcal{I}} = D - BA^{-1}B^{\mathsf{T}}, \quad \eta_{\mathcal{I}} = -BA^{-1}\mathring{y}_{kt}, \quad f_{\mathcal{I}} = \mathring{y}_{kt}^{\mathsf{T}}\left(\mathbf{I} - A^{-1}\right)\mathring{y}_{kt} \tag{17}$$

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

$$|\Lambda_{\mathcal{I}}| = \det\left(A - B^{\mathsf{T}}D^{-1}B\right)\det A^{-1}\det D \tag{18}$$

Combining Eqs. 17 and 18, the marginal image likelihood is

$$p\left(I_t \,|\, k_t, I_{t-1}\right) = (2\pi)^{\frac{-I}{2}}\,|\Lambda_{\mathcal{I}}|^{\frac{1}{2}}\exp\frac{-1}{2}\left(\overline{I_t^{\mathsf{T}}}\Lambda_{\mathcal{I}}\overline{I_t} + \overline{I_t^{\mathsf{T}}}\eta_{\mathcal{I}} + f_{\mathcal{I}}\right) \tag{19}$$

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. 13 is computationally efficient. This is because in Eq. 19 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. 17 and 18 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. 13 is to normalize it by dividing by the sum of the evaluated likelihoods of Eq. 13 for each k_t.

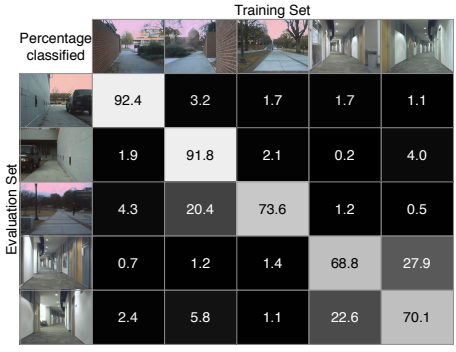
5 Experimental Results

In this section we provide experimental evidence to support the claims that *a)* our method efficiently classifies between multiple coarse path shapes and *b)* that to accomplish this it is important to use information from image motion instead of image appearance, to capture differences in structure. To support these claims we perform a quantitative evaluation and comparison with a classifier using Gist features. Please see the end of this section for implementation details and parameters.

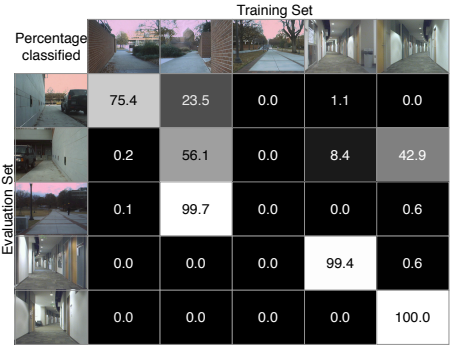
All datasets contain vibration, yawing, and side-to-side motions of the platform, which violate the *ideal* linearity described in Section 3, but fit the *approximate* linearity and are thus handled by our method.

5.1 Quantitative Evaluation

We learned linear optical flow templates for the environments exemplified by the top row of thumbnails of Figure 2, and performed inference on the environments exemplified by the left column of thumbnails. For the first three environments, the coarse environment shapes are similar between the training and



(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%)

Fig. 2: 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.

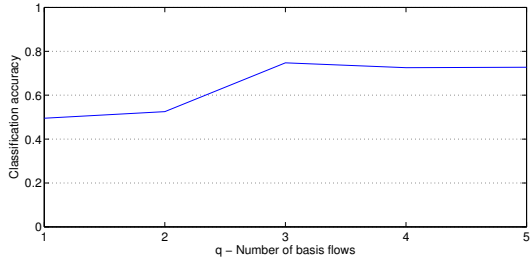


Fig. 3: Classification accuracy for linear optical flow templates learned with various numbers of basis flows, i.e. latent variable dimensionalities q, learned and evaluated with the same datasets as in Figure 2.

testing datasets, but there are differences such as the height of the wall and distance to it varying. Additionally, the texture and image appearance is quite different. The last 2 environment types are left and right curving hallways (which we consider two different discrete types) with the same curvature. Texture and appearance differ less between training and testing sets in the curved hallways, they are from different building floors. The distance from the robot to the wall varies with side-to-side motions and yawing motions in all videos, and there is also camera vibration. We empirically selected to learn templates with $q = 3$ basis flows as this provided the highest accuracy.

Figure 2 shows the confusion matrices for our method and a neural network classifier using Gist features. The Gist features were computed by the software[1] accompanying [18]. We selected the subset of Gist features suggested by [18], and trained a neural network with 200 and 100 node hidden layers for 500 epochs (verifying that error on a holdout set did not increase during training) using Weka [29].

Figure 3 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 higher accuracy on 'left wall', 'right wall', and 'walkway' highlights 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 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.

5.2 Implementation Details and Parameters

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, $p\left(\lambda_{kti}{=}1\right) = 0.95$. 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 4.2 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 the Gauss-Newton method.

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

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

and basis flows scaled down from the original 640×480. Additionally, for optimization and inference (i.e. in all ranges over i in Section 4), 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 4 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 [30].

6 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. Frazzoli, E., Dahleh, M., Feron, E.: A hybrid control architecture for aggressive maneuvering of autonomous helicopters. In: Proceedings of the IEEE Conference on Decision and Control. Volume 3. (1999) 2471–2476
2. Schouwenaars, T., Mettler, B., Feron, E., How, J.: Robust motion planning using a maneuver automation with built-in uncertainties. In: Proceedings of the American Control Conference. Volume 3. (2003) 2211–2216
3. Hoiem, D., Efros, A.A., Hebert, M.: Geometric context from a single image. In: International Conference of Computer Vision (ICCV), IEEE (2005)
4. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: Proceedings of the European Conference on Computer Vision. (2008)
5. Sturgess, P., Alahari, K., Ladicky, L., Torr, P.: Combining appearance and structure from motion features for road scene understanding. In: Proceedings of the British Machine Vision Conference. (2009)
6. Roberts, R., Potthast, C., Dellaert, F.: Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2009)

7. Lalonde, J., Vandapel, N., Huber, D., Hebert, M.: Natural terrain classification using three-dimensional ladar data for ground robot mobility. Journal of Field Robotics **23** (2006) 839–861
8. Huertas, A., Matthies, L., Rankin, A.: Stereo-based tree traversability analysis for autonomous off-road navigation. In: IEEE Workshops on Application of Computer Vision, WACV/MOTIONS'05. Volume 1. (2005) 210–217
9. Michels, J., Saxena, A., Ng, A.Y.: High speed obstacle avoidance using monocular vision and reinforcement learning. In: Intl. Conf. on Machine Learning (ICML). (2005)
10. Khan, Y., Komma, P., Zell, A.: High resolution visual terrain classification for outdoor robots. In: IEEE ICCV Workshop on Challenges and Opportunities in Robot Perception. (2011)
11. Becker, F., Lenzen, F., Kappes, J.H., Schnörr, C.: Variational recursive joint estimation of dense scene structure and camera motion from monocular high speed traffic sequences. In: International Conference on Computer Vision. (2011)
12. Geiger, A., Lauer, M., Urtasun, R.: A generative model for 3d urban scene understanding from movable platforms. In: Computer Vision and Pattern Recognition (CVPR), Colorado Springs, USA (2011)
13. Flint, A., Murray, D., Reid, I.: Manhattan scene understanding using monocular, stereo, and 3d features. In: Proceedings of the International Conference on Computer Vision. (2011)
14. Tsai, G., Xu, C., Liu, J., Kuipers, B.: Real-time indoor scene understanding using bayesian filtering with motion cues. In: Proceedings of the International Conference on Computer Vision. (2011)
15. Nourani-Vatani, N., Borges, P.V.K., Roberts, J.M., Srinivasan, M.V.: Topological localization using optical flow descriptors. In: Proceedings of the 1st IEEE Workshop on Challenges and Opportunities in Robotic Perception, with ICCV'2011. (2011)
16. Mozos, O.M., Burgard, W.: Supervised learning of topological maps using semantic information extracted from range data. In: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS). (2006) 2772–2777
17. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision **42** (2001) 145–175
18. Siagian, C., Itti, L.: Rapid biologically-inspired scene classification using features shared with visual attention. IEEE Transactions on Pattern Analysis and Machine Intelligence **29** (2007) 300–312
19. Swadzba, A., Wachsmuth, S.: Indoor scene classification using combined 3D and Gist features. In: Asian Conference on Computer Vision, Springer (2010) 201–215
20. Ackerman, C., Itti, L.: Robot steering with spectral image information. IEEE Transactions on Robotics **21** (2005) 247–251
21. Pugeault, N., Bowden, R.: 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)
22. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. CVPR (2005) 524–531
23. L. Lazebnik, C.S., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2006)

24. Pandey, M., Lazebnik, S.: Scene recognition and weakly supervised object localization with deformable part-based models. In: Proceedings of the International Conference on Computer Vision. (2011)
25. Heeger, D., Jepson, A.: Subspace methods for recovering rigid motion I: Algorithm and implementation. International Journal of Computer Vision **7** (1992) 95–117
26. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence. Volume 3. (1981)
27. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society **39** (1977) 1–38
28. Dellaert, F., Thrun, S., Thorpe, C.: Jacobian images of super-resolved texture maps for model-based motion estimation and tracking. In: IEEE Workshop on Applications of Computer Vision (WACV). (1998)
29. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. SIGKDD Explorations **11** (2009)
30. Hutchings, B., Nelson, B., West, S., Curtis, R.: Optical flow on the ambric massively parallel processor array (MPPA). In: IEEE Symposium on Field Programmable Custom Computing Machines (FCCM). (2009) 141–148