Learning General Optical Flow Subspaces for Egomotion Estimation and Detection of Motion Anomalies

Richard Roberts
Georgia Institute of Technology

Christian Potthast
Georgia Institute of Technology

Frank Dellaert
Georgia Institute of Technology

Note: This is a draft; camera-ready version to be available April 1, 2009.

Abstract

This paper deals with full-frame estimation of optical flow in a generalized imaging system by exploiting probabilistic subspace constraints on the flow. We deal with the extended motion of the imaging system through an environment that we assume to have some degree of statistical regularity. For example, in autonomous ground vehicles the structure of the environment around the vehicle is far from arbitrary. We exploit this regularity to predict the perceived optical flow due to platform motion. The subspace constraints hold not only for perspective cameras, but in fact for a very general class of imaging systems, including catadioptic and multiple-view systems. Using minimal assumptions about the imaging geometry, we derive a probabilistic subspace constraint that captures the statistical regularity of the scene geometry relative to an imaging system. We propose an extension to probabilistic PCA (Tipping and Bishop, 1999) as a way to learn this subspace from a large amount of recorded imagery, and demonstrate its use in conjunction with a sparse optical flow algorithm. To deal with the sparseness of the input flow, we use a generative model to estimate the full-dimensionality subspace using only the observed flow measurements. Additionally, to identify and cope with image regions that violate the subspace constraints, such as moving objects or gross flow estimation errors, we employ a per-pixel Gaussian mixture outlier process. We demonstrate results of finding the optical flow subspaces and employing them to estimate full-frame flow and to recover camera motion, for a variety of imaging systems in several different environments.

1. Introduction

This paper deals with estimation of full-frame optical flow from sparse flow measurements in a generalized imaging system, in the case where the imaging system is moving through an environment with some degree of statistical regularity. We automatically and robustly find the linear subspace of the flow, and use the subspace constraints at runtime to find outlying optical flow vectors in new frames and predict unobserved flow. We then use the projection of sparse flow onto the subspace to recover the motion of the imaging system through the environment.

We are interested in approximate monocular visual odometry as a replacement for more costly or limited methods of obtaining incremental platform motion. Laser scanners, or LIDAR, with which one can perform laser scan matching, tend to be heavy, expensive, and require much power. Wheel odometry, while very accuracte indoors, is unreliable due to wheel slippage in outdoor environments. All but the most costly Intertial Measurement Units (IMU) quickly accumulate velocity errors, and therefore must be fused with absolute position or velocity measurements.

Geometric stereo visual odometry, as described by Matthies [15], relies on finding image features that match between frames, and then finding the camera motion that best explains the movements of the features between frames and cameras in a stereo head. Most current systems employ fast transform solving in a RANSAC harness, followed by optimization of pose using in-lying features, for example in [17, 16, 1]. Nistér *et al.* report errors of about 12 m over a 360 m course [17]. Others have used a ground plane assumption to compute monocular visual odometry, for example [19, 4].

These geometric methods, while extremely accurate, assume perspective cameras for which lens distortion can be modelled. Additionally, they are computationally demanding, making them challenging to implement on low-power systems.

In contrast, we recover egomotion directly from sparse optical flow using subspace constraints of the flow that hold for a very general class of imaging systems, while simultaneously identifying outliers in the sparse flow. The subspace constraints hold under the assumption of regularity of scene depth, as is typically the case for robots operating outdoors and in urban streets.

Optical flow estimation in general suffers from the *aperature problem*. Because each neighborhood of pixels can potentially have a different motion in the image, optical flow is typically computed on small windows (e.g. [14]) thoughout the image, but without global constraints cannot be evaulated in regions with ambiguous texture. Global smoothness constraints (e.g. [9]) are designed to propagate flow to these ambiguous regions, but such constraints make strong assumptions about the scene and the imaging system.

More recent work in computing optical flow has applied global constraints that arise from the optics of the imaging system and the structure of the scene. These usually assume a perspective camera, and model motion in the image as an affine or perspective transformation, for example [2, 11]. Some of these methods can apply differing motion to separate regions of the image, for example [12, 3]. Irani exploits linear subspace constraints that hold over several frames [10]. Others use PCA to find the linear subspace automatically, e.g. [6], but still require optical flow to be computed over entire frames.

2. Generalized imaging system and subspace

In this section we describe the class of imaging systems with which we are concerned, and show that the optical flow therein lies near a linear subspace, constant for all incremental platform motion. We also show how to recover the platform motion from the optical flow.

2.1. Collection of local cameras

We deal with a generalized imaging system where each pixel neighborhood can be approximated by local projective camera, which measures both intensity *and* an optical flow vector for that single pixel. Grossberg and Nayar [7] derive a generalized imaging model based on "ray pixels", or *raxels*. They suggest a ray surface that ensures the function mapping rays to pixels is smooth. In order to compute local optical flow, we actually only require the mapping from pixels to rays to be piece-wise smooth, as discontinuities are discarded as outliers.

2.2. Linearity of the subspace

In this generalized imaging model, the mapping from platform motion to optical flow is linear when scene depth for each local camera is constant. Irani shows that for incremental motion over several frames, the optical flow lies in a linear subspace for arbitrary structure and camera motion [10]. This subspace approximation holds for a few frames and depends on the scene structure. Our goal is slightly different, in that we aim to recover a subspace for optical flow that can be assumed *constant over the entire camera trajectory.* The main idea is to exploit the regularity of scene depth with respect to the camera, which actually holds in many applications.

We now show that for every pixel the local optical flow t_j is linear in the incremental platform motion δ, i.e.,

$$t_j(\delta, Z) \approx H\delta$$

where Z is the scene depth and H is a 2×6 matrix defining the linear subspace in which the flow t lives. To do this, consider the flow induced by the incremental motion δ_j of a *local* camera. Using the incremental rotation approximation, we have [10, 13]

$$t_j(\delta, Z) \triangleq \begin{bmatrix} \dot{u}_j \\ \dot{v}_j \end{bmatrix} \approx f\frac{1}{Z}\begin{bmatrix} \tau_j^x \\ \tau_j^y \end{bmatrix} + f\begin{bmatrix} \phi_j^y \\ -\phi_j^x \end{bmatrix} \qquad (1)$$

where f is the focal length, and τ_j^x, τ_j^y, ϑ_j^x, and ϑ_j^y are the components of the incremental translation and rotation of the local camera.

The key point is that, while Eq. 1 depends on scene depth, we can write it as the multiplication of a *typical flow* associated with the *expected* inverse depth $\mu_j = E\{1/Z\}_{P(Z_j)}$, and a multiplicative component $(1 + n_j)$ with expected value 1.0 due to deviations from the typical scene depth:

$$\begin{bmatrix} \dot{u}_j \\ \dot{v}_j \end{bmatrix} \approx \begin{bmatrix} f\mu_j & 0 & 0 & 0 & f & 0 \\ 0 & f\mu_j & 0 & -f & 0 & 0 \end{bmatrix} \begin{bmatrix} \tau_j \\ \phi_j \end{bmatrix} (1+n_j)$$

The incremental motion δ_j of the local camera is in turn a function of the incremental motion of the base frame, which can be approximated by a 6×6 linear mapping F_j:

$$\delta_j \triangleq \begin{bmatrix} \tau_j \\ \phi_j \end{bmatrix} = f_j(\delta) \approx F_j\delta$$

Hence, concatenating

$$\begin{aligned} t_j(\delta, Z) &\triangleq \begin{bmatrix} \dot{u}_j \\ \dot{v}_j \end{bmatrix} \\ &\approx \begin{bmatrix} f\mu_j & 0 & 0 & 0 & f & 0 \\ 0 & f\mu_j & 0 & -f & 0 & 0 \end{bmatrix} F_j\delta = H_j\delta \end{aligned}$$

The optical flow in a generalized imaging system is thus in a linear subspace, with dimensionality equal to the degrees-of-freedom of the platform motion. This linear subspace is constant for constant depth and optics.

Although it may seem at first to be quite a limiting assumption, we in fact can often approximate inverse depth as a constant. This assumption works well when there is considerable regularity of depth relative to the robot in the environment. First, the relative depth to the ground plane is always a constant for planar motion. Additionally, locations above the ground plane are also approximately at constant depth in hallways and urban canyons. Because our method detects and ignores outliers, points that violate the regularity do not significantly affect results.

2.3. Recovering egomotion

Because the mapping from incremental motion to optical flow is linear, we can easily learn this mapping to recover egomotion from the optical flow.

Given the subspace coordinates $x_i \in \mathbb{R}^q$ of each i^{th} frame of optical flow, i.e. the projection of the optical flow onto the q-dimensional flow subspace, there is an unknown linear mapping from the subspace coordinates to the platform motion. To learn this mapping, we find the matrix $\mathbf{M}_x \in \mathbb{R}^{q \times q}$ and the vector $M_c \in \mathbb{R}^q$ that satisfy

$$\mathbf{M}_x x_i + M_c = \delta_i, \tag{2}$$

for every frame i, where $\delta_i \in \mathbb{R}^q$ is the known incremental motion of the platform in the i^{th} frame. To find $\mathbf{M}_x$ and M_c, we rearrange this equation into q linear systems, each of the form $\mathbf{A}y = b$:

$$\begin{bmatrix} x_1^{\mathrm{T}} & 1 \\ & \vdots \\ x_m^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{M}_x^k & M_c^k \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} \delta_1^k \\ \vdots \\ \delta_m^k \end{bmatrix},$$

where $\mathbf{M}_x^k$ is the k^{th} row of $\mathbf{M}_x$, M_c^k is the k^{th} element of M_c, and δ_i^k is the k^{th} element of the known platform motion for frame i. Thus each linear system specifies the mapping from the subspace coordinates to one of the platform degrees-of-freedom. For each frame, we find the subspace coordinates x_i from the sparse flow by iterating (8) and (9), which corresponds to the E-step of the EM algorithm we develop in Section 3.4.

We found it necessary to solve these systems using a robust method, such as iteratively-reweighted least squares, as there were frames in our data sequences where either the optical flow or the ground truth were completely incorrect.

After training by learning $\mathbf{M}_x$ and M_c, we can estimate the egomotion for new frames of sparse flow simply by finding the subspace coordinates, again by iterating (8) and (9), and then evaluating (2) to obtain the egomotion δ. The probability of each flow vector being an inlier, from (9) is $\langle z_i \rangle$. Estimating egomotion is very effecient, as normally only a few iterations are required for the estimate of x_i to converge.

3. Finding the flow subspace

We present an extension of probabilistic PCA [18] to robustly find the principal subspace. The robustness of this principal subspace method is not limited to the traditional case where entire samples are discarded as outliers. Instead, we can find dimensions (i.e. per-pixel flow vectors) of each sample that are not consistent with the values of the other dimensions.

De la Torre and Black [5] also developed an "intra-sample" robust PCA method, in which they minimize an energy function that weights each pixel with a per-image-pixel analog outlier process. We instead present a generative model for intra-sample robust PCA, which models each pixel as a Gaussian mixture model of either the expected value on the subspace, or zero-mean noise.

The idea of exploiting patterns in optical flow using PCA has arisen in other contexts. Fleet *et al.*, for example used PCA to learn a parametrized model of optical flow on images containing deforming bodies, including human motion. They used this model to predict optical flow and also to classify movements in the subspace [6].

3.1. Training data

Given a training video sequence, we compute a downsampled optical flow field over each pair of frames. To do so we divide the images into a grid of 20×20 pixel cells, then track the strongest Harris corner [8] in each cell using the Lucas-Kanade algorithm [14]. We employ a weak threshold on corner response to prevent tracking textureless regions.

Each pair of frames yields an observation vector t_i, filled with the concatenated horizontal and vertical optical flow components. For flow fields of size $w \times h$, the length of each observation vector is $d = 2wh$.

Because the flow field is in fact sparse, there is also a set S_i of the "seen" indices of t_i, where optical flow is available. Indices not in S_i are missing, and we save computation by ignoring them in calculations, as described below.

3.2. Generative model

In a generative model for the optic flow, each vector is either an inlier correlated to the other optic flow vectors in the frame, or an outlier of zero-mean Gaussian noise:

$$t_{ij} = \begin{cases} W_j x_i + \mu_j + \epsilon_v & , z_{ij}^v \\ \epsilon_f & , z_{ij}^f \end{cases}, \tag{3}$$
$$\text{where } \epsilon_v \sim \mathcal{N}(0, \sigma_v) \text{ and } \epsilon_f \sim \mathcal{N}(0, \sigma_f)$$

where t_{ij} is the j^{th} optic flow component from the i^{th} frame. z_{ij}^v and z_{ij}^f are mutually exclusive binary variables indicating whether the optic flow component is an inlier or outlier (v for valid and f for false). ϵ_v and ϵ_f are zero-mean Gaussian random variables modelling the expected noise for inliers, which should be relatively small, and the expected noise for outliers, which should be relatively large.

We model inliers as arising from linear combinations of the basis flows, W, with the latent coefficients, x. As with PCA, we also center the data before finding basis flows, so we include the mean, μ. As with PPCA, we estimate the variance σ_v^2 of the inlier noise, which will measure how well the model explains the training data.

The key observations we make about this model are:

1. The model is similar to PCA: if all vectors were inliers, the the basis flows would span the principal subspace, as shown in [18].
2. When conditioned on the basis flows and latent coefficients, each observation vector is drawn from a mixture model of two *spherical* normal distributions.
3. Although the flow vectors of each frame are correlated through the basis flows, all flow components are mutually independent when conditioned on the basis flows and latent variables.

3.3. Maximum likelihood formulation

As we observed above, when given W, μ, and x, each t_{ij} is conditionally independent. This assumption allows us to solve for the basis flows and latent variables even in the presence of missing data and outliers. The conditional distribution over a single component of a single flow vector is the mixture model

$$t_{ij} | x_i z_{ij} \theta \sim \mathcal{N}\left(t_{ij};\, \bar{t}_{ij},\, \sigma_v^2\right)^{z_{ij}^v} \mathcal{N}\left(t_{ij};\, 0,\, \sigma_f^2\right)^{z_{ij}^f}, \quad (4)$$

where $\bar{t}_{ij} = W_j x_i + \mu_j$, $\theta = \left\{W, \mu, \sigma_v^2, \sigma_f^2, \pi_v, \pi_f\right\}$, and π_v and $\pi_f = 1 - \pi_v$ are the mixing coefficients of the mixture models. We note that there is in fact a different mixture model for every flow vector and component, but that they are related through the parameters θ.

As with PPCA, there is a standard Gaussian prior on x:

$$x \sim \mathcal{N}\left(\mathbf{0},\, \mathbf{I}_{q\times q}\right). \quad (5)$$

As with a traditional Gaussian mixture model, $p\left(z_{ij}^v\right) = \pi_v$ and $p\left(z_{ij}^f\right) = \pi_f$.

We are now ready to formulate the maximum likelihood problem. We wish to obtain a ML solution over the joint distribution of the entire model,

$$p\left(\theta x z | t\right) \propto p\left(\theta\right) \prod_i p\left(x_i\right) \prod_{j \in S_i} L\left(t_{ij} | x_i z_{ij} \theta\right) p\left(z_{ij}\right). \quad (6)$$

Note the product over $j \in S_i$, by which we only consider vectors in the training data that were "seen", and ignore those that were missing. We can ignore these missing values because of the conditional independence between flow vectors, and doing so significantly saves computation.

3.4. EM algorithm

The complete log-likelihood of (6), using the distributions above is

$$\begin{aligned}\mathcal{L}\left(\theta x z | t\right) = \sum_i \log \mathcal{N}\left(x_i;\, 0,\, 1\right) + \sum_{j \in S_i} \\ z_{ij}^v \left(\log \pi_v + \log \mathcal{N}\left(t_{ij};\, \bar{t}_{ij},\, \sigma_v^2\right)\right) \\ + z_{ij}^f \left(\log \pi_f + \log \mathcal{N}\left(t_{ij};\, 0,\, \sigma_f^2\right)\right). \quad (7)\end{aligned}$$

Finding the expectation of this log-likelihood with respect to both x and z simultaneously would be intractable, but we can instead first find the expectation with respect to x, and then with respect to z, comprising a generalized EM algorithm. We thus obtain update equations by taking the derivatives of (7) w.r.t. the parameters, and solving for each parameter:

$$\mu_j = \left(\sum_{i \in S_j} \left\langle z_{ij}^v \right\rangle\right)^{-1} \sum_{i \in S_j} \left\langle z_{ij}^v \right\rangle \left(t_{ij} - W_j x_i\right)$$

$$\sigma_v^2 = \left(\sum_{i,\, j \in S_i} \left\langle z_{ij}^v \right\rangle\right)^{-1},$$

$$\sum_{i,\, j \in S_i} \left\langle z_{ij}^v \right\rangle \left(y_{ij}^2 - 2 y_{ij} W_j \left\langle x_i \right\rangle + \operatorname{tr}\left(\left\langle x_i x_i^{\mathrm{T}} \right\rangle W_j^{\mathrm{T}} W_j\right)\right)$$

$$W_j = \sum_{i \in S_j} \left\langle z_{ij}^v \right\rangle \sigma_v^{-2} \left\langle x_i \right\rangle^{\mathrm{T}} y_{ij} \left(\sum_{i \in S_j} \left\langle z_{ij}^v \right\rangle \sigma_v^{-2} \left\langle x_i x_i^{\mathrm{T}} \right\rangle\right)^{-1}$$

where $y_{ij} = t_{ij} - \mu_j$. S_i is as above, while S_j is the set of *frame* indices whose j^{th} flow component was not missing.

For the purposes of our experiments, we constrain $\sigma_f^2 = 10$, but this outlier variance could easily be estimated online as well.

We find the expectation $\langle x_i \rangle$ with respect to its posterior $p\left(x | t z \theta\right) \propto p\left(t | x z \theta\right) p\left(x\right)$ with Bayes' law and the distributions (4) and (5). Combining the distributions over each t_{ij} into a multivariate distribution over t_i with diagonal covariance Λ_i^{-1} simplifies finding the distribution over x_i. Following, $p\left(x | t z \theta\right)$ has mean and covariance

$$\begin{gathered}\left\langle x_i \right\rangle = \left(W_{Si}^{\mathrm{T}} \Lambda_i W_{Si} + \mathbf{I}_{q\times q}\right)^{-1} W_{Si}^{\mathrm{T}} \Lambda_i y_i, \quad (8) \\ \left\langle x_i x_i^{\mathrm{T}} \right\rangle = \left(W_{S_i}^{\mathrm{T}} \Lambda_i W_{S_i} + \mathbf{I}_{q\times q}\right)^{-1} + \left\langle x_i \right\rangle \left\langle x_i \right\rangle^{\mathrm{T}}, \\ \text{where } \Lambda_i = \sigma_v^{-2} \mathbf{I}_{q\times q} z_i.\end{gathered}$$

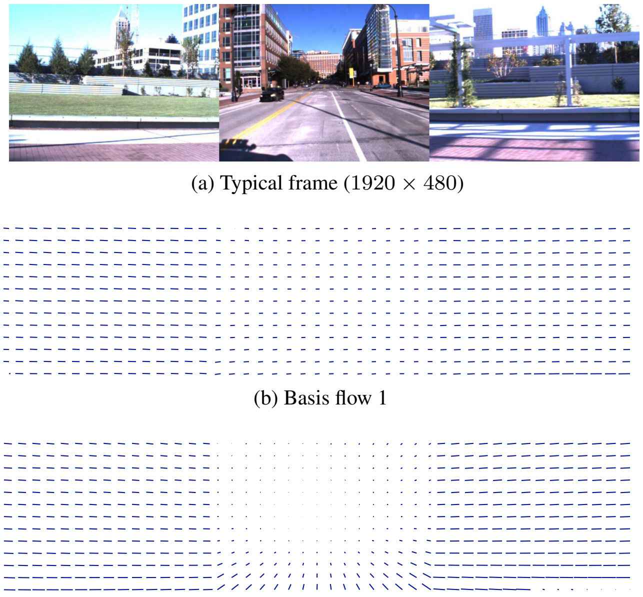
The expectation $\left\langle z_{ij}^v \right\rangle$ is easier, as it is the simply the proportion of probability of the flow component belonging to the inlier distribution:

$$\left\langle z_{ij}^v \right\rangle = \frac{\mathcal{N}\left(t_{ij};\, \hat{t}_{ij},\, \sigma_v^2\right)}{\mathcal{N}\left(t_{ij};\, \hat{t}_{ij},\, \sigma_v^2\right) + \mathcal{N}\left(t_{ij};\, 0,\, \sigma_f^2\right)}, \quad (9)$$

where $\hat{t}_{ij} = W\left(W^{\mathrm{T}} W\right)^{-1} \left(W^{\mathrm{T}} W + \sigma_v^2 I_q\right) \langle x_i \rangle$.

4. Experiments

In this section, we present experiments that illustrate the linear flow subspace for several imaging systems, show the robustity of our method to outliers, illustrate estimation of full-frame flow, and recover platform motion.

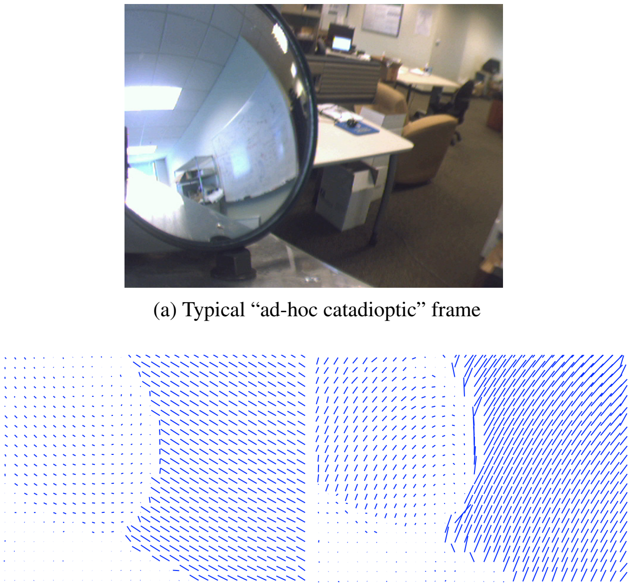


(a) Typical frame (1920 × 480)

(b) Basis flow 1

(c) Basis flow 2

Figure 1: Basis flows spanning flow subspace for the 3-camera outdoor driving sequence.



(a) Typical "ad-hoc catadioptic" frame

(b) Basis flow 1

(c) Basis flow 2

Figure 2: Typical frame (640 × 480) and basis flows for ad-hoc catadioptic system. The basis flows clearly show the differing optical properties of the imaging system. Optical flow is useful both in the mirror and in the view past the mirror. In the lower-left part of the image, however, the camera sees only the plate on which it is mounted, which moves with the camera and thus has no optical flow.

4.1. The flow subspace

Figure 1 shows the basis flows spanning the linear optical flow subspace for the 3-camera outdoor driving sequence. The vehicle had 2 degrees-of-freedom, arising from its steering rate and speed, and thus we constrained search to a 2-dimensional flow subspace. While the vehicle did pitch and roll slightly due to suspension movement, but this motion is small and erratic, so we do not wish to model it. We observe that the first basis flow approximately corresponds to the vehicle rotating, while the second basis flow approximately corresponds to driving forwards. Also, the basis flows capture the depth regularity of the sequence, as in the second basis flow, the flow vectors lower in the image, which correspond to points on the ground plane closer to the camera, displace more than the points higher in the image, which correspond to more distant points. On the other hand, in the first basis flow, corresponding to rotation, the flow vectors have approximately equal magnitude in each camera.

Figure 2 shows basis flows for a catadioptic system in which the camera partially looks into a mirror, and partially looks past the mirror. Additionally, the lower-left corner of the frame is occupied by the plate to which the camera and mirror are rigidly attached, so this corner contains no optical flow. The basis flows found by our method clearly reflect these three image regions with differing optics.

Figure 3 illustrates the importance of robustness to outliers when finding the flow subspace. We found basis flows in a training sequence in which people walked in front of the camera, creating out-lying optical flow. This sequence was recorded with a standard projective camera in pure rotation, so the flow vectors should be paralell, but the people walking in the scene negatively impact the results of standard non-robust PPCA. Our method, which is robust to individual outlying optical flow vectors within frames of optical flow, correctly ignores the inconsistent motion and yields parallel optical flow vectors in the basis flows.

4.2. Full-frame flow

The multi-camera outdoor driving data set consists of tiled images, and flow fields between frames are quite sparse and contain errors, as shown in Figure 4a. Our method estimates the latent variables for a pair of frames using the observed sparse optical flow vectors, while identifying out-lying flow vectors, and then reconstructs full-frame optical flow, shown in Figure 4b, using the estimated latent variables and the basis flows. Figure 5 shows similar results for the "ad-hoc catadioptic system".

In frames containing moving objects, gross sparse flow estimation errors, or structure that is unusually close to or unusually far from the camera, robustness of our method to outliers is very important. Figure 6 shows the inliers and

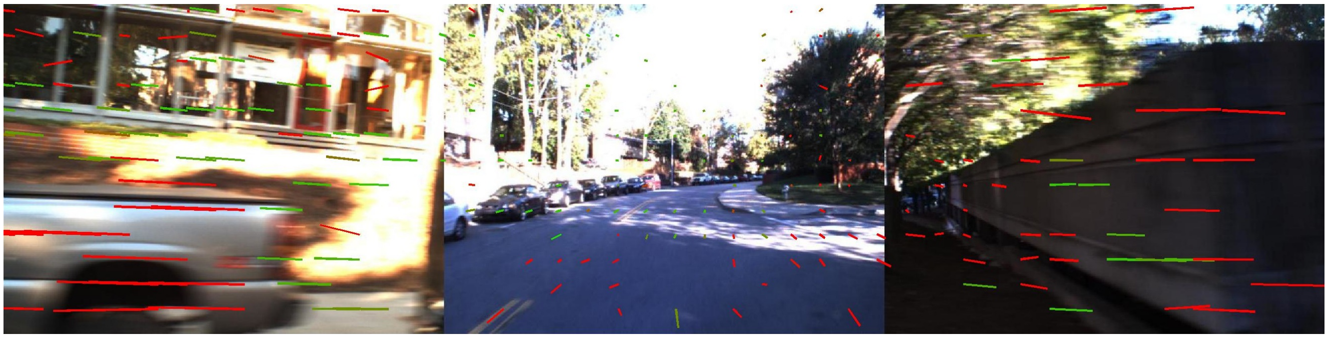
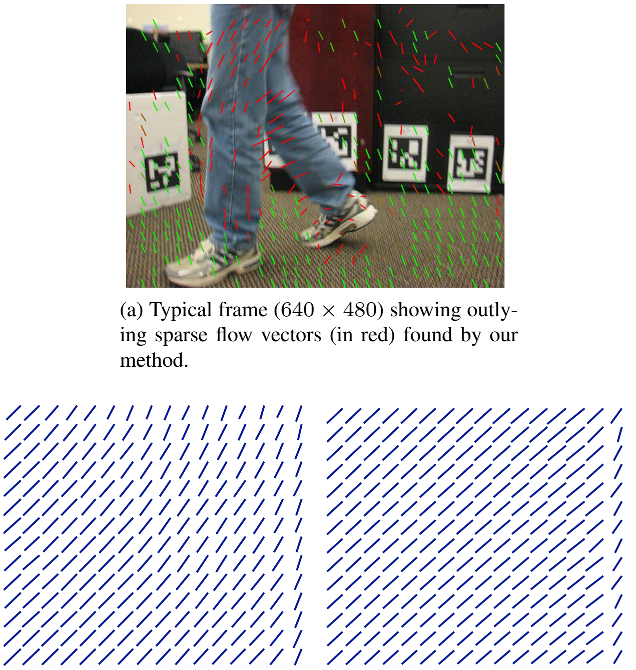


Figure 6: Detection of outliers in the sparse optical flow in a frame from the 3-camera outdoor driving sequence. Colored lines are sparse optical flow, ranging from green when $p\,(\text{inlier}) = 1$, to red when $p\,(\text{inlier}) = 0$. Sparse flow vectors that are inconsistent with the linear flow subspace have low inlier probability. Vectors on the moving pickup truck, in the textureless regions of the road, and on the very close and very far structure of the wall are labelled as outliers.

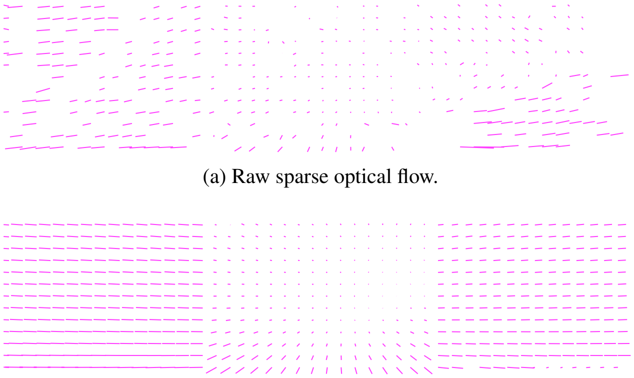


(a) Typical frame (640 × 480) showing outlying sparse flow vectors (in red) found by our method.

(b) One of the two basis flows found by standard PPCA.

(c) The same basis flow found by our method.

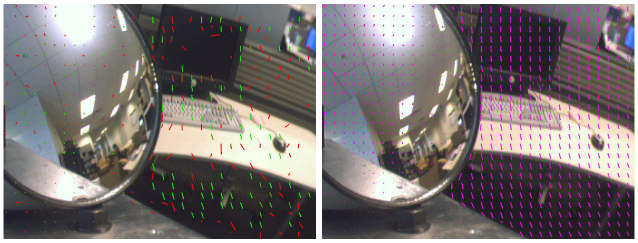
Figure 3: Comparison of basis flows found by standard PPCA and by our method, a) for a yawing and pitching camera, with outlying flow vectors caused by people walking in front of the camera while it was moving. b) In the basis flow found by standard PPCA, the vectors are not parallel, as they should be for this dataset. c) In the same basis flow found by our robust method, the vectors are properly much closer to parallel. This figure only shows the second of two basis sets for this sequence.



(a) Raw sparse optical flow.

(b) Full-frame estimated optical flow.

Figure 4: Typical raw sparse optical flow field from the multi-camera outdoor driving dataset, and the corresponding full-frame flow estimated by our method.



(a) Raw sparse optical flow

(b) Estimated full-frame flow

Figure 5: Typical raw sparse optical flow field from the adhoc catadioptic system dataset, and the corresponding full-frame flow estimated by our method.

outliers that are present in a typical frame. Green vectors are inliers, while red vectors are outliers. In the portion of the image from the left-facing camera, the vectors on the moving pickup truck are labelled as outliers. In the forward-facing camera, the vectors from poor feature tracks on the

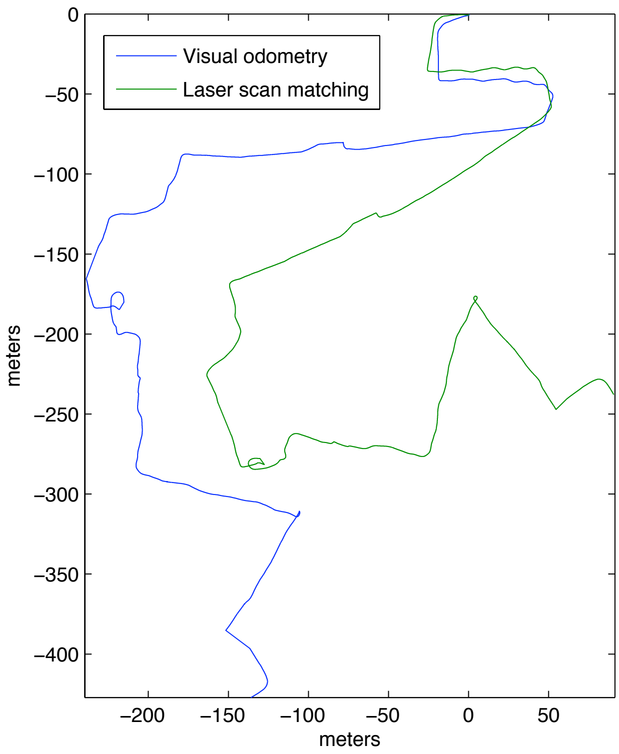


Figure 7: The calculated trajectory by this monocular visual odometry method, and by laser scan matching. In the upper left corner the trajectory should be parallel before and after the u-turn, as visual odometry correctly estimates. Neither method successfully closes the loop.

textureless portion of the road are mostly ignored. In the right-facing camera, the structure of the wall that is very close and very far from the camera is ignored, while the vectors on the structure near the average distance are labeled as inliers. Note that our method actually computes the probability of a flow vector being an inlier, as indicated on the figure by colors between green and red.

4.3. Ego-motion estimation

We obtained platform motion using laser scan matching for a single-camera outdoor sequence. We used the motion found by laser scan matching as training data to learn the linear transformation from subspace coordinates to motion, as described in Section 2.3. Figure 7 shows the trajectory of the platform as obtained from laser scan matching and from optical flow. In the upper left corner the trajectory should be parallel before and after the u-turn, as visual odometry correctly estimates, although neither method successfully closes the loop. We explain why this discrepancy likely occurs in Section 5.

Task	Computation time
Compute sparse flow	19.1 ms
Estimate full-frame flow	6.8 ms
Estimate ego-motion	0.2 ms
Allowable for real-time (30 fps)	33.0 ms

Table 1: Execution time of our prototype code, per frame, on 1920×480 frames, and 45×13 flow fields.

5. Discussion

We chose to train the supervised learning of egomotion estimation using laser scan matching, a common method for obtaining platform motion in the abscence of stereo visual odometry. We were surprised to see that ego-motion estimated from optical flow actually appeared to be more accurate than the “ground truth” that supervised the learning of ego-motion estimation, as described in the U-turn in the upper-right corner of Figure 7. We believe that the fact we enforce a *linear* subspace explains this result. We can see, that laser scan matching sometimes over-estimated turns, and sometimes under-estimated them. This is most likely due to pitching and rolling of the platform, which would change the viewpoint of the laser, and to moving objects.

During training, the method of supervised learning in Section 2.3, is constrained to a linear model, and thus would average the over- and under- estimation of turns by scan matching, leaving only the bias with respect to turning one direction or the other, which the trajectory results show is close to zero.

At runtime, visual odometry is robust to small pitching and rolling when estimating the latent variables, and is fairly robust to moving objects. We used the iterative closest point (ICP) algorithm to laser scan matching, which is sometimes robust to fast-moving objects, but cannot be robust against changes in viewpoint caused by pitching and rolling.

An on-line implementation of this method constitutes future work. Estimation of optical flow at runtime is effecient, requiring only a few iterations of projection onto the basis flows, and estimating the probability of each vector belonging to one of two Gaussian distributions.

An informal evaulation of execution time, in Table 1, reveals that our prototype code, with sparse flow extraction in C++ and full-frame flow and ego-motion estimation in MATLAB, is faster than real-time.

We showed earlier that we can label moving objects in the scene, as they constitute outliers. In fact, we are determining the *probability* of each vector belonging to the dominant image motion, or to zero-mean noise. In future work, we wish to show that we can also track multiple moving objects by incorporating methods from the motion tracking community. Black and Anandan, for example, use robust statistics to fit image regions to different motion models [3]. The linear subspace found by our method could serve as such a motion model, allowing estimation of separate subspace coordinates for each image region.

References

[1] M. Agrawal and K. Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive gps. In *ICPR*, 2006. 1

[2] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, volume 588, pages 237–252. Springer, 1992. 2

[3] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, January 1996. 2, 8

[4] J. Campbell, R. Sukthankar, I. Nourbakhsh, and A. Pahwa. A robust visual odometry and precipice detection system using consumer-grade single vision. In *ICRA*, 2005. 1

[5] F. De la Torre and M. Black. Robust principal component analysis for computer vision. *ICCV'01*, 1:362–369, 2001. 3

[6] D. Fleet, M. Black, Y. Yacoob, and A. Jepson. Design and Use of Linear Models for Image Motion Analysis. *International Journal of Computer Vision*, 36(3):171–193, 2000. 2, 3

[7] M. Grossberg and S. Nayar. A general imaging model and a method for finding its parameters. In *Proc. International Conference on Computer Vision*, volume 2, pages 108–115, 2001. 2

[8] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the Fourth Alvey Vision Conference*, volume 15, 1988. 3

[9] B. Horn and B. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17(1-3):185–203, 1981. 2

[10] M. Irani. Multi-frame correspondence estimation using subspace constraints. *International Journal of Computer Vision*, 48(3):173–194, 2002. 2

[11] M. Irani and P. Anandan. Direct recovery of planar-parallax from multiple frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1528–1534, November 2002. 2

[12] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *European Conference on Computer Vision*, pages 282–287. Springer, 1992. 2

[13] H. Longuet-Higgins and K. Prazdny. The Interpretation of a Moving Retinal Image. *Proceedings of the Royal Society of London. Series B, Biological Sciences (1934-1990)*, 208(1173):385–397, 1980. 2

[14] 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. 2, 3

[15] L. Matthies. *Dynamic Stereo Vision*. PhD thesis, Carnegie Mellon University, 1989. 1

[16] K. Ni and F. Dellaert. Stereo tracking and three-point/one-point algorithms – a robust approach in visual odometry. In *International Conference on Image Processing*, 2006. 1

[17] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *IEEE Conference on Computer Vision and Patterns Recognition*, volume 1, pages 652–659, 2004. 1

[18] M. Tipping and C. Bishop. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999. 3, 4

[19] H. Wang, K. Yuan, W. Zou, and Q. Zhou. Visual odometry based on locally planar ground assumption. In *Information Acquisition, 2005 IEEE International Conference on*, page 6, 2005. 1