Transforming Camera Geometry to A Virtual Downward-Looking Camera: Robust Ego-Motion Estimation and Ground-Layer Detection

Qifa Ke and Takeo Kanade
Computer Science Department
Carnegie Mellon University
{ke+,tk}@cs.cmu.edu

Abstract

This paper presents a robust method to solve the two coupled problems: ground layer detection and vehicle ego-motion estimation, which appear in visual navigation. We virtually rotate the camera to the downward-looking pose in order to exploit the fact that the vehicle motion is roughly constrained to be planar motion on the ground. This camera geometry transformation, together with planar motion constraint, will: 1) eliminate the ambiguity between rotational and translational ego-motion parameters, and 2) improve the Hessian matrix condition in the direct motion estimation process. The virtual downward-looking camera enables us to estimate the planar ego-motions even for small image patches. Such local measurements are then combined together, by a robust weighting scheme based on both ground plane geometry and motion compensated intensity residuals, for a global ego-motion estimation and ground plane detection. We demonstrate the effectiveness of our method by experiments on both synthetic and real data.

1 Introduction

Ego-motion estimation and ground plane detection have many applications, such as visual navigation, computer vision based driving assistance, and 3D environment map reconstruction. In this paper we address the case of a single camera rigidly mounted on a car moving in traffic scenes that includes cluttered background including other static or moving objects on the ground plane. It is difficult to apply traditional Structure from Motion algorithms here since they usually require estimating the depth for such cluttered background. To overcome such difficulty, planes in the scene have been used for ego-motion estimation [19, 13].

Ground plane is of special interest. Methods to obtain ground plane include 2D dominant motion estimation [12] and layer extraction [14, 26, 5, 28, 27, 20, 24, 15, 16]. These approaches can be classified into two categories: top-down approaches and bottom-up approaches. Top-down approaches either assume that the ground plane is a dominant plane, or assume that the scene can be approximated with a few planar layers who simultaneously compete for layer support. In our traffic scenarios, the ground plane is not necessary a dominant plane, and the cluttered background is hard to be modelled with a small number of planar layers. It is therefore hard to apply the top-down approaches here. In the bottom-up approaches, images are first divided into small patches, and local measurement (such as 2D image transformation) for each patch is then computed. These local measurements are then grouped into layers. Due to the typical forward motion in vehicle moving, it is necessary to use projective homography for local 2D measurements. Given *small* local support area and low texture on the road (ground), the recovery of projective homography is not reliable due to large number of unknown parameters, small field of view, and ambiguities among its parameters.

In this paper, we assume the camera is calibrated such that its focus length and its relative pose with respect to the vehicle is known. In such a particular setup, the ground plane normal is explicitly constrained too [1]. We can therefore use ego-motion, instead of projective homography, as the local measurement. Given an image patch that is *assumed* to be on the ground, the estimated ego-motion is the local measurement of such image patch. Using ego-motion as the local measurement is an improvement over using projective homography, since it exploits the ground plane geometry. However, estimating ego-motion based on small image patch still suffers from ambiguities among its parameters due to small field of view [2, 7].

To overcome the above difficulty, we exploit the fact that the vehicle motion can be approximated by planar motion on the ground. Such planar motion is of great practice importance and has been used in structure from motion and camera calibration [18, 4], and vehicle ego-motion estimation [22]. In this paper, we use a virtual downward-looking camera to exploit the planar motion constraint. Thinking of a virtual downward-looking camera on planar motion has the following advantages: 1) It eliminates the ambiguity between rotational and translational ego-motion parameters; 2) It improves the Hessian matrix condition in the direct

[1] We do not need to know the distance from the camera to the ground due to the scale ambiguity between the camera translation and scene depth.

1063-6919/03 $17.00 © 2003 IEEE

motion estimation process; 3) It induces image motions that are linear in terms of image coordinates, and therefore can be reliably estimated.

The virtual camera is used to collect the local measurements, i.e., to estimate the planar ego-motions based on small image patches. Such local measurements are then combined together, by a robust weighting scheme for the global ego-motion estimation and ground plane detection. Regularization is then applied for the recovery of the remaining small non-planar motions.

2 Ego-motion estimation

In this section, we describe the direct method to estimate the vehicle ego-motion with respect to a small image patch that is assumed to be on the ground.

2.1 Ego-motion model

Given a sequence of images $I_0, I_1, ..., I_N$ under a perspective camera with internal matrix of $diag(f, f, 1)$, we want to compute the camera ego-motion between the reference image I_0 and another image $I_i, i = 1, 2, ..., N$. The *incremental* image motion at an image point $\mathbf{p} = (x, y)^\top$ in I_i is given by (see [11]):

$$\mathbf{v}_i(\mathbf{p}) = \mathbf{B}_p \mathbf{\Omega}_i + \frac{1}{Z(\mathbf{p})} \mathbf{A}_p \mathbf{T}_i \tag{1}$$

where $\mathbf{\Omega}_i = (\omega_X, \omega_Y, \omega_Z)_i^\top$ and $\mathbf{T}_i = (T_X, T_Y, T_Z)_i^\top$ are the camera rotational and translational velocity, $Z(\mathbf{p})$ is the 3D scene depth at Point $\mathbf{p}$,

$$\mathbf{B}_p = \begin{bmatrix} -\frac{xy}{f} & (f + \frac{x^2}{f}) & -y \\ -(f + \frac{y^2}{f}) & \frac{xy}{f} & x \end{bmatrix} \tag{2}$$

$$\mathbf{A}_p = \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} \tag{3}$$

If we are given a 3D plane $\mathbf{n}^\top \mathbf{P} + d = 0$ with $\mathbf{n} = (n_1, n_2, n_3)^\top$ the plane normal and $\mathbf{P} = (X, Y, Z)^\top$ the 3D coordinate of points on the plane, then we can rewrite Eq.(1) as:

$$\mathbf{v}_i(\mathbf{p}) = \mathbf{B}_p \mathbf{\Omega}_i + \frac{\mathbf{n}^\top \mathbf{F}}{d} \mathbf{A}_p \mathbf{T}_i \tag{4}$$

where $\mathbf{F} = (-\frac{x}{f}, -\frac{y}{f}, -1)^\top$.

Eq.(4) shows that there is a scale ambiguity between d and the camera translation $\mathbf{T}_i$, which means that we can only recover the direction of the camera translation. Without loss of generality, we set $d = -1$ in our experiments.

2.2 Direct estimation of ego-motion

As has been pointed out in [22], in typical traffic scenarios, direct method [11, 8, 6, 17] is more preferable than optical-flow based approach [1, 10, 23, 21] for ego-motion estimation. The reason is that the road usually has weak texture or linear image structure, while the cluttered background including moving objects often contains many feature points.

Given calibrated camera and ground plane normal, we use direct method to estimate the incremental ego-motion based on the brightness constancy assumption, by minimizing the sum square difference (SSD) with respect to the incremental camera motion parameters $\Theta = (\mathbf{\Omega}_i, \mathbf{T}_i)$:

$$\begin{aligned} E(\Theta) &= \sum_p \left[I_i(\mathbf{p} + \mathbf{v}_i(\mathbf{p}, \Theta)) - I_0(\mathbf{p}) \right]^2 \\ &\approx \sum_p \left[\mathbf{g}_p^\top \mathbf{J}_p^\top \Theta + e_p \right]^2 \end{aligned} \tag{5}$$

where $e_p = I_i(p) - I_0(p)$ is the temporal difference at Pixel p, $\mathbf{g}_p^\top = \nabla I_i(\mathbf{p})$ is the image gradient at Pixel $\mathbf{p}$ in image I_i, and $\mathbf{J}_p$ is the Jacobian at p:

$$\mathbf{J}_p = \frac{\partial \mathbf{v}_i(\mathbf{p})}{\partial \Theta} = \begin{bmatrix} \mathbf{B}_p^\top \\ \frac{1}{Z(p)} \mathbf{A}_p^\top \end{bmatrix} = \begin{bmatrix} \mathbf{B}_p^\top \\ \frac{\mathbf{n}^\top \mathbf{F}}{d} \mathbf{A}_p^\top \end{bmatrix} \tag{6}$$

From Eq.(5), we can see that every pixel inside the image patch with non-zero intensity derivative makes a contribution to the final solution of Θ. To achieve robustness to outliers, the contribution of each pixel should be weighted according to some robust criteria. For example, robust estimator uses the residual e_p to determine the weight $w_p = w(e_p) = \frac{\dot{\rho}(e_p)}{e_p}$, where $\rho(\cdot)$ is some robust M-estimator.

The weighted least square solution of Eq.(5) is given by:

$$\Theta = \mathbf{L}^{-1} \mathbf{b} \tag{7}$$

where

$$\mathbf{L} = \sum_p w_p \mathbf{J}_p \mathbf{g}_p \mathbf{g}_p^\top \mathbf{J}_p^\top \tag{8}$$

is the *Hessian*,

$$\mathbf{b} = \sum_p (-w_p e_p \mathbf{J}_p \mathbf{g}_p) \tag{9}$$

is the *accumulated residual*.

Once we recover the incremental camera motion parameters $\mathbf{\Theta} = (\mathbf{\Omega}_i, \mathbf{T}_i)$, we perform an incremental update to the ego-motion $\mathbf{M}_i$:

$$\mathbf{M}_i \leftarrow \mathbf{M}_i \begin{bmatrix} \mathbf{R}(\mathbf{\Omega_i}) & \mathbf{T}_i \\ 0 & 1 \end{bmatrix} \tag{10}$$

where $\mathbf{R}(\mathbf{\Omega_i})$ is the incremental rotation matrix given by the Rodriguez's formula:

$$\mathbf{R}(\mathbf{\Omega}) = \mathbf{I} + [\tilde{\mathbf{n}}]_\times \sin\theta + [\tilde{\mathbf{n}}]_\times^2 (1 - \cos\theta) \tag{11}$$

where $\theta = \|\Omega\|$, and

$$[\tilde{\mathbf{n}}]_\times = \frac{1}{\theta} \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}. \tag{12}$$

The overall direct ego-motion computation is an iterative Gauss-Newton gradient decent process. Each iteration consists of the following three steps:

1. Compute the incremental motion parameters (Eq.(7)).
2. Perform the incremental update to the ego-motion $\mathbf{M}_i$ (Eq.(10)).
3. Warp the image I_i towards the reference image I_0, using the homography induced by the ground plane $(\mathbf{n}, d)$ under current ego-motion: $\mathbf{H} = \mathbf{K}(\mathbf{R}(\mathbf{\Omega}) - \frac{\mathbf{T}}{d}\mathbf{n}^\top)\mathbf{K}^{-1}$, where $\mathbf{K} = \text{diag}(f, f, 1)$ is the camera internal matrix.

3 Camera models for planar ego-motion estimation

There are several difficulties in estimating the full vehicle ego-motion based on a *small* image patch:

- During a short period of time, the vehicle undergoes approximately planar motion. For a camera rigidly mounted on such vehicle [2], its ego-motion consists of a rotation around an axis vertical to the ground plane, and two translations parallel to the ground plane. Therefore, full ego-motion model contains more parameters than necessary. Estimating such diminishing parameters are inherently ill-conditioned.
- There are inherent ambiguities between rotation and translation. Given a small image patch, therefore small field of view (FOV), it is hard to differentiate the w_X-induced flow from the T_Y-induced flow, and the w_Y induced flow from the T_X-induced flow, respectively [2, 7]. These inherent ambiguities introduce elongated valley in the SSD error function [3], resulting in slow convergence and bad local minima.

It is therefore necessary to exploit the planar motion constraint. To do so, we divide the six ego-motion parameters into two triples. The first triple consists of the planar motion parameters, and the second triple consists of the diminishing non-planar motion parameters that can be ignored at the stage of local measurement. In the following, we introduce two virtual cameras and analyze how the selection of camera models affects the effectiveness in exploiting planar motion constraint.

Virtual cameras can be achieved by rectifying the images using the homography induced by the ground plane and the relative pose between the original camera and the virtual camera. Doing so requires camera calibration for the camera rotational pose with respect to the vehicle. We assume the camera is fixed with respect to the vehicle, which means that the calibration can be done before hand (see the Appendix for a simple calibration method). It is important to keep the virtual cameras always on the same plane so that the camera motions among the rectified images are still planar motions.

[2]The camera can have any orientation, but is otherwise fixed w.r.t. the vehicle body.

3.1 Virtual forward-looking camera

In a typical setting, the camera is mounted on the vehicle looking at the ground at some angle, as shown in Fig.(7). A simple way to make use of the planar motion constraint is to virtually rotate the camera such that its optical axis (Z axis) points forward horizontally and its XZ plane parallel to the ground plane, as has been done in [22]. We will call it the *forward-looking* camera.

The planar ego-motion parameters are then reduced to $\Theta_f = (w_Y, T_X, T_Z)$. There still exists ambiguity between w_Y and T_X. In [22], the dominant camera motion set is chosen to be (w_X, w_Y, T_Z). But in real experiments we have observed non-negligible T_X, especially when the vehicle is changing lanes or turning. Moreover, the camera motion is not longer planar due to w_X.

In the coordinate frame of the forward-looking camera, the normal of the ground plane is $(0, 1, 0)$, and the Jacobian w.r.t. Θ_f is:

$$\mathbf{J}_p = \frac{\partial \mathbf{v}_i(\mathbf{p})}{\partial \Theta_f} = \begin{bmatrix} f + \frac{x^2}{f} & y & -\frac{xy}{f} \\ \frac{xy}{f} & 0 & -\frac{y^2}{f} \end{bmatrix}^\top \quad (13)$$

In addition to the ambiguity between w_Y and T_X, the above Jacobian also indicates the following problems:

- It is usually hard to estimate w_Y and T_Z within a small FOV since they introduce image motions that are second order polynomial terms of the image coordinate (x, y).
- The Hessian matrix is determined by both the image texture and the Jacobian. When the texture is low,the second order terms in the Jacobian will contribute to a badly-conditioned Hessian matrix.

Coordinate normalization and translation are useful technique to improve the matrix condition number [9]. In our case, coordinate normalization does not change the condition of the Hessian matrix, since every element in the Jacobian is multiplied by a same constant [3]. Translating the coordinates to center around $(0, 0)$ will improve the matrix condition. Doing so effectively translates the camera such that its optical axis passes through the center of the input image patch. In the forward-looking camera, it is impossible to do so given an image patch on the ground that is parallel to the camera optical axis.

3.2 Virtual downward-looking camera

The above analysis on forward-looking camera geometry motivates us to rotate and translate (parallel to the ground plane) the camera geometry such that we think of a virtual camera whose optical axis is vertical to the ground plane

[3]Notice that f also needs to be scaled according to the normalization to preserve the correctness of Eq.(1).

and passing through the center of input image patch. We call it the *downward-looking* camera.

In the coordinate frame of downward-looking camera, the normal of the ground plane is $(0, 0, 1)$. The dominant motion becomes $\Theta_d = (w_Z, T_X, T_Y)$, and the Jacobian w.r.t. Θ_d is:

$$\mathbf{J}_p = \frac{\partial \mathbf{v}_i(\mathbf{p})}{\partial \Theta_d} = \begin{bmatrix} -y & f & 0 \\ x & 0 & f \end{bmatrix}^\top \tag{14}$$

The advantages of using the downward-looking camera are:

- The above Jacobian consists of only zero and first order polynomial terms, which, together with the virtual camera translation so that its image coordinates are center around $(0, 0)$, will result in a well-conditioned Hessian matrix even when the road has low texture.
- There is not inherent ambiguities among the parameters in Θ_d. It is easy to differentiate the flow induce by w_Z from the flow induced by (T_X, T_Z). Indeed, Θ_d can be reliably estimated since they induce image motions that are linear in terms of image coordinate (x, y) (no perspective distortion).

Notice that equally treating the pixels in the rectified image is equivalent to give larger weights to pixels (in the original un-rectified image) that correspond to points further-away on the ground plane, due to the perspective distortion (front-shorten) in the un-rectified image. We can adjust such scene-dependent weighting by non-uniform image sampling. Also notice that translating the camera to look at the patch center effectively enlarges the camera field of view (FOV). We avoid the degenerate case of infinite rectified image area by using only the image pixels below the horizon line (vanishing line of the ground plane), since pixels above the horizon line in the image are obvious non-ground pixels. Given the camera pose relative to the vehicle, it is straightforward to calculate the horizon line (see Appendix for details).

4 Ground plane detection and global ego-motion estimation by virtual downward-looking camera

This section describes a robust technique to combine locally estimated ego-motions for ground plane detection and global ego-motion estimation. The general framework of the algorithm is:

1. Bootstrap from local estimations: Divide the image into small $n \times n$ patches [4]. For each patch, estimate an ego-motion (Section 3.2) and compute its robust weights based on both geometry and intensity residuals (Section 4.1).
2. Combine the local estimations according to their weights for global ego-motion estimation, including the non-planar motions.
3. Recompute the robust weight based on current ego-motion.

Step 1 is an important bootstrap step to provide a good initialization for further global estimation. Step 2 and 3 are the two iterative steps. In our experiments, we have found one or two iterations are enough, due to the accurate local ego-motion estimation by the downward-looking camera. The ground plane is detected based on the final weights.

4.1 Geometry based robust weighting

Traditional robust weighting uses motion compensated pixel intensity residuals e_p, i.e., $w_p = w(e_p)$ in Eq.(7). The residual e_p depends on both geometry and texture. Pixels not on the ground plane but with low texture will also have low residuals when compensated by the motion corresponding to the ground plane, and will be given large weights if weighting is purely based on intensity residuals. When the ground layer has low texture, the inclusion of those false pixels will affect the final ego-motion estimation. We should exclude such false pixels by exploiting the ground plane geometry in the robust weighting.

For each patch in the image, we initialize its plane normal as the ground plane normal, then refine its plane normal (Section 4.1.1) under the currently estimated ego-motion. If the patch is in fact on the ground, the refined plane normal will be close to the ground normal due to accurate initialization. Otherwise, we will end up with a plane normal that is distinct from the plane normal of the ground [5].

Our final weighting scheme use both the intensity residuals, and the angle between the re-estimated plane normal $\mathbf{n}$ and the ground normal $\mathbf{n}_g$:

$$w_p = w(e_p, \theta) \tag{15}$$

where e_p is the intensity residual, $\theta = arccos\left(\frac{\mathbf{n}^\top \mathbf{n}_g}{\|\mathbf{n}\| \cdot \|\mathbf{n}_g\|}\right)$, and $w(\cdot, \sigma)$ is the robust weighting with scale σ that is set to the robust standard deviation (see [5]) by $\sigma = 1.4826 \cdot median_p|e_p|$.

4.1.1 Compute plane normal

This section describes the direct method to estimating the plane normal based on current ego-motion. We re-use Eq.(7) and the corresponding algorithm in Section 2.2, except that the unknowns are the plane normal $\mathbf{n} = (n_1, n_2, n_3)$ instead of ego-motion Θ. We therefore need to derive the new Jacobian $\mathbf{J}_p = \frac{\partial \mathbf{v}_i(\mathbf{p})}{\partial \mathbf{n}}$. Given the ego-motion and the ground plane equation, we prefer using the exact homography to represent $\mathbf{v}_i(\mathbf{p})$, instead of using the

[4] We use overlap image patches.

[5] We do not care if such non-ground plane normal is actually correct, as long as it is distinct from the normal of the ground plane.

instantaneous representation in Eq.(4). The reason is the following. In each step of incremental ego-motion estimation, the instantaneous representation is a good approximation since the incremental ego-motion is very small. But once the final ego-motion Θ is recovered, instantaneous representation is no longer a good approximation, especially when multiple frames are used.

Suppose the initial plane normal is $\mathbf{n}$, we want to compute the incremental plane normal update $\mathbf{m}$ to $\mathbf{n}$. The homography $\mathbf{H}$ induced by the updated plane $(\mathbf{n}+\mathbf{m})\mathbf{P}+d=0$ is:

$$\begin{aligned}\mathbf{H} &= \mathbf{K}(\mathbf{R}(\mathbf{\Omega}) - \frac{\mathbf{T}}{d}\mathbf{n}^\top)\mathbf{K}^{-1} - \mathbf{K}\frac{\mathbf{T}}{d}\mathbf{m}^\top\mathbf{K}^{-1} \\ &= \tilde{\mathbf{R}} - \mathbf{K}\frac{\mathbf{T}}{d}\mathbf{m}^\top\mathbf{K}^{-1} \end{aligned} \quad (16)$$

where $\mathbf{K} = \text{diag}(f, f, 1)$ is the camera internal matrix.

Denote $\mathbf{r}_i^\top$ the i-th row of $\tilde{\mathbf{R}}$, and $[\tilde{x}, \tilde{y}] = [\frac{\mathbf{r}_1^\top\mathbf{p}}{\mathbf{r}_3^\top\mathbf{p}}, \frac{\mathbf{r}_2^\top\mathbf{p}}{\mathbf{r}_3^\top\mathbf{p}}]$. The incremental image motion at point $\mathbf{p} = (x, y, 1)^\top$ is:

$$\mathbf{v}_i(\mathbf{p}) = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{h}_1^\top\mathbf{p}}{\mathbf{h}_3^\top\mathbf{p}} - \tilde{x} \\ \frac{\mathbf{h}_2^\top\mathbf{p}}{\mathbf{h}_3^\top\mathbf{p}} - \tilde{y} \end{bmatrix} \quad (17)$$

where $\mathbf{h}_i^\top$ is the i-th row vector of $\mathbf{H}$.

The Jacobian $\mathbf{J}_p$ with respect to $\mathbf{m}$ is: $\mathbf{J}_p = \frac{\partial \mathbf{v}_i(\mathbf{p})}{\partial \mathbf{m}} =$

$$\frac{\text{diag}(x, y, 1)}{d\mathbf{r_3}^\top\mathbf{p}} \begin{bmatrix} \frac{\tilde{x}}{f}T_Z - T_X & \frac{\tilde{x}}{f}T_Z - T_X & \tilde{x}T_Z - fT_X \\ \frac{\tilde{y}}{f}T_Z - T_Y & \frac{\tilde{y}}{f}T_Z - T_Y & \tilde{y}T_Z - fT_Y \end{bmatrix}^\top$$

At each iteration, the plane normal is updated by:

$$\mathbf{n} \leftarrow \mathbf{n} + \mathbf{m}$$

The new plane normal is then plugged into Eq.(16) to compute the new homography for the next iteration.

4.2 Recovering remaining non-planar motion parameters

After the planar ego-motions have been recovered, we can estimate other small non-planar motions, which might exhibit due to vehicle bouncing or non-planar road condition. Under the coordinate frame of the downward-looking camera, the non-planar motion set is $\Theta_2 = (w_X, w_Y, T_Z)$. The Jacobian $\mathbf{J}_p$ w.r.t. the non-planar motion parameters Θ_2 is:

$$\mathbf{J}_p = \frac{\partial \mathbf{v}_i(\mathbf{p})}{\partial \Theta_2} = \begin{bmatrix} -\frac{xy}{f} & f + \frac{x^2}{f} & -x\frac{\mathbf{n}^\top\mathbf{F}}{d} \\ -(f + \frac{y^2}{f}) & \frac{xy}{f} & -y\frac{\mathbf{n}^\top\mathbf{F}}{d} \end{bmatrix}^\top$$

Estimating Θ_2 is inherently ill-conditioned since it induces very small or diminished image motions that are second order polynomial terms of image coordinates. Nevertheless, small or diminishing motions mean that it is safe to apply strong regularization to improve the condition. The regularized cost function is: $E(\Theta_2) =$

$$\sum_{\mathbf{p}} \left[\tilde{I}_i(\mathbf{p} + \mathbf{v}_i(\mathbf{p}, \Theta_2)) - I_0(\mathbf{p})\right]^2 + \lambda \sum_{\mathbf{p}} \mathbf{v}_i(\mathbf{p}, \Theta_2)^2 \quad (18)$$

where $\tilde{I}_i$ is the image I_i warped by the homograpy induced by the ground plane under current ego-motion Θ_1. The second summation term is the regularization term, which states that the image motion induced by parameter set Θ_2 must be small. $\lambda \geq 0$ is a constant parameter. A larger λ enforces stronger regularization.

By setting $\frac{\partial E(\Theta_2)}{\partial \Theta_2} = 0$, the weighted least square solution is:

$$\Theta_2 = \left[\sum_{\mathbf{p}} w_p \mathbf{J}_p(\mathbf{g}_p\mathbf{g}_p^\top + \lambda\mathbf{I})\mathbf{J}_p^\top\right]^{-1} \sum_{\mathbf{p}} (-w_p e_p \mathbf{J}_p \mathbf{g}_p) \quad (19)$$

Enforcing the regularization is equivalent to "virtually improve" the texture, as shown by the diagonal matrix $\lambda\mathbf{I}$ in Eq.(19), and will therefore improve the condition of Hessian matrix.

5 Experimental results

5.1 Local planar ego-motion estimation

This section presents the experimental results on estimating the planar ego-motion based on small image patches, which is an important bootstrap step for further global ego-motion estimation and ground plane detection. To deal with large motion, in the experiments we use a multi-resolution Gaussian pyramid of the input images. In all experiments, we only use the image pixels below the horizon line, since pixels above the horizon line in the image are obvious non-ground pixels.

5.1.1 Synthetic case

To compare different motion models, we use a synthetic image sequence with ground truth. Fig.(1) shows the two synthesized images, where the camera simulates a moving vehicle on the ground plane by simultaneously moving forward and turning left (around an axis at some distance to the vehicle and vertical to the ground). The normal of the ground plane and the camera focus length are known.

The synthetic case in Table (1) quantitatively compares the condition number of the Hessian matrix and the recovered ego-motion parameters using four different motion models. The image patch used to compute the ego-motion is indicated by the rectangle in Fig.(1a).

From Table (1), we can see that removing the diminishing parameters greatly improves the condition of Hessian matrix, since diminishing parameters are inherently ill-conditioned. As a result, the 8-parameter model has the worst condition since its number of unknown parameters is far more than necessary. The downward-looking camera improves the condition number by orders of magnitudes,

	synthetic case (Fig.(1))		real case (Fig.(3))	
	condition num.	ego-motion	condition num.	ego-motion
8-parameter	1.6312e+006	N/A	1.7618e+006	N/A
full ego-motion	3.0674e+003	$\begin{bmatrix} 0.2162^\circ & -0.1360^\circ & -1.5926^\circ \\ -0.0301 & 0.0226 & -0.1264 \end{bmatrix}$	8.5083e+004	$\begin{bmatrix} -0.7710^\circ & -0.1182^\circ & 0.2130^\circ \\ -0.2001 & -0.2367 & 0.0636 \end{bmatrix}$
forward-looking	2.1349e+002	$[-0.4725^\circ, 0.0058, 0.0903]$	4.5254e+003	$[0.0108^\circ, -0.0064, 0.0595]$
downward-looking	8.4357e+000	$[-0.9991^\circ, -0.0181, 0.1066]$	5.5469e+001	$[-0.0840^\circ, -0.1222, 0.2497]$

Table 1. Ego-motion estimation and condition of Hessian (larger condition number means worse condition). For synthetic case, the ground truth of ego-motion is: $(w_Y, T_X, T_Z) = (-1.0^\circ, -0.0175, 0.1)$, in the coordinate frame of forward-looking camera. Translations are measured by the unit of image height. The motion parameters of the 8-parameter model do not directly indicate the ego-motion parameters, and are not shown here.

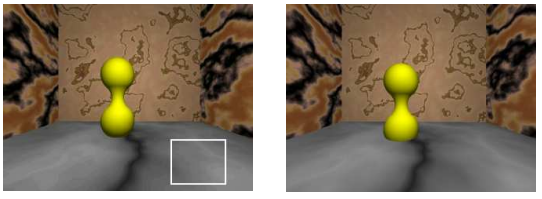
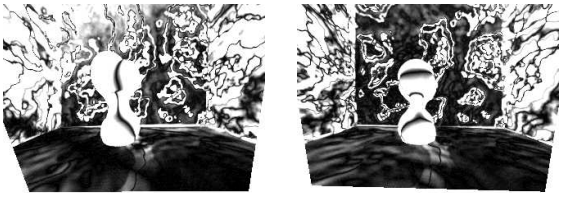
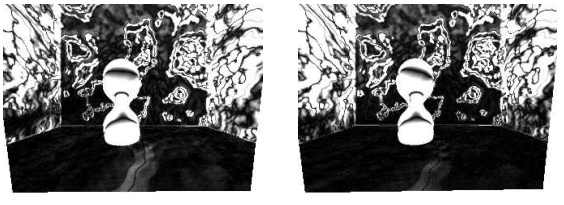


Figure 1. Synthesized images where the ground plane has low textures. The rectangle shows one of the patch used to compute the camera ego-motion.



(a): 8-parameter (b): full ego-motion



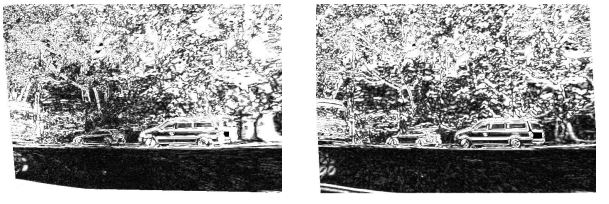
(c): forward-looking (d): downward-looking

Figure 2. Motion compensated residual images by motions from Table (1). The residuals are scaled up by a factor of 4 for visibility.

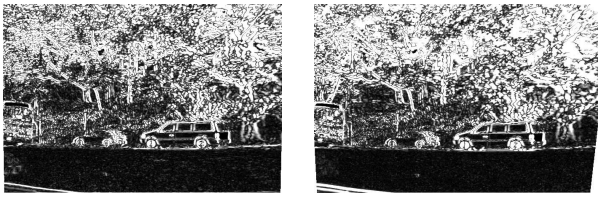
and recovers the most accurate ego-motion, which supports our observations in Section 3.2. The forward-looking model performs better than the full ego-motion model. But it appears that part of the left-turn has been confused by left-translation in the forward-looking model.

Fig.(2) shows the motion-compensated residual images for qualitative comparison. As we can see, pixels inside the used rectangle are well-compensated in all models, but only the downward-looking camera fully compensates all pixels in the ground plane, which means that it actually recovers a good global motion model based on a small image patch.

Figure 3. Real images with low textures on the ground plane, and moving cars/bus in the background.



(a): 8-parameter (b): full ego-motion



(c): forward-looking (d): downward-looking

Figure 4. Motion compensated residual images. The residuals are scaled up by a factor of 4 for visibility. Notice the residuals of lane-marks at the bottom left, and the residuals of car dash-board right below the lane-marks. The downward-looking camera model compensates the lane marks best, and shows correct parallax on the car dash-board.

5.1.2 Real case

In this subsection, we use real images to compare the performance of ego-motion estimation based on small image patches. Fig.(3) shows the images we use, where the camera is put on a car that is simultaneously moving forward and turning left (around an axis at some distance to the vehicle and vertical to the ground). The rectangle shows the image patch we randomly select to compute the ego-motion. It is quite a challenging task due to the very low texture of

Figure 5. Traffic scene in city with cluttered background containing moving cars. The road has weak or linear textures.

the road and the small image patch.

The last two columns in Table (1) show the condition number of the Hessian matrix and the recovered ego-motion. As we can see, the downward-looking camera has the best condition number, and its recovered ego-motion correctly indicates that the car is moving forward and turning left. The forward-looking camera model does not recover the correct left-turn motion, which appears to be caused by the confusion between w_Y and T_X. The full ego-motion model has large non-planar motions, which is obviously incorrect.

The motion compensated residual images in Fig.(4) qualitatively show the performance. As we can see, all motion models well compensate the pixels inside the rectangle that are used for estimation, but only the downward-looking camera compensates all the pixels on the ground, as can be indicated by the yellow lane marks at the bottom-left of the images. The darker pixels at the very bottom-left of the images (right below the yellow lane marks) in Fig.(3) are part of the car dash-board, and their corresponding residuals in the downward-looking camera model show correct parallax. We also use the shape of the image boundaries to indicate the ego-motion. As we can see, only the downward-looking camera has correct shape corresponding to forward and left-turn motions.

5.2 Ground layer detection and global ego-motion estimation

In this section, we show the results of ground layer detection and the global ego-motion estimation. Fig.(3) and Fig.(5) show the two image sequences we used in this experiment. The roads have either very weak texture, or linear image structure. The backgrounds are cluttered and contain moving objects.

Fig.(6) shows the experimental results. The first row is the result on Fig.(3), and the second row is the result on Fig.(5). Fig.(6b) shows the weights (see Eq.(15)) indicating the ownership (ground layer or non-ground layer) of the pixels. Outliers, such as moving cars (and their shadows), buildings, and trees on the side, are clearly indicated by low weights. Fig.(6c) shows the detected ground layer using a simple histogram-based threshold scheme. The lane marks on the road are included in the ground layer although their colors are quite different from the majority pixels of the road plane. The car dash-board at the bottom-left in the first image sequence, and the trees and moving cars on the ground in both image sequences, are excluded due to significantly lower weights. Notice that some of the outliers, such as the trees at the right side of the road in the second sequence, have very low texture and therefore low intensity residuals, but still be excluded due to the fact that their geometries (plane normals) are significantly different from the ground plane normal. Fig.(6d) shows the motion compensated residuals by the global ego-motion estimated based on the weights in (b). As we can see, the pixels on the road are well compensated, while pixels from other objects, such as the buildings, trees, and the moving cars with their shadows, show correct parallax.

6 Conclusion

Vehicle ego-motion estimation and ground layer detection are challenging tasks due to low texture on the road and the non-linear perspective distortion. By ways of virtual camera, we have made use of the constraint that the vehicle is undergoing planar motion on the ground. Enforcing such constraint is necessary to avoid the estimation of diminishing parameters that are ill-conditioned. By using virtual downward-looking camera, we further improve the condition of the Hessian matrix, and eliminate the ambiguities among the unknown parameters, which are linear in terms of image coordinates and can be reliably estimated. Together with a geometry-based robust weighting scheme, we have shown promising results on vehicle ego-motion estimation and ground layer detection.

We have assumed that the camera focus length is known. In practice, we only require a rough initialization of the focus length, since the error in the focus length only introduces systematic bias on the estimated ego-motion, but does not affect the ground layer detection. We can therefore use the algorithm presented in this paper to derive the ground plane. Then use the detected ground plane to calibrate the camera [25] to correct the bias in ego-motion.

References

[1] G. Adiv. Determining 3-d motion and structure from optical flow generated by several moving objects. *PAMI*, 7(4):384–401, July 1985.

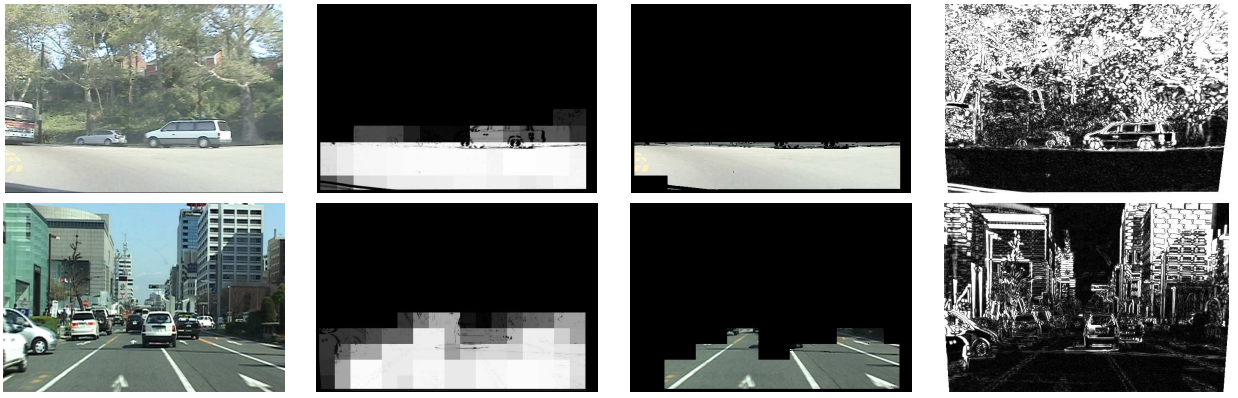
[2] G. Adiv. Inherent ambiguities in recovering 3-d motion and structure from a noisy flow field. *PAMI*, 11(5), May 1989.

[3] Y. Aloimonos. Harmonic computational geometry: A new tool for visual correspondence. In *BMVC 2002*, 2002.

[4] M. Armstrong, A. Zisserman, and R. I. Hartley. Self-calibration from image triplets. In *ECCV96*.

[5] S. Ayer and H. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *ICCV95*.

[6] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV92*.



(a) (b) (c) (d)

Figure 6. Ground layer detection and global ego-motion estimation. (a): reference frame of the input images; (b): weights indicating the ownership of pixels (brighter means larger weight); (c): detected ground layer using weights in (a); (d): motion compensated residuals by the global ego-motion. The residuals are scaled up by a factor of 4 for visibility.

[7] K. Daniilidis and H. Nagel. The coupling of rotation and translation in motion estimation of planar surfaces. In *CVPR93*, pages 188–193, 1993.

[8] K. Hanna. Direct multi-resolution estimation of ego-motion and structure from motion. In *MOTION91*, 1991.

[9] R. Hartley. In defence of the 8-point algorithm. In *ICCV95*.

[10] D. Heeger and A. Jepson. Subspace methods for recovering rigid motion i: Algorithms and implementation. *IJCV*, 7(2):95–117, January 1992.

[11] B. Horn and E. Weldon, Jr. Direct methods for recovering motion. *IJCV*, 2(1):51–76, June 1988.

[12] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *ECCV92*.

[13] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. In *CVPR94*, 1994.

[14] A. Jepson and M. Black. Mixture models for optical flow computation. In *CVPR93*.

[15] Q. Ke and T. Kanade. A subspace approach to layer extraction. In *CVPR 2001*.

[16] S. Khan and M. Shah. Object based segmentation of video using color, motion and spatial information. In *CVPR01*.

[17] R. Mandelbaum, G. Salgian, and H. Sawhney. Correlation-based estimation of ego-motion and structure from motion and stereo. In *ICCV99*, pages 544–550, 1999.

[18] S. Maybank. Theory of reconstruction from image motion, springer. *BERLIN*, 93:1992.

[19] S. Negahdaripour and B. Horn. Direct passive navigation. *PAMI*, 9(1):168–176, January 1987.

[20] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV'98*.

[21] S. Soatto and P. Perona. Recursive 3-d visual-motion estimation using subspace constraints. *IJCV*, 22(3).

[22] G. Stein, O. Mano, and A. Shashua. A robust method for computing vehicle ego-motion. In *IEEE Intelligent Vehicles Symposium (IV2000)*, 2000.

[23] T. Tian, C. Tomasi, and D. Heeger. Comparison of approaches to egomotion computation. In *CVPR'96*.

[24] P. Torr, R. Szeliski, and P. Anandan. An integrated bayesian approach to layer extraction from image sequences. In *ICCV99*.

[25] W. Triggs. Autocalibration from planar scenes. In *ECCV'98*.

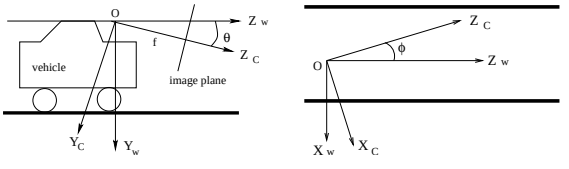
[26] J. Wang and E. Adelson. Representing moving images with layers. *IEEE Trans. on Image Processing*, 3(5), 1994.

[27] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *CVPR97*.

[28] Y. Weiss and E. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *CVPR96*.

Appendix: calibration of camera look-at point

Fig.(7) shows the coordinate frames of the vehicle (X_W, Y_W, Z_W) and the camera (X_C, Y_C, Z_C). We must know the camera look-at point [6] *w.r.t. the vehicle* in order to virtually rotate the camera desired pose. The look-at point is defined, in the frame of (X_W, Y_W, Z_W), as the intersection point of axis Z_C and the plane $Z_W = 1$. Since the camera is fixed w.r.t. the vehicle, the look-at point is fixed too. To derive the look-at point, we drive the car along a straight road with parallel lane marks, and take a few images. If the optical axis Z_C is identical to the axis Z_W, the vanishing point in each image, $v = (x_v, y_v)$, of the parallel lane marks will be coincident with the camera principal point $c = (x_c, y_c)$ [7]. Therefore the look-at point is $\left(\frac{x_v - x_c}{f}, \frac{y_v - y_c}{f}\right)$, where f is focus length. Each image gives an estimation of the look-at point. If using multiple images, we use the mean of them. The horizon line is then defined as $y = y_v$. Automatic techniques have been developed by researchers to compute vanishing points. In our experiments, the lane marks are semi-automatically identified.



(a): side view (b): top view

Figure 7. Coordinate frames.

[6] We assume x_C is parallel to the ground plane. Violation of such assumption will only introduce a constant bias in the rotation around the camera axis, which cancels out when computing the relative ego-motion among the views.

[7] Assumed to be at the image center.