Parameterizing Homographies

CMU-RI-TR-06-11

Simon Baker, Ankur Datta, and Takeo Kanade

The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

The motion of a plane can be described by a homography. We study how to parameterize homographies to maximize plane estimation performance. We compare the usual 3×3 matrix parameterization with a parameterization that combines 4 fixed points in one of the images with 4 variable points in the other image. We empirically show that this 4pt parameterization is far superior. We also compare both parameterizations with a variety of direct parameterizations. In the case of unknown relative orientation, we compare with a direct parameterization of the plane equation, and the rotation and translation of the camera(s). We show that the direct parameterization is both less accurate and far less robust than the 4-point parameterization. We explain the poor performance using a measure of independence of the Jacobian images. In the fully calibrated setting, the direct parameterization just consists of 3 parameters of the plane equation. We show that this parameterization is far more robust than the 4-point parameterization, but only approximately as accurate. In the case of a moving stereo rig we find that the direct parameterization of plane equation, camera rotation and translation performs very well, both in terms of accuracy and robustness. This is in contrast to the corresponding direct parameterization in the case of unknown relative orientation. Finally, we illustrate the use of plane estimation in 2 automotive applications.

1 Introduction

The motion of a plane can be described by a homography. This is true for a moving plane imaged by a static camera. It is true for a static plane imaged by a moving camera. It is true for the apparent motion of a plane between 2 cameras. It is true whether the camera(s) are calibrated or not.

We investigate how to parameterize a homography to maximize plane estimation performance. The usual approach is to parameterize a homography with the elements of the 3×3 homography matrix. We compare the 3×3 parameterization with a parameterization (the 4pt parameterization) that combines 4 fixed points in one of the images with 4 variable points in the other image. We compare these parameterizations empirically and show that the 4pt parameterization is significantly better, both in terms of robustness and in terms of plane estimation accuracy.

We also compare these 2 parameterizations with a variety of direct parameterizations [Negahdaripour and Horn, 1987]. We consider 3 different cases: (1) **Unknown Relative Orientation:** A plane imaged by 2 intrinsically-calibrated cameras when the relative orientation of the cameras is unknown, (2) **The Fully Calibrated Case:** A plane imaged by 2 fully calibrated cameras, and (3) **A Moving Stereo Rig:** A plane imaged simultaneously, at 2 time instants, by 2 fully calibrated cameras as the plane or the (rigidly-connected) cameras move.

In the case of unknown relative orientation, we compare the 3×3 and 4pt parameterizations with the direct parameterization of the plane equation, and the rotation and translation of/between the camera(s). We empirically show that the direct parameterization is more accurate than the 3×3 parameterization, but less accurate than the 4pt parameterization. More significantly, the direct parameterization is far less robust than the other 2 parameterizations. The reason for the poor robustness is the inherent ambiguity between rotation and translation in ego-motion estimation [Longuet-Higgins, 1984, Adiv, 1989, Daniilidis and Nagel, 1993]. We show how this behavior can be explained using a measure of independence of the Jacobian images.

The direct parameterization in the fully calibrated setting, sometimes known as "Layered Stereo" [Baker *et al.*, 1998], is far more robust than both the 3×3 parameterization and the 4pt parameterization. This improved robustness is to be expected because the direct parameterization

just consists of 3 parameters of the plane equation, compared to 8 unknowns for the other 2 parameterizations. The direct parameterization and the 4pt parameterization are equally accurate. The 3×3 parameterization is far less accurate, however.

In the case of a moving stereo rig we find that the direct parameterization of the plane equation, camera rotation, and translation (9 parameters) is just as robust as an independent parameterization of the plane equation at each time instant (3 parameters at each time), and even more accurate. In contrast to the case of unknown relative orientation, the extra constraints between the calibrated images eliminate any ambiguity between rotation and translation across time.

Finally, in Section 4.2 we illustrate the use of plane estimation in 2 automotive applications: (1) the generation of a "Top Down View" of the road surface and (2) the estimation of car egomotion and the creation of a road mosaic.

2 Background

2.1 Geometry of Planes and Homographies

We assume that the i^{th} camera can be modeled using a standard projective model [Faugeras, 1993]:

$$\mathbf{u}_i \;=\; \mathbf{P}_i \mathbf{x} \quad \text{where} \quad \mathbf{P}_i = \mathbf{K}_i \left[\mathbf{R}_i \; \mathbf{T}_i\right] \tag{1}$$

where $\mathbf{u}_i = (u_i, v_i, 1)^{\mathrm{T}}$ are 2D homogeneous image coordinates, $\mathbf{x} = (x, y, z, 1)^{\mathrm{T}}$ are 3D homogeneous world coordinates, $\mathbf{P}_i$ is the 3×4 projective camera matrix, $\mathbf{K}_i$ is the 3×3 matrix of intrinsic camera parameters, $\mathbf{R}_i$ is the rotation of world coordinate system relative to the camera, $\mathbf{T}_i$ is the location of the world coordinate origin in the camera reference frame (the camera center is located at $-\mathbf{R}_i^{-1}\mathbf{T}_i$ in the world coordinate system), and equality is defined up to scale in the 2D projective space $\mathcal{P}^2$ [Faugeras, 1993].

Given two cameras $\mathbf{P}_i$ and $\mathbf{P}_j$, we would like to model the relationship between coordinates $\mathbf{u}_i$ in one camera and the corresponding coordinates $\mathbf{u}_j$ in the other camera. Following [Baker *et al.*,

1998], Equation (1) can be partially inverted to give:

$$\mathbf{x} \;=\; \mathbf{P}_i^* \mathbf{u}_i + \lambda \mathbf{p}_i \tag{2}$$

where $\mathbf{P}_i^* = \mathbf{P}_i^{\mathbf{T}}(\mathbf{P}_i \mathbf{P}_i^{\mathbf{T}})^{-1}$ is the 4×3 pseudo-inverse of $\mathbf{P}_i$, λ is an unknown scalar, and

$$\mathbf{p}_i = \begin{pmatrix} -\mathbf{R}_i^{-1}\mathbf{T}_i \\ 1 \end{pmatrix} \tag{3}$$

is the epipole, a vector in the null-space of $\mathbf{P}_i$; i.e. $\mathbf{P}_i \mathbf{p}_i = \mathbf{0}$. Suppose that all of the points under consideration lie on the plane $\mathbf{n}^{\mathrm{T}}\mathbf{x} = 0$ where $\mathbf{n} = (n_x, n_y, n_z, n_d)^{\mathrm{T}}$. The first 3 components (n_x, n_y, n_z) are the normal to the plane. The fourth component n_d (once the vector is normalized such that $n_x^2 + n_y^2 + n_z^2 = 1$) is the perpendicular distance of the plane from the origin. The scale λ can then be solved from:

$$\mathbf{n}^{\mathrm{T}}\mathbf{x} \;=\; \mathbf{n}^{\mathrm{T}} \left[\mathbf{P}_i^* \mathbf{u}_i + \lambda \mathbf{p}_i\right] \;=\; 0. \tag{4}$$

Solving this equation for λ, substituting the result into Equation (2), projecting the result into camera j using Equation (1), and rearranging gives:

$$\mathbf{u}_j \;=\; \mathbf{P}_j \left[\left(\mathbf{n}^{\mathrm{T}}\mathbf{p}_i\right)\mathbf{I} - \mathbf{p}_i \mathbf{n}^{\mathrm{T}}\right] \mathbf{P}_i^* \mathbf{u}_i. \tag{5}$$

The expression:

$$\mathbf{H}_{i,j} \;=\; \mathbf{P}_j \left[\left(\mathbf{n}^{\mathrm{T}}\mathbf{p}_i\right)\mathbf{I} - \mathbf{p}_i \mathbf{n}^{\mathrm{T}}\right] \mathbf{P}_i^* \tag{6}$$

is a 3×3 matrix, defined up to scale, which is known as a homography. The homography $\mathbf{H}_{i,j}$ maps coordinates $\mathbf{u}_i$ in camera i to coordinates $\mathbf{u}_j = \mathbf{H}_{i,j}\mathbf{u}_i$ in camera j.

2.2 Image Alignment

Image alignment, the process of moving and deforming an image to match another image as closely as possible, dates back to the Lucas-Kanade algorithm [Lucas and Kanade, 1981]. The translational model of [Lucas and Kanade, 1981] was subsequently generalized to a variety of other warps in [Bergen *et al.*, 1992]. See [Baker and Matthews, 2004] for a recent overview. Image alignment consists of minimizing an expression like:

$$\sum_{\mathbf{u}_i} [I_i(\mathbf{u}_i) - I_j(\mathbf{H}_{i,j}\mathbf{u}_i)]^2 \tag{7}$$

with respect to the parameters of the homography $\mathbf{H}_{i,j}$. Suppose the parameters are stored in the parameter vector $\mathbf{h}$. The Lucas-Kanade (forwards additive [Baker and Matthews, 2004]) algorithm to minimize the image alignment goal in Equation (7) then consists of iteratively computing updates to the parameters:

$$\Delta\mathbf{h} \;=\; \mathrm{Hess}^{-1} \sum_{\mathbf{u}_i} \left[\nabla I_j \frac{\partial \mathbf{H}_{i,j}}{\partial \mathbf{h}}\right]^{\mathrm{T}} [I_i(\mathbf{u}_i) - I_j(\mathbf{H}_{i,j}\mathbf{u}_i)] \tag{8}$$

where Hess is the Hessian matrix:

$$\mathrm{Hess} \;=\; \sum_{\mathbf{u}_i} \left[\nabla I_j \frac{\partial \mathbf{H}_{i,j}}{\partial \mathbf{h}}\right]^{\mathrm{T}} \left[\nabla I_j \frac{\partial \mathbf{H}_{i,j}}{\partial \mathbf{h}}\right] \tag{9}$$

and then updating the parameters appropriately. The expression $\frac{\partial \mathbf{H}_{i,j}}{\partial \mathbf{h}}$ is the Jacobian of the homography. The Jacobian consists of a pair of images, (1 for the u direction and 1 for the v direction) for each parameter in $\mathbf{h}$. The expression $\nabla I_j \frac{\partial \mathbf{H}_{i,j}}{\partial \mathbf{h}}$ consists of an image for each parameter in $\mathbf{h}$. The images in $\nabla I_j \frac{\partial \mathbf{H}_{i,j}}{\partial \mathbf{h}}$ are known as the steepest-descent images for the reasons described in [Baker and Matthews, 2004].

3 Parameterizations

The subject of this paper is the estimation of the plane equation $\mathbf{n}$ using homographies. The most straightforward 2-step approach can be summarized as followed:

1. Use image alignment to estimate the homography $\mathbf{H}_{i,j}$. Any of the algorithms in [Baker and Matthews, 2004] could be used.

2. Estimate the plane equation $\mathbf{n}$ from $\mathbf{H}_{i,j}$. For example, the algorithm in [Faugeras and Lustman, 1988] could be used.

The second of these steps requires knowledge of the intrinsic camera parameters $\mathbf{K}_i$ and $\mathbf{K}_j$. The homography $\mathbf{H}_{i,j}$ only has 8 free variables because it defined up to scale. The plane equation $\mathbf{n}$ has 3, the relative orientation of the cameras has 3, and the translation between the cameras also has 3. Even if we know the intrinsic camera parameters $\mathbf{K}_i$ and $\mathbf{K}_j$, the best we can hope for is to estimate is the relative orientation of the cameras (3 unknowns), the plane equation $\mathbf{n}$ (3 unknowns), and the direction of translation (2 unknowns). If we define the distance between the cameras to be 1 unit (plane estimation is not possible unless there is a non-zero translation), the other quantities (i.e. the distance to the plane) can be recovered in those units.

In summary, recovery of the plane equation is only possible in Step 2 if we know the intrinsic calibration of the cameras $\mathbf{K}_i$ and $\mathbf{K}_j$ (although homography estimation does not.) Since the focus of this paper is plane estimation, we assume that $\mathbf{K}_i$ and $\mathbf{K}_j$ are known in the rest of this paper.

3.1 Unknown Relative Orientation

In this section we assume that the relative orientations and translations (or ego-motion) of the cameras ($\mathbf{R}_i$, $\mathbf{R}_j$, $\mathbf{T}_i$, and $\mathbf{T}_j$) are unknown. This scenario corresponds to: (1) A single camera moving along an unknown path through a rigid scene and capturing 2 images, (2) 2 cameras capturing images of a static scene, or (3) 2 synchronized cameras capturing images of a dynamic scene at the same time.

3.1.1 3x3 Parameterization

The most straightforward approach is to use the 2 step algorithm described above. In particular, the homography is usually parameterized with the elements of the 3×3 matrix:

$$\mathbf{H}_{i,j} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}. \tag{10}$$

There are a number of ways of dealing with the fact that the 3×3 matrix has 9 elements, but the homography only has 8 free parameters. One simple approach is to hard-code $h_9 = 1$. An almost equivalent approach taken in [Shum and Szeliski, 2000] is to maintain an estimate of the full 3×3 matrix, but only compute an 8 parameters update to the 3×3 matrix as follows:

$$\begin{pmatrix} 1+\Delta h_1 & \Delta h_2 & \Delta h_3 \\ \Delta h_4 & 1+\Delta h_5 & \Delta h_6 \\ \Delta h_7 & \Delta h_8 & 1 \end{pmatrix} \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \tag{11}$$

Here, we take the first approach of hard-coding $h_9 = 1$ (i.e. set $\mathbf{h} = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8)^{\mathrm{T}}$) because for the homographies in our experiments h_9 is never close to 0.

Once $\mathbf{H}_{i,j}$ has been computed, the plane equation can be computed by solving Equation (6). In [Faugeras and Lustman, 1988] a closed-form solution to this equation is presented. In this paper, we solve Equation (6) using a Gauss-Newton non-linear optimization algorithm. In the experiments in Section 4 we ensure that this algorithm is initialized close to the correct answer, and check that the algorithm does not diverge wildly.

Since we can only solve for the relative orientation and translation, we set $\mathbf{R}_i = \mathbf{I}$, $\mathbf{T}_i = \mathbf{0}$ and just solve for $\mathbf{R}_j$ and $\mathbf{T}_j$. As mentioned above, it is still not possible to estimate all 9 unknowns (3 in $\mathbf{n}$, 3 in $\mathbf{R}_j$, 3 in $\mathbf{T}_j$) from the 8 non-linear constraints. We define the distance between the cameras to be 1 unit and estimate the other parameters (i.e. the distance to the plane) in terms

of those units. We optimize for the 3 free parameters of $\mathbf{n}$, the 3 free parameters of $\mathbf{R}$, and the 2 free parameters of $\mathbf{T}$ using the approach in [Shum and Szeliski, 2000]; i.e. we maintain full representations of $\mathbf{n}$ (4 numbers), $\mathbf{R}$ (9 numbers), and $\mathbf{T}$ (3 numbers), but only solve for 3, 3, and 2 parameter updates respectively. The updates to $\mathbf{n}$ consist of 2 rotations of the plane normal (n_x, n_y, n_z) and an additive update to n_d. The update to $\mathbf{R}$ consists of 3 rotations and the update to $\mathbf{T}$ 2 rotations.

3.1.2 4pt Parameterization

Equation (10) is not the only way to parameterize a homography. Another way is to parameterize it in terms of 4 pairs of corresponding points in the 2 images. Suppose that $\mathbf{u}_i^k = (u_i^k, v_i^k, 1)^{\mathrm{T}}$ for $k = 1, 2, 3, 4$ are 4 *fixed* points in image i and $\mathbf{u}_j^k = (u_j^k, v_j^k, 1)^{\mathrm{T}}$ are 4 *variable* points in image j such that:

$$\mathbf{H}_{i,j}\mathbf{u}_i^k \;=\; \mathbf{u}_j^k. \tag{12}$$

The 8 parameters $\mathbf{h} = (u_j^1, v_j^1, u_j^2, v_j^2, u_j^3, v_j^3, u_j^4, v_j^4)^{\mathrm{T}}$ can then be used to parameterize the homography. It is relatively simple to interchange between the 3×3 parameterization and the 4 point parameterization, to compute the Jacobian $\frac{\partial \mathbf{H}_{i,j}}{\partial \mathbf{h}}$, and to update the warp appropriately [Baker and Matthews, 2004].

Once the 8 parameters $(u_j^1, v_j^1, u_j^2, v_j^2, u_j^3, v_j^3, u_j^4, v_j^4)^{\mathrm{T}}$ have been computed, the plane equation can then be computed by solving the following set of equations:

$$\mathbf{P}_j \left[\left(\mathbf{n}^{\mathrm{T}} \mathbf{p}_i \right) \mathbf{I} - \mathbf{p}_i \mathbf{n}^{\mathrm{T}} \right] \mathbf{P}_i^* \mathbf{u}_i^k \;=\; \mathbf{u}_j^k, \tag{13}$$

where $k = 1, 2, 3, 4$. Each of these equations, once scale-normalized, provides 2 constraints. As above, we solve the 8 constraints for the 8 unknowns of $\mathbf{n}$ (3 unknowns), $\mathbf{R}_j$ (3 unknowns), and the direction of $\mathbf{T}_j$ (2 unknowns) using the same Gauss-Newton non-linear optimization algorithm.

3.1.3 Direct nRT Parameterization

The above parameterizations both require 2 steps: (1) compute the parameters of the homography using image alignment, and (2) solve the for plane equation (plus rotation and direction of the translation.) An alternative is to parameterize the homography directly with the plane equation $\mathbf{n}$, the rotation matrix $\mathbf{R}_j$, and the direction of the translation $\mathbf{T}_j$; i.e. set $\mathbf{h} = (\mathbf{n}, \mathbf{R}_j, \mathbf{T}_j)^{\mathrm{T}}$. The plane equation can then be computed in a single step by performing the image alignment of Equations (7–9), the so called "direct method" [Negahdaripour and Horn, 1987]. Intuitively, it might be argued that this nRT parameterization gives a 1 step algorithm and so should perform the best. In Section 4 we perform experiments to test this idea.

3.2 The Fully Calibrated Case

In this section we assume that the relative orientations and translations of the cameras ($\mathbf{R}_i$, $\mathbf{R}_j$, $\mathbf{T}_i$, and $\mathbf{T}_j$) are known; i.e. have been calibrated. This scenario corresponds to a pair of synchronized calibrated cameras simultaneously capturing images of a dynamic scene, or 2 unsynchronized calibrated cameras capturing images of a static scene.

3.2.1 3x3 Parameterization

The 3×3 parameterization of the homography can still be used in the calibrated setting when using image alignment to estimate the homography in Step 1. In Step 2, Equation (6) can also still be used to compute $\mathbf{n}$. Since $\mathbf{R}_i$, $\mathbf{T}_i$, $\mathbf{R}_j$, and $\mathbf{T}_j$ are known, however, the constraints on $\mathbf{n}$ in Equation (6) are linear. We solve them with (unweighted) least squares.

3.2.2 4pt Parameterization

The 4pt parameterization of the homography can still be used in the calibrated setting. Equation (13) can also still be used to compute $\mathbf{n}$. Again, since $\mathbf{R}_i$, $\mathbf{T}_i$, $\mathbf{R}_j$, and $\mathbf{T}_j$ are known, Equation (13) provides a (different) set of linear constraints on $\mathbf{n}$ which we solve with least squares.

3.2.3 Direct n Parameterization: Layered Stereo

In the fully calibrated case, it is also possible to parameterize the homography directly using $\mathbf{n}$, $\mathbf{R}_j$, and $\mathbf{T}_j$. Since $\mathbf{R}_i$, $\mathbf{T}_i$, $\mathbf{R}_j$, and $\mathbf{T}_j$ are known, however, the only unknowns are the 3 parameters of $\mathbf{n}$. The resulting direct algorithm for estimating $\mathbf{n}$ using image alignment is the "Layered Stereo" algorithm of [Baker *et al.*, 1998]. In [Baker *et al.*, 1998] it is argued that using this "n-parameterization" is better than using any other because only 3 unknown parameters need to be solved for. No empirical validation of this claim was presented however. In Section 4 we empirically investigate this claim.

3.3 Moving Stereo Rig

Suppose now that the synchronized, fully calibrated, stereo rig of Section 3.2 captures 2 pairs of images as either the scene, the stereo rig, or both move.

3.3.1 Independent n Parameterizations

The simplest approach to dealing with the 2 time steps is to parameterize the problem independently at each time. For example, the direct parameterization of the homography using the plane equation $\mathbf{n}$ (Section 3.2.3) could be used separately at each time instant, in total 6 parameters.

3.3.2 nRT Parameterization: Layered Motion-Stereo

A more sophisticated approach is to add additional parameters, in particular, the single global 3D rotation $\mathbf{R}$ and translation $\mathbf{T}$ of the stereo rig. The effect of this global motion can then be used to update the camera matrices:

$$\mathbf{P}_i = \mathbf{K}_i \left[\mathbf{R}_i \, \mathbf{T}_i\right] \;\rightarrow\; \mathbf{P}_{i'} = \mathbf{K}_i \left[\mathbf{R}_{i'} \, \mathbf{T}_{i'}\right] \tag{14}$$

where $\mathbf{R}_{i'} = \mathbf{R}_i\mathbf{R}$ and $\mathbf{T}_{i'} = \mathbf{T}_i - \mathbf{R}_i^{-1}\mathbf{T}$. It is then possible to define 6 homographies (and their inverses) between the 2 pairs of images:

$$\mathbf{H}_{i,j}, \mathbf{H}_{i',j'}, \mathbf{H}_{i,i'}, \mathbf{H}_{j,j'}, \mathbf{H}_{i,j'}, \mathbf{H}_{i',j} \tag{15}$$

compared to only the 2 ($\mathbf{H}_{i,j}$ and $\mathbf{H}_{i',j'}$) that are possible without introducing the global rotation $\mathbf{R}$ and translation $\mathbf{T}$. These homographies can then be used to set up a single large image alignment problem to simultaneously estimate $\mathbf{n}$, $\mathbf{R}$, and $\mathbf{T}$. We refer to this approach as Layered Motion-Stereo, a generalization of Layered-Stereo [Baker *et al.*, 1998].

There are then 2 major possible benefits of using Layered Motion-Stereo once over using Layered Stereo independently at the 2 times instants: (1) 3 times as many image constraints can be used in the direct estimation of the parameters using image alignment, and (2) the global rotation $\mathbf{R}$ and translation $\mathbf{T}$ can be useful parameters to estimate in their own right. See Section 4.2. On the other hand, (1) 9 parameters must be estimated (3 in $\mathbf{n}$, 3 in $\mathbf{R}$, and 3[1] in $\mathbf{T}$) compared to 6 for the independent parameterization (3 for $\mathbf{n}$ at the first time instant and 3 at the second time instant), and (2) the computational cost of the higher number of parameters and higher number of constraints is dramatically more. In the following section, we empirically compare these 2 parameterizations to see whether the additional constraints outweigh the larger number of parameters, or vica versa.

4 Experiments

4.1 Quantitative Evaluation

We now present the results of an empirical comparison of the various parameterizations. In this evaluation, the image I_i is a real image of a road scene, like that at the top of Figure 5. The other image(s) I_j, ($I_{i'}$, and $I_{j'}$) are generated synthetically from I_i using the ground-truth $\mathbf{H}_{i,j}$ (etc), generated from the ground-truth $\mathbf{n}$, $\mathbf{R}_j$, $\mathbf{T}_j$, etc. We conduct experiments to compute 2 measures

[1]It is possible to estimate all 9 parameters of $\mathbf{n}$, $\mathbf{R}$, and $\mathbf{T}$. Because there is more than 1 homography we are not limited to 8 constraints.

of performance:

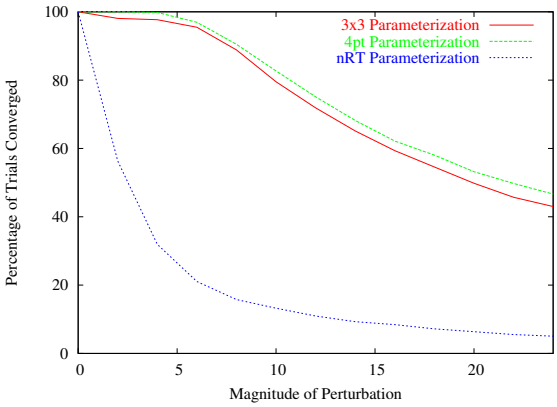
Robustness: We measure the robustness of image alignment (Step 1 in the 2-step algorithm) by perturbing the initial estimate of $\mathbf{n}$ by larger and larger amounts. For each magnitude of perturbation, we generate 2500 trials, and count the percentage of trials that converge.

Accuracy: We measure the accuracy of plane estimation (Step 2 in the 2-step algorithm) by adding more and more (white, Gaussian) noise to I_i and I_j. The initial estimate of $\mathbf{n}$ is also perturbed by a small amount, large enough that the algorithms have to update their parameters to get a reasonable estimate of $\mathbf{n}$, but small enough that the algorithms all converge most of the time. For each noise level, we compute RMS errors in the estimation of $\mathbf{n}$, checking that each of the 1000 trials actually converged. We compute two error measures: (1) the error in the estimate of the plane equation normal (n_x, n_y, n_z) measured in Radians and (2) the error in the estimate of n_z in meters. The ground-truth relative translation between the cameras is 0.38 meters and is used to set the scale of the results for n_z.

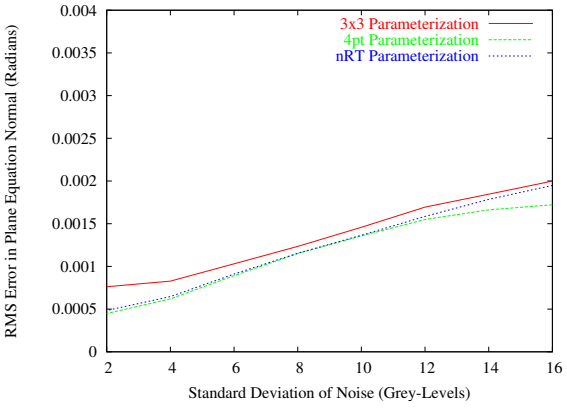
4.1.1 Unknown Relative Orientation

The results in the case of unknown relative orientation are included in Figure 1. The robustness results in Figure 1(a) show that the 4pt parameterization is the most robust. The 3×3 parameterization is next best, with the nRT parameterization having very poor robustness. The accuracy results in Figures 1(b–c) also show that the 4pt parameterization performs the best, followed by the nRT parameterization.

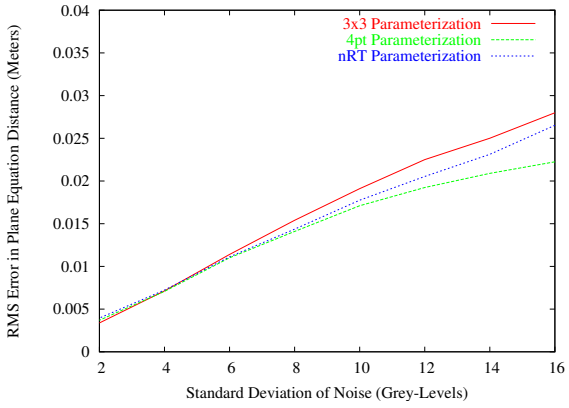
The poor performance of the nRT parameterization is not surprising. The cause is the well-known ambiguity between rotation and translation in ego-motion estimation [Longuet-Higgins, 1984, Adiv, 1989, Daniilidis and Nagel, 1993]. Previous authors have attempted to characterize the effect of this ambiguity on the image alignment problem by computing the conditional number of the Hessian matrix [Ke and Kanade, 2003]. The condition number alone, however, does not provide a good explanation of the performance in Figure 1. Although the 4pt parameterization



(a) Image Alignment Robustness



(b) Accuracy of (n_x, n_y, n_z) Estimation



(c) Accuracy of n_z Estimation

Figure 1: Unknown Relative Orientation: (a) The robustness results show that the 4pt parameterization is the best, followed by the 3×3 parameterization. The nRT parameterization has very poor robustness. See Figure 2 for an explanation. (b-c) The 4pt parameterization gives the most accurate estimates of $\mathbf{n}$ followed by the nRT parameterization and the 3×3 parameterization.

does have the best condition number, the condition number of the 3×3 parameterization is far, far worse than that of the nRT parameterization. The condition number of any of the parameterizations can also be changed by several orders of magnitude by changing the "units" of the parameters. We empirically verified that this change in units doesn't effect the performance significantly (so long as the numerical precision of the computer is exceeded.)

A better explanation is included in Figure 2. In this figure, we plot an independence measure of the Jacobians $\frac{\partial \mathbf{H}_{i,j}}{\partial \mathbf{h}}$ for the 3 difference parameterizations, a measure that is independent of the units. In particular, for each parameter in $\mathbf{h}$, we compute what fraction of it that lies in the subspace spanned by the other 7 parameters in $\mathbf{h}$. The results in Figure 2 show the strong coupling between

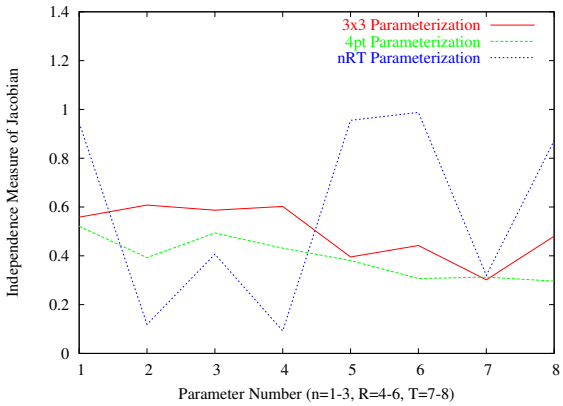


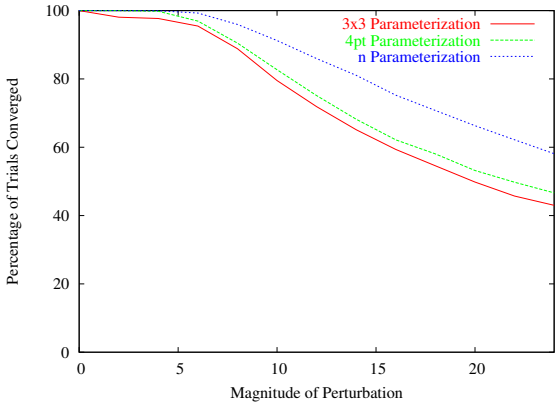
Figure 2: An explanation of the very poor performance of the nRT parameterization in Figure 1. For each parameter in $\mathbf{h}$, we compute what fraction of it lies in the subspace spanned by the other 7 parameters. The results show that there are 4 parameters (2 in $\mathbf{n}$, 1 in $\mathbf{R}_j$, and 1 in $\mathbf{T}$) that are not very independent of each other. This coupling is a manifestation of the well-known ambiguity between rotation and translation in ego-motion estimation [Longuet-Higgins, 1984, Adiv, 1989, Daniilidis and Nagel, 1993].

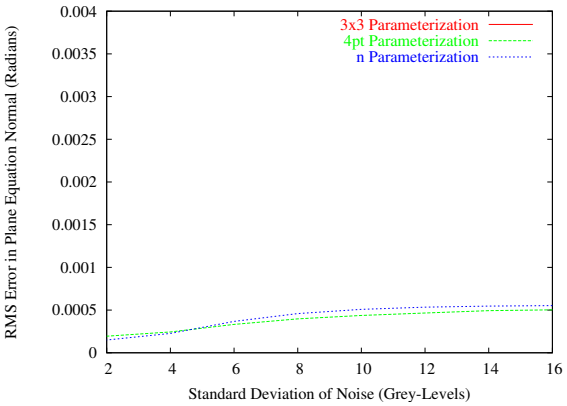
2 of the parameters in $\mathbf{n}$ (numbers 2 and 3) with 1 of the parameters in $\mathbf{R}$ (number 4), and 1 of the parameters in $\mathbf{T}$ (number 7).

4.1.2 The Fully Calibrated Case

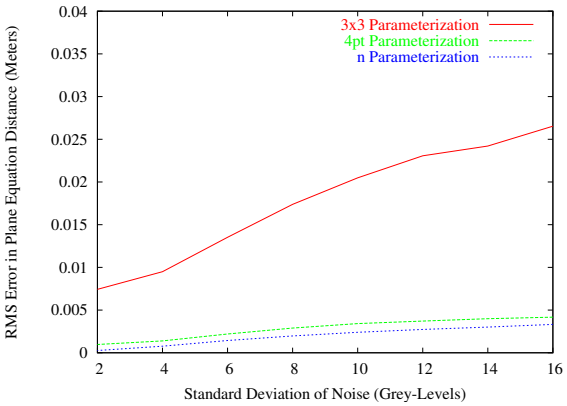
The results in the fully calibrated case are included in Figure 3. The robustness results in Figure 3(a) show that the n parameterization is the most robust. These results validate the claims in [Baker *et al.*, 1998] that just optimizing over the 3 free parameters in $\mathbf{n}$ is preferable to optimizing over the 8 parameters of the homography (either 4pt or 3×3.) The accuracy results in Figure 3(b–c) show that the n parameterization and the 4pt parameterization are similar, with the n parameterization perhaps being marginally better overall. The results for the 3×3 parameterization are far, far worse. The results for the angle of the road plane (n_x, n_y, n_z) are so poor, they don't fit on the same graph without changing the scale. (The scales of Figures 1(b), 3(b), and 4(b) are kept the same for easy comparison.) The results are around 2 orders of magnitude worse, at around 0.02-0.06. Solving Equation (6) using least squares is a very poor way of estimating $\mathbf{n}$.

(a) Image Alignment Robustness





(b) Accuracy of (n_x, n_y, n_z) Estimation

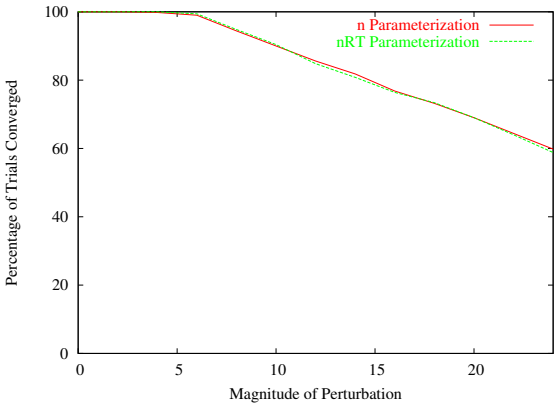


(c) Accuracy of n_z Estimation

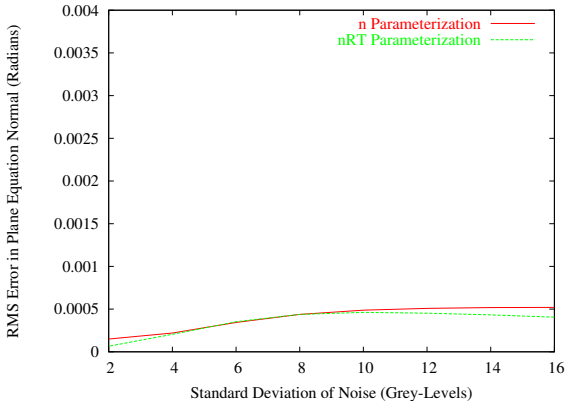
Figure 3: The Fully Calibrated Case: (a) The n parameterization is substantially more robust than the 4pt or 3×3 parameterizations, validating the claims in [Baker *et al.*, 1998]. (b–c) The accuracy of the 3×3 parameterization is far, far worse than that of the other 2 parameterizations. (b) The results for the angle of the road plane (n_x, n_y, n_z) are around 2 orders of magnitude worse, at around 0.02-0.06.

4.1.3 Moving Stereo Rig

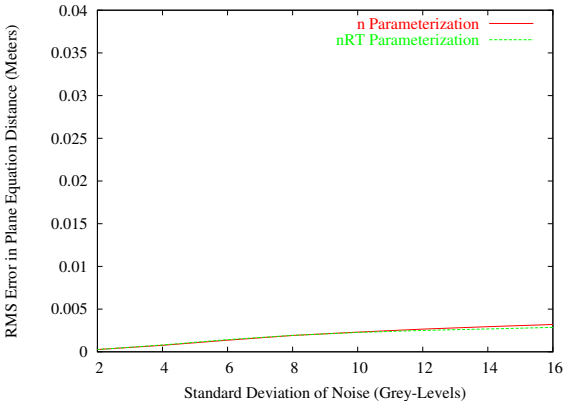
In the fully calibrated case, the robustness of the n parameterization with 3 parameters is far better than that of both the parameterizations with 8 parameters. In the case of a moving stereo rig (see Figure 4), the performance of the 2 parameterizations is almost identical in terms robustness. Although the nRT parameterization has 9 parameters and the n parameterization only 6 parameters (3 for each time instant), the nRT parameterization can use more image data. It appears that the extra data roughly balances the extra parameters. The accuracy of the nRT parameterization is, however, slightly better than that of the n parameterization.



(a) Image Alignment Robustness



(b) Accuracy of (n_x, n_y, n_z) Estimation



(c) Accuracy of n_z Estimation

Figure 4: Moving Stereo Rig: (a) The nRT and n parameterizations perform very similarly terms of robustness. The extra parameters in the nRT parameterization (9 parameters) when compared to the n parameterization (6 parameters, 3 at each time instant), appear to be roughly balanced by the extra image constraints that can be used. (b–c) The plane estimation accuracy of the nRT parameterization is slightly better than that of the n parameterization.

4.2 Qualitative Results

Our motivating example consists road plane estimation from cameras inside a car. This task is particularly difficult because of the lack of texture on the road. It is therefore important to use the best possible parameterization.

4.2.1 Top Down View Generation

Our first application is the generation of a "Top Down View" of the road. Example results for the 3×3 parameterization, the 4pt parameterization, and the n parameterization in the fully calibrated

Figure 5: Top Down View Generation: A comparison of the performance of the 3×3 parameterization, the 4pt parameterization, and the n parameterization in the fully calibrated setting. Each of the parameterizations is used to estimate the road plane, which is then used to generate a synthetic "Top Down View" by warping appropriately. The task is particularly difficult because of the lack of texture on the road and the oblique angle between the camera and the road plane. The image above is the first frame in an 800 frame movie. That movie clearly shows that the 3×3 parameterization performs poorly. The n parameterization performs, if anything, slightly better than the 4pt parameterization.

setting are included in Figure 5. The image in this figure is the first frame of an 800 frame movie. The movie shows that plane estimation using the 3×3 algorithm (left panel) is very inaccurate, verifying the results in Figure 3(b–c). The results for the 4pt parameterization (center panel) are far better. The movie clearly demonstrates that using Equation (13) for plane estimation is far better

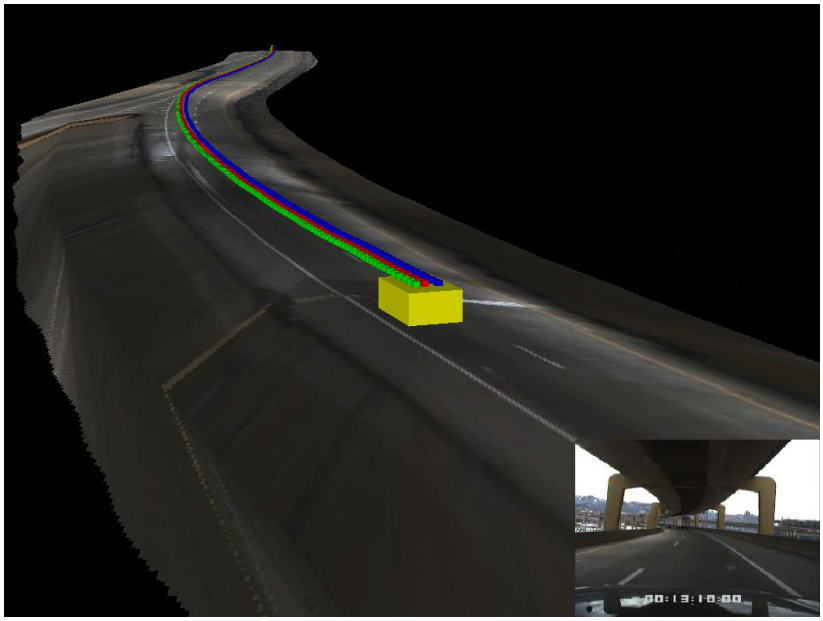


Figure 6: Road Mosaicking and Car Ego-Motion Estimation: An example frame from a movie illustrating the use of the nRT parameterization (Layered Motion-Stereo) to generate a mosaic of the road plane from the "Top Down Views" in Figure 5. The ego-motion of the cameras is displayed by rendering the 3 cameras across time as red, green, and blue cuboids.

than using Equation (6). The results for the n parameterization (right panel) are fairly similar to those for the 4pt parameterization.

4.2.2 Road Mosaicking and Car Ego-Motion

As mentioned in Section 3.3.2 one benefit of the nRT parameterization over the n parameterization in the case of a moving stereo rig is that the global motion of the stereo rig (in this case the motion of the stereo rig is the same as the ego-motion of the car) is computed. As illustrated in Figure 6, the computed ego-motion can be used to generate a mosaic of the road on the ground plane from the Top Down Views in Figure 5. The ego-motion of the cameras is also displayed.

5 Discussion

We have shown that the usual 3×3 parameterization of the homography is not the best. The 4pt parameterization performs better, both in terms of robustness and plane estimation accuracy. It is

an open question, however, whether this parameterization is the best. It is also an open question, as to what is the best choice for the fixed parameters $(u_i^1, v_i^1, u_i^2, v_i^2, u_i^3, v_i^3, u_i^4, v_i^4)^{\mathrm{T}}$. Making progress on such questions requires a way of quantifying how "good" a parameterization is. In Figure 2 we proposed the independence of the Jacobian images as such a measure. Previous authors have proposed the conditional number of the Hessian matrix [Ke and Kanade, 2003] instead. The independence of the Jacobian images has two main advantages over the condition number of the Hessian matrix: (1) it is independent of units of the parameters; i.e. scaling the parameters by an arbitrary amount leaves the measure unchanged, and (2) it is independent of the (texture in the) input images; it is therefore a measure of the parameterization alone. The independence of the Jacobian images is probably not the best such measure, however. The relative performance of the 3×3 and 4pt parameterizations is not explained by Figure 2. The formulation of a better measure is left as future work.

The case of unknown relative orientation corresponds to the well studied problem of planar ego-motion estimation [Longuet-Higgins, 1984, Negahdaripour and Horn, 1987]. Even though there are well-known ambiguities in the nRT parameterization [Longuet-Higgins, 1984, Adiv, 1989, Daniilidis and Nagel, 1993], it is sometimes claimed that the direct (1 step) approach [Negahdaripour and Horn, 1987] is better than the 2 step approach of first estimating the homography and then estimating the plane equation $\mathbf{n}$ and ego-motion $\mathbf{R}$, $\mathbf{T}$. Our empirical evaluation of the accuracy measure in Figures 1(b–c) is only partially consistent with this thinking. The direct nRT parameterization outperforms the 3×3 parameterization. However, this argument ignores: (1) the robustness measure and (2) the fact that the 3×3 parameterization of the homography is not the only possibility. The rest of the results in Figure 1 indicate that the problem of planar ego-motion estimation may be one of the rare cases where a 2-step algorithm may be better than a 1-step (direct) algorithm. Splitting the algorithm into 2 allows the better parameterizations of the homography (such as the 4pt parameterization) in the difficult image alignment step. The well-known ambiguities in the nRT parameterization [Longuet-Higgins, 1984, Adiv, 1989, Daniilidis and Nagel, 1993] are thereby avoided during image alignment. The second step of

solving for the the plane equation $\mathbf{n}$ and ego-motion $\mathbf{R}$, $\mathbf{T}$ can be performed without any of the difficulties caused by low texture in the images using Equation (13).

In the fully calibrated case, we empirically validated the claims in [Baker *et al.*, 1998] that the n parameterization (3 parameters) is more robust than using an 8 parameter homography. However, the accuracy of the n parameterization is only slightly better than that of the 4pt parameterization. We also generalized "Layered Stereo" [Baker *et al.*, 1998] to "Layered Motion-Stereo" in the case of a moving stereo rig. Our quantitative results show that the robustness of the 9 parameter nRT parameterization is very similar to the $3 \times 2 = 6$ parameter n parameterization, performed independently at each of the 2 time instants. The additional image constraints appear to roughly balance the extra number of parameters to be solved for. The accuracy of the nRT parameterization is significantly better, however. As a practical matter, note that the ego-motion ambiguities for a single camera [Longuet-Higgins, 1984, Adiv, 1989, Daniilidis and Nagel, 1993] do not appear in the case of a moving stereo rig. In automotive applications like "Top Down View" generation, road mosaicking, and car ego-motion estimation, it is probably worthwhile using a calibrated stereo rig.

Acknowledgments

The research described in this paper was supported by DENSO Corporation.

References

[Adiv, 1989] G. Adiv. Inherent ambiguities in recovering 3-D motion and structure from a noise flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5), 1989.

[Baker and Matthews, 2004] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221 – 255, 2004.

[Baker *et al.*, 1998] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[Bergen *et al.*, 1992] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision*, 1992.

[Daniilidis and Nagel, 1993] K. Daniilidis and H. Nagel. The coupling of rotation and translation in motion estimation of planar surfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1993.

[Faugeras and Lustman, 1988] O.D. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.

[Faugeras, 1993] O.D. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.

[Ke and Kanade, 2003] Q. Ke and T. Kanade. Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[Longuet-Higgins, 1984] H.C. Longuet-Higgins. The visual ambiguity of a moving plane. *Proceedings of the Royal Society of London B*, 223, 1984.

[Lucas and Kanade, 1981] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artifical Intelligence*, 1981.

[Negahdaripour and Horn, 1987] S. Negahdaripour and B.K.P. Horn. Direct passive navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):168–176, 1987.

［Shum and Szeliski, 2000］H.-Y. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 16(1):63–84, 2000.