Motion From Structure: Robust Multi-Image, Multi-Object Pose Estimation

By

John Flynn

Video and Image Processing Lab
Department of Electrical Engineering and Computer Science
Univ. of California/Berkeley

Motion From Structure: Robust Multi-Image, Multi-Object Pose Estimation

By

John Flynn

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Avideh Zakhor
Research Advisor

(Date)

* * * * * * *

Professor David Forsyth
Second Reader

(Date)

ABSTRACT

We propose a robust algorithm for tracking camera motion in unconstrained urban environments. We achieve this by using both an attached laser scanner, which gives sparse 3D structure for each frame, and image-based correspondences from a video camera. The combination of a set of 3D points and their associated projections in each frame can be used to find the motion between frames. We propose a new robust multi-object, multi-image pose estimation algorithm and show how to use this algorithm to accurately track the motion of a moving vehicle with an attached camera and 2D laser scanners. Our algorithm is based on a RANSAC formulation and is thus robust to incorrect 3D data and image occlusions, as evinced by tests using synthetic and real data.

1. INTRODUCTION

There is a growing demand for three-dimensional (3D) models of urban environments for many applications, including urban planning, virtual reality and propagation simulation of radio waves for the cell phone industry. Currently, acquisition of 3D city models is difficult and time consuming. At the Video and Image Processing Lab, we have developed a system for fast model acquisition. The system consists of laser range scanners and a camera, mounted on a moving vehicle. However, the accuracy of the resulting models hinges on recovering the vehicle's position and orientation accurately at all times. In this paper we propose a new pose estimation technique that uses images and laser scan data to accurately and robustly estimate the pose.

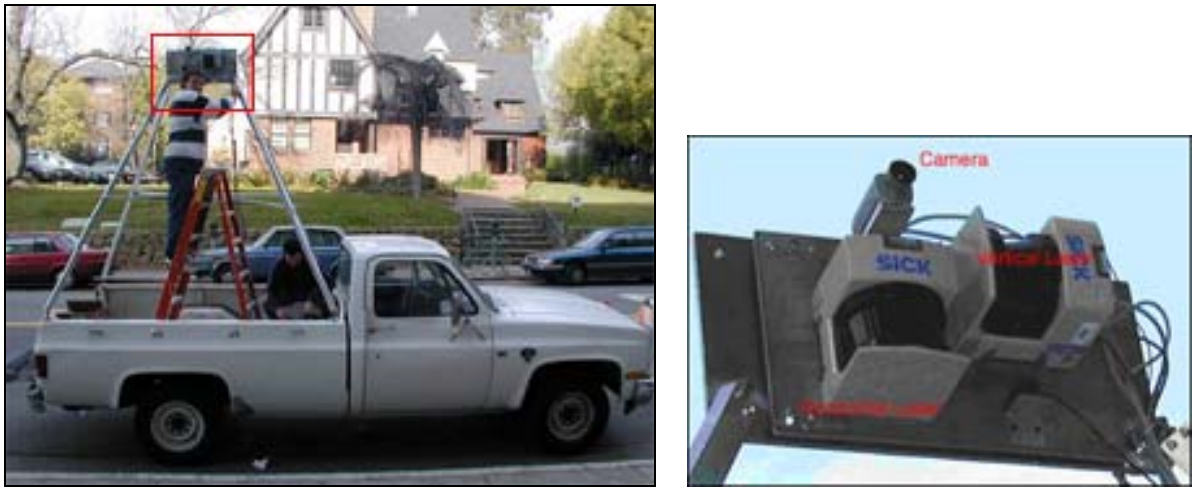
Before discussing the pose estimation algorithm, we begin in Section 2 with a description of the acquisition system. In Section 3 we discuss laser *scan-matching*, which is another technique for estimating the pose, and can also be used as an input to the proposed algorithm. In Section 4 we discuss camera calibration, which relates 3D points in the world to their projections in an image, and is thus a necessary prerequisite to our algorithm. Section 5 presents the proposed pose estimation technique, while Section 6 presents results on both real and synthetic data. Section 7 concludes the paper and discusses possible improvements and extensions.

2. THE CITY MODELING PROJECT

2.1 Introduction

There is a growing demand for three-dimensional (3D) models of urban environments for many applications, including urban planning, virtual reality and propagation simulation of radio waves for the cell phone industry. Currently, acquisition of 3D city models is difficult and time consuming. Commercially available models typically take months to create and usually require significant manual intervention. This process not only results in high costs inhibiting broad use of the models, but also makes it impossible to use them for applications where the goal is to monitor changes over time, such as detecting damage or possible danger zones caused by catastrophes such as earthquakes, land slides, and hurricanes.

As part of the city-modelling project at the Video and Image Processing Lab, we have developed an experimental set-up set up that is capable of rapidly acquiring 3D and texture data of entire streets from the ground level by using two fast 2D laser scanners and a digital camera. The apparatus consists of horizontal and vertical 2D laser scanners and a video camera, mounted on a truck as shown in Figure 1. A PC controls the instruments and stores the data as it is captured. The scanners capture the geometry of the scene and the camera captures the textural information.



(a) *(b)*

Figure 1: The data acquisition. (a) Shows the vehicle used for data acquisition. The region within the red box is the sensor plate. (b) Shows a close-up of the sensor plate.

2.2 System Overview

We will only present a brief overview of the system here; a more detailed description is available in [5].

The data acquisition system can be divided into two parts: a sensor module and a processing unit. The sensor module, shown in Figure 1(b), consists of two 2D laser scanners mounted with their scanning planes at 90 degrees and a digital camera; the processing unit consists of a dual processor PC, large hard disk drives, and additional electronics for power supply and signal shaping.

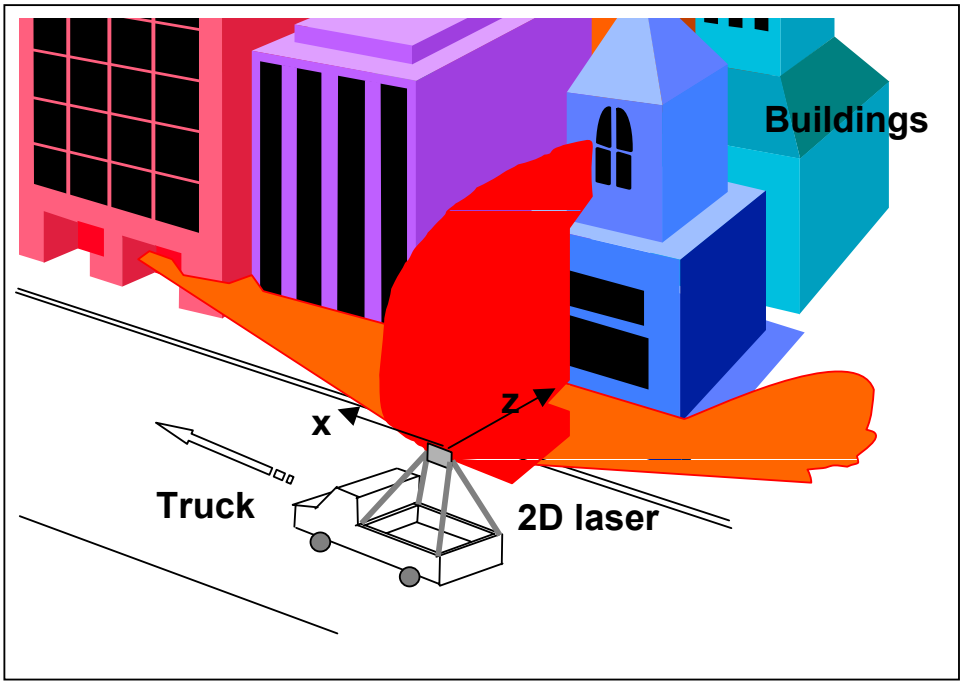


Figure 2: The model acquisition system

In order to avoid obstacles such as cars and pedestrians, we mount the sensor module on a rack, so that it is at a height of approximately 3.6 meters. The processing unit is mounted on the truck bed. The scanners have a 180^0 field of view with a resolution of 1^0, a range of 80 meters and an accuracy of ±6 centimetres; the acquisition time for a scan is 6.7 milliseconds. Scans are captured at a rate of 75Hz. Both 2D scanners are facing the same side of the street; one is mounted vertically with the scanning plane orthogonal to the direction of the vehicle's motion, the other is mounted horizontally with the scanning plane parallel to the ground, as shown in Figure 2. While the vehicle is moving, the vertical scanner captures the shape of the building facades, whereas the horizontal scanner measures the shape in a plane parallel to the ground, and is used for position estimation as described later.

The camera is mounted with the scanners, with its line of sight approximately parallel to the intersection between the orthogonal scanning planes. The laser scanners and camera are synchronized by a common trigger signal, this means that every camera image is captured at exactly the same time as both a horizontal and vertical laser scan. The camera is triggered every 15 scans, thus images are captured at a rate of 5Hz, this means at a driving speed of about 30km/h, there is approximately 1.8m between successive images. Since the camera and laser scanners are triggered by the same signal, the transformation between the laser scan points and the camera's coordinate frame is fixed.

The advantage of our model acquisition set-up is rapid scene capture, but the accuracy of the captured data hinges on finding the correct position and orientation of the vehicle in space. If the position and orientation is not known accurately then captured 3D points will be placed in incorrect positions in the world. Sensors such as GPS would not provide the accuracy needed, and are known to fail in urban environments. We use a combination of laser scan matching and the vision-based method outlined in this paper to overcome the pose estimation problem. In the next section we describe the scan-matching algorithm, and discuss some of its limitations.

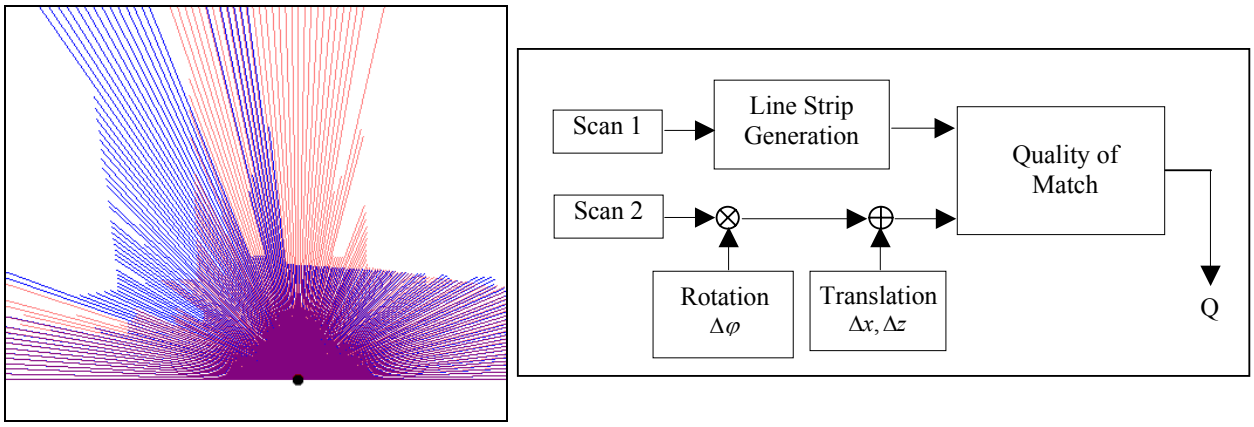
3. POSITION ESTIMATION USING SCAN MATCHING

Scan matching is used to estimate how the acquisition vehicle moves in space. Although the scan matching algorithm was developed in [5], it is worth reviewing here, since it is an alternative pose estimation method, albeit one with some limitations. In addition the estimated pose from scan matching is used as an initialisation for the proposed algorithm.

3.1 The Scan-Matching Algorithm

The central idea behind the scan-matching algorithm is that nearby horizontal scans have significant overlap. This means that a scan, $S_{t+\Delta t}$, taken at time $t+\Delta t$, measures many of the same features as the scan S_t taken at time t,as can be seen in Figure 3(a). However in $S_{t+\Delta t}$, these features will have been observed from a different position and orientation in space. If it is assumed that the vehicle only moves within the ground plane then the change in position and orientation between the two scans can be represented by a set of three parameters, a translation, Δx,

Δz and a rotation $\Delta\varphi$. The three parameters Δx, Δz, $\Delta\varphi$ can be estimated by maximising the alignment of scans S_t and $S_{t+\Delta t}$.



(a) *(b)*

Figure 3: Illustration of Scan Matching. (a) Two nearby scans have a high degree of overlap, this can be exploited to find the rotation and translation between nearby scans. (b) Shows the flow chart for the scan-matching algorithm.

The flow chart for the scan-matching algorithm is shown in Figure 3(b). To measure the alignment between scans we connect the scan points of S_t with line segments, apply the rotation and translation to $S_{t+\Delta t}$, and measure how close the transformed scan points are to the line segments of S_t. Of course, even at the correct set of parameters, there will be some outliers that are not close to any line segments; this is overcome by using a robust cost function that downweights the importance of outliers. The maximisation is achieved by searching the parameter space from a coarse to fine scale. Finally, steepest descent is used to further maximise the alignment. Successive $\Delta x, \Delta z, \Delta\varphi$ are then summed to arrive at a complete path for the vehicle.

3.2 Limitations of Scan Matching

Scan matching results in accurate motion estimates when the motion is in plane, i.e. on flat ground with no pitching or rolling of the truck, but cannot capture the small bumps and rolls, the out of plane motion, that are inevitable on a real vehicle, on real terrain. For the same reasons, large out of plane motions, for example when the vehicle drives up a hill, are also impossible to detect using scan-matching. These motions are important as they affect texture mapping of the resulting 3D models and affect all future processing of the 3D and intensity data. For stereo reconstruction and image based rendering applications local pose accuracy is very important [16]. The full 3D position and orientation of the truck at all times is thus necessary for accurate models and is a prerequisite for further processing. The central idea in this thesis is the development of a pose estimation technique that uses the images and the scans to compute the full 3D pose.

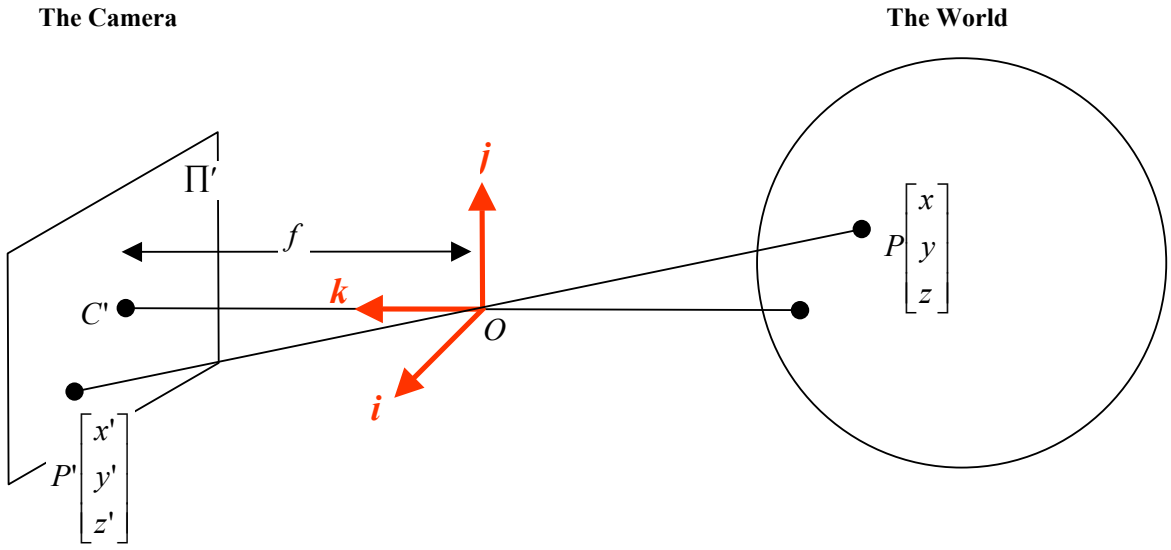
In the next section we discuss camera calibration, which relates 3D points in the world to their projections in an image.

4. CAMERA CALIBRATION

In order to relate the images to the 3D points from the laser scanners the camera first needs to be calibrated. Camera calibration is necessary for texture mapping, our pose estimation algorithm and for general stereo. In this section we discuss our camera model and how we estimate its parameters.

4.1 The Camera Model

The relationship between 3D points in the world and their projections in a camera image is determined by the *camera model*. Our camera model is a pinhole model. The relationship between a point in the world and its projection is easily derived by referring to Figure 4, which is a diagram of the imaging process, assuming the pinhole model.



(a) *(b)*

Figure 4: The Imaging Process

O is the origin of the coordinate system, $\prod'$ is the camera projection plane, at $z = f$, C' is the camera centre of projection and f is the focal length. P is a point in the world and P' is its projection in the image.

From similar triangles:

$$x' = \lambda x \tag{4.1}$$

$$y' = \lambda y \tag{4.2}$$

$$f = -\lambda z \Rightarrow \lambda = -f/z \tag{4.3}$$

$$\Rightarrow x' = -f/z \; x \tag{4.4}$$

$$\Rightarrow y' = -f/z \; y \tag{4.5}$$

There will also be some translation and scaling to convert to actual *u,v* pixel coordinates. The pinhole camera may be specified completely by the *intrinsic camera matrix*, *A*, which is specified by 6 parameters:

$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

$$\alpha = -fk_u \tag{4.7}$$

$$\beta = -fk_v \tag{4.8}$$

k_u, k_v represent the effective number of pixels per centimeter along the *u* and *v* axes. λ is the skewness factor that accounts for the fact that the camera axes may not be orthogonal, it is usually close to zero for real cameras. u_0, v_0 are the pixel coordinates of the principle point, u_0, v_0 are usually very close to the centre of the image.

4.2 Radial Distortion

Real cameras are not perfect pinhole cameras. In particular, radial distortion, which causes straight lines to appear curved, is a significant effect that is not explained by the simple pinhole camera model. Radial distortion is a linear displacement of image points radially to or from the centre of the image, caused by the fact that objects are at different angular distances from the lens undergo different magnifications. Our camera has a very wide angle lens so radial distortion is particularly significant as can be seen in Figure 6(a). Fortunately, radial distortion can be easily modelled and compensated for.

Radial distortion displaces points in the following way, assuming a second order model:

$$\breve{x} = x + x(k_1 r^2 + k_2 r^4) \tag{4.9}$$

$$\breve{y} = y + y(k_1 r^2 + k_2 r^4) \tag{4.10}$$

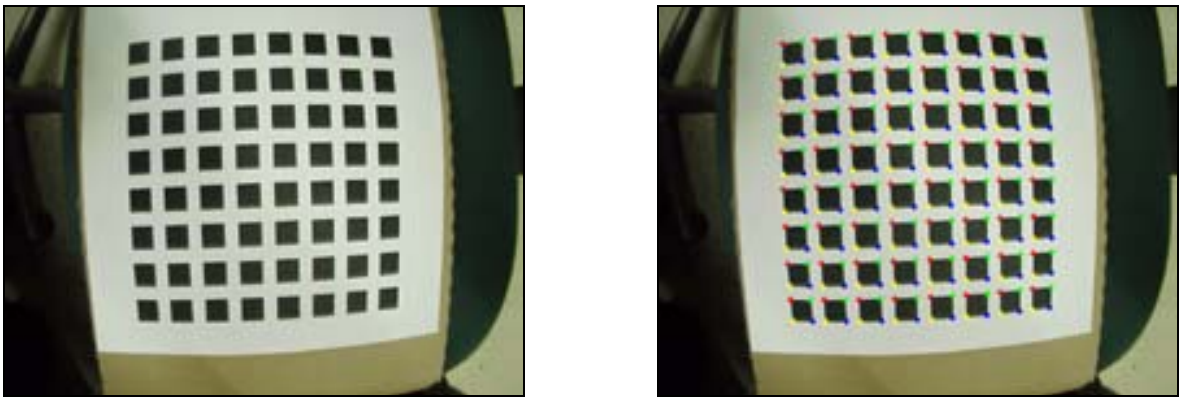
k_1, k_2 are the parameters of the radial distortion, if they are known then the radial distortion can be effectively reversed. r is the distance from the principle point. The coordinates of the principle point is thus an implicit parameter of the radial distortion equations.

4.3 Estimating the Camera Parameters

Camera calibration is the process of determining the parameters of the camera model. Since our model is the pinhole model with radial distortion we need to find both the parameters of the intrinsic camera matrix and the parameters for radial distortion.

To estimate both sets of parameters we use a tool developed in [17]. Although the complete details are beyond the scope of this thesis, the basic idea is that if a camera views a planar object in two different positions and/or rotations then a *homography* can be found that relates points on the plane in the first view to points in the second view. This homography is a function of both the relative rotation and translation between the two views and the intrinsic camera matrix *A*. Each image that views the plane imposes additional constraints on *A*. If enough views are used then *A* is completely determined, and can be found by solving a set of linear equations. If more than enough views are used then *A* is over-determined and a least squares solution can be found. Of course, the effect of radial distortion must be removed before computing the homography, so the parameters of radial distortion must be found also. These can be found by minimising the pixel error induced by the homography. The minimisation requires a starting point close to the minimum, so *A* is first found analytically assuming no radial distortion and this is used as the starting point.

To ease detecting points on the plane, a calibration pattern is used. The pattern we use is a black and white checkboard pattern, as shown in Figure 5(a). The corners of the square are used as points on the plane. We have developed a simple algorithm that detects the corners of the squares in an image. Firstly, black squares are detected using simple image processing techniques. This gives a rough estimate of the position of the corners. Their position is further refined to sub-pixel accuracy using a corner detector.



(a) *(b)*

Figure 5: The pattern used for camera calibration. The original pattern is shown in (a), the detected corners are shown in (b)

These corners were then use as input to the calibration algorithm described above. The results for distortion removal are shown below in Figure 6. The image on the left shows significant radial distortion, the image on the right shows the same image after the effect of the radial distortion has been removed. In the original image straight lines are curved, after correction the lines are straight.

(a) *(b)*

Figure 6: The results of camera calibration. The image in (a) shows significant radial distortion. (b) shows the same image after the radial distortion has been removed, using the estimated parameters.

In the next section we show how to use the calibration information and the laser scans to estimate the full 3D pose of the vehicle.

5. POSE ESTIMATION USING IMAGES

The problem of pose estimation is a central one in computer vision. Given a set of images, taken by a set of cameras, or one camera undergoing some unknown motion, pose estimation is the problem of determining the correct position and orientation for each camera or frame. Much research has focused on the structure from motion problem, which has proven to be difficult. This approach generally starts with an uncalibrated camera and simultaneously estimates structure and motion. Typical structure from motion algorithms estimate the motion between successive frames via the fundamental matrix [2,6,9,13,15], and then perform a bundle adjustment over many frames [10,11,12,14,16]. This approach is computationally intensive, and is difficult to make robust.

Compared with the general structure from motion problem, object pose estimation is generally more tractable. Object pose estimation is the problem of estimating the relative transformation, a rotation and translation, between a known 3D object, in the form of a set of 3D points, and a single camera viewing the object. A numerical solution to this problem was originally presented in [8] and a fully projective refinement was added in [1]. This paper generalises the work of [1] to handle multiple objects and multiple images. A RANSAC-like formulation is also introduced to improve robustness.

Our pose estimation algorithm exploits the fact that the scan points for a particular scan are visible in many images. For every image that is captured using our system, there is an associated horizontal and vertical laser scan recorded at exactly the same time as the image. Since the laser scanner is two-dimensional the return point from it is represented in distance and angle. Thus, the 3D coordinates of the reflected points by the laser scanner are known with respect to the coordinate system of the laser scanner, and hence the coordinate system of the camera. This is because there is a fixed transformation between each laser scanner and the camera, as they are both mounted on a plate as shown in Figure 1(b). This means that a subset of points in each of the captured camera frames have known 3D coordinates with respect to the camera's coordinate system. By camera's coordinate system, we mean the relative position and orientation of the camera. There is thus *structure* associated with each camera's coordinate system. If this structure can be identified in nearby views then the relative rotation and translation between the nearby views can be estimated. This is in some ways the dual of a typical scene reconstruction algorithm, where the motion is known and the structure is found. We refer to our algorithm as *multi-object* multi-image pose estimation. Multi-object refers to the fact that we have a separate set of points, or object, associated with each of the images, as opposed to a single object visible in many images.

(a) *(b)* *(c)*

Figure 7: The projected scan points. Image (b) shows an image and its associated horizontal and vertical scan points projected into the image(red crosses) using the initial rough transformation obtained from scan matching. Images (a) and (c) are nearby images. The blue crosses show the projections of the scan points from image(b) in image(a) and image(c). By matching projections across all images, an optimal pose estimate can be found. In this set the horizontal scan clearly does not project to the same points (particularly in image (c)), as the vehicle begins to go down hill.

5.1 Summary of Proposed Method

Since every image has an associated horizontal and vertical laser scan, sparse structure in each image is known in the form of a relatively small number of 3D points, approximately 200. If the corresponding points could be identified in nearby images in the sequence, then object-based pose estimation algorithms could be used to find the pose. The corresponding points in the nearby images can be found using an initial rough estimate of the transformations combined with standard image correspondence techniques. In our application the initial transformation comes from laser scan-matching. It is important to note that the 3D points that are associated with each image may not be easily identifiable in the nearby images, i.e. they will probably not be 'feature' points, and

may be occluded in some images, so incorrect matches will be common; the pose estimation technique must be robust to these outliers.

RANSAC [3] has been shown to be effective at estimating the epipolar geometry between two images, even in the presence of large numbers of outliers [13]. RANSAC can be used to fit models to data robustly because it does not aim to minimise an error metric such as squared distance, like many parameter estimation methods, but instead aims to maximise a *consensus*. A point is said to be *consenting* when it fits the parameters. The consensus is the number of points that are consenting. The measure of when a point fits the parameters is application specific; in the epipolar geometry estimation problem a point is said to be consenting when it is within one or two pixels of its epipolar line.

A naive RANSAC pose estimation algorithm would randomly pick the minimum number of correspondences across all N views, solve for all $N-1$ transforms, and then find the consensus across all N views. In practice, this approach is unlikely to work well; this is because if the probability of choosing enough good points, those that actually correspond, between two images is p, then the probability of randomly picking enough good points in the N views such that the $N-1$ transforms can be found is of the order of p^N. This can be very small if N is large, requiring many iterations of choosing points, solving for the transformation and checking the consensus. This could quickly become computationally infeasible. To overcome this problem, our algorithm only calculates the pose between two views at a time, but the consensus is measured across all N images. A final step in our algorithm minimises the projected error across all N images, among consenting points, giving the maximum likelihood solution, assuming Gaussian error in inlier positions.

5.2 Mathematical Preliminaries

A point $P_{m,i}$ is the m^{th} known 3D point, in homogenous coordinates, associated with image i. A 3D point is defined to be associated with an image if it has been captured by the laser scanner at the same time as the image is captured. The coordinates, $P^j_{m,i}$, of this point in the coordinate system of image j is given by:

$$P^j_{m,i} = \mathbf{T}_j \mathbf{T}_i^{-1} P^i_{m,i} \tag{5.1}$$

where $\mathbf{T}_k$ is the transformation between camera k and some reference coordinate system, consisting of a rotation, $\mathbf{R}_k$ and translation, $\vec{t}_k$. The coordinates of the projection, $p^j_{m,i}$ of $P^i_{m,i}$ in image j, is thus given by:

$$p^j_{m,i}(1) = P^j_{m,i}(1) / P^j_{m,i}(3) \tag{5.2a}$$

$$p^j_{m,i}(2) = P^j_{m,i}(2) / P^j_{m,i}(3) \tag{5.2b}$$

The projected points above are on the focal plane of the normalised camera with focal length unity. It is trivial to convert from this camera coordinate system to actual *u,v* pixel coordinates if the camera calibration is known and vice-versa.

The coordinates of the projection of an arbitrary 3D point, $P_{m,i}$, in image *j*, are thus a function of $P^i_{m,i}$, $\mathbf{T}_i$ and $\mathbf{T}_j$. Each $\mathbf{T}_k$ is a function of six independent variables, t^x_k, t^y_k, t^z_k, the translation components, and $\theta_k, \phi_k, \alpha_k$, the rotation components, expressed in Euler angles.

Each $p^j_{m,i}$ is thus a function of 12 different parameters. Let the measured projected position, found using image correspondences, of $P^j_{m,i}$ in image *j* be $q^j_{m,i}$. The aim of multi-image, multi-object pose estimation is to minimise:

$$\sum_i \sum_m \sum_{j,j\neq i} \left\| p^j_{m,i} - q^j_{m,i} \right\| \tag{5.3}$$

over all, t^x_k, t^y_k, t^z_k, the translation components, and $\theta_k, \phi_k, \alpha_k$. For *N* images this is a minimisation problem in *6(N-1)* variables; one image is chosen as the base coordinate frame.

On initial inspection, the form of (5.3) appears similar to that found in the bundle adjustment literature. The difference is that in bundle adjustment each 3D point is in the world coordinate system and thus the projection of a 3D point in any image, assuming intrinsically calibrated cameras, is only a function of a single transformation matrix, from the world coordinate system to the camera's coordinate system. In our problem the projection depends on two transformations, thus the form of (5.3) is more complicated.

For estimating the transform between a pair of images, (5.3) reduces to

$$\sum_m \left\| p^j_{m,i} - q^j_{m,i} \right\| \tag{5.4}$$

This is a much simpler function, as image *i* can be chosen as the base coordinate frame and thus $\mathbf{T}_i$ becomes the identity and (5.4), for fixed $P_{m,i}$, is a function of only the six components of $\mathbf{T}_j$. This is single-object single-image pose, or *extrinsic camera calibration*, and has been addressed in [1,8].

Let the *consensus* of a given parameter set be defined as the number of projection points that are consenting. A projection point, $p^j_{m,i}$ is said to be *consenting*, if it is close to its measured point $q^j_{m,i}$.

$$\left\| p^j_{m,i} - q^j_{m,i} \right\| < \varepsilon \tag{5.5}$$

where ε is a chosen parameter, dependant on the amount of expected noise in inlier feature positions; in this paper we choose five pixels. The consensus, for a given set of parameters, is thus given by:

$$C \equiv \sum_{i} \sum_{m} \sum_{j, j \neq i} I(\left\| p_{m,i}^{j} - q_{m,i}^{j} \right\| < \varepsilon) \tag{5.6}$$

where *I(statement)* is the indicator variable, which is one when the enclosed statement is true and zero otherwise. A RANSAC algorithm chooses the set of parameters to maximise (5.6), and then only keeps the set of points that are consenting. (5.3) is then minimised only within these consenting points.

Equation (5.6) is a function of all the transformation matrices, and is expensive to compute for many images. However if a single image's transformation matrix is changed then the consensus can be updated efficiently. Let *EX(k)* be defined as the extent of image *k*, the set of all images that have a *connection* to image *k*. Two images are connected if at least one of the 3D points in one image is visible in the other image. Although *EX(k)* can be computed exactly, in practice it is chosen to be all the images nearby to image *k*. If the transformation matrix of image *k* is changed then only a consensus *change* need be computed. This consensus change only involves *EX(k)* other images and is much easier to compute than (5.6). The terms of (5.6) involving image *k,* and its transformation matrix $\mathbf{T}$, are given by:

$$C(k, \mathbf{T}) \equiv \sum_{j \in EX(k)} \sum_{m} I(\left\| p_{m,k}^{j} - q_{m,k}^{j} \right\| < \varepsilon) + \sum_{j \in EX(k)} \sum_{m} I(\left\| p_{m,j}^{k} - q_{m,j}^{k} \right\| < \varepsilon) \tag{5.7}$$

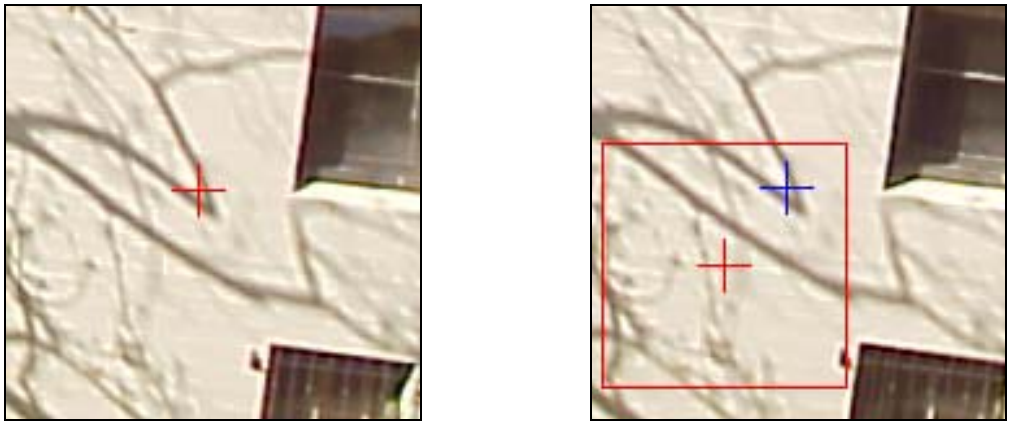
If image *k*'s transformation matrix is changed to $\mathbf{T}'$ then the consensus change is given by:

$$\Delta C = C(k, \mathbf{T}') - C(k, \mathbf{T}) \tag{5.8}$$

where $\mathbf{T}$ is the previous transformation matrix.

The new value of (5.6) can be computed as:

$$C_{new} = C_{old} + \Delta C \tag{5.9}$$



(a) *(b)*

Figure 8: The red cross on the left, in image (a), shows the projection of a scan point associated with image (a), the red cross on the right shows the projection of the scan into a second image, image b, using the rough transformation. The blue cross in image (b) is the correspondence found using image matching.

5.3 The Proposed Algorithm

Our proposed algorithm works as follows, given a set of *N* images with associated 3D points in each image, and a rough estimate of the transformation parameters:

1. For each image *i*, each 3D point associated with image *i* is first projected into image *i*, giving $p^i_{m,i}$. This gives the true projection of the point in image *i*; since the 3D point is defined in the image's coordinate frame, the projection of this point in the other images may not be accurate, as the pose is only roughly known.
2. Each 3D point $P_{m,i}$ in view *i* is then projected into every other view *j*, using the initial estimate of the transformation matrices; this results in $p^j_{m,i}$. A search window of a certain size, depending on the uncertainty in the rough transformation matrices, around $p^j_{m,i}$ is searched to find the best match to $p^i_{m,i}$, in terms of intensities; this gives $q^j_{m,i}$, the measured projection as shown in Figure 8. The sum of squared colour differences is used as the matching criterion. If $P_{m,i}$ is visible in view *j* then $P_{m,i}$ is said to have a correspondence in view *j*. Steps 1 and 2 are repeated for all *N* images.
3. The initial consensus is found using (5.6), using the rough initial transformations; the maximum consensus is set to the initial consensus.
4. A pair of images, *a* and *b*, are chosen from the *N* views, such that *a* has at least three correspondences in *b*.
5. Three correspondence pairs ($P_{m,a}, q^b_{m,a}$) are chosen at random. Using these pairs (5.4) is minimised to find $\mathbf{T}_{a\to b}$, using the previous estimate of the transformation as an initialisation. This means that the transformation for image *b* is now given by $\mathbf{T}'_b = \mathbf{T}_{a\to b}\mathbf{T}_a$.
6. Using $\mathbf{T}_b$ and $\mathbf{T}'_b$, the consensus change given by (5.8) is computed. If the consensus change is positive, meaning the overall consensus has increased, $\mathbf{T}_b$ is set to $\mathbf{T}'_b$ and the maximum consensus is updated according to (5.9).
7. Steps 4-6 are repeated a large number of times, depending on the expected number of outliers.
8. Finally, (5.3) is minimised using steepest descent among the consenting points to find the optimal set of transformations, in a maximum likelihood sense.

If a set of rough transformations is unavailable, then the search window can be made very large. Alternatively the previous transformation can be used as an initialisation. For our application laser scan-matching gives a good estimate of the initial transformations.

The minimisation of (5.3) in step 8 is roughly $O(n^3)$ where *n* is the number of images and thus quickly be comes computationally unfeasable. When the number of images is large the sequence can be split into smaller blocks and this final minisation can be performed within each block.

If step 5 results in a vehicle motion that is incompatible with the vehicle physics, for example if the resulting motion means that the truck dropped 5 metres in a fifth of a second, then the transformation is rejected immediately without checking the consensus.

6. RESULTS

The proposed algorithm was evaluated on synthetic and real data. Synthetic data was used to quantify the effect of outliers and noise on the algorithm. Real data was used to evaluate the performance of the algorithm in a real world situation.

6.1 Synthetic Data

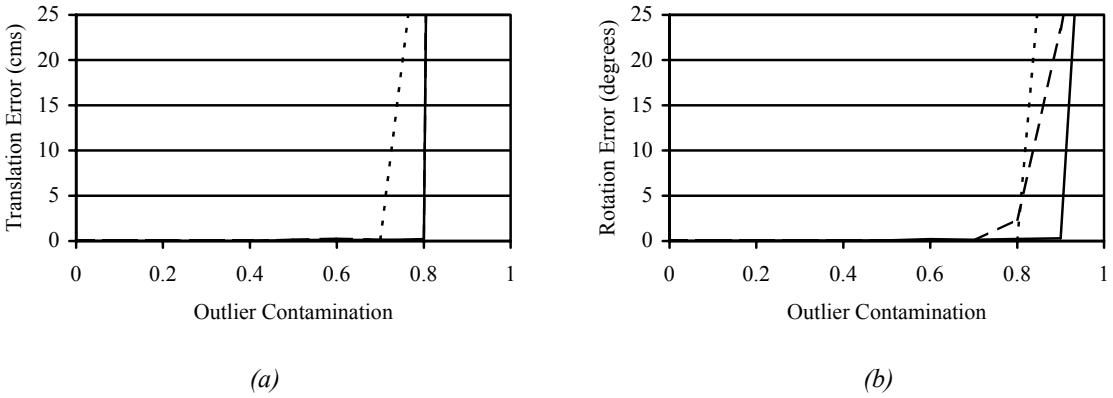
For the 'synthetic' data, real laser data associated with five pictures was used; projecting the real data using a set of known transformations generated correspondences. Thus perfect correspondences were generated; meaning (5.3) was zero. Using these perfect correspondences allows the effect of outliers and noise to be modelled in a controlled way. Firstly, the outliers were added to the correspondence data; varying numbers of outliers were added, set by a probability μ. The meaning of the parameter μ is as follows; each correspondence was discarded, with probability μ, and replaced with an *x,y* value within the range of its correlation window, which was chosen to be *100x100* centered at the actual projection point. This simulated the effect of incorrect correspondences. The effect of Gaussian noise in the correspondence data was also modelled, by adding noise with increasing variance, ρ^2, to the correspondence data. Gaussian noise was added to both the *x* and *y* coordinates of the projected image points. The transformations were then estimated using the proposed algorithm, and the results were compared with the known transformations. The initial value for each transformation for the minimisation of (5.3) and (5.4) was set to the identity transformation. In general we found that convergence was not an issue for our algorithm, the algorithm always converged if there was enough data available, i.e. if the consensus was high enough.

Let the k^{th} actual transformation matrix be T_k and the k^{th} estimated transformation matrix be $\widetilde{T}_k$. Let the actual translational components be t_k^x, t_k^y, t_k^z, and the estimated translational components be $\widetilde{t}_k^x$, $\widetilde{t}_k^y$, $\widetilde{t}_k^z$. The absolute

difference between these gives the error, $\varepsilon_{t,k}^{x}, \varepsilon_{t,k}^{y}$, $\varepsilon_{t,k}^{z}$. The average translation error is the average over all three translation components and images, $\overline{\varepsilon}_t$. All translational components are measured in centimetres.

Let the actual rotation be R_k, and the estimated rotation be $\widetilde{R}_k$. To represent the error in the rotation component of the transformation an 'error rotation' is found. This is the rotation that needs to be applied to the estimated rotation matrix to get the actual rotation matrix. In other words, $R_{e,k} = R_k \widetilde{R}_k^{-1}$. This error rotation is then parameterised into angle-axis form, and the absolute angle of $R_{e,k}$, ω_e is used as a measure of the error in rotation. The average rotation error over all images is thus $\overline{\omega}_e$.

The motion for the tests using synthetic data was designed to mimic real motion experienced by the truck. The motion was mainly translational in the direction of the vehicle. Some rotation in the ground plane was also tested, this would correspond to the vehicle turning a corner. All sets had relatively little rotation around the other axes, along with small vertical motion; this is typical of the vehicle's motion. The results are summarised, for different data sets, in Figures 9 and 10 below.



(a) *(b)*

Figure 9: Effects of outliers on the proposed algorithm, with 0 noise variance, on three different image sequences. (a) Shows the average translational error as a function of outlier contamination. (b) Shows the average rotation error as a function of outlier contamination. Each sequence consisted of five images.

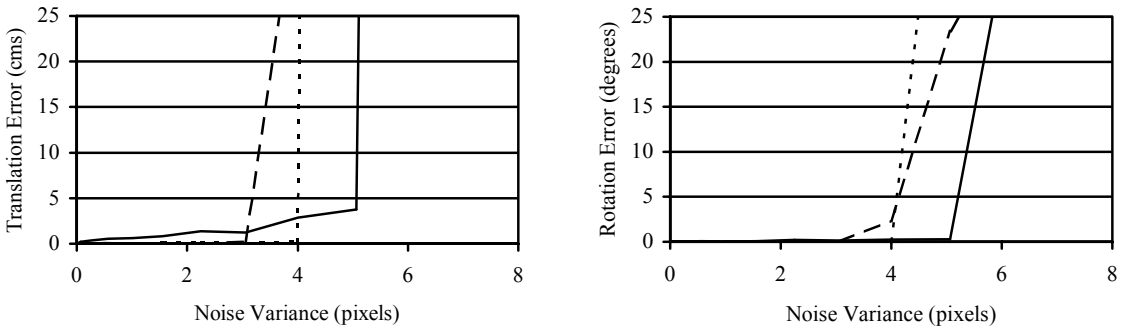
Figure 9 shows the average translation error, $\overline{\varepsilon}_t$, and average rotation error, $\overline{\omega}_e$, as a function of the outlier contamination μ. The plots show that the algorithm is robust to up to 70% contamination by outliers, with reasonable results even achieved for 80% contamination. Above 70% outlier contamination the errors become very large, and are in fact off the charts. However, even at high outlier probabilities, only some of the transformations become completely incorrect, while others are found correctly. This is probably because more of the outliers, by chance, are added to one image. The transformation for this image is found incorrectly, but the rest of the transformations are accurate. This is an interesting effect, since it shows that if one image has so little correct correspondences, for example if something temporarily occludes the entire frame, that its transformation cannot be found it does not cause the entire algorithm to fail; the other transformations are still found correctly. This does not

occur if the views are processed pair-wise first and then brought together at the end, as is common in structure from motion. This shows the advantage of processing many images simultaneously.

To demonstrate this effect further, we simulated the effect of there being no reliable correspondence data for one of the images in a synthetic sequence. We added 95% outliers to only the correspondence data that involved image 2. Outliers were added to both the projections of the scan points associated with the other images into image 2 and to the projection of the scan points associated with image 2 into the other images. This effectively meant that there was no linkage between image 2 and the other images. No other outliers were added to the remaining correspondence data. We then estimated the pose using (a) a pair-wise approach, and (b) our multi-image algorithm. The pair-wise approach maximises the consensus between pairs of images first, between image 0 and image 1, image 1 and image 2, etc. and then calculates the final solution by minimising (5.3). The results are shown in Table 1 below, where the average translation and rotation error are shown for different images. Using the pairwise method, the pose of image 1 is computed correctly, but the poses of image 2, 3 and 4 are incorrect. Using our multi-image algorithm only image 2 is incorrect. By maximising the consensus across all images instead of just in pairs, our algorithm has effectively ignored the erroneous correspondence data involving image 2.

Pair-wise Approach			**Proposed Multi-Image Approach**		
Image No	ε_t	ω_e	**Image No**	ε_t	ω_e
1	0.000659	0	1	0.262869	0.2307
2	751.7887	58.2784	2	351.3307	26.02139
3	1446.845	20.13024	3	0.01691	0.149353
4	1443.991	20.12973	4	0.115966	0.310904

Table 1: Advantage of multi-image approach over pair wise approach.



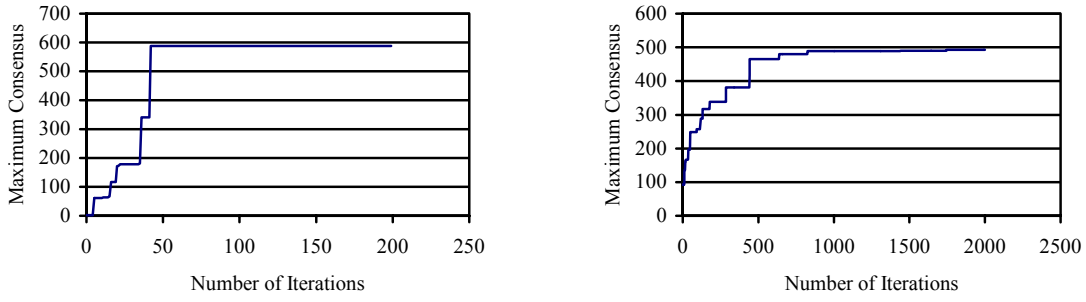
(a) *(b)*

Figure 10: Effects of noise on the proposed algorithm, with zero outlier probability, on three different image sequences. (a) Shows the average translational error as a function of the noise variance. (b) Shows the average rotation error as a function of the noise variance. Each sequence consisted of five images.

Figure 10 shows the average translation error, $\overline{\varepsilon_t}$, and average rotation error, $\overline{\omega_e}$, as a function of the noise variance ρ^2. The plots indicate good robustness to noise. The algorithm does not catastrophically fail until the

noise reaches relatively high levels. Failure does not occur until the noise variance reaches approximately 4. Lower levels of noise negligibly affect the accuracy of the algorithm.

We also studied convergence of our algorithm for real and synthetic data. Figure 11(a) shows a plot of maximum consensus as a function of iteration number for 50% outlier contamination on synthetic data. Our algorithm has found the maximum consensus after approximately 50 iterations. The identity matrix is used as initialisation for all images, so the starting consensus is very low. Figure 11(b) shows the same plot for real data; a sequence of five images is used. The maximum consensus has been found after roughly 1000 iterations. Although the percentage of outliers in both cases is roughly the same, convergence for the real data sequence took longer. This is because in the real data the ouliers are not distributed uniformly across all the images. Some scans are occluded in some images, thus there are more outliers in some images than in others. Our algorithm chooses image pairs with a uniform probability; it is possible that convergence would be improved if the algorithm adaptively chose image pairs based on the number of outliers in each image. This would mean images with high numbers of outliers would be sampled more often than images with low numbers of outliers. This method has not been tested as algorithm execution time is not a primary concern in our application.



(a) *(b)*

Figure 11: The consensus as a function of iteration number. (a) The correspondence data was corrupted with 50% outliers and zero noise, the maximum consensus before corrupting the data was 1100. The maximum consensus found by our algorithm was 588, and was reached after 50 iterations. (b) On real data the algorithm takes longer to converge, in this case roughly 1000 iterations were required before convergence.

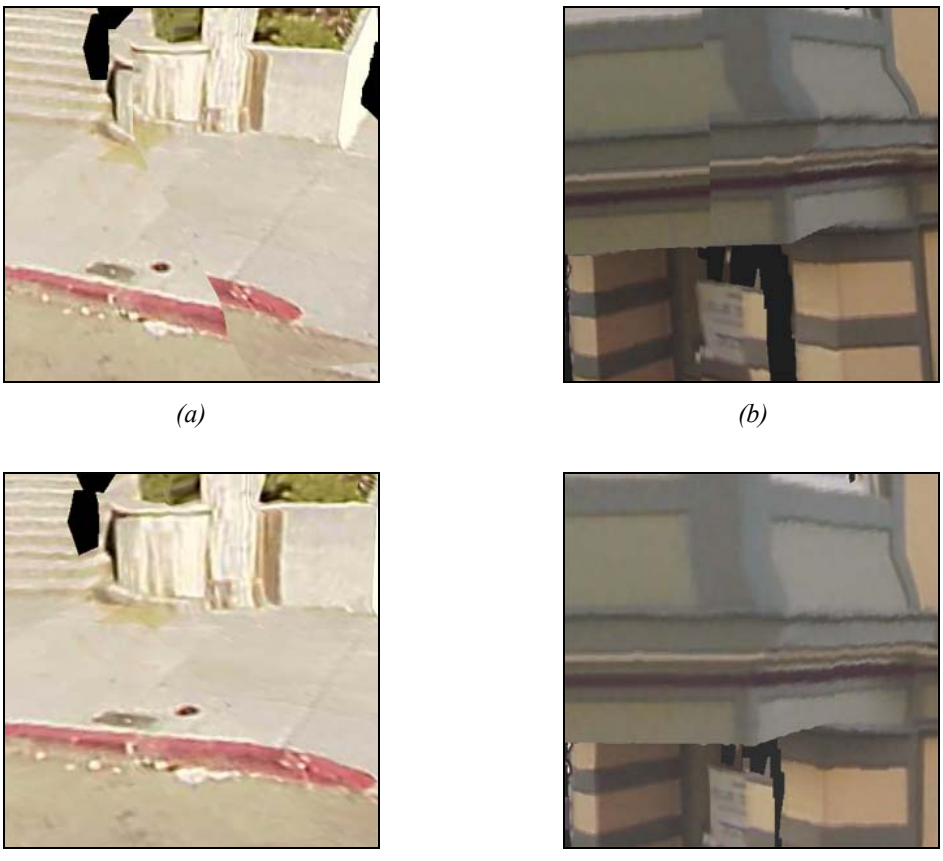
6.2 Real Data

Two sequences of real image data were used to test the algorithm. The first sequence, the Ihouse sequence consisted of 500 images. The second sequence, the downtown sequence consisted of 3500 images. We processed all 3000 images simultaneously to maximise the consensus given by (5.6). (5.3) was minimised by splitting the sequence into blocks of ten images as mentioned previously. The transformation from the laser scan-to-scan match is used as the initial estimate of the transformation and a search window of 100x100 pixels is used. This allows for significant errors in the laser scan-to-scan match. Our algorithm has performed approximately 3000 iterations per image to find the maximum consensus. It took a 2Ghz Pentium 4 PC approximately three days to process all 3000 images.

(a) *(b)*

Figure 12: Reprojections of model points in the images. (a) The left images show the reprojections of selected model points in two images with the original rough pose estimation. Projections of the same 3D point are drawn in the same colour in both images. The original pose is clearly incorrect. (b) The right images show the same points projected with the new pose, found using the proposed algorithm. The images clearly match the model well.

The pose estimates are clearly improved in a local sense, as can be seen in Figure 12. Crosses of the same colour correspond to projections of the same 3D point. A 3D point should be photo-consistent across all images if the pose is correct. The projections of the same 3D point are not photo-consistent [7] using the original pose estimates as shown in Figure 12(a), but are photo-consistent using the pose estimates calculated with our algorithm as shown in Figure 12(b). This photo-consistency is important for texture mapping and stereo algorithms.



(a) *(b)*

(c) *(d)*

Figure 13: Texture Mapped Models. (a) and (b) show portions of a textured mapped model created using the pose computed from scan matching. A significant seam is visible in both images. (c) and (d) show the same portions of a model created using the pose from our algorithm. The seams are not visible in the second model.

To demonstrate the effect of locally incorrect pose on the final model, we created two texture-mapped models using our system. The first model was created using the pose computed from scan matching. The second model was created using the pose computed by our algorithm. Figure 13 shows some detail from both models. Clearly, seams are visible in the texture mapping in the first (top) model while no seams are visible in the second (bottom) model. The entire model is not visible in any single image so many images are used to texture-map a model. Seams occur when the texture-mapping switches between consecutive images. If the transformation between consecutive images is not known accurately then the images do not match at the join. When there is any vertical motion the scan matching algorithm cannot accurately estimate the pose and seams occur. Our algorithm, however, estimates the full pose and thus the seams are no longer visible.

We quantified how well our algorithm removed seams by making a number of models and texture mapping them using both the pose estimates from scan matching and the full pose estimates from our algorithm. For each model, we compared the appearance of the seams on the model created using the original pose estimates from laser scan matching and the model created using the improved pose estimates based on the algorithm proposed in Section 5. We judged the seams as completely removed, improved, no change and worse. The results are summarised in Table 2. As can be seen, on average, 80% of the seams were completely removed, the rest were mostly unaffected, while a small percentage appeared worse.

Model Number	Original Number of Seams	Completely Removed	Better	No Change	Worse
1	48	42	0	4	2
2	15	8	0	7	0
3	29	25	2	2	0
4	24	18	1	5	0
5	33	24	0	9	0
6	18	15	0	3	0
7	34	28	1	3	2
Total	201	160	4	33	4
Percentage		80	2	16	2

Table 2: Improvement in the number of visible seams using the proposed algorithm.

The globally recovered pose for the Ihouse sequence, is shown along with the captured point cloud in Figure 14. In Figure 15, the recovered vehicle path is shown coloured blue, overlaid on an aerial image of the corresponding area. The original path obtained from scan matching is also shown colour red. A rigid rotation and translation, followed by uniform scaling, has been applied to both paths so that they best fit the aerial image. This is equivalent to estimating the pose and focal length of the camera that captured the aerial image. Qualitatively the shape of the vehicle's path as recovered by our algorithm appears to be more accurate than the original path computed by scan matching; the path computed using our algorithm keeps to the road outlines, while the original path does not.

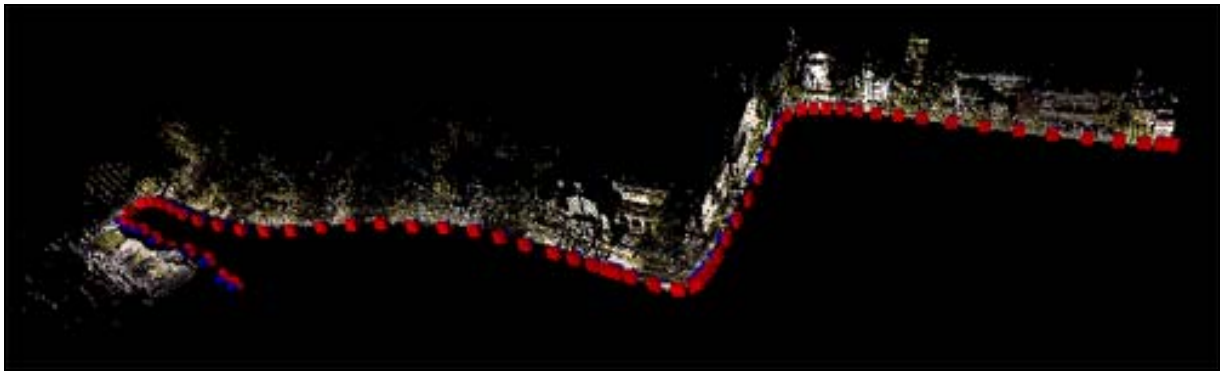


Figure 14: The Ihouse sequence: *The recovered pose. The red and blue boxes represent the cameras. Only a subset of the cameras is shown. The captured point cloud is coloured using the image data and displayed.*

The main errors in the original path are largely confined to three areas, as shown in Figure 15. The original path obtained from laser scan matching is shown in red, the path obtained using our algorithm is shown in blue. The areas in the yellow boxes are areas that pose difficulty for the scan-matching algorithm. In these areas the horizontal scanner sees mainly trees. Trees are highly irregular and successive scans see different outlines of the trees, thus scan matching is not accurate. The images in these areas show significantly more features than just the trees, and our proposed algorithm is able to accurately estimate the pose in these areas. Also, there are significant hills in this region, and scan-matching cannot capture this type of motion, so the path from scan-matching can never be correct in this region. Although the path computed using our algorithm appears correct, it is difficult to quantify the accuracy since ground truth data is not available.

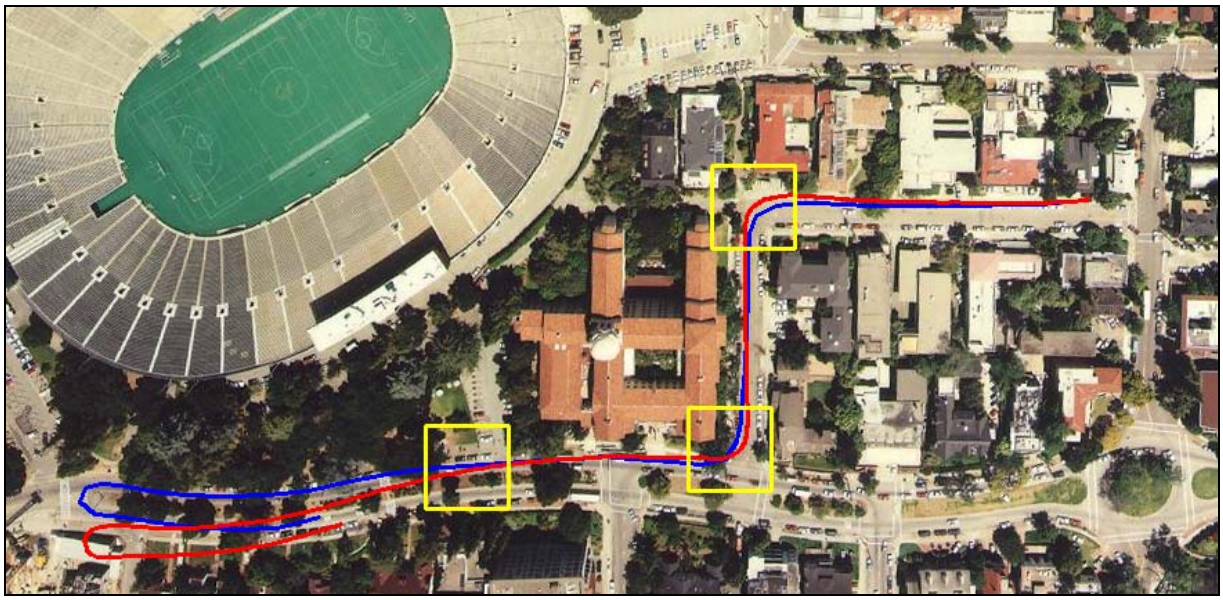


Figure 15: The Ihouse sequence: Plots of vehicle motion in the ground plan, overlaid on an aerial photo of the area. The original path obtained from laser scan matching is shown in red, the path obtained using our algorithm is shown in blue. The problem areas for the scan-matching algorithm are highlighted.

The recovered vehicle path for the downtown sequence, projected into an aerial image of the corresponding area, is shown in Figure 16. The recovered path using the proposed algorithm is coloured in blue; the path computed using laser scan-matching is shown in red. As can be seen, the computed paths from both schemes are inaccurate. The path computed using our proposed image based algorithm shows no significant improvement over the original path. The main errors for both techniques are due to drift; this sequence is considerably longer than the Ihouse sequence and the effect of drift is important. As shown in [4] the drift problem can be mitigated using global information such as digital roadmaps.

Figure 16: The downtown sequence: Plots of vehicle motion in the ground plan, overlaid on an aerial photo of the area. No colour photo was available. The original path obtained from laser scan matching is shown in red, the path obtained using our algorithm is shown in blue. A gross error occurs at the highlighted region, for reasons discussed below.

The main errors in the path occur at corners. In general, corners are difficult for our algorithm. At a corner the visibility of points changes rapidly as shown in Figure 17, there are thus many occluded scan points, and the possibility of correspondence error increases. If in addition the scene at a corner contains many other occluders, then the algorithm can give completely incorrect pose estimates. In addition, points that are scanned at an oblique angle, which occurs when the vehicle turns around a corner, are unreliable. This is because the laser diverges; its spot size is approximately 30cm at 20m. When an oblique surface is scanned the laser has a range of possible depth values that fall within this spot, and the value reported back becomes unreliable.

It should be noted that the scan-matching algorithm performs particularly well in the downtown area. There are no significant hills in the area so the in-plane assumption is not violated. In addition, the area contains many polygonal surfaces, which aid the scan-matching algorithm.

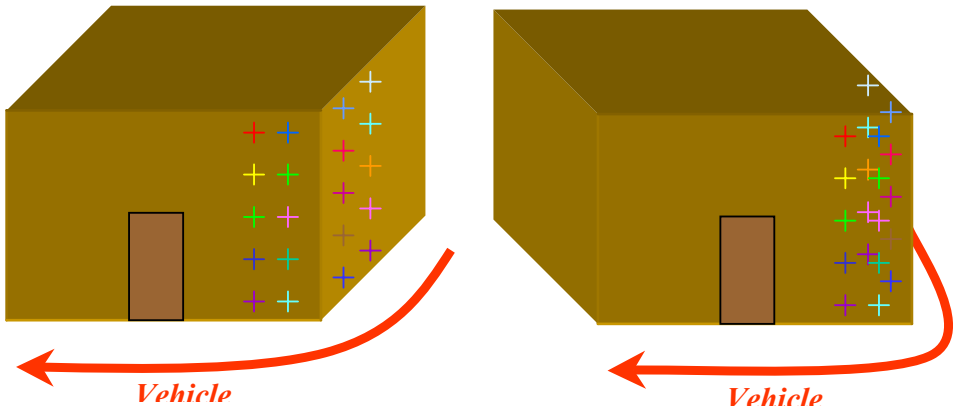


Figure 17: Diagram showing problems at corners. In (a), as the vehicle rounds the corner, both sides of the building are visible. A short time later, in (b) one side of the building has become invisible, and many correspondences will become occluded. This will lower the consensus and may cause problems if the scene is otherwise complicated. There are many occluding objects and the overall scene is relatively complicated.

6.4 Real Data With No Initialisation

As discussed, our algorithm performs well when there is a good initial estimate of the transformations available, such as those obtained by laser scan-matching. To verify our algorithm further we calculate the vehicle's path without any initial estimate of the transformations.

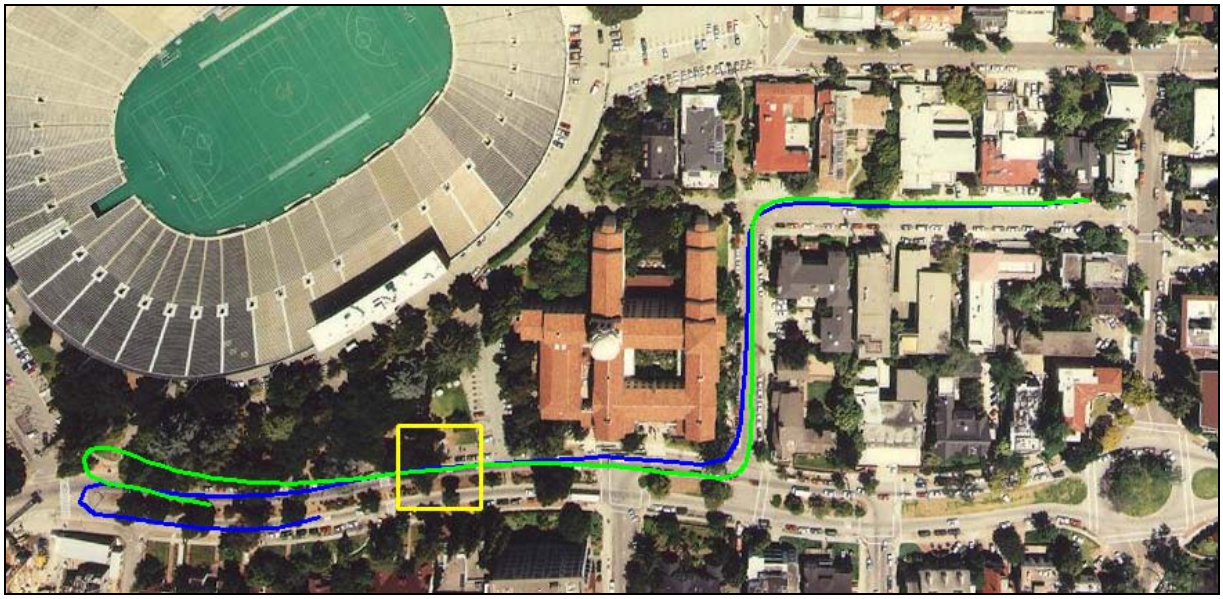


Figure 18: The Ihouse sequence: Plot of computed vehicle motion in the ground plane, computed using no initial transformations, overlaid on an aerial photo of the area. The path obtained using no initial transformations is shown in green, the path obtained using laser scan-matching as initialisation is shown in blue. The main error seems to occur in the highlighted area shown.

We split the sequence into blocks and run our algorithm within each block. For each block, we predict the initial transformation, based on the previous block. The transformation is predicted using simple linear extrapolation. In

the first block the vehicle is assumed to be at rest. In addition, we use an adaptive search window that calculates the size of the search window based on the predicted speed of the vehicle. Also, if the maximum consensus found becomes very low for a particular block, we increase the window size and run the algorithm again for that particular block. This is necessary to correctly deal with areas where the predicted transformation is not a good estimate for the actual transformation. The computed path using this method is shown in Figure 18 in green. The computed path using the laser scan-matching as initialisation is also shown in blue. It can be seen that the two paths are similar, but the path computed using no initial estimate of the transformation diverges from the true path towards the end of the sequence. The main error seems to have occurred in the highlighted area. This is probably because in this area the images show mainly trees and little else; trees are problematic for correspondence finding algorithms, due to the high numbers of self-occlusions and their non-Lambertian properties. Thus, fewer correct correspondences are found, and the pose estimates become less accurate.

7. CONCLUSIONS AND FUTURE WORK

A new robust multi-image multi-object pose estimation algorithm has been presented. It has been shown to be robust to large numbers of outliers and significant amounts of noise, in tests with synthetic data. On real data the algorithm also works well; the match between the 3D data and images is clearly improved, and algorithm maintains accurate pose over long distances.

In the future we will explore further the use of this technique without using the scan matching as an initial estimate of the pose. If the pose could be estimated accurately and robustly without using the horizontal laser scanner then the horizontal scanner could be removed completely from our system, significantly reducing the cost and complexity.

We are currently developing a system to model both sides of the street simultaneously by using four scanners and two cameras. The extra camera and scanners will be placed on the opposite side of our vehicle to the current camera and scanners, and will model the opposite side of the street. Our algorithm can be easily extended to handle this new situation, and robustness and accuracy would improve as the extra scanners and camera would provide more sources of data.

With any pose estimation technique that is based on relative motion drift inevitably occurs, as seen for the downtown sequence. Recently we have obtained aerial laser data, it is possible that this data could be used to correct the full pose estimates in a similar manner to the method of [4], which corrects the pose estimates only in the ground plane using aerial images. In addition, it is possible that invariants in the image sequence, such as the angle of vertical lines or planes, could aid in global correction.

REFERENCES

[1] Carceroni R. L. and C. M. Brown. "Numerical Methods for Model-Based Pose Recovery". Technical. Report 659, *Computer Science Department, The University Of Rochester, N.Y.*, August 1998.

[2] O. Faugeras. "What Can be Seen in Three Dimensions with an Uncalibrated Stereo Rig?" *Proc. European Conference on Computer Vision*, pages 563--578 Santa Margherita Ligure, Italy, May 1992.

[3] M. A. Fischler and R. C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Comm. ACM, 24(6):381-395, 1981.

[4] C. Früh and A. Zakhor, "3D Model Generation for Cities using Aerial Photographs and Ground Level Laser Scans" to be presented in *Computer Vision and Pattern Recognition Conference*, Kauai, Hawaii, December, 2001.

[5] C. Früh and A. Zakhor, "Fast 3D Model Generation in Urban Environments", *IEEE Conf. on Multisensor Fusion and Integration for Intelligent Systems*, Baden-Baden, Germany, 2001, p. 165-170.

[6] R.I. Hartley. "Estimation of Relative Camera Positions for Uncalibrated Cameras". *Proc. European Conference on Computer Vision*, pages 579-587, Santa Margherita Ligure, Italy, May 1992.

[7] K. N. Kutulakos and S. M. Seitz, "A Theory of Shape by Space Carving". *Proc. 7th International Conference Computer Vision*, pp. 307-314, Corfu, Greece, 1999.

[8] D. Lowe. "Fitting Parameterized Three-Dimensional Models to Images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441-450, May 1991.

[9] Q.-T. Luong, R. Deriche, O. Faugeras, and T. Papadopoulo. "On Determining the Fundamental Matrix: Analysis of Different Methods and Experimental Results". Technical Report RR-1894, INRIA, Sophia Antipolis, France, 1993.

[10] C. J. Poelman and T. Kanade. "A Paraperspective Factorization for Shape and Motion Recovery". *Proc. of the 3rd European Conference on Computer Vision*, volume B of *Lecture Notes in Computer Science*, pages 97–108, Stockholm, Sweden, May 1994.

[11] P. Sturm and W. Triggs. "A Factorization Based Algorithm for Multi-Image Projective Structure and Motion". *Proceedings of the 4th European Conference on Computer Vision*, pages 709–720, Cambridge, UK, Apr. 1996.

[12] C. Tomasi and T. Kanade. "Shape and Motion from Image Streams under Orthography: a Factorization Method". *The International Journal of Computer Vision*, 9(2):137–154, 1992.

[13] P. H. S. Torr and D.W. Murray, "The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix," *Int. J. Computer Vision*, vol. 24, no. 3, pp. 271–300, 1997.

[14] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. "Bundle adjustment — a modern synthesis". In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, pages 298–372, Corfu, Greece, Sept. 1999.

[15] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. “A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry”. *Artificial Intelligence Journal*, 78:87–119, Oct. 1995.

[16] Z. Zhang and Y. Shan. “Incremental Motion Estimation Through Local Bundle Adjustment”. Technical Report MSR-TR-01-54, Microsoft Research, May 2001.

[17] Z. Zhang. “Flexible Camera Calibration By Viewing a Plane From Unknown Orientations”. *Proc. 7th International Conference Computer Vision*, pp 666-673, Corfu, Greece, 1999.