2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Automatic Calibration of Multi-Modal Sensor Systems using a Gradient Orientation Measure

Zachary Taylor, Juan Nieto and David Johnson
University of Sydney, Australia
{z.taylor, j.nieto, d.johnson}@acfr.usyd.edu.au

***Abstract*— A novel technique for calibrating a multi-modal sensor system has been developed. Our calibration method is based on the comparative alignment of output gradients from two candidate sensors. The algorithm is applied to the calibration of the extrinsic parameters of several camera-lidar systems. In this calibration the lidar scan is projected onto the camera's image using a camera model. Particle swarm optimization is used to find the optimal parameters for this model. This method requires no markers to be placed in the scene. While the system can use a set of scans, unlike many existing techniques it can also automatically calibrate the system reliably using a single scan. The method presented is successfully validated on a variety of cameras, lidars and locations. It is also compared to three existing techniques and shown to give comparable or superior results on the datasets tested.**

I. Introduction

Lidar sensors can provide invaluable information for allowing mobile robots to perceive the world around them. While these systems have come a long way in recent years, they still have significant constraints on their range, resolution and update rate. Many of these problems can be overcome by combining the system with a colour camera. The high update rate, resolution and additional colour information offered by the camera gives the robotic system a much richer view of the world. This combined system however relies on the sensors being calibrated so that the lidar's point cloud can be projected onto the camera's image. An example of this projection is shown in Figure 1.

This calibration is far from trivial due to the very different modalities via which the two sensors operate. Because of these difficulties, on most mobile robots that fuse multiple sensor modalities the sensors are manually calibrated. This is usually done using reflective markers, checker-boards or by painstakingly hand labelling large numbers of points. These methods are slow, labour intensive and often produce results with significant errors. Once this calibration has been completed it is assumed that it remains unchanged while the robot is operating. In practice however this is a poor assumption as the calibration is degraded due to the robot's motion, particularly for mobile robots working in rough environments. This means that for these systems to be able to operate autonomously for extended periods of time a robust method that can automatically recalibrate the sensors without requiring the system to stop its current operations is required.

This paper introduces a novel metric, the *gradient orientation measure* (GOM) for evaluating the alignment between lidar and camera systems. This metric is used to automatically calibrate the extrinsic parameters of a camera for use with a lidar system. The process can be performed on a single scan of an arbitrary environment without any markers, checkerboards or other indicators placed in the scene. The paper also addresses the issue of how successful the calibration is by creating an estimate for the accuracy of the result.

Fig. 1. A lidar's point cloud with a cameras image projected onto it.

Specifically, this paper presents the following contributions:

- The introduction of a new metric for evaluating the alignment of a camera's image with a lidar scan.
- The evaluation and comparison of four different techniques for performing the calibration. Levinson and Thrun's method [1], Pandey *et al*'s method [2], our own normalized mutual information based method [3] and the introduced GOM method.
- The evaluation of the use of bootstrap resampling for use in predicting the error present in the calibrations.

II. Related work

Work in this area can be roughly divided into two categories. The calibration of mobile ground based systems that use a low resolution lidar (usually the Velodyne HDL-64E) to make a series of scans as they move around the environment and systems that produce a single high resolution scan of an area from a fixed location.

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

A. Mobile systems

Until recently [1], [2], [4] no methods existed for lidar-camera calibration on mobile ground based platforms that did not rely on markers being added to the scene. However recently the field has seen significant development with several methods for the calibration being proposed. A method proposed by G. Pandey *et al* attempts to register a Velodyne lidar with panoramas created by a ladybug panoramic camera [2]. Their method operates by using the known intrinsic values of the camera combined with its estimated extrinsic parameters to generate an image from the lidar data which is coloured by its intensity values. The mutual information between this generated image and the actual camera image is used to evaluate the alignment with the assumption that when this measure is maximized the system is perfectly calibrated. The optimisation of the extrinsic camera parameters is done by the Barzilai-Borwein (BB) steepest gradient ascent algorithm. To reduce the number of local maximums and improve the robustness of the alignment process the mutual information taken is the average of a set of scan-image pairs.

A similar method that makes use of the same sensors and metric was also developed by R. Wang *et al* [4]. The only significant difference in the methods is that R. Wang *et al* makes use of the Nelder-Mead downhill simplex method for optimisation.

An approach presented by Levinson and Thrun, makes use of the fact that the change in colour intensity in an image is often associated with a change in depth [1]. The approach generates an edge image from the camera image by taking the maximum difference between each pixel and its neighbour. Edges are extracted from the lidar by taking the difference in depth between successive points and removing the points below a set threshold. Once this has been done an image is generated from the lidar data using the cameras parameters. An element-wise multiplication of these images is performed, assuming that when the sum of this is maximised the two sensors are correctly aligned. There is also some extra processing done to improve the robustness and convergence of the method.

B. High resolution systems

A recently proposed method by H. Li *et al* makes use of edges and corners [5]. Their method works by constructing closed polygons from edges detected in both the lidar scan and images. Once the polygons have been extracted they are used as features and matched to align the sensors. The method was only intended for and thus tested using aerial photos of urban environments.

A. Mastin *et al* achieved registration of an aerial lidar scan by creating an image from it using a camera model [6]. The intensity of the pixels in the image generated from the lidar scan was either the intensity of the laser return or the height from the ground. The images were compared using the joint entropy of the images and optimisation was done via downhill simplex. The method was only tested in an urban environment where buildings provided a strong relationship between height and image colour.

For the alignment of fixed ground based scans in urban environments a large number of methods exist that exploit the detection of straight edges in a scene [7], [8]. These straight lines are used to calculate the location of vanishing points in the image. While these methods work well in cities and with images of buildings they are unable to correctly register natural environments due to the lack of strong straight edges.

From a more theoretical view on the calibration [9] looked into different techniques for generating a synthetic image from a 3-D model so that mutual information would successfully register the image with a physical photo of the object. They used NEWUOA optimization in their registration and looked at using the silhouette, normals, specular map, ambient occlusion and combinations of these to create an image that would robustly be registered with the real image. They found surface normals and a combination of normal and ambient occlusion to be the most effective.

III. Method

A block diagram of our approach is shown in Figure 2. The image is first converted to grey scale and histogram equalisation is performed on it. For the data given by the lidar the points are coloured by the intensity of return and histogram equalisation is performed on them. The magnitude and orientation of the gradient at each point is then calculated for both the grey scale image and lidar data. Once this has been done a camera model is used to create a 2-D image from the point cloud. The gradient orientation measure (GOM) is used to compare the camera's image with the generated image. This process is repeated for changing extrinsic camera model parameters using particle swarm optimisation. The optimisation continues until all the particles converge and a global maximum for the GOM between the images is found. Note that the method is able to work with both a single image-scan pair or with a set of scans.[1]

A. Gradient calculation

The gradient magnitude and orientation of the camera image is calculated using the Sobel operator. Calculation of the gradient for the lidar point cloud is slightly more challenging. We wish the gradient orientation and magnitude of the points in the lidar data to be from the perspective of the camera so that the orientations will match up with those in its images. The simplest way to calculate these gradients is to first use the camera model to generate an image from the lidar points and use a standard Sobel operator on this generated 2-D image. Unfortunately this method gives poor results. This is because as the point cloud is fairly sparse, many pixels in the image have no corresponding lidar point. An issue also occurs that as the points are forced into a discrete grid issues caused by aliasing also occur, creating patterns in the calculated magnitude and orientation of the gradient.

Instead the gradient of the points are calculated as follows. The points are first projected onto a sphere that is centred at

[1]All the code used for our method is publicly available online at http://zacharytaylor.github.io/Multimodal-Calib/

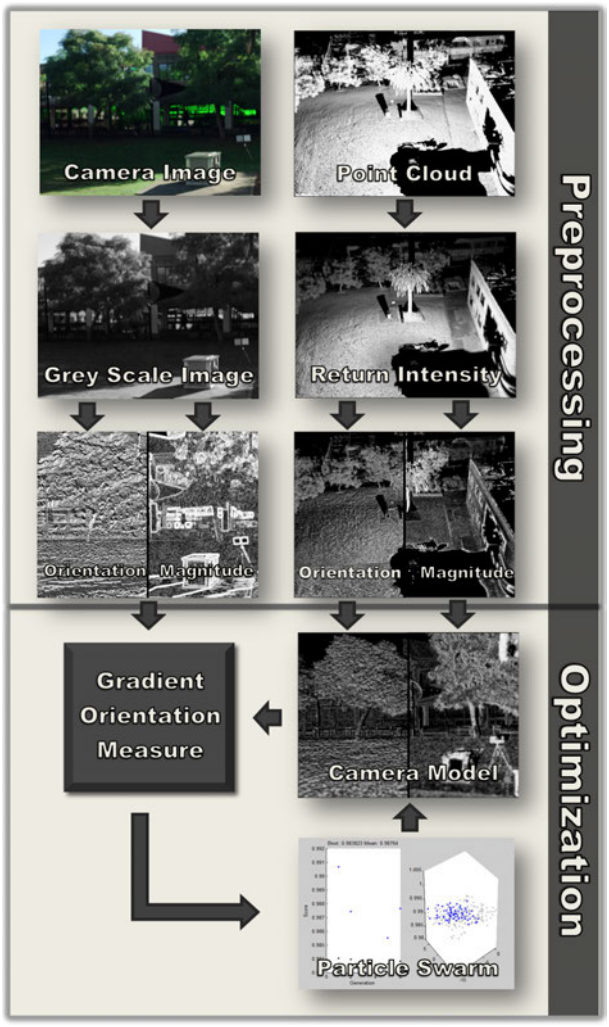

Fig. 2. Overview of alignment method

the estimated location of the camera using Equations 1 and 2.

$$x_{sphere} = arccos(\frac{z}{\sqrt{x^2+y^2+z^2}}) \tag{1}$$

$$y_{sphere} = arctan(\frac{y}{x}) \tag{2}$$

A sphere is used rather than the plane in image generation as with a plane points in front of and behind the camera can be projected onto the same location. Each point on the sphere has the 8 nearest neighbours to it calculated before the gradient is calculated using Algorithm 1.

As the gradient is dependent on the location of the camera it requires re-estimation every time the camera's extrinsic parameters are changed. However as this process is fairly computational expensive for the purpose of gradient calculation in our process it is assumed that $parameters_{initial} \approx parameters_{final}$. This assumption allows the gradients to be pre-calculated and gives a massive reduction in the computational cost. In our experiments this assumption did not appear to negatively impact the accuracy of the method.

Let
p_x, p_y, p_v be the current points x-position, y-position and intensity-value, respectively
n_x, n_y, n_v be the neighbouring points x-position, y-position and intensity-value, respectively
$g_x, g_y, g_{mag}, g_{or}$ be the gradient in the x-direction, gradient in the y-direction, gradient-magnitude and orientation, respectively

$g_x = 0$;
$g_y = 0$;
$g_{mag} = 0$;
for *neighbouring point n* **do**
  $x = p_x - n_x$;
  $y = p_y - n_y$;
  $v = p_v - n_v$;
  $g_x = g_x + \frac{vx}{8}$;
  $g_x = g_y + \frac{vy}{8}$;
  $g_{mag} = g_{mag} + |\frac{v}{8}|$;
end
$g_{or} = arctan2(g_y, g_x)$

Algorithm 1: lidar gradient calculation

B. Gradient Orientation Measure

The formation of a measure of alignment between two multi-modal sources is a challenging problem. Features in one source can be weak or missing from the other. A reasonable assumption when comparing two multi-modal images is that if there is a significant change in intensity between two points in one image then there is a high chance there will be a large change in intensity in the other modality. This correlation exists as these large changes in intensity usually occur because of a difference in the material or objects being detected; changes which generally affect most sensor modalities.

The gradient orientation measure (GOM) exploits these differences to give a measure of the alignment. GOM was inspired by a measure proposed by Josien P. W. *et al* in [10] for use in medical imaging registration. The presented measure however has significant differences as Josien P. W. *et al's* method is unnormalized, uses a different calculation of the gradients strength and is combined with mutual information.

GOM operates by calculating how well the orientation of the gradients are aligned between two images. For each pixel it gives a measure of how aligned their direction α is by Equation 3.

$$\alpha_j = cos(2(or_{1,j} - or_{2,j})) + 1 \tag{3}$$

Where $or_{i,j}$ is the orientation of the gradient in image i at point j. The difference in angle is multiplied by two as a change going from low to high intensity in one modality may be detected as going from high to low intensity in the other modality. This means that for two aligned images the two corresponding gradients may be out of phase by 180

degrees.

Sharper gradients represent features that are more likely to be preserved between images. The stronger gradients also mean that the direction of the gradient calculated will be less susceptible to noise and thus more accurate. This means that these points should be given an increased weight. This is accomplished by multiplying α by a factor μ. Where μ is simply the product of the two gradient magnitudes at that point as shown in Equation 4.

$$\mu_j = mag_{1,j} * mag_{2,j} \tag{4}$$

Where $mag_{i,j}$ is the magnitude of the gradient in image i at point j. Summing the value of all of these points results in a measure that is dependent on the alignment of the gradients. An issue however is that this measure will favour maximising the strength of the gradients present in the overlapping regions of the sensor fields. To correct for this the measure is normalised by dividing it by the sum of all of the gradient magnitudes, μ. This gives the final measure which is shown in Equation 5.

$$GOM = \frac{\sum_{j=1}^{n} \mu_j \alpha_j}{2 \sum_{j=1}^{n} \mu_j} \tag{5}$$

The measure has a range from 0 to 1, where if 0 every gradient in one image is perpendicular to that in the other and 1 if every gradient is perfectly aligned. Something of note is that if the two images were completely uncorrelated we would expect the measure to give a value of 0.5.

While this method achieves alignment by matching the orientation of gradients, it only requires that a small fraction of the points to undergo significant intensity changes in both modalities to achieve accurate registration. This occurs as if there is a strong feature that is only present in one modality its gradient strength and direction is either uncorrelated or only weakly correlated with the gradients in the other modality. This causes the GOM value for the points of this unmatched feature to give a value of around 0.5. This means that for most scenes the total GOM score will only be significantly greater then 0.5 when features present in both modalities are being correctly aligned.

C. Mutual Information

Mutual information (MI) is the most common technique used in multimodal registration. While the proposed method does not make use of it, so many of the methods in this field (including those GOM is compared against) make use of MI we felt it necessary to include a brief overview to highlight the differences with our approach.

MI is a measure of how similar one signal is to another. It was first developed in information theory using the idea of Shannon entropy [11]. Shannon entropy is a measure of how much information is contained in a signal and its discrete version is defined in Equation 6 [12].

$$H(X) = H(p_X) = \sum_{i=1}^{n} p_i log(\frac{1}{p_i}) \tag{6}$$

Where X is a discrete random variable with n elements and the probability distribution $p_X = (p_1, ..., p_n)$. For this purpose $0log\infty = 0$.

If two distributions are independent then their joint distribution is equal to the sum of their individual distribution. MI uses this to give a measure of the signals dependence by taking the difference between the independent and joint distributions of the entropy. It is defined in Equation 7.

$$MI(M,N) = H(M) + H(N) - H(M,N) \tag{7}$$

where $H(M,N)$ is the joint entropy which is defined in Equation 8.

$$H(M,N) = H(p(m,n)) = \sum_{m} \sum_{n} p(m,n) log(\frac{1}{p(m,n)}) \tag{8}$$

When used for registration purposes MI can be influenced by the total amount of information contained in images causing it to favour images with less overlap [13]. This is solved by using a *normalized mutual information metric* (NMI) defined in Equation 9.

$$NMI(M,N) = \frac{H(M) + H(N)}{H(M,N)} \tag{9}$$

In practice, for images, the required probabilities $p(M)$, $p(N)$ and $p(M,N)$ are usually estimated using a histogram of the distribution of intensity values. As NMI operates using these histograms of intensities, it makes no use of any spatial information or structure present in the image. This is the main difference between NMI and GOM. GOM works with the gradient of points rather than their intensity and so the value of neighbouring points and any structure present in the image is taken into account.

D. Camera model

To convert the lidar data from a list of 3-D points to a 2-D image that can be compared to the camera's images, the points are first passed into a transformation matrix

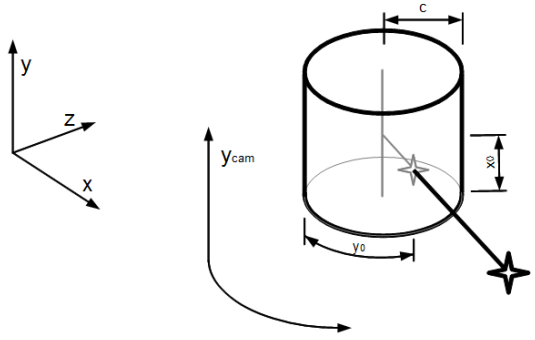

Fig. 3. Cylinder model used to represent hyperspectral camera

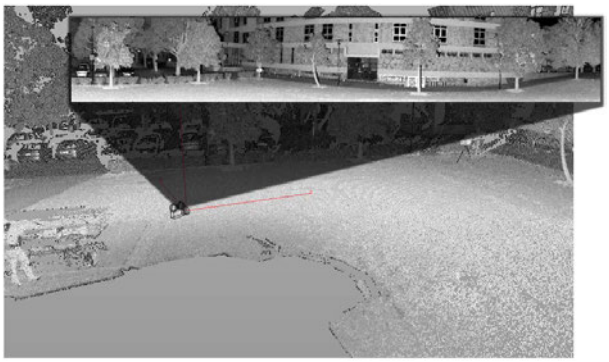

Fig. 4. Image generated by placing camera model in lidar point cloud

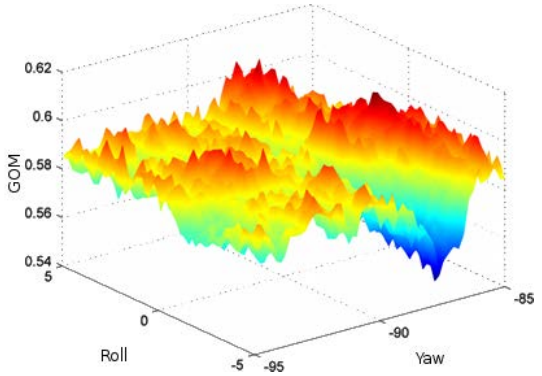

Fig. 5. Example of GOM values for changing roll and yaw

that aligns the camera's and the world axis. After this has been performed, one of two basic camera models is used. For standard cameras a pin-hole camera model is used as defined in Equations 10 and 11. For some of our datasets the images were obtained from a Hyperspectral camera. Hyperspectral cameras are sensitive to hundreds of different electromagnetic frequencies rather than just the three (red, blue and green) that standard cameras use. Hyperspectral cameras operate slightly differently to regular cameras as the image of the world is built up by rotating a single vertical line array. To account for this, a panoramic camera model that projects the points onto a cylinder must be used. A rough depiction of this is shown in Figure 3. This model projects the points using Equations 12 and 13 [14],

$$x_{cam} = x_0 - \frac{cx}{z} + \Delta x_{cam} \tag{10}$$

$$y_{cam} = y_0 - \frac{cy}{z} + \Delta y_{cam} \tag{11}$$

$$x_{cam} = x_0 - c\arctan(\frac{-y}{x}) + \Delta x_{cam} \tag{12}$$

$$y_{cam} = y_0 - \frac{cz}{\sqrt{x^2+y^2}} + \Delta y_{cam} \tag{13}$$

where

$\mathbf{x}_{cam}$, $\mathbf{y}_{cam}$ are the x and y position of the point in the image.

x, y, z are the coordinates of points in the environment.

c is the principle distance of the model.

$\mathbf{x}_0$, $\mathbf{y}_0$ are the location of the principle point in the image.

Δ**x** , Δ**y** are the correction terms used to account for several imperfections in the camera.

A depiction of how the camera model operates on a point cloud can be seen in Figure 4.

E. Generated image

As images are uniform grids, creating an image from the lidar can result in aliasing issues as discussed in Section III-A. It also results in a many-to-one mapping of the lidar points to pixels in the generated image resulting in a significant loss of information. To prevent these issues the generation of a traditional image from the lidar data is only done for visualisation purposes. Instead the points of the lidar are kept in a list and when they are projected using the camera model their position is not discretised. To get the matching points from the camera image linear interpolation is performed at the coordinates given by the lidar's point list.

F. Optimisation

Depending on the assumptions made by the camera model and the accuracy of the initial scan's position the problem has up to nine variables to solve. This search space is also highly non-convex with a large amount of local maximums. An example of the typical shape of GOM for a single scan alignment is plotted in two dimensions in Figure 5.

The fairly large range that the correct values can lie in coupled with the local maximums mean that simple gradient assent type methods as used by others to solve image lidar registration [6], [2] cannot be used here. To solve these problems particle swarm optimisation is used [15], [16]. Particle swarm optimisation works by placing an initial population of particles randomly in the search space. On each iteration a particle moves to a new location based on three factors: i) it moves towards the best location found by any particle, ii) it moves towards the best location it has ever found itself and iii) it moves in a random direction. The optimiser stops once all particles have converged. The process of registration is shown in Algorithm 2.

IV. Accuracy Estimation

The calibration of a system is of limited use without some indication of how accurate the calibration was. To provide an estimate of this accuracy we applied a bootstrap resampling technique [17] to the estimation of the extrinsic parameters.

In bootstrap resampling new samples are generated by randomly selecting elements from the original scans with replacement until the new sample is the same size as the original. These new samples are then run through the same optimisation process as the original until all of them have converged. It is assumed that the optimised parameters obtained are all independent of one another and that the collection of the samples parameters are normally distributed. This is used to find an estimate for the standard deviation

Let
$\boldsymbol{r^i(t)}$ be the position of particle i at time t
$\boldsymbol{v^i(t)}$ be the velocity of particle i at time t
$\boldsymbol{p_n^{i,L}}$ be the local best of the ith particle for the nth dimension
$\boldsymbol{p_n^g}$ be the global best for the nth dimension
$\boldsymbol{n \in 1, 2, ...N}$
$\boldsymbol{t}$ be the time
$\boldsymbol{\Delta t}$ be the time step
$\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ are the cognitive and social factor constants
$\boldsymbol{\phi_1}$ and $\boldsymbol{\phi_2}$ are two statistically independent random variables uniformly distributed between 0 and 1
$\boldsymbol{w}$ be the inertial factor

for *each itertaion* l **do**
  if $f(r^i(l+1)) > f(p^{i,L}(l))$ **then**
    $p^{i,L}(l+1) = r^i$
  end
  if $f(r^i(l+1)) > f(p^g(l))$ **then**
    $p^g(l+1) = r^i$
  end
  $v_n^i(t+\Delta t) =$
  $wv_n^i(t) + c_1\phi_1[p_n^{i,L} - x_n^i(t)]\Delta t + c_2\phi_2[p_n^g - x_n^i(t)]\Delta t$
  $r_n^i(t+\Delta t) = r_n^i(t) + \Delta t v_n^i(t)$
end

Algorithm 2: Particle swarm algorithm

of each parameter. For each scan 10 samples are generated as through trial and error this number was found to give a reasonable balance between accuracy and run time.

V. Results

The method was tested on two different datasets. The first dataset was obtained next to the Australian Centre for Field Robotics (ACFR) building. It made use of a rotating hyperspectral camera in combination with a high density Riegl lidar to make a single accurate scan of an area. The second dataset is the Ford Campus Vision and Lidar Dataset [18]. This dataset combines a series of Velodyne scans with a 360° panoramic camera mounted on top of a moving vehicle.

For all datasets the particle swarm optimiser was started with 200 particles and run until the particles all converged to within 0.01 degrees and 0.01 m in all dimensions of each other. This usually took 100 to 200 iterations

For each dataset GOM was compared against two other methods. The G. Pandey *et al* method that was presented in [2] using the source code they provided online. The second was a method we had previously developed for use in aligning scans of mine faces [3]. This method uses the same initialisation and optimisation as GOM but uses normalised mutual information (NMI) as its measure.

For the Ford dataset an implementation that made use of Levinson's alignment metric [1] was also compared against. This was optimised using the same framework as the GOM and NMI methods. Levinson's method was not tested on the ACFR dataset as a step in the method requires an average of all the images to be subtracted from each image. This means that Levinsons method cannot be run on a single image-scan pair.

Fig. 6. Setup used to collect ACFR data

The code was written in Matlab with mex files written in C and CUDA created for the generation of the lidar images and MI calculations. The code was run on a Dell latitude E6150 laptop with an Intel i5 M520M CPU and a NVS3100 GPU. Each function evaluation took around 0.01 seconds. The total runtime for the code was 3 to 10 minutes for the ACFR dataset and 10 to 20 minutes for a series of 20 image-scan pairs used in the Ford dataset.

A. ACFR experiment

A Specim hyperspectral camera and Riegl VZ1000 lidar scanner were mounted on top of a Toyota Hilux and used to take a series of four scans of the ACFR building from the park next to it, the setup is shown in Figure 6.

This dataset did not provide any intrinsic parameters for the camera, because of this it was assumed that the principle point was in the centre of the image and there was no distortion. The focal length parameter for the camera was estimated by hand and then optimized for at the same time as the extrinsic values. The code provided by G. Pandey *et al* was modified slightly to allow it to be used with a panoramic camera. This provided method did not optimise for focal length. To overcome this issue when the method was run a focal length was provided. As the actual focal length was unknown the focal length given by the most accurate of the other methods was used instead.

The search space for the optimiser was constructed assuming the following:

- The roll, pitch and yaw of the camera were within 10, 5 and 5 degrees respectively of the lasers.
- The cameras principal distance was within 30 pixels of correct (for this camera principal distance $\approx$ 770).
- The x, y and z coordinates were within one metre of correct.

B. ACFR results

No accurate ground truth is available for the ACFR dataset. To overcome this issue and allow a quantitative evaluation of the accuracy of the method to be performed 10 points in

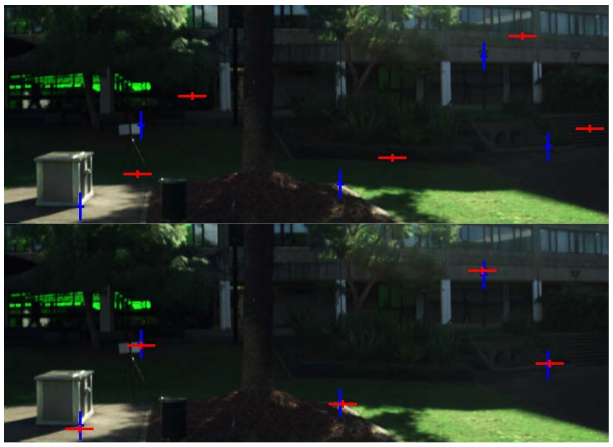

Fig. 7. Hand labelled points for a section of image-scan pair 1. Points on the lidar scan are labelled with red crosses and points on the image with blue crosses. The top image shows the initial calibration and the bottom image the result of GOM calibration

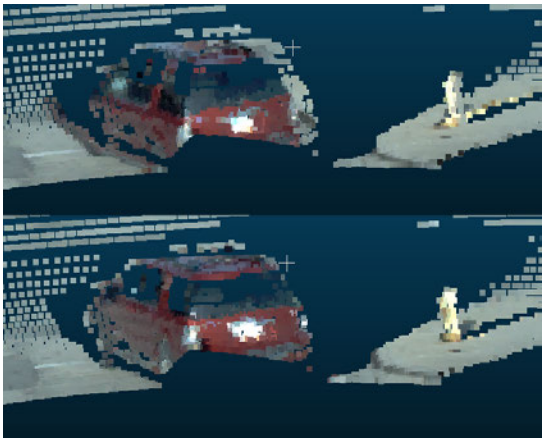

Fig. 8. A section of the Ford datasets velodyne scan coloured by the projected camera image during optimization. The initial alignment is shown on top, with the alignment after optimization below.

Scan	Inital	GOM	NMI	G. Pandey *et al*
1	59.2	3.0	128.3	61.1
2	49.7	5.9	9.5	43.6
3	13.0	7.4	6.1	13.7
4	24.1	3.2	10.8	6.4
Average	**36.5**	**4.9**	**39.9**	**31.2**

TABLE I
ACCURACY COMPARISON OF DIFFERENT METHODS ON ACFR DATASET. ALL DISTANCES IN PIXELS

each scan-image pair were matched by hand. An example of this is shown in Figure 7. An evaluation as to the accuracy of the method was made by measuring the distance in pixels between these points on the generated images. The results of this are shown in Table I

For this dataset GOM significantly improved upon the initial guess for all four of the tested scans. The NMI method usually also performed a reasonable job, however on the first scan set it converged to an incorrect solution. G. Pandey *et al* method performed poorly on this dataset usually giving results not significantly different than the initial conditions. This was probably due to the optimiser used quickly finding and becoming trapped in a local maximum near where it started. This would have occurred as the method was designed primarily for use on groups of scans simultaneously (as in the Ford dataset) relying on the combination of the scans to smooth out these local maximums.

C. Ford experiment

The Ford campus vision and lidar dataset is a dataset provided by G. Pandey *et al* [18]. The test rig was a Ford F-250 pickup truck that had a ladybug panaspheric camera and Velodyne lidar mounted on top of it. The dataset contains scans obtained driving around downtown Dearborn, Michigan USA. The intrinsic parameters of the cameras and velodyne had accurate calibration provided. It is also the dataset G. Pandey *et al*'s method was developed on making it the ideal dataset on which to test our method against this existing technique. A subset of 20 scans was chosen for use in testing the methods. The initial parameters used were those provided with G. Pandey *et al*'s method. The search space used for the particle swarm optimiser was as follows

- The roll, pitch and yaw of the camera were within 3 degrees of correct.
- The x, y and z coordinates were within 0.3 metres of correct.

D. Ford results

The Ford dataset does not have a ground truth to compare the results of the calibration against. However a measure of the accuracy can still be obtained through the use of the ladybug camera. The ladybug camera consists of 5 different cameras all pointing in different directions (excluding the camera pointed directly upwards). The extrinsic location and orientation of each of these cameras is very accurately known with respect to each of the other cameras. This means that if the calibration is performed for each camera independently the error in their relative location and orientation will give a strong indication as to the methods accuracy.

All five cameras of the ladybug were calibrated independently, an example of the process is shown in Figure 8. The error in each cameras relative position to each other camera is found and the average error is shown in Tables II and III.

In these scans it can be seen that the GOM method and Levinson's method exhibit fairly similar performance on this dataset. The NMI method tended to perform slightly worse with most errors slightly larger than the GOM or Levinson method. In this test the Pandey method gave mixed results as it had the most accurate distance estimates but the least accurate angle estimates.

E. Accuracy estimation

For the GOM method the Ford dataset was rerun using the bootstrapping method outlined in Section IV to give an estimate of the standard deviation of the error. An estimate of

	Roll	Pitch	Yaw	**Average angle**
GOM	0.1822	0.3923	0.2366	**0.2704**
NMI	0.2840	0.1233	0.7621	**0.3898**
Lev	0.1297	0.4461	0.6114	**0.3957**
Pan	0.9849	0.3371	1.0553	**0.7925**

TABLE II

AVERAGE ROTATIONAL ERROR BETWEEN TWO ALIGNED LADYBUG CAMERAS. ALL ANGLES IN DEGREES

	X	Y	Z	**Average distance**
GOM	0.0629	0.0520	0.0453	**0.0534**
NMI	0.2953	0.2226	0.1354	**0.2178**
Lev	0.0469	0.0485	0.1328	**0.0761**
Pan	0.0354	0.0353	0.0147	**0.0285**

TABLE III

AVERAGE TRANSLATIONAL ERROR BETWEEN TWO ALIGNED LADYBUG CAMERAS. ALL DISTANCES IN METRES

the actual standard deviation of the error in the cameras was generated by assuming the error was normally distributed and independent for each camera pair. These results are shown in Table IV.

	Roll	Pitch	Yaw	X	Y	Z
Error	0.1829	0.3746	0.3128	0.0900	0.0735	0.0838
Actual σ	0.1106	0.3392	0.3516	0.0773	0.0549	0.0453
Bootstrap σ	0.1987	0.5111	0.3430	0.0661	0.0725	0.0804

TABLE IV

AVERAGE STANDARD DEVIATION OF THE ERROR BETWEEN TWO ALIGNED LADYBUG CAMERAS

For this dataset the standard deviation that bootstrapping provided served fairly well as a prediction of the actual error. While there was some difference between the estimated and bootstrapped SD the errors were always of the same order of magnitude providing a fair indication of how reliable the alignment was. This demonstrates that the bootstrapping method can be used to provide an indication of how accurate a systems calibration has been.

VI. CONCLUSION

A method for calibrating the extrinsics of two multi-modal sensors has been developed. The method has been demonstrated by applying it to lidar camera calibration. This novel technique makes use of the orientation of the gradient of points. The method has been shown to achieve accuracies comparable or superior to state of the art techniques currently in use for calibration of Velodyne scanners on mobile platforms. It has also been shown to work well on aligning a single high resolution scan of an area. This calibration using a single scan is a challenging problem that most current solutions are unable to handle or commonly give poor results on. The method does not rely on any sensor or modality specific features making it appropriate for use in a wide range of situations.

ACKNOWLEDGMENT

This work has been supported by the Rio Tinto Centre for Mine Automation and the Australian Centre for Field Robotics, University of Sydney.

REFERENCES

[1] J. Levinson and S. Thrun, "Automatic Calibration of Cameras and Lasers in Arbitrary Scenes," in *International Symposium on Experimental Robotics*, 2012, pp. 1–6.

[2] G. Pandey, J. R. Mcbride, S. Savarese, and R. M. Eustice, "Automatic Targetless Extrinsic Calibration of a 3D Lidar and Camera by Maximizing Mutual Information," *Twenty-Sixth AAAI Conference on Artificial Intelligence*, vol. 26, pp. 2053–2059, 2012.

[3] Z. Taylor and J. Nieto, "A Mutual Information Approach to Automatic Calibration of Camera and Lidar in Natural Environments," in *the Australian Conference on Robotics and Automation (ACRA)*, 2012, pp. 3–5.

[4] R. Wang, F. P. Ferrie, and J. Macfarlane, "Automatic registration of mobile LiDAR and spherical panoramas," *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 33–40, Jun. 2012.

[5] H. Li, C. Zhong, and X. Huang, "Reliable Registration of Lidar Data and Aerial Images without Orientation Parameters," *Sensor Review*, vol. 32, no. 4, 2012.

[6] A. Mastin, J. Kepner, and J. Fisher III, "Automatic registration of LIDAR and optical images of urban scenes," *Computer Vision and Pattern Recognition*, pp. 2639–2646, 2009.

[7] S. Lee, S. Jung, and R. Nevatia, "Automatic integration of facade textures into 3D building models with a projective geometry based line clustering," *Computer Graphics Forum*, vol. 21, no. 3, 2002.

[8] L. Liu and I. Stamos, "A systematic approach for 2D-image to 3D-range registration in urban environments," *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, 2007.

[9] M. Corsini, M. Dellepiane, F. Ponchio, and R. Scopigno, "Image to Geometry Registration: a Mutual Information Method exploiting Illumination-related Geometric Properties," *Computer Graphics Forum*, vol. 28, no. 7, pp. 1755–1764, 2009.

[10] J. P. Pluim, J. B. Maintz, and M. a. Viergever, "Image registration by maximization of combined mutual information and gradient information." *IEEE transactions on medical imaging*, vol. 19, no. 8, pp. 809–14, Aug. 2000.

[11] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, "Mutual-information-based registration of medical images: a survey," *Medical Imaging, IEEE*, vol. 22, no. 8, pp. 986–1004, 2003.

[12] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[13] C. Studholme, D. L. Hill, and D. J. Hawkes, "An overlap invariant entropy measure of 3D medical image alignment," *Pattern recognition*, vol. 32, no. 1, pp. 71–86, Jan. 1999.

[14] D. Schneider and H.-G. Maas, "Geometric modelling and calibration of a high resolution panoramic camera," *Optical 3-D Measurement Techniques VI*, 2003.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[16] S. M. Mikki and A. a. Kishk, *Particle Swarm Optimization: A Physics-Based Approach*, Jan. 2008, vol. 3, no. 1.

[17] J. Kybic, "Bootstrap resampling for image registration uncertainty estimation without ground truth." *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 19, no. 1, pp. 64–73, Jan. 2010.

[18] G. Pandey, J. R. Mcbride, and R. M. Eustice, "Ford Campus Vision and Lidar Data Set," pp. 1–6, 2008.