Global Localisation in Real and Cyberworlds Using Vision

Nghia Ho, Ray Jarvis
Monash University, Australia
{nghia.ho, ray.jarvis}@eng.monash.edu.au

Abstract

This paper demonstrates how a robot can perform global localisation given a rich 3D map of an outdoor environment previously obtained via a laser range scanner and registered colour imagery. The robot is equipped with a panoramic mirror and is able to localise itself by matching sensory data from similar poses in the real and cyberworld. Some global localisation experiments are performed in a small outdoor test environment. The localisation is more accurate than a conventional civilian GPS.

1 Introduction

Global localisation is the capability of estimating the position of a robot given a model of the environment and sensor readings. Here the environment is fully known and can be modelled offline. This problem differs from the more recently active research of simultaneous localisation and mapping, better known as SLAM [Leonard and Durrant-Whyte, 1991]. In SLAM, the environment is initially unknown and the robot builds the map and localises simultaneously, usually in real-time. Although what has been achieved from SLAM is quite impressive, only rarely are there environments that are not known to some extent. For the majority of applications where a prior model can be obtained, global localisation has an advantage. Since the acquisition of the prior model is not under any real-time constraint, it can be processed offline with no restrictions on 'fine tuning' of the map. Global localisation can also solve the kidnapped robot problem, where a robot is randomly placed in an environment and is still able to localise, something not readily achievable with SLAM.

This paper presents work done on globally localising a robot in an environment modelled by a 3D laser scanner and registered coloured imagery using a panoramic mirror as the robot's sensor. The laser scanner/camera produces a very rich and dense model represented by a 3D point cloud with colour. This in itself presents a challenge to process such a vast amount of data. The choice of a vision sensor for the robot is attractive because it is relatively inexpensive, passive in operation and produces rich data, though requiring more processing than a time-of-flight sensor like a laser or sonar. The advantage of having a panoramic mirror that has a horizontal viewing angle of 360 degree is obvious.

[Jogan and Leonardis, 1999] presents a vision based localisation method using panoramic images. They build an indoor appearance map by taking images at every 60 cm. The images are then projected onto eigenspace to reduce the data. Our work follows the same concept of building an appearance map by sampling the environment at various locations, except we do it in a cyberworld created by a laser range scanner. The images obtained from the cyberworld undergo data reduction using Haar wavelet decomposition. There is no real-time constraint placed on our robot. It operates in a stop go fashion, so computational efficiency is not a critical issue. The same can be said for processing the appearance map as it is done offline.

The paper is organised as follows. Section 2 will give a brief overview of the robot platform used. Section 3 will discuss how the appearance map is produced from the point cloud. Section 4 will describe capturing and processing the panoramic images. Section 5 will describe how the processed panoramic images are matched against the appearance map. Section 6 will describe the global localisation algorithm. Section 7 contains experimental results. Section 8 will discuss the experimental results and future work needed. Finally, section 9 will conclude the paper.

2 Hardware

The robot platform is an ER1 robot as shown in Figure 1. It has a panoramic mirror mounted on top with a Canon Powershot S3IS digital camera. The digital camera offers more control over a common video camera such as aperture, shutter speed, exposure compensation and higher quality images (6MP). Since there is no real-time constraint the high resolution digital camera is a suitable choice, image acquisition time being approximately 1-2 seconds. The camera is remotely controlled via the open source gphoto2 program[1] .

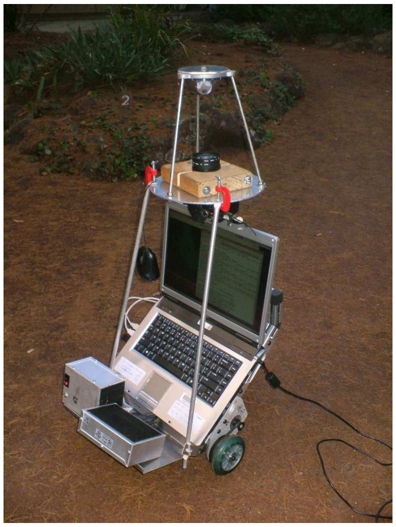


Figure 1: ER1 robot platform

3 Map Acquisition

The 3D cyberworld is created using a Riegl LMS Z420i terrestrial laser scanner and a high resolution digital camera. This scanner has an effective range between 2-800 metres, 360x80 degrees field of view. Colour information for each 3D point is obtained via a Nikon D100 camera mounted on top. The environment of interest is scanned at various locations to build a complete map. The dense point cloud from each scan are then registered together using in-house software based on the Trimmed Iterative Closest Point (TrICP) algorithm [Chetverikov *et al.*, 2002]. Each scan is registered to the next nearest scan with one scan chosen as the reference point. Our test environment was scanned at 5 different locations. The registered point clouds contained over 36 million points occupying about 850MB of disk space (uncompressed 32 bit floats). Given the size of the data, it is infeasible to render a scene at any given pose fast enough for localisation using particle filters. The frame rates achieved were well under 1 frame per second on a Nvidia 6800GT graphics card. For global localisation we need to generate many hypothesis, which can be anywhere in the 100,000s. It would be impractical to wait over 100,000 seconds for a single update. Clearly, we need to compress the dense 3D map into something more manageable.

The point cloud is post processed by representing as much of the environment using planes and texture, which are more memory efficient. The plane fitting was done using RANSAC as described in [Ho and Jarvis, 2007]. This reduced the point cloud to about 400MB (~53% compression over the raw point cloud), 160MB allocated to texture and the rest being points. Our test environment consisted a lot of vegetation that can't be fitted by planes well. This fusion of texture and point map is then sampled at point on a every 0.25 metre spaced rectangular grid and 10 degrees rotation to create the final appearance map database.

3.1 Panoramic Mirror in Cyberworld

To generate the appearance map in cyberworld, an accurate simulation of the panoramic mirror used by the robot is required. The mirror has the same profile found in [Chahl, 1998] with an elevation gain of 7. Figure 2 shows the mirror's parameters and co-ordinate system used in equation 1. The polar equation describing this mirror is:

$$r^4 = \frac{r_0^4}{cos4\theta_a} \tag{1}$$

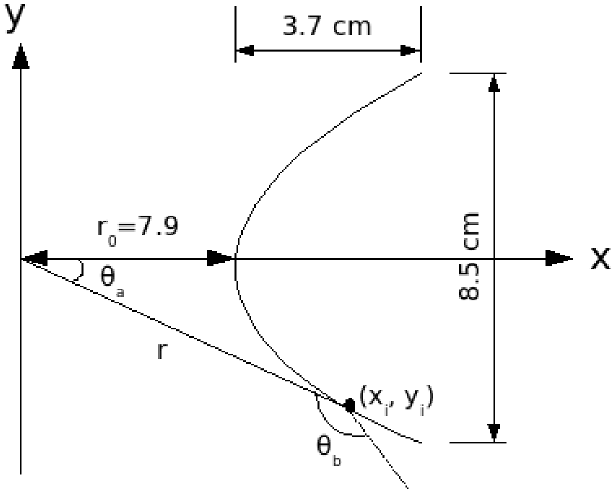
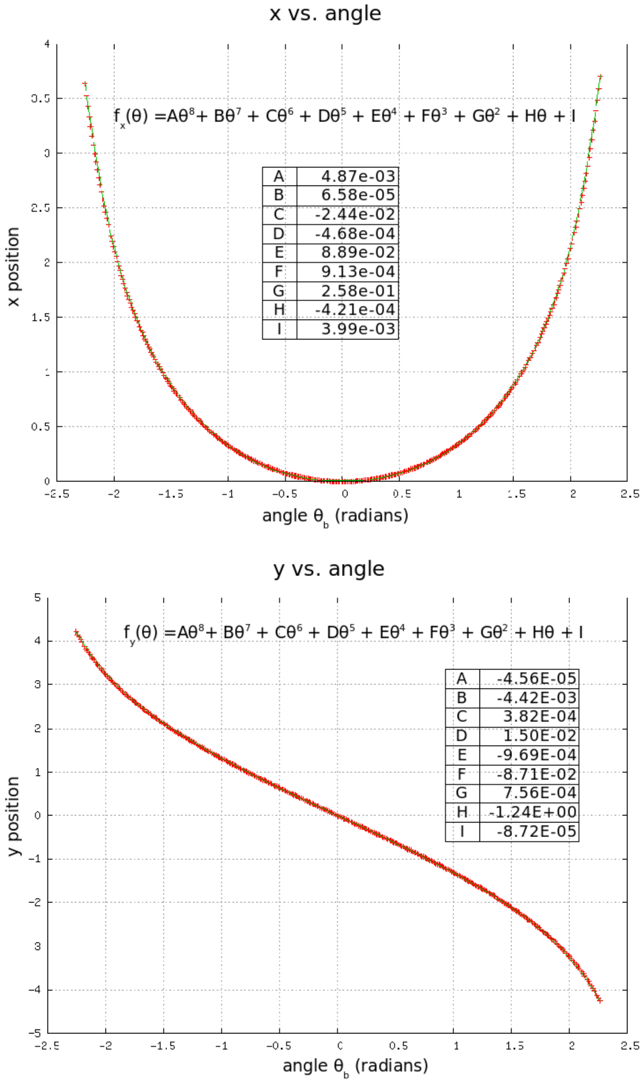


Figure 2: Mirror's parameters and co-ordinate system

The variable r_0 was found experimentally by running a Matlab script that optimised this variable until it fitted the profile of the mirror. To visually confirm the accuracy of the model, the shape of the profile was printed out on paper and compared to the mirror. Given a geometric model of the mirror we are interested in finding an equation to back project 3D points onto the mirror. Finding an analytical solution proved difficult so an approximation was used instead. This was done in simulation by projecting rays from the camera onto the mirror

[1] http://www.gphoto.org

and seeing where it intersected with the mirror and reflected. An 8th degree polynomial was used to fit a relationship between the angle of reflection ,θ_b, and location (x_i, y_i) where it occurs on the mirror. Figure 3 shows the fitted data for the back projection. All functions return x and y values relative to the apex of the mirror not the co-ordinate system in Figure 2. Given the 2D position on the mirror, finding the 3D position is trivial given an angle around the optical axis of the mirror.



A	4.87e-03
B	6.58e-05
C	-2.44e-02
D	-4.68e-04
E	8.89e-02
F	9.13e-04
G	2.58e-01
H	-4.21e-04
I	3.99e-03

A	-4.56E-05
B	-4.42E-03
C	3.82E-04
D	1.50E-02
E	-9.69E-04
F	-8.71E-02
G	7.56E-04
H	-1.24E+00
I	-8.72E-05

Figure 3: Back projection equations

The panoramic mirror is simulated in OpenGL using GLSL vertex/fragment shaders[2]. Figure 4 shows an example panoramic image generated from the cyberworld. Three grey vertical bars are added to represent the metal legs of the tripod occluding the mirror.

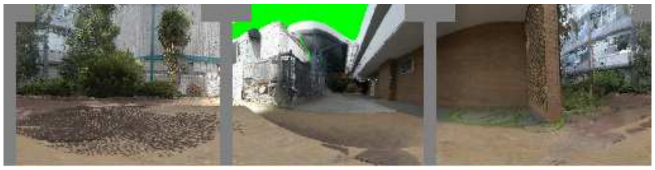


Figure 4: Example of a panoramic image in cyberworld.

4 Panoramic image

The panoramic images are captured at 640x480 instead of the full 2816x2112 because we end up using a downsample image for image matching, the high resolution is unnecessary. We plan to make use of the high resolution for a future idea that will be mentioned in the discussion. To unwrap the image, a linear mapping is applied as shown graphically in Figure 5. The resolution of the unwrapped image is 512x128 pixels.

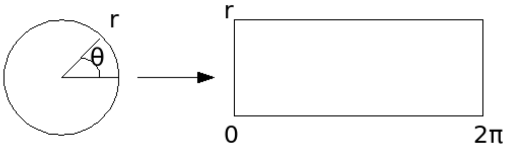


Figure 5: Unwrapping a panoramic image

4.1 HDR imaging

A common problem faced when taking panoramic images outdoor is the low dynamic range of the camera. This is particularly noticeable for scenes with a mix of dark and bright regions. The sky tends to oversaturate while areas cast with a shadow undersaturate. To produce an image with a larger dynamic range, we adopt standard techniques found in high dynamic range (HDR) imaging [Debevec and Malik, 1997]. Typically, multiple images of the same scene are taken at different exposure level and combined to produce a single HDR image. The HDR image range exceeds that of a standard monitor display and requires tone mapping to compress it for displaying purposes.

For our application, only two images are required. One at -2EV and the other at +2EV. This is adequate in highlighting very bright and dark areas. We use the open source program qtpfsgui[3] with the Reinhard02 [Reinhard, 2002] tone mapping operator to create the HDR images. The program was modified to run on the command line. An example is shown in Figure 6. This HDR operation takes between 1-3 seconds.

[2] http://www.opengl.org/documentation/glsl/

[3] http://www.qtpfsgui.sourceforge.net/

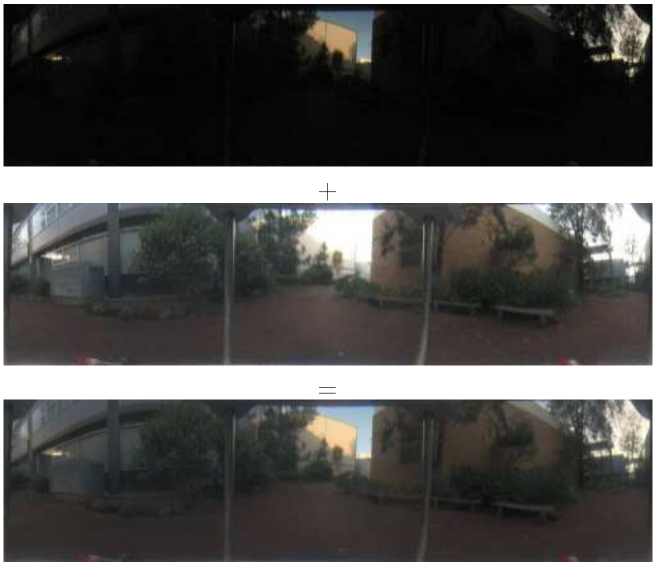


Figure 6: HDR panoramic image. Top is taken at -2EV, middle at +2EV, and bottom is the final HDR image

Algorithm 1 Haar Wavelet decomposition in 1D

```
procedure DECOMPOSE(A : array[0..h-1])
    $A \leftarrow A \;/\; \sqrt{h}$
    while h > 1 do
        for $i \leftarrow 0$ to $h-1$ do
            $A'[i] \leftarrow$ (A[2i] + A[2i+1]) / $\sqrt{h}$
            $A'[h+i] \leftarrow$ (A[2i] - A[2i+1]) / $\sqrt{h}$
        end for
        $A \leftarrow$A'
    end while
end procedure
```

5 Image matching

A multi-resolution Haar wavelet approach based on [Jacobs *et al.*, 1995] was applied to create a signature vector for each pose in the database. The pseudo code for the Haar wavelet decomposition in 1D is given in Algorithm 1. To decompose a 2D image, the rows are decomposed first then the columns. Haar wavelets are very fast to compute and simple to implement.

Our approach for creating the signature is as follows:

Algorithm 2 Wavelet signature

1. Downsample image to 512x128 (unwrapped panoramic)
2. Convert image to grey scale
3. Perform a 2D Haar wavelet decomposition
4. Keep the first 64x16 coefficients
5. Quantise coefficients to [-1,1] (negative, positive)

The signature is the quantised set of 64x16 coefficients, which is a vector of 1024 in length. The signature can be thought of as encoding light to dark and dark to light intensity transition at different resolutions. The first coefficient in the signature is the average intensity of the entire image (not used but kept anyway). This signature can be efficiently stored as an array of 1-bit values since the coefficients are quantised to two levels. Each signature takes up 128 bytes. Our database of over 100,000 poses totals to about 15MB. To measure the similarity between a query and target image, Algorithm 3 is used.

Algorithm 3 Similarity measure

```
procedure GETWEIGHT(query : array[64][16], target : array[64][16])
    for $y = 0$ to $15$ do
        for $x = 0$ to $63$ do
            $bin = GetLevel(x, y)$
            $histogram[bin] \leftarrow histogram[bin] + (query[x][y] = target[x][y])$
        end for
    end for
    $score = \prod_i sigmoid(b_i + w_i \times histogram[i])$
    $return\ score$
end procedure
```

The wavelet coefficients are grouped into 5 bins by GetLevel, which returns the resolution level of the coefficient at (x,y). This is the same weighting scheme used in [Zhuang and Ouhyoung, 1997]. A graphical representation is shown in Figure 7. Each colour represents a spatial bin at that particular level and are numbered on the top.

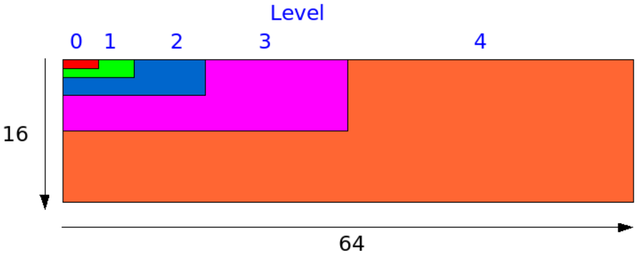


Figure 7: Wavelet coefficient weighting scheme

The weights and bias value were found by using logistic regression from a set of ground truth training data. The training set consisted of 996 images classified into two classes, match and mismatch. The match data set consisted of 6 images at known poses. While the mismatch were images randomly chosen from the appearance map that were at least 5 metres away from the 6 images. We found that treating each histogram bin independently as an individual probability produced better discrimination in comparison to doing

$$score = sigmoid\left(b + \sum_i w_i \times histogram[i]\right)$$

The weights used are summarised in Table 1.

Level	Bias (b_i)	Weight (w_i)
0	-6.153	0.371
1	-8.131	0.407
2	-13.175	0.284
3	-15.332	0.096
4	-38.797	0.080

Table 1: Weights used

6 Global Localisation

Although, given a panoramic image, we could simply perform a database lookup and find an estimated pose from the best scoring image, this would not be very robust, particularly for a large database. Since we have a mobile platform, continuity is exploited by matching a sequence of images instead. For this, a particle filter is employed to perform global localisation [Thrun *et al.*, 2001; Rekleitis, 2004]. 100,000 particles were initialised uniformly across the map. Areas with obstacles that the robot can't physically move to are automatically detected based on height thresholding of the point cloud. For each movement of the robot an update was performed. Updating 100,000 particles takes about 5 seconds on a Pentium M 1.8Ghz. A summary of the algorithm used is given:

Algorithm 4 Global localisation using particle filter

```
begin loop
  1. move robot
  2. apply motion model to particles
  3. capture panoramic image
  4. update particles' weight using GetWeight
  5. resample particles
end loop
```

A Gaussian noise model for wheel odometry was used for the motion model and was obtained empirically. The noise is a percentage of the total distance the robot has moved or rotated.

$$\varepsilon_t = N\left(\mu = -0.043,\, \sigma = 0.015\right) \quad (2)$$

$$\varepsilon_r = N\left(\mu = -0.005,\, \sigma = 0.045\right) \quad (3)$$

$$\varepsilon_\theta = \Delta\theta + \varepsilon_r \Delta\theta \quad (4)$$

ε_t and ε_r are the translation and rotation error respectively. N is a Gaussian function with mean μ and standard deviation σ. $\Delta\theta$ and Δt are relative rotation and translation the robot has performed and (x_i, y_i, θ_i) are absolute position and orientation.

$$\bar{X}_{i+1} = \begin{pmatrix} x_{i+1} \\ y_{i+1} \\ \theta_{i+1} \end{pmatrix} = \begin{pmatrix} x_i + (\Delta t + \varepsilon_t \Delta t)\, cos\,(\theta_i + \varepsilon_\theta) \\ y_i + (\Delta t + \varepsilon_t \Delta t)\, sin\,(\theta_i + \varepsilon_\theta) \\ \theta_i + \varepsilon_\theta \end{pmatrix} \quad (5)$$

The pose is estimated using the weighted mean:

$$\bar{X} = \sum_i X_i w_i \quad (6)$$

Alternative estimations are the best particle and robust mean. The best particle is the particle with the highest weight. One disadvantage of choosing the best particle is that it introduces discretisation errors. The robust mean does a weighted mean in a small window around the best particle. It has the advantage of selecting the mode of distribution (for multimodal distribution) and reduces discretisation error. However, if the particles have converged tightly then the weighted mean and robust mean produce results with insignificant numerical difference.

7 Experimental Results

The robot is placed at a random location and programmed to move in a straight line for 6 metres, updating every metre, with the exception of experiment 2 that ran for 8 metres. The position where it stopped is recorded by using a tape measure relative to natural landmarks in the environment as reference points. This experiment was performed 6 times at various locations. It is important to point out that this not a kidnap robot experiment which is something to be implemented in future work. The particle filter algorithm is manually reset every time the robot is randomly displaced. Figure 8 shows the particles for experiments 1 at each update stages, with the image from the panoramic mirror shown on the bottom right. As observed, the particles converge quickly.

To determine the error in localisation, we ran the global localisation 50 times since the particle filter is based on random sampling, hence the outcome will always be slightly different. For every run, the estimated pose is compared to the recorded ground truth using Euclidean distance. The distance from the ground truth and estimated position is defined as the error. The errors are aggregated and the mean calculated for each experiment. Table 2 summarises the results for all the experiments. Figure 9 shows all the ground truth location for all experiments.

Experiment no.	True (x,y,θ)	Est (x,y)	Est θ
1	(2.8, 0.8, 18°)	(2.5, 0.8)$\pm$0.3	$18^\circ \pm 2^\circ$
2 (8 metres)	(-6.5, -1.2, -77°)	(-6.3, 0)$\pm$1.2	$-74^\circ \pm 5^\circ$
3	(-7.0, 1.1, -70°)	(-6.3, 1.5)$\pm$0.8	$-76^\circ \pm 6^\circ$
4	(0.3, -0.8, 8°)	(-0.8, -0.7)$\pm$1.1	$10^\circ \pm 2^\circ$
5	(9.0, 2.8, 22°)	(8.1, 2.6)$\pm$0.9	$29^\circ \pm 7^\circ$
6	(-10.0, 4.1, 130°)	(-9.5, 2.7)$\pm$1.5	$129^\circ \pm 2^\circ$

Table 2: Summary of global localisation experiments. All values are expressed in metres.

8 Discussion

The particles converge rapidly towards the true position for all experiments. The error in localisation is, however, not small enough to navigate narrow paths reliably yet. This can be improved by fusing the robot with a close range sensor to detect obstacles and augment the particle filter. Despite that, the localisation is more accurate than a conventional civilian GPS, which can have uncertainties anywhere in the 10s of metres. Currently, we are extending the environment by adding more laser scans and increasing the environment size.

One of the major disadvantage of the appearance map method is that it disregards the 3D information from the point cloud. The point cloud is a precise metric model and it would be a waste to not exploit it. At the moment, we are investigating ways of incorporating the 3D information. One idea is to do feature matching across the panoramic image and the appearance map. For each pixel in the appearance map it can be traced back to a 3D point. Given a list of 3D points and their 2D correspondence it is possible to perform a triangulation, treating each matched pixel like a static beacon. This is where the high resolution images have an advantage. The high resolution allows more features to be detected while at the same time providing higher correspondence accuracy, assuming the appearance map is also high resolution. Another idea is to extract 3D information from the sequence of images as the robot moves. Matching in 3D space should be more robust but likely be more complex.

Occlusion and lighting conditions have not been considered and is something for future work. The wavelet signature has the potential to handle occlusion because it is made up of local spatial information at various resolutions. An occluded region would only introduce a bad match for that area only. The signature, in theory, is inherently robust to global illumination since it only considers dark to light and light to dark intensity transition. An image undergoing constant change in illumination would still have the same signature except for the first coefficient, which we don't use anyway.

One advantage of the the appearance map method is its generality. It can be extended to higher dimensions if computational efficiency is not an issue. One could even extend the localisation to 3D space, such as with a helicopter robot.

9 Conclusion

We have presented a global localisation method based on appearance maps generated from a laser range scanner. The localisation is more accurate than a conventional civilian GPS but less accurate than using a direct sensor such a laser. Although we have not exploited any 3D information from the point cloud it is something that we are investigating. Nevertheless, we have achieved successful results when working in 2D space only and can expect the accuracy to be further improved if fused with 3D information.

References

[Chahl, 1998] Javaan Singh Chahl. Panoramic surveillance system. *U.S. Patent 5,790,181*, 1998.

[Chetverikov *et al.*, 2002] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. In *Proc. International Conf. on Pattern Recognition, Quebec, Canada. IEEE Comp. Soc.*, 2002.

[Debevec and Malik, 1997] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 369–378, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[Ho and Jarvis, 2007] Nghia Ho and R. A. Jarvis. Large scale 3d environmental modelling for stereoscopic walk-through visualisation. *3DTV Conference*, 2007.

[Jacobs *et al.*, 1995] Charles E. Jacobs, Adam Finkelstein, and David H. Salesin. Fast multiresolution image querying. *Computer Graphics*, 29(Annual Conference Series):277–286, 1995.

[Jogan and Leonardis, 1999] Matjaz Jogan and Ales Leonardis. Panoramic eigenimages for spatial localisation. In *Computer Analysis of Images and Patterns*, pages 558–567, 1999.

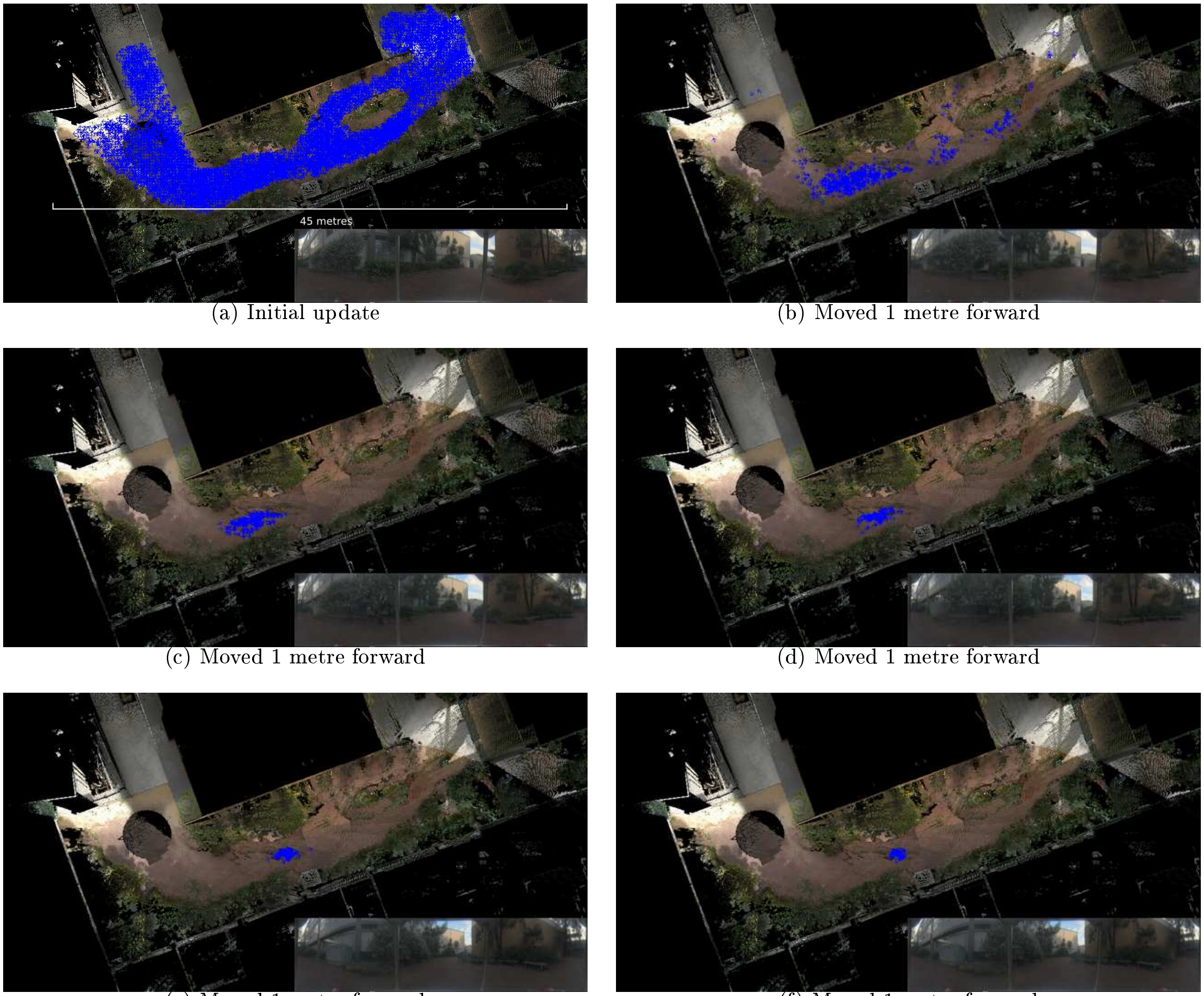
[Leonard and Durrant-Whyte, 1991] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on*, pages 1442–1447 vol.3, 1991.

[Reinhard, 2002] Erik Reinhard. Parameter estimation for photographic tone reproduction. *J. Graph. Tools*, 7(1):45–52, 2002.

[Rekleitis, 2004] Ioannis M. Rekleitis. A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University, 3480 University St., Montreal, Québec, CANADA H3A 2A7, 2004.

[Thrun *et al.*, 2001] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2001.

[Zhuang and Ouhyoung, 1997] Zheng-Yun Zhuang and Ming Ouhyoung. Novel multiresolution metrics for content-based image retrieval. In *PG '97: Proceedings of the 5th Pacific Conference on Computer Graphics and Applications*, page 105, Washington, DC, USA, 1997. IEEE Computer Society.



(a) Initial update

(b) Moved 1 metre forward

(c) Moved 1 metre forward

(d) Moved 1 metre forward

(e) Moved 1 metre forward

(f) Moved 1 metre forward

Figure 8: Results for experiment 1.

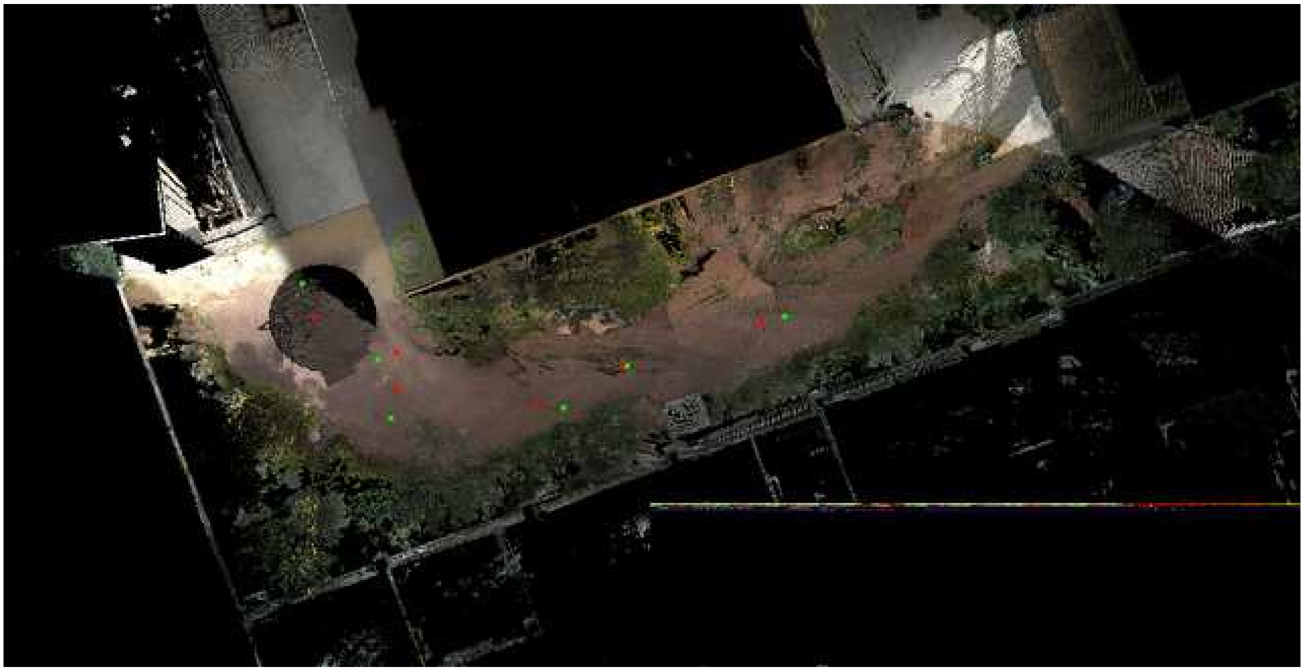


Figure 9: Ground truth and estimated position for all 6 experiments.