Empirical Calibration Method for Adding Colour to Range Images

Craig Robertson and Robert B. Fisher
Division of Informatics, University of Edinburgh
e-mail: cr@igs-pla.net

Abstract

This paper describes a cheap and easy method of capturing colour images in order to map textures accurately onto range data. By using an empirical mapping algorithm it avoids systematic errors arising from the use of incomplete parametric camera models.

1. Introduction

Acquisition of $2\frac{1}{2}$d or 3d measurements is a field where much work has been successfully done, see [4] for a survey of techniques. We are also aware that there are systems that can capture colour as well as range but not as inexpensively as the system described in this paper. Accurately mapping texture onto images is either technically difficult [6], expensive (for example colour scanners from 3dscanners, MENSI, K2T etc. are many thousands of dollars) or both. Since our group is involved in reverse-engineering we capture good quality range images with a resolution of up to 100 points/mm^2 with errors having SD of around 15 microns. The system we use does not give us the facility for capturing texture for these points however. This paper describes a simple method for sampling colour and accurately mapping it to acquired range points using a \$40 web-cam.

2. Equipment Configuration

The placement of the colour camera relative to the scanning head is shown in figure 1. In practice, one could use any type of camera mount, provided it remains stable relative to the scanning head.

Shown in figure 2 is the configuration of the scanner head relative to the camera. The camera needs no precise orientation with respect to the scanner co-ordinate system, as described in the next section.

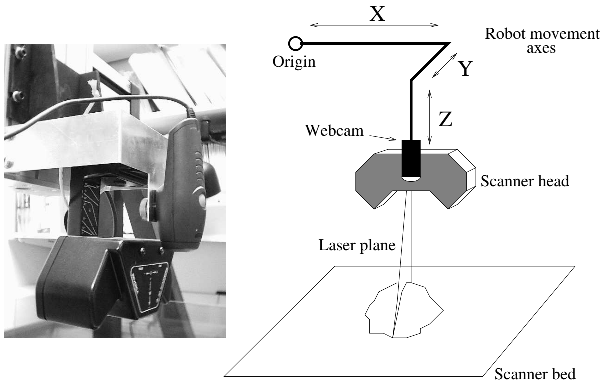


Figure 1. Equipment Configuration (a) Photo (b) Schematic

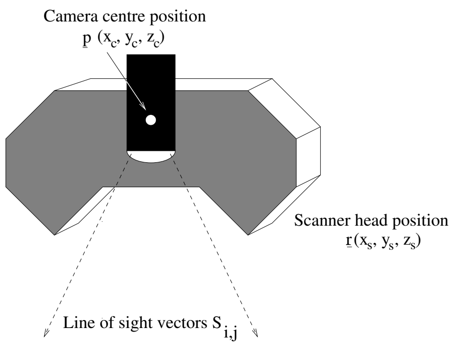


Figure 2. Equipment Configuration

3. Calibration and Mapping

3.1. Line of Sight Calibration

In order to calibrate the colour camera we use an unconventional method which generates the line of sight through every pixel in the camera image plane [8, 7, 9] by function interpolation over many landmark sites. Although this is reasonably expensive in disk space, it has the immediate advantage that there is no need to model any distortion phenomena because they are implicitly accounted for. The overall accuracy of the calibration is then only limited by the repeatability of camera positioning.

The method consists of the following steps:

1. The calibration grid is generated with black circles of known radius at a known separation. A single grey circle is approximately in the centre of the grid to remove ambiguity (fig.3).

2. Two images are taken of the calibration grid at distances separated by a known amount (say 50 mm). The first image is higher than the second.

3. Circle centres are computed for each image (shown as white dots fig. 3). If there are M in the first image and N in the second than clearly $M > N$ since the camera is farther away. This will naturally give rise to greater accuracy in the areas where corresponding circles are found.

4. We now fit six fourth order functions of the form:
   $f(i,j) = a_0 i^4 + a_1 i^3 j + a_2 i^2 j^2 + a_3 i j^3 + a_4 j^4 +$
   $a_5 i^3 + a_6 i^2 j + a_7 i j^2 + a_8 j^3 +$
   $a_9 i^2 + a_{10} i j + a_{11} i j^2 +$
   $a_{12} i + a_{13} j + a_{14}$
   as follows:

   - $f_1(u_{1,i}, v_{1,i}) = X_{1,i}$
   - $f_2(u_{1,i}, v_{1,i}) = Y_{1,i}$
   - $f_3(u_{1,i}, v_{1,i}) = Z_{1,i}$
   - $f_4(u_{2,j}, v_{2,j}) = X_{2,j}$
   - $f_5(u_{2,j}, v_{2,j}) = Y_{2,j}$
   - $f_6(u_{2,j}, v_{2,j}) = Z_{2,j}$

   Where $(u_{1,i}, v_{1,i})$ are the $i \in M$ circle centre positions in the first image, $(u_{2,j}, v_{2,j})$ are the $j \in N$ circle centre positions in the second image and $(X_{1,j}, Y_{1,j}, Z_{1,j})$ are the real positions of the circle centres in the first image (known relative to the grey circle), with their corresponding equivalents in the second image. Of course, we know the values for Z accurately because the distances from the calibration table were known *a priori*.

5. We now have interpolative functions to generate all of the line of sight rays for the pixels in the calibration grid. For a given position in the line of sight table, $L(p,q)$, we can generate $\underline{a}_{p,q} = (f_1(p,q), f_2(p,q), f_3(p,q))$ and $\underline{b}_{p,q} = (f_4(p,q), f_5(p,q), f_6(p,q))$ from the six functions above and thus generate the normalized line of sight vector between them $L(p,q) = \frac{\underline{a}_{p,q} - \underline{b}_{p,q}}{||\underline{a}_{p,q} - \underline{b}_{p,q}||}$.

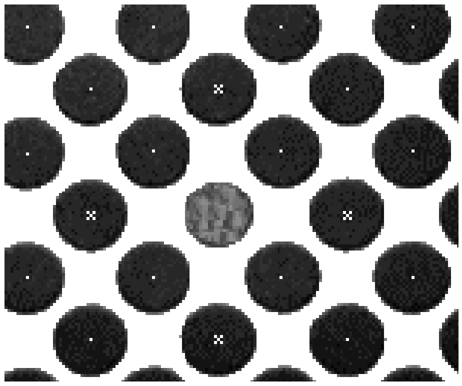


Figure 3. Centre-point Estimation. Note the Xs, which signify the row and column of the circle used for correspondence.

3.2. Colour Registration

In order to registred colour with range data we need to establish the camera centre with respect to the robot coordinate system. The translation vector for this can be found by scanning another calibration object, shown in figure 4. This object consists of mapping pins (radius 2.5 mm) laid out to cover most of the field-of-view of the colour camera and staggered in such a way that there is no overlap. In our case we use 21 pins. Each pin centre is found from the $2\frac{1}{2}$d range data using a sphere-finding variant of the RANSAC algorithm [2] (constrained between 2.0 mm and 3.0 mm radius). Once the sphere data is identified, it is cut out and a precise least-squares fit is performed to give us the 21 sphere centres $C_i, i \in [0, ..., 20]$ whose radius and centre are in the robot coordinate system.

To find the camera position, we now move the robot to a known, convenient position in the robot coordinate system, $\underline{p}$, and take a colour image as shown in figure 4. The spheres will appear in this image as circles and we can find their centres using a simple extraction algorithm (their centres do not overlap and they can be sorted by either X or Y coordinates to ensure a unique ordering).

The camera centre can now be found by back projecting from the known sphere centres along the lines of sight corresponding to the vectors in L where the circle centres are found, as shown in figure 4. Empirically, we have found this process to be stable provided several estimates are taken from sets of three non-colinear points, although relatively Z is the least stable. Standard deviations in mm on the shown estimates are ($\sigma_x = 0.2344, \sigma_y = 0.2305, \sigma_z = 1.0887$). This estimate for Z is, however, least important for mapping colour onto the range since the angular resolution for the texture is poor relative to the resolution of the range image.

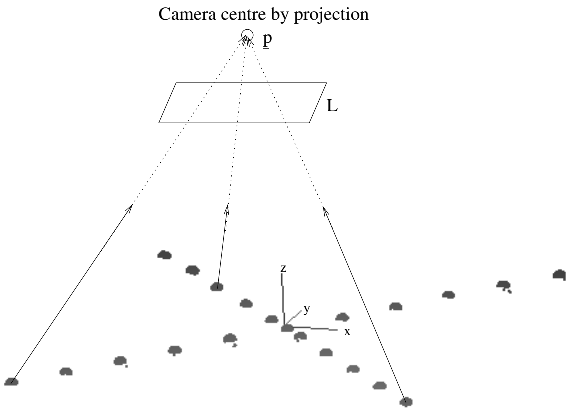


Figure 4. Camera Centre by Projection

3.3. Mapping RGB to XYZ

In order to map RGB to XYZ we have used a simple algorithm:

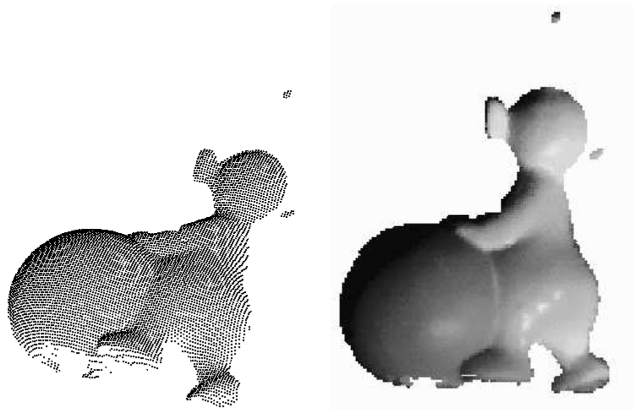
1. Compute the translation from the range sensor where the colour image was taken for calibration, $\underline{r}$ to the projected camera centre $\underline{p}$. This gives the translation vector between the robot position and the camera centre, $\underline{t} = \underline{r} - \underline{p}$. Now for any robot movement, we know the exact position of the colour camera centre.
2. Scan the object to obtain the range image.
3. Take a colour image at a known robot position, r_{new}.
4. Calculate the camera centre position, $\underline{p}_{new} = \underline{r}_{new} - \underline{t}$.
5. For all range points, $\underline{x}$ in the range image, calculate the vector between the camera centre and the range point, $\underline{q} - \underline{p}_{new} - \underline{x}$.
6. Search for the vector in the line of sight table L which is closest to this vector and use the colour of the corresponding image pixel.

This algorithm does not really make allowances for non-convex objects seen from the point of view of the computed camera centre, causing some range points on the back of objects to acquire colour from the front side. More complex (possibly more accurate) algorithms would be able to correct this.

4. Results

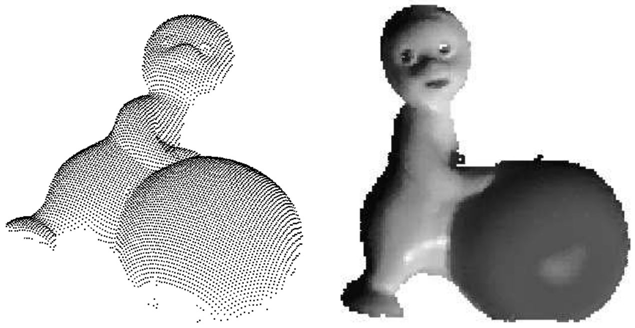
4.1. Plastic Figure

The results of mapping several colour images to range images is shown in figures 5 to 9. The figure itself is approximately 75 mm high and 45 mm wide. The range images are scanned at a resolution of $0.5mm \times 0.5mm$ in the XY plane. It can be seen that even small details are matched with good accuracy, for example eyes in figures 6, 7 and 8, the foot-pads in figure 5, 7 and 9 and the ear colouring in figure 8.



(a) Range image 1 (b) Attributed range image 1

Figure 5. Plastic figure 1



(a) Range image 2 (b) Attributed range image 2

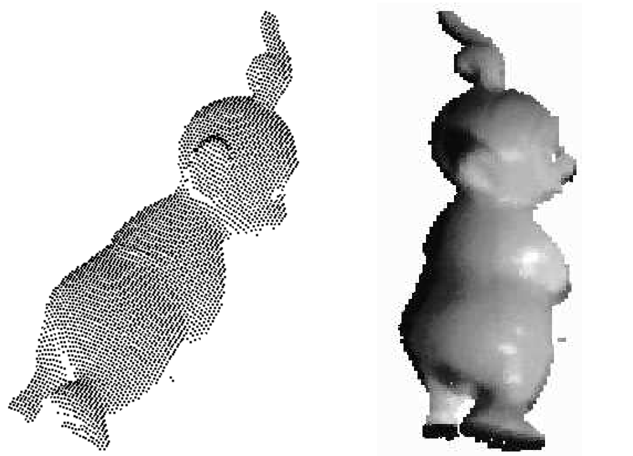
Figure 6. Plastic figure 1

Each image contains around 5,000 range points and the search takes around 30 seconds to complete a mapping of every range point to its corresponding colour. Since the search is local, this time is linearly scalable.

5. Conclusions and Further Work

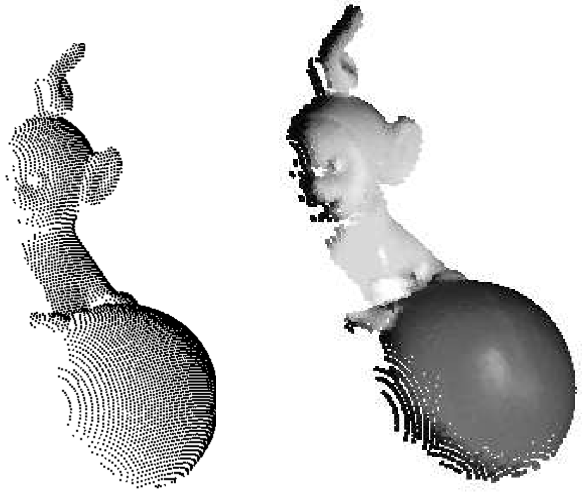
We have described a calibration method for mapping colour images to range points using a simple equipment configuration and two simple algorithms. In practice, the method takes very little time to both calibrate and perform the mapping.

Although the results we have obtained are good, the stability of the method depends on the exact identification of the projected camera centre, so a more robust method of



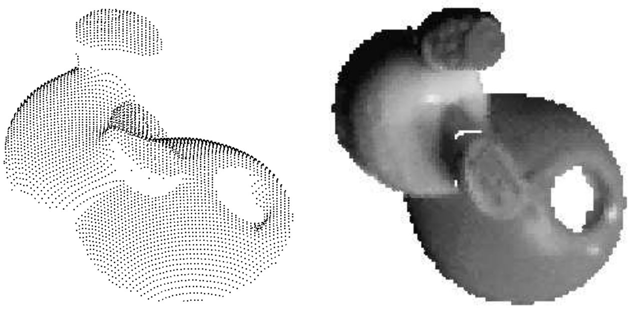
(a) Range image 3 (b) Attributed range image 3

Figure 7. Plastic figure 1



(a) Range image 5 (b) Attributed range image 5

Figure 8. Plastic figure 1



(a) Range image 6 (b) Attributed range image 6

Figure 9. Plastic figure 1

identifying this might be helpful. Obviously, using a much higher resolution camera (3Mpixel cameras are fairly common now) is also possible, to get higher quality textures. A more efficient mapping algorithm is clearly required, that could discriminate possibly ambiguous texturing between points.

Registration of coloured range images is a reasonably new field but is something that we are investigating currently. Pairs of range images have been registered [1], multiple images have been registered [3] and even several attributed range images have been registered [5] although only shapes with rotational symmetry were demonstrated, but the fusion of several unconstrained attributed range images has not so far been accomplished.

Acknowledgements

The work presented in this paper was funded by a UK EPSRC grant number GR/H86905.

References

[1] P. J. Besl and N. D. McKay. A method for registration of 3d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[2] R. C. Bolles and M. A. Fischler. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Technical Note 213, Artificial Intelligence Center, SRI International, Menlo Park, California*, 1980.

[3] D. Eggert, A. W. Fitzgibbon, and R. B. Fisher. Simultaneous registration of multiple range views for use in reverse engineering. *Proc. Int. Conf. on Pat. Recog*, pages 243–247, August 1996.

[4] H. R. Everett. Survey of collision avoidance and rangeing sensors for mobile robots. *Revision 1, US Navy Oceans Systems Center, San Diego, CA*, 1992.

[5] G. Godin, D. Laurendeau, and R. Bergevin. A method for the registration of attributed range images. *Proceedings of the Third International Conference on 3-d Digital Imaging and Modeling*, pages 179–186, May/June 2001.

[6] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999.

[7] E. Trucco and R. B. Fisher. Acquisition of consistent range data using local calibration. *Proceeding IEEE Int. Conference on Robotics and Automation*, pages 3410–3415, 1994.

[8] E. Trucco, R. B. Fisher, and A. W. Fitzgibbon. Direct calibration and data consistency in 3-d laser scanning. *Proc. British Machine Vision Conference, BMVC '94*, pages 489–498, September 1994.

[9] E. Trucco, R. B. Fisher, A. W. Fitzgibbon, and D. K. Naidu. Calibration, data consistency and model acquisition with a 3-d laser striper. *Int J. of Computer-Integrated Manufacturing*, 11(4):292–310, 1998.