A Method for Efficiently Re-estimating Camera Distortion Parameters

Ruby van Rooyen and Neil Muller

iThemba LABS, P. O. Box 722, Somerset West, 7129, South Africa

Abstract

At iThemba LABS, we use stereo vision techniques to accurately position the a patient for proton therapy. We chose to use conventional zoom lenses due to cost reasons. However these lenses have a high distortion factor. Also, due to maintenance work and other activities, we cannot assume that the lenses will not be disturbed between sessions. Thus we need to have a simple and efficient manner to recalculate the distortion parameters of the lenses before each treatment session.

1. Introduction

1.1. Problem Description

Proton radiotherapy is a useful treatment method for a number of lesions. The dose distribution properties of the proton beam allow for high doses to be delivered to the target volume while keeping the dose to the surrounding tissue to a minimum. Due to the high cost associated with this treatment, it is often reserved for lesions that are difficult to treat with conventional radiotherapy techniques, especially lesions close to critical structures. For more information see for example [1].

iThemba LABS has been involved with proton therapy for over ten years. Due to cost restrictions, iThemba LABS uses a fixed beam-line to deliver the proton dose, and uses a robotic manipulator to position the patient. The position of the patient during setup and treatment is monitored by a number of cameras and stereo techniques are used to calculate the patient's position at any time. A critical issue is the high positioning accuracy required. For further discussion on the system and some of the previous work on the vision aspects see [2], [3] and [4].

Amongst other things, for the degree of accuracy required, we need to have an accurate model of the distortion due to the lenses. Since the lenses used are conventional zoom lenses, distortion is a major factor in the system.

Furthermore, although we can accurately measure distortion before the cameras are mounted in the treatment room, the close proximity of the cameras to the the working area means that we cannot assume that the camera parameters do not change over time. Thus a number of additional checks are needed to ensure that the distortion model is correct and to update this model if need be. Since these checks will be done by the radiographers supervising the treatment, they need to be both simple and reasonably fast. A low degree of user involvement in this check is also desirable.

2. Obtaining initial distortion model

2.1. Distortion Pattern

In the design of an automated distortion correction technique, the main objectives of feature detection is accuracy, efficiency and robustness [5]. This can be achieved by keeping the complexity of the feature detection method as low as possible. Thus, we use circular features as the perspective projection of a circle is always a circle or an ellipse. Sub-pixel accuracy of the feature location can be achieved real-time and the complexity of the method is linear with the number of pixels [2].

Distortion is most clearly seen as the curvature of lines in a grid pattern. Consequently, we have designed a similar pattern for the computation of the distortion parameters (shown in figure 1). The distortion pattern is constructed by uniformly spacing 8 mm diameter circular targets in a rectangular grid. A group of three torus shaped targets are placed at various row positions in the upper left corner of the pattern. This allow for fast calculation of the orientation vector and the repetition of the pattern allows for the use of various zoom settings. To ensure high accuracy, the colours of the circles are selected such that a high contrast between the circles and the background is obtained [5].

2.2. Distortion Correction

2.2.1. Distortion Model

To get the corrected pixel coordinates of a feature that is to be used for 3D reconstruction, a lens model is used to compute the distortion parameters required [6]. This model is given by

$$(x_u, y_u) = (x_d, y_d) + \delta(\kappa, P)\,,$$

where (x_u, y_u) are the undistorted image coordinates, (x_d, y_d) are the distorted input coordinates and (κ, P) are the respective radial and tangential distortion parameters.

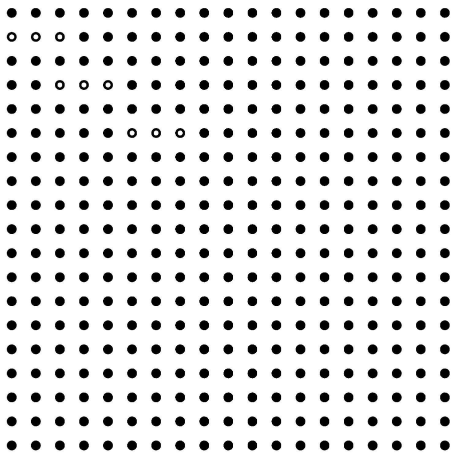

Figure 1: Distortion Correction Pattern.

The distortion factors given by:

$$\delta_x(\kappa,P)i = \overline{x}_d \sum_{l=1}^{\infty} \kappa_l \overline{r}_d^l + \left[P_1(\overline{r}_d + 2\overline{x}_d^2) + 2P_2\overline{x}_d\overline{y}_d\right] \left[1 + \sum_{l=1}^{\infty} P_{l+2}\overline{r}_d^l\right]$$

and

$$\delta_y(\kappa,P) = \overline{y}_d \sum_{l=1}^{\infty} \kappa_l \overline{r}_d^l + \left[2P_1\overline{x}_d\overline{y}_d + P_2(\overline{r}_d + 2\overline{y}_d^2)\right] \left[1 + \sum_{l=1}^{\infty} P_{l+2}\overline{r}_d^l\right]$$

where $\overline{x}_d = x_d - x_p$ and $\overline{y}_d = y_d - y_p$ with (x_p, y_p) the principle point and $\overline{r}_d^2 = \overline{x}_d^2 + \overline{y}_d^2$.

Although most distortion models ignore tangential distortion, it has been shown that, by allowing the centre of radial distortion (c_{xr}, c_{yr}) to be different from the principle point (x_p, y_p) of the lens, a good approximation for the decentring distortion is obtained [7]. Thus the distortion model becomes:

$$(x_u, y_u) = (x_d, y_d) + (\overline{x}_d, \overline{y}_d)\delta(\kappa) \text{ and}$$

$$\overline{r}_d^2 = \overline{x}_d^2 + \overline{y}_d^2 = (x_d - c_{xr})^2 + (y_d - cyr)^2 \,,$$

$$\delta(\kappa) = \kappa_1\overline{r}_d^2 + \kappa_2\overline{r}_d^4 + \dots \,,$$

where κ_1 and κ_2 are the first and second radial distortion parameters.

Numerical stability requires that the second order term κ_1 be found first [8], even if the higher order terms are required.

2.2.2. *Distortion Algorithm*

The distortion correction model will be used as a filter rather than being applied to the whole of the image for correction.

The off-line process used to calculate the distortion model parameters and the distortion centre is similar to the algorithms suggested in [9].

Distortion Algorithm:

1. Extract the distorted dot pattern [2].
2. Calculate the dot centroids in the distorted image. [2]
3. Find the point correspondence between the distortion centroid and the centroids in the corrected image.
4. Estimate an initial distortion centre and corrected image centre [9].
5. Calculate the expansion coefficients and image centre.
6. Adjust the distortion parameters to reduce the error.
7. Repeat from 5 until convergence.

An initial approximation of the distortion centre is estimated by interpolating the point where the curvature is zero. The expansion coefficients are calculated by finding the polynomial transform that orients the centroids of the grid dots to a straight line along the x and y direction:

$$y_{ij} = b_1^x x_{ij} + b_{0i}^x \,; 1 \le j \le K_i \,,\ 1 \le i \le L_x \,,$$

where (x_{ij}, y_{ij}) is the centroids of the dots on the corrected image, L_x is the number of rows in the grid and K_i is the number of dots in row i. This calculation is similar for the y direction.

Since the dot pattern conforms to a strictly rectangular grid, we assume that all the grid lines (along a particular axis) have the same slope. Thus b_1^x represents the horizontal slope for all grid lines in the x direction of the corrected image, and b_{0i}^x is the different intercept value of each grid line [9].

For the mapping from radial distortion space to corrected space, a non-linear least squares optimisation algorithm is used to fit the distorted grid lines to straight lines [9],[7].

After the model parameters are computed they must be used for real-time distortion correction. We construct a look-up tabel to map distorted input coordinate to the corrected output coordinates. This look-up table is calculated from an inverse mapping of the distortion correction model, and gives coordinates corresponding to pixel values in the distorted image [9].

3. Updating the Model

3.1. Vault environment

We use the same texture printed on a portable planar object. This printed example will be held up to each camera in turn. Obviously the translation and rotation of this object relative to the camera will not be consistent between the cameras and across sessions. We assume that the distortion parameters observed in the treatment room are close to those calculated outside the treatment room. As we will check the model frequently, we can alternatively use the previous session's calculations as the starting information.

We can easily extract straight lines from the resulting image, using the grid arrangement of the markers. Using these straight lines, we can easily test whether the distortion model is still valid.

Given that we only sparsely sample the lines, and that we are no longer assured that we have good coverage of the camera view, we can encounter problems using the purely line-based approaches to modelling the distortion correction. Noise issues due to the treatment room environment and damage to the cameras can also impact on the estimation if we restrict ourselves to purely straight lines. However, we can use the entire object to calculate the updated distortion parameters, using the method described in [10] and [11].

To do this, we first estimate the transformation parameters and then optimise over both the transformation and distortion parameters to minimise the error between the predicted image and the observed image.

3.2. Estimate of Transformation parameters

We know that the observed image is distorted. However, since we model the distortion as radial, and we know the approximate centre of distortion, we can safely assume that the distortion near this point is small. This allows us to use linear models to estimate the transformation parameters.

For our purposes, we assume a pinhole camera model. The transformation in homogeneous coordinates is given by

$$\lambda \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = \begin{bmatrix} f & s & u_x \\ 0 & \frac{f}{d} & u_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 & T_x \\ r_4 & r_5 & r_6 & T_y \\ r_7 & r_8 & r_9 & T_z \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 0 \\ 1 \end{bmatrix}$$

where $\begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}$ is a rotation matrix, $\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$ is the translation vector and $\begin{bmatrix} x_p \\ y_p \\ 0 \\ 1 \end{bmatrix}$ are the points on the distortion pattern. It is known that determining the full transformation from point correspondences on a plane is under-determined (although it was shown in [12] that it is possible given suitable additional geometric information). However, we only need the 2D projective transformation given by

$$\lambda \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = M \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ m_7 & m_8 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix}.$$

(The final matrix entry is set to 1 to ensure uniqueness). Given four point correspondences, this can be easily solved (see [13] for example).

In our case, since we are dealing with a regular grid, we do not need image registration. Given the image coordinates four points forming a single square of the grid, $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}, \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}$ and $\begin{bmatrix} x_4 \\ y_4 \end{bmatrix}$ we can solve for the transformation using the following system

$$\begin{bmatrix} \lambda_1 x_1 & \lambda_2 x_2 & \lambda_3 x_3 & \lambda_4 x_4 \\ \lambda_1 y & \lambda_2 y_2 & \lambda_3 y_3 & \lambda_4 y_4 \\ \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \end{bmatrix} = M \begin{bmatrix} 0 & d & 0 & d \\ 0 & 0 & d & d \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Since all other points on the grid can be expressed as $\begin{bmatrix} \alpha d \\ \beta d \end{bmatrix}$ where α, β are integers, this transformation is accurate up to an unknown translation (It is also accurate only up to a rotation of $90°$, but since the grid is symmetrical on both x and y axes, this is not important). Since the distortion correction is concerned with the matching of straight lines, this unknown translation does not effect any of the resulting calculations and can be ignored.

3.3. Optimisation

This estimate for the transformation is sensitive to noise (which can be compensated for by using multiple points) and will include some error since we neglect the effect of distortion. Thus we need to optimise both this transformation and the distortion parameters to ensure a good fit. Given a set of observed points $\mathbf{o}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, i \in [1, N]$ on the image, and the corresponding grid points $\begin{bmatrix} \alpha_i d \\ \beta_i d \end{bmatrix}$ where the α's and β's can easily be calculated by either counting points or calculating distances from the points used to obtain the transformation, we define the error function

$$E(\kappa, P, M) = \sum_{i=1}^{N} \left(\mathbf{o}_i - \left(M \begin{bmatrix} \alpha_i d \\ \beta_i d \end{bmatrix} + \delta(\kappa, P) \right) \right)^2$$

where $\delta(\kappa, P)$ is the distortion function defined earlier. The parameters of E are the distortion parameters and the projective transformation.

We assume that the distortion parameters, while not static, vary slowly with time. Thus we can use the already calculated distortion model as a starting point for optimisation. The estimated transformation parameters are used as a starting point for the true transformation parameters. Thus we can be reasonably confidant that optimisation will start from a position close to the global minimum. Conventional optimisation techniques can be used to obtain the final parameters with comparative ease. The usual problems associated with this type of high-dimensional optimisation are eliminated by ensuring that the initial estimate is close to the correct starting point.

We optimise over this sparse set of points and then use this result as a starting point for the optimisation over the full image. Since the initial optimisation is over a limited number of grid points, the overall complexity is fairly small and the calculation can be done quickly. The final optimisation should start from very close to the correct solution and converge quickly.

4. Conclusions

We show how we can easily update the distortion model for our cameras using a simple method. This allows us to detect and compensate for changes to the distortion parameters that may occur over time. As the distortion correction calculations can significantly impact on the overall accuracy of the system, the ability to easily correct these parameters if needed is a necessary component of the system design.

5. References

[1] S. Webb, *The physics of three-dimensional radiotherapy: Confromal radiotherapy, radiosurgery and treatment planning*, Insitute of Physics Publishing, Bristol and Philadelphia, 1993.

[2] R. van Rooyen, "Fast, robust detection of circular retro-reflective targets," in *Proceeding of the Fourteenth Annual Symposium of the Pattern Recognition Association of South Africa*, Nov. 2003, pp. 21–26.

[3] Evan de Kock, Brian O'Kennedy and Neil Muller, "Calibrating a stereo rig and ct scanner with a single calibration object," in *Vision, Modeling and Visualization 2002*, G. Greiner, H. Niemann, T. Ertl, B. Girod and H.-P. Seidel, Eds., 2002, ISBN 3-89838-034-3.

[4] Evan de Kock, Neil Muller, Denys Maartens, Jan van der Merwe, Deon Muller, Ruby van Rooyen, Andre van der Merwe, Jan Eksteen, Neil von Hoesslin, Dirk Wagener and Jan Hough, "Integrating an industrial robot and multi-camera computer vision systems into a patient positioning system for high-precision radiotherapy," in *35th International Symposium on Robotics*, Mar. 2004.

[5] Jun-Sik Kim, Ho-Won Kim and In-So Kweon, "A camera calibration method using concentric circles for vision applications.," in *The 5th Asian Conference on Computer Vision*, Jan. 2002.

[6] Sing Bing Kang, "Semiautomatic methods for recovering radial distortion parameters from a single image," Tech. Rep. CRL 97/3, Digital Equipment Corporation, Cambridge Research Lab, May 1997.

[7] G. Stein, "Lens disortion calibration point correspondences," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 602–608.

[8] Ben Tordoff and David W. Murray, "Violating rotating camera geometry: The effect of radial distortion on self-calibration," in *15th International Conference on Pattern Recognition*, Anil K. Jain, Svetha Venkatesh and Brian C. Lovell, Eds. IEEE, 2000, pp. 1423–1427, IEEE Computer Society.

[9] Chao Zhang, James P. Helferty, Geoffrey McLennan and William E. Higgins, "Nonlinear distortion correction in endoscopic video images," in *2000 IEEE International Conference on Image Processing*, Sept. 2000, pp. 439–442.

[10] T. Tamaki, T. Yamamura and N. Ohnishi, "Correcting distortion of image by image registration," in *The 5th Asian Conference on Computer Vision*, Jan. 2002.

[11] T. Tamaki, T. Yamamura and N. Ohnishi, "Unified approach to image distortion," pp. 584–587, 2002.

[12] V. Fremont and R. Chellali, "Direct camera calibration using two concentric circles form a single view.," in *International Conference on Artifical Reality and Telexistence*, Dec. 2002.

[13] R. Szeliski, "Visual mosaics for virtual environments," *IEEE Computer Graphics and Applications*, vol. 16, no. 2, pp. 22–30, Mar. 1996.