# Collective Calibration of Active Camera Groups

Paul Chippendale, Francesco Tobia
*ITC-irst, 38100 POVO (TN)*
chippendale@itc.it, tobia@itc.it

## Abstract

*Until recently, traditional approaches to the task of camera calibration have relied on the use of accurate grid patterns, or strategically placed targets. Such approaches can prove time consuming and often require expert supervision. For ambient intelligence type applications, it is unwise to rely on the consistency of camera positions and orientations; consequently a fully autonomous camera calibration procedure is preferred.*

*In this paper we present an auto-calibration system for the estimation of the extrinsic properties of active cameras lying within indoor, self-observable groups. Neither prior knowledge of camera locations, nor a calibration-pattern is required, just a few basic parameters concerning the physical appearance of the cameras. Groups of active cameras can be calibrated within minutes, by exploiting their precise control mechanisms. As no supervision is required, camera deployment configurations can be changed or new cameras added easily.*

## 1. Introduction

In this paper we are concerned with the problem of calibrating collective groups of active cameras, which are free to pan, tilt and zoom over a wide range, but which remain in one geographical location. A camera is said to be fully calibrated if both the intrinsic and extrinsic parameters are known. This paper will not address the acquisition of the intrinsic calibration, although we have already developed this capability in a complementary paper [2], where an automatic method for deriving the intrinsic parameters of active cameras is described. When used in conjunction with the algorithms described in this paper, a complete auto-calibration system can be realised. As the intrinsic parameters of the active cameras are static, they need only be acquired once (either in situ or in the lab) and have no bearing on camera network recalibration based on camera relocation or reorientation.

Until recently, traditional approaches to the task of camera calibration have relied on the use of accurate grid patterns, or strategically placed beacons (such as a checker board [1]). The multi-camera approach introduced by Svoboda [3] can be used to calibrate groups of static cameras and conventionally requires a calibration video to be recorded where a bright light or similar point source target is moved around the room which is observed simultaneously by at least two cameras. This technique however, can be time consuming, and only provides good calibration in the room volume where the target object was well observed. Conversely, our calibration technique maintains its calibration for the entire room space because the calibration targets are widespread. We propose a self-calibration approach using multiple observations from spatially dispersed cameras viewing the same scene from greatly differing aspects. This type of deployment of active cameras occurs frequently in a variety of circumstances ranging from indoor surveillance situations to the monitoring of sporting events. In this paper we will concentrate more closely on ambient intelligence applications and more specifically on a scenario where a group of more than two active cameras are located within a room. In a home environment, cameras are easily moved or knocked, resulting in a loss of extrinsic calibration. Furthermore, as it may take several attempts to obtain an adequate video surveillance coverage within the environment, cameras may have to be moved around the observation space many times.

For typical people tracking applications, it is not always necessary to have single pixel accuracy for camera calibration and often the benefits of swift self-calibration far outweigh any tiny losses in location precision.

This paper is organised as follows: In Section 2, our method for acquiring the extrinsic calibration of multiple active cameras is explained. In Section 3, the system's calibration accuracy will be illustrated using a human sized calibration target placed randomly around the room, and then finally in Section 4, some

0-7803-9385-6/05/$20.00 ©2005 IEEE

conclusions about the system's advantages and drawbacks will be made.

## 2. Calibration Approach

The sensors used during our experiments were the Sony SNC-RZ30P network cameras (see Figure 1). Although there is no constraint placed upon the system stating that all of the cameras should be identical, it is necessary that a few basic physical characteristics for each camera type are known prior to self-calibration. This information relates to parameters such as the physical outer lens diameter (or any other physical characteristic of the camera – e.g. a frontal image of the camera from literature is sufficient), the field of view, and PTZ constraints; all easily obtained from manufacturer's specifications.

**Figure 1: Sony SNC-RZ30P camera**

The deployment of the cameras within the environment can be arbitrary, with only a single minor constraint; each camera must be able to see at least one other to form an observable chain, i.e. visually linking all of the cameras. In reality, this constraint is trivial, because to achieve suitable video coverage of an environment, cameras must inherently be placed in favourable positions which are conversely viewable from most of the environment.

### 2.1. Mutual Camera awareness

Rather than following the conventional approach of using a calibration object, our system uses the physical and mechanical properties of the cameras themselves to calculate separation distances and angular metrics. From the manufacturer's specifications, the diameter of the Sony SNC-RZ30P lens (i.e. the outermost visible glass surface which we will be using as a calibrated target) is 30mm and the angular rotation of a single pan point (i.e. its smallest possible movement) is 0.001211 radians or 0.069 degrees.

To more effectively describe the calibration procedure, a test case of four cameras, designated with tags 197, 198, 199 and 200, will be used in this paper.

Initially, all of the active cameras are instructed to zoom, pan and tilt to a predetermined starting configuration depending on their current role in the calibration process. The first camera in the list (in our case 197) is designated the *watcher*, and will attempt to locate as many of the other cameras as possible, whilst all of the other cameras (here: 198, 199 and 200) become the *movers*, having their tilt values initialised to 0º whilst commanded to rotate their pan angles at a controlled rate to become temporally visible. As a *mover*, a camera is instructed to modify its pan setting from full left to full right (-135º to +135º for the Sony) and then back again; and so forth until it is commanded to do otherwise.

In the case of a *watcher*, of which there can only be one at any given time, its pan and tilt settings are modified in accordance with a search pattern. Due to the field of view limitations of conventional PTZ Cameras (at unity zoom the SNC-RZ30P has 45˚ width by 34˚ height) it is highly unlikely that all of the *movers* would lie within the initial field of view of each *watcher*. Consequently, a panning/tilting strategy has been devised that adjusts the *watcher's* pan and tilt values systematically in order to try to locate the *movers* efficiently. In addition to field of view considerations, the selection of the *watcher's* pan and tilt search pattern relates to the physical dimension of the entire camera group as well as to the maximum pixel resolutions of the CCD sensors. Given these factors, the maximum distance that another camera's lens (or movement activity) can be detected, given the constraint that a lens cannot be detected if it appears less that $s$ = 5 pixels (where $s$ is the minimum recognisable lens size in pixels) in the *watcher's* field of view, is defined in Equation 1: where $l$ is the linear distance between cameras, $d$ is the camera lens diameter, $w$ is the horizontal pixel resolution of the sensor and $\alpha$ and $\beta$ are the horizontal and vertical angular fields of view of the sensor.

**Equation 1: Maximum Camera separation**

$$l = \frac{(wd/s)}{(2\tan(\theta_\alpha/2))}$$

In our case, given that the maximum working resolution of the SNC-RZ30P is 640x480 pixels ($w$=640) and the lens diameter is $d$=3cm, if we set the search zoom level to unity (making the field of view $\theta_\alpha$= 45˚), then the maximum distance of separation that a moving camera can be detected $l$=463 cm. However, as we are dealing with active cameras which have an optical zoom capability, the simplest solution to extending the detectable distance of camera separation is by increasing the optical zoom by a factor of 2.

Hence, by doubling the optical zoom to 2, the range rises to *l*=927cm. On the down side, this decreases the horizontal and vertical fields of view also by a factor of two. Consequently, it may take four times the number of search sectors to locate all of the cameras, depending on their relative locations.

For a camera configuration with an estimated maximum separation distance under 463cm, a zoom factor unity can be employed, leading to the creation of a pan sector sequence for the *watcher* as follows: 0˚, 40.5˚, -40.5˚, -81˚, 81˚, 121.5˚, -121.5˚, -135˚ and 135˚. This covers the 270˚ operational pan range in 40.5˚ steps to ensure a 10% visual overlap of the 45˚ field of view.

The tilt value of the *watcher* is controlled in a similar fashion to that of the pan, although its value is only adjusted upon the unsuccessfully completion of the entire pan sector scan. The tilt sector is initially set to a slightly tilted-upward looking orientation $\theta_\beta/4$, working on the hypothesis that all of the cameras are roughly situated at the same height above the ground plane. This tilted-upward looking orientation is aimed at minimising possible motion noise from users in the room during calibration and thus speeding up the calibration process. In practice, however, room noise does not bring about calibration failure, as the tell-tale *mover* signatures do not occur naturally in the environment. Allowing for the same 10% overlap of coverage, creates a tilt sector search pattern as follows: 8.5˚, -22.1˚, 39.1˚.

The searching procedure begins with the *watcher* assuming its default pan and tilt states (in our case as the estimated maximum separation distance is under 450cm, the pan is 0º and the tilt is 8.5º). All of the *movers* are then instructed to oscillate left-right-left. To determine whether one of the *movers* can be seen by the *watcher*, signed Motion History Images (MHIs) [4] are initiated immediately after each sector adjustment. The intensity values in the MHIs are indicative of the time at which pixels last witnessed motion and thus represent a temporally decaying composite of motion observed over time. Positive and negative MHIs are used in combination in order to detect both the velocity and direction of candidate *mover's*. In Figure 2, red pixels indicate dark to light temporal activity (from the Positive MHI) and the green pixels illustrate light to dark activity.

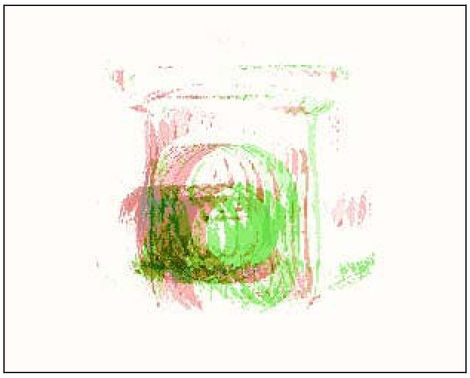

**Figure 2: Compound MHI of a *Mover* as seen from the *Watcher (Red – Positive MHI, Green - Negative)***

As the rotational/panning rate of the *movers* is known (through feedback from the *mover's* driver servo), if a *mover* is present in the field of view of the *watcher*, in only a short period of time (typically the time taken for a complete left-right-left rotation cycle) the compound MHI will reveal a distinctive motion pattern. The target signature is that of a green leading area followed by a red wake - created by the dark lens moving across the pale coloured camera body during *mover* rotation (see Figure 2 for an example).

Upon the detection of a potential *mover* in the compound MHI, its activity density is also assessed to see if it lies above a threshold. Where the density and movement pattern satisfies the target criteria, an AreaZoom[1] is conducted by the *watcher* in order to take a much closer look at its target. After successive AreaZooms, a zoom level near to the maximum optical limit of the camera will be reached. At this stage a *mover* is verified and identified (in terms of its ID tag) through the methodical pausing of all of the *movers* in the list until the blob under observation decays in harmony with the request. In the event of a positive identification, the *mover* is not restarted during the current *watcher's* cycle.

Whether the detection was successful or not, the *watcher* zooms out again and continues to look for further *movers* in the list that are visible in the same field of view. After a default time period, if no potential *movers* are located in the current sector, the *watcher's* position is adjusted to observe the next.

The detection procedure continues around all of the sectors until all *mover*s have been located or the entire search pattern has been expended. This procedure is then repeated in-turn, until all cameras have played the role of *watcher*.

[1] AreaZoom is a command function for the SNC-RZ30P that adjusts the pan, tilt and zoom values according to a rectangle of pixels in the current image

Once completed, the angular displacement of each-camera-to-each-other, and to its own zero pan and tilt position, should have been derived. In the event that a complete set of angular data has not been gathered (possibly due to difficult lighting conditions or occlusions), given that sufficient other observations were successful, it is a straightforward process to derive the missing information using simple geometry.

## 2.2. Inter-angular & Inter-distance Estimation

Extrinsic camera parameters define the relationship between the camera reference system and world coordinates. In our work, we chose the world coordinates solidal to one of the camera orientations at rest (*i.e. a reference camera*). This choice is arbitrary, but for our purpose - multi-camera system interaction - the mainly goal is to obtain the relationship (relative position and orientation) between the cameras, despite the 3D representation. If needed, we can change the system to a more meaningful one (i.e. a room corner) by localizing in 3D space the new oriented origin and applying the inverse transform (thus furnishing us with a complete extrinsic calibration).

Since we are assuming the cameras are in an observable chain, the whole calibration can be obtained by solving the relative position and orientation between each two "linked" cameras. Roto-translation matrices relative to the reference can be obtained by opportunely concatenating these results.

The calibration problem is reduced to the estimation of the roto-translation matrix between two cameras which can see each other. We can further split the problem in to two, since the computation of the rotation and the translation are independent.

### 2.2.1. Camera-Camera Distance

The distance between the two cameras can be obtained using geometrical constraint and our a priori knowledge about the two object's shape - as stated previously we choose to exploit the visible lens.

Camera settings at the end of the awareness stage grant us sufficient zoom to quickly detect lens position and diameter using circle Hough transforms. In this way it is possible to iteratively point and zoom each camera at the other, leaving the centre of the target lens in their principal point (see Figure 3). In common indoor environments, in which cameras are placed 2-5 meters away, this procedure ends with the lens target being about 50-100 pixel radius at full zoom.

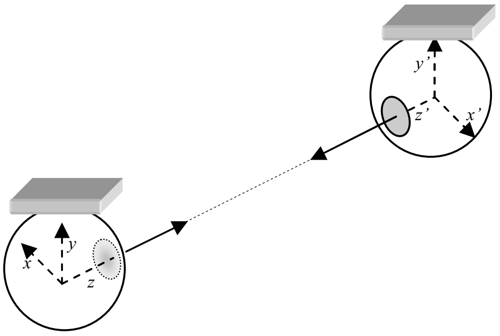

**Figure 3: cameras and system coordinates**

In a pinhole model hypothesis - as is our assumption - if an object lies on a plane parallel to the image plane, the ratio between its real dimension and its pixel projection on the image can provide an estimate of camera-camera separation if the exact optical zoom magnification is known. However, as it is not known, a different approach is adopted. The *watcher* is panned by precise tiny angles and the corresponding positions of the target lens centre is stored. Virtual right triangles[2] are then constructed with known vertex angles $\theta$ and opposite sides, $b$, (in pixels). The ratio between observed lens diameter in pixels and in mm (a priori) gives us the conversion factor needed to estimate the camera-camera distance (see Equation 2 and Figure 4).

**Equation 2: Camera-camera distance**

$$b_{mm} = b_{pxl}\frac{r_{mm}}{r_{pxl}} \qquad b \text{ base length, } r \text{ lens radius (mm and pixel)}$$

$$h_{mm} = \frac{b_{mm}}{\tan\theta} \qquad h \text{ intra - camera distance, } \theta \text{ pan angle (radians)}$$

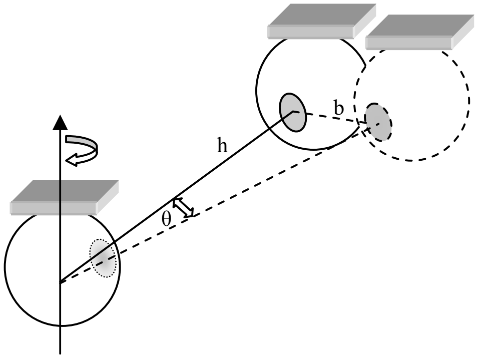

**Figure 4: Panning the *watcher* by θ is equivalent to using a larger target**

[2] The base is not flat as it lies on a spherical surface. However, as the angle is so small, (<1° in a typical indoor setting) such an approximation is valid

The main source of error in this procedure is the lens radius estimation in the image, done by a Hough transform; in our tests this usually incurs a 3% error, but is significantly lowered (more than halved) through the usage of angular displacements in different directions.

The vector translation is calculated using the distance as the norm and the watcher orientation as the vector orientation. Since this is the distance between the *watcher's* optical centre to the target lens, we must also add the distance between target lens and the target optical centre (found in the camera specifications).

### 2.2.2. Camera to Camera Orientation

The best way to describe the rotational extrinsic parameters of an active camera is to use two different entities, one to describe the transform when it is at rest, and another one to describe the transform from the rest position to the current one; this second matrix is defined exclusively by the camera control parameters, while the first - the one which we wish to derive - stores all of the information about the physical configuration. In this way the calibration (obtained by combining these two matrices) is easily updated when the camera changes its orientation.

This rotation matrix can be seen as a composition of different rotations. Indeed it represents the movements that the *watcher* has to do (starting from rest position) to match the orientation of the target camera (when it is also at its rest position). This kind of movement can be obtained by the following steps[3], in which the watcher:

1. pans and tilts to look straight at the target (watcher and target cameras now share the same Z axis direction, as represented in Figure 5)
2. rolls to match the different mounting skew (watcher and target cameras now share the same XYZ axis direction)
3. pans by an angle of PI (watcher and target cameras now share the same XYZ axis)
4. pans and tilts backwards to the target's rest position

Most of this information was acquired during the previous phase: the distance computation, where we commanded each camera to look straight at the other storing the corresponding parameters; rotation 1 and 4 can be directly derived by these values. Rotation 3 is trivial, so at this point the only angle we need to solve is the relative roll ρ.

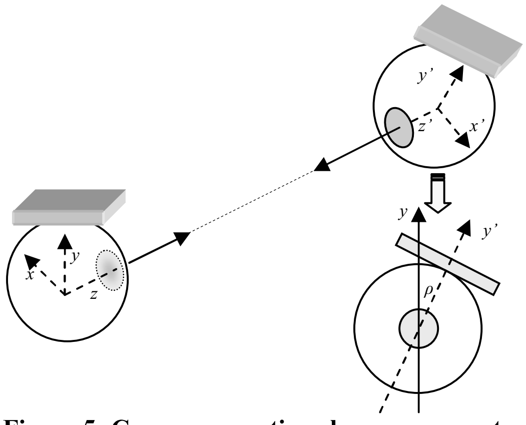

**Figure 5: Camera mounting skew assessment**

The rotation along the (common) Z axis is given by the relative position of *y'* (see Figure 6), and this direction can be obtained by looking at the positions of the target lens at different tilts (the lens centre projection always belongs to the *y'* axis projection). However, lens centre estimation is not trivial since the tilted lenses are distorted by projection and are no longer circular. Consequently, we choose to intersect the lens shape with a circle centred about the origin; the two intersections define a line with an angular coefficient equal to tan (ρ). As before, the procedure is noisy, but likewise can be improved by repeating the measure with different tilt angles.

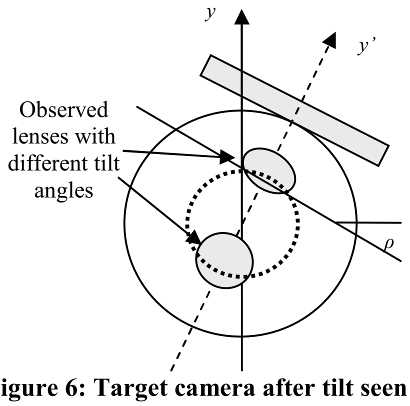

**Figure 6: Target camera after tilt seen by *watcher***

Once ρ is known, the relative rotation matrix between the two cameras is finally solved as stated above.

## 3. Experimental Results

Before the accuracy of the calibration is illustrated, it is important to make a short note about the speed at which the extrinsic calibration can be completed. In practice experiments have shown that to calibrate a 4 cameras system takes around 8 minutes from scratch.

[3] Pan, tilt and zoom are the rotations along the current X, Y, Z axes respectively

Having a rough knowledge of camera positions (e.g. a recalibration given minor displacements) enables a much faster calibration, as it is possible to restrict the awareness stage to given sectors.

To evaluate the performance of the extrinsic calibration procedure, a mannequin of height 160cm and an arm length of 68cm was placed in a variety or positions and orientations around the room. Spherical targets, easily visible, were placed on the mannequin to form a T-shape (see Figure 7). In order to demonstrate solely the calibration performance, the target spheres were manually segmented from the 2D images from each camera and then reconstructed in 3D using the extrinsic calibration gathered by the system.

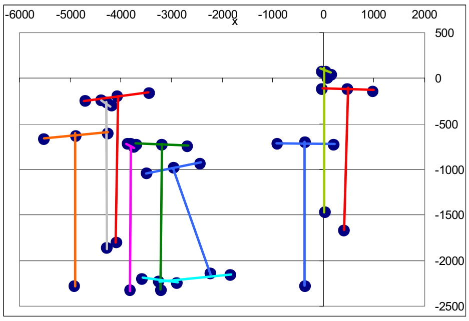

**Figure 7: Mannequin test positions in room**

For each mannequin position each of the cameras in the room was panned and tilted until all of the target spheres were visible. Then, through 3D reconstruction, the inter target distances were measured and compared to the ground truth. To further explore the accuracy and usability of the calibration system, intentionally the number of cameras used for the reconstruction process was reduced to three and then two. The average percentage difference from the ground truth data to the observed data can be seen in Table 1.

| Cameras | 4 | 3 | 2 |
|---|---|---|---|
| % Error | 0.4 | 1.2 | 1.6 |

**Table 1: Percentage error**

As can be seen, across the entire room the average error is 0.4%, this relates to about 0.5cm error over the height of the mannequin. This error increases slightly to 2.5cm when only two cameras are used for reconstruction.

These errors obtained from randomly placed targets provide us with a good estimation of distance distortion typical of our calibrated system. We can expect the same error magnitude in usual localization problems.

## 4. Conclusions

Firstly, it is important to note that a human operator is not required during calibration and also that the room can still be occupied during the calibration procedure. This is crucial in a domestic environment where camera configurations may be changed often. Also, as the algorithm is designed to operate in the indoor environment, an error of around 1% (depending on camera coverage) is in the order of a few centimetres – negligible for many tracking purposes.

The main source of precision error in the system is the lens radius estimation stage, where each pixel error implies 1-2% error. Consequently, we decided to use a simple model; assumptions such as: 1. pinhole camera; 2. optical centre is the physical pivotal point of the cameras; 3. principal point constant whilst zooming; 4. no angular errors in the motor mechanisms are considered negligible.

Admittedly, the Sony SNC-RZ30P cameras used for our experiments provide an ideal physical target for lens segmentation, and it would be difficult for the calibration to function in a darkly painted room, with dark looking cameras; but by utilising the colour channels, it may be possible to distinguish moving cameras from their surroundings. Similarly, by utilising a different physical characteristic of the camera, perhaps its entire physical appearance, a template matching algorithm could instead be utilised for detection.

## 5. Acknowledgements

This work was funded by the European Commission within the CHIL project (http://chil.server.de).

## 6. References

[1] 'Camera Calibration Toolbox for Matlab': http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.

[2] 'Automatic Lens Distortion Estimation for an Active Camera', Oswald Lanz, ICCVG 2004, International Conference on Computer Vision and Graphics, September 22-24, 2004, Warsaw, Poland

[3] 'Quick Guide to Multi-Camera Self-Calibration, Tomá Svoboda', Computer Vision Lab, Swiss Federal Institute of Technology, Zürich, BiWi-TR-263, Version 0.2, August 20, 2003

[4] 'Motion Characterization Using Gradient Histograms', http://www.cc.gatech.edu/~kwatra/computer_vision/project/report.html