Proceedings

Of the

Seventh ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2013)

Palm Spring, CA

October 29 – November 1st 2013

Parameter-Unaware Autocalibration for Occupancy Mapping

David Van Hamme*, Maarten Slembrouck*, Dirk Van Haerenborgh*,
Dimitri Van Cauwelaert*, Peter Veelaert* and Wilfried Philips*†
*Ghent University
TELIN dept., IPI/iMinds
Ghent, Belgium
†Senior member of IEEE

***Abstract*—People localization and occupancy mapping are common and important tasks for multi-camera systems. In this paper, we present a novel approach to overcome the hurdle of manual extrinsic calibration of the multi-camera system. Our approach is completely parameter unaware, meaning that the user does not need to know the focal length, position or viewing angle in advance, nor will these values be calibrated as such. The only requirement to the multi-camera setup is that the views overlap substantially and are mounted at approximately the same height, requirements that are satisfied in most typical multi-camera configurations.**

The proposed method uses the observed height of an object or person moving through the space to estimate the distance to the object or person. Using this distance to backproject the lowest point of each detected object, we obtain a rotated and anisotropically scaled view of the ground plane for each camera. An algorithm is presented to estimate the anisotropic scaling parameters and rotation for each camera, after which ground plane positions can be computed up to an isotropic scale factor. Lens distortion is not taken into account. The method is tested in simulation yielding average accuracies within 5cm, and in a real multi-camera environment with an accuracy within 15cm.

I. Introduction

Occupancy mapping is a common and important task in many multi-camera vision applications. An occupancy map is a two-dimensional representation of a scene in which the position of foreground objects is marked. Applications which require such a map include surveillance, smart rooms, video conferencing and sports analysis. Considerable effort has been expended to develop accurate and robust methods for calculating occupancy maps [1]–[4].

All of these methods require the camera system to be minutely calibrated. The intrinsic parameters are typically obtained by analysing a number of views of a planar checker board pattern [5] or polka dot pattern [6] in various orientations. This method often requires a person to stand on a ladder to provide the camera with a good view of the pattern. To calculate the extrinsic parameters, common calibration procedures require placement of specific calibration patterns or manual entry of tie points [7] or walking around slowly with an easy-to-track light source such as a coloured flash light or capped laser pointer [8].

This research was made possible through iMinds, an independent research institute founded by the Flemish government

In recent years, efforts have been made to fully automate the calibration process. Significant works include Sinha *et al.* [9], who are able to calibrate a dense multi-camera system to a very high accuracy, but depend on very accurate silhouette extraction. This is usually achieved by the use of green screen or similarly uniform background, which limits the real-world usability outside the lab. Another promising approach was presented by Dellaert *et al.* [10], but was only demonstrated using outlier-free hand-picked features.

The reliance of occupancy mapping on accurate but cumbersome calibration procedures is a significant drawback for many non-critical applications (e.g. statistical behaviour monitoring), where it is acceptable to sacrifice accuracy for ease of use. In this paper, we present a novel approach to calibration that uses the projected height of a person in the different camera views to relate their image coordinate system to the world axes. This eliminates the need for a specific calibration procedure: the user can just walk around the room and the system calibrates automatically. No prior knowledge about the cameras is used. The only requirements to the multi-camera setup are that all cameras are at approximately the same height, and there is a substantial area of overlap between the cameras. In typical configurations, both requirements are easily satisfied. Lens distortion is not taken into account as, in our experience, it has proven not to be a significant factor for standard lenses. It does however preclude the use of extremely wide angle (fisheye) lenses.

In section II we will explain how the projected height of an object can be used to calculate the image-to-world coordinate transform required for occupancy mapping. In section III we will present a practical algorithm to compute this transform. Results of both simulations and real-world testing are presented in section IV. Finally, conclusions about the proposed method are drawn in section V.

II. Projected height as estimator for camera pose

In this section we will analyse how the projected height of an object (i.e. the difference in camera image Y coordinates between the top and bottom point) can be used to reproject image points to world coordinates. We use a pinhole camera model [11] for this analysis, as it offers the clearest perspective on the technique we will use.

Assume we have a camera with zero roll angle (i.e. the horizontal image sensor axis is parallel to the ground plane),

and zero pitch angle (i.e. the vertical image sensor axis perpendicular to the ground plane). It can be easily seen from Figure 1 that the projection height h' of a vertical object of actual height h is inversely proportional to its distance d to the focal point F of the camera:

$$h' \propto \frac{h}{d}. \tag{1}$$

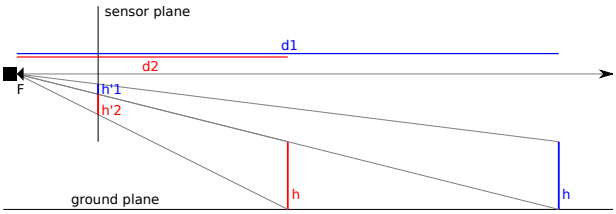

Figure 1: Relation of height and distance in the zero pitch, zero roll case (side view).

In other words, we can calculate the relative distances between different observations of the object from the ratio of their projection height. The emphasis is on the word *relative*: without knowing the distance d for at least one observation, we cannot obtain absolute distance values. The distances to the camera are only determined up to a scale factor. If the actual height h of the object is known, this is also sufficient to determine the scale factor.

In order to backproject the image points to the world ground plane, we need to know not just the distances associated with the image points, but also the focal length f of the camera. Once depth and focal length are known, this defines the intersection of the line through the image point and the focal point with the world ground plane (Figure 2). Note that the world coordinates of these intersection points are relative to the camera location and orientation: the obtained world ground plane maps for different cameras will be rotated and translated versions of each other.

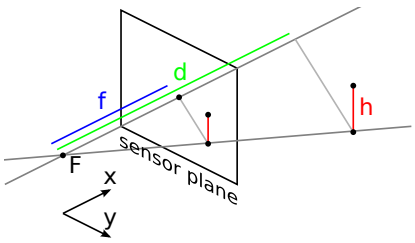

Figure 2: Backprojection of image point onto world plane, using known distance d and focal length f_c.

If we know the height h of the object or the distance d for one of the observations, but not the focal length f, we can determine the absolute distances, i.e. the ground plane Y coordinates, but not the X coordinates: they will only be known up to a scale factor. If neither h, d or f are known, the Y axis will also only be defined up to a scale factor. In this case the X and Y axis will be differently scaled versions of the actual (rotated) world axes. A simulated example is shown in Figure 3. In this simulation we assumed a focal length equal to half the sensor width to compute the ground plane coordinates. The actual focal length of the simulated camera was 30% longer. This results in the the backprojected coordinates being slightly stretched in the X direction.

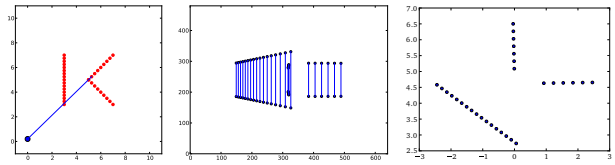

Figure 3: Reconstruction of relative object positions using projection height in the zero pitch case (simulated). The left image shows the camera position (blue dot) and actual world object positions (red dots). The center image shows the view from the camera, and the right image the reconstructed positions, an anisotropically scaled version of the (rotated) actual positions.

In the case of a camera with non-zero pitch, the situation complicates somewhat. As shown in Figure 4, the projection height h' is no longer proportional to the actual height h, but to its projection h_p on the plane γ through the bottom point perpendicular to the principal axis, with the focal point F as center of projection:

$$h' \propto \frac{h_p}{d} \tag{2}$$

In which h_p can be seen as the perpendicular projection h_a of h on γ extended by a length h_b as a consequence of the projection through the focal point F. Note that the ratio of h_a to h is constant: it only depends on the pitch angle of the camera and not on the location of the object. The ratio of h_b to h however depends on the image coordinates of the projection of the top point of the object. For objects that lie in the bottom half of the image, h_b is negative.

If we ignore the non-proportionality of h_b for now, we can still determine ground plane positions up to two scale factors for X and Y, similar to the zero pitch scenario. The distances between observations are still proportional to their projection heights.

Because of the dependence of h_b on image location, the estimated depths will be inaccurate. The error increases with increasing downward camera pitch and with increasing lens aperture. For a wide angle lens with 90°vertical aperture on a

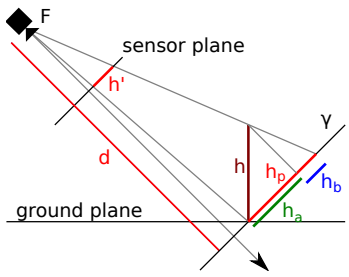

Figure 4: Projected height vs. distance in the nonzero pitch, zero roll case (side view).

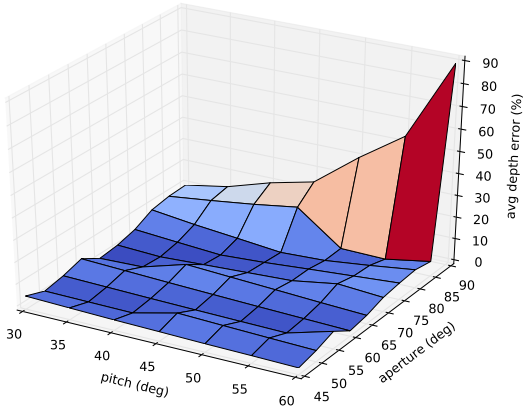

Figure 5: Mean error on the depth estimate versus camera pitch and vertical aperture angle.

Table I: Analysis of mean depth errors. Aperture range is split due to rapid deterioration of accuracy for apertures over 80°.

aperture (deg)	45-80	85-90
mean error (%)	2.81	21.77
error std (%)	1.92	23.21
max error (%)	7.37	89.18

camera pointing 45°downward, the average error on the depth estimation over all image points is 11.7% and the maximum error is 29.6% for objects at the bottom of the image.

If the focal length f in pixels and pitch angle α (negative for a camera pointing downwards from horizontal) are known, the length of h_b can be determined. Let c_y denote the image Y coordinate of the principal point and t_y the image Y coordinate of the top of the object. The length h_b is then given by

$$h_b = h\, sin(\alpha) \frac{(c_y - t_y)}{f} \tag{3}$$

and the exact relative depths can be calculated. If we do not know f or α, the best we can do is calculate h_b based on what we assume are average values $\overline{f}$ and $\overline{\alpha}$. For an occupancy application, we may assume that common pitch angles are between 30°and 60°, and common vertical aperture angles between 45°and 90°, since the cameras must adequately cover the room or yard. Using these ranges to calculate an approximate length $\overline{h_b}$, the mean depth estimation errors are shown in Figure 5 and Table I. It can be seen that the errors are only significant for apertures above 80°. This means that our workaround of using typical values is not valid for fisheye lenses. We may note that the pinhole camera model is a very poor approximation for this type of lenses anyway, which would severely compromise the results by itself.

An important final note concerns camera roll. We have assumed a roll angle of zero in the above discourse, which will not usually be the case in real applications. Fortunately, the roll can be easily corrected when observing vertical objects. It is sufficient to have a single observation which lies on a line that passes close to the center of the camera image. The angle between this line (the extension of the observation) through the center and the vertical axis is the roll angle, and the image can simply be rotated to achieve the zero roll scenario.

To summarize, so far we have established that for each camera in a typical multi-camera setup we can use the projection heights of multiple observations of the same object in different positions to obtain an anisotropically scaled and rotated version of the real world plane axes. Or, put differenty, we can determine the mapping from image coordinates to world plane coordinates up to a rotation and two scaling factors using no prior knowledge about the camera whatsoever, just by comparing the projection heights of two or more observations of the same object in a different location.

In the next section, an algorithm is proposed to determine the rotation and scaling factors for each camera by comparing the observations of the invidual cameras.

III. Algorithm

In the previous section, we have built a world coordinate map for each camera that is an anisotropically scaled and rotated version of the real world plane. In this section, a method is outlined to estimate the rotation angles and the X and Y scale factors for each camera. The core idea is that if we find scale factors for each camera so that their coordinate maps are rotated versions of each other, then these coordinate maps will automatically be uniformly scaled versions of the actual world coordinates. In order to find the correct scale factors for each camera, we must be able to compare the observations of the same object by different cameras. This is easily achieved when the cameras are reasonably synchronised and only one object is observed at a time during the calibration phase.

Because of the non-linear relation between the world plane maps between the cameras (due to the rotation) we will resort to an exhaustive search in stages. One camera is taken as a reference: its X scale factor is assumed 1 and its rotation 0°. The Y scale factor for the reference camera, and the X and Y scale factors and rotations of all other cameras are then estimated simultaneously. The core algorithm to achieve this is outlined below, in which $P_{n,j}$ are the preliminary world coordinates of observation j for camera n.

This is essentially an exhaustive search in which for each Y scale factor $scale_{y,1}$ of the reference camera, all combinations $scale_{y,n}$ and $scale_{x,n}$ are tried for the other cameras, their rotations computed, and the sum of least squares distances between their scaled and rotated points with the reference camera's points minimized. To make this tractable, the search intervals for the scale factors are first iterated in large steps to identify the rough location of the minimal cost solution, after which a smaller interval is iterated in finer steps to further pinpoint its location.

Note that the choice of reference camera does not matter. The algorithm will converge on the correct Y/X scale ratio for each camera regardless of which camera was chosen as reference.

Only scale factors between 0.5 and 2 are evaluated, on the assumption that the estimated focal length f to calculate the initial world coordinates cannot be more than twice too short

Algorithm 1 Algorithm to estimate the scale factors for each camera

```
mincost = 1e10
for scale_{y,1} ∈ [0.5, 2] do
  apply scale_{y,1} to all P_{1,j}
  for n = 2 to N do
    cost = 0
    for scale_{y,n} ∈ [0.5, 2] do
      apply scale_{y,n} to all P_{n,j}
      for scale_{x,n} ∈ [0.5, 2] do
        apply scale_{x,n} to all P_{n,j}
        calculate rotation between cam 1 and cam n from
        a pair of points in both views
        apply this rotation to all P_{n,j}
        cost+ = Σ_j |P_{n,j} − P_{1,j}|^2
      end for
    end for
    if cost < mincost then
      mincost = cost
      save scale_{y,1} and the best scale_{y,n} and scale_{x,n} for
      each cam
    end if
  end for
end for
```

or too long. This holds true for the vertical aperture ranges considered in section II.

Now that for each camera the scale factors and rotation are known, they define the mapping between image and world coordinates up to a global isotropic scale factor. There is no way to determine the exact X scale factor corresponding to the reference camera without knowing at least one distance or the height of the object. In an occuppancy mapping context, the average height of a person can be used to compute an approximation of the scale factor that is sufficiently close for practical applications. In a surveillance application for example it is not as much the absolute position of a person that matters, but its proximity to the limits of the area. The limits of the area can be determined automatically by observing a person who walks along these limits.

The performance of our estimation method will be examined in the next section.

IV. Results

We have established that it is possible to find the mapping between each camera's image coordinates and the world coordinates, up to a scale factor and rotated relative to the first camera. In this section we will investigate the accuracy of this method in simulation, and prove its viability with a real-world experiment.

A. Simulation

To estimate the impact of our assumptions in section II and the relation between the number of observations and the accuracy of the obtained mapping, we have conducted the following simulation. Four cameras are randomly chosen in a 2 by 2 metre region in the corners of a 10 by 10 metre room. The camera height is random between 2 and 5 metres. The focal length of each camera is randomly chosen to correspond with a vertical aperture angle between 45°and 80°. The cameras are pointed at a random point on the ground in a 2 by 2 metre region in the centre of the room, yielding pitch angles between 18°and 56°. A number of observations are simulated as 1.8 metre high vertical line segments in the middle 5 by 5 metres of the room (where the camera views will generally all overlap).

For each camera, the mapping from image to world coordinates is computed as described in sections II and III. The optimal scaling and rotation of these world coordinates (which are relative to the first camera) to the original simulated world are determined and applied. The mean euclidian distance between the world coordinates of each observation in each camera and the actual position of the object is then computed. This is repeated 100 times for each number of observations (5, 10, 20 and 40 observations). The results are summarized in table II. Visualizations of a good and a poor result are shown in Figures 6 and 7.

Table II: Results of simulated experiment.

# observations	5	10	20	40
mean err (m)	0.056	0.053	0.051	0.045
std err (m)	0.084	0.072	0.052	0.040

In the simulated experiments, the method proves accurate to approximately 5 centimetres regardless of the the number of observations. The standard deviation on the error decreases for higher numbers of observations. The errors are caused by two separate effects. The first effect is the impact of the unknown actual length of h_b as decribed earlier. The more the focal length and pitch differ from their assumed average values, the lower the maximum theoretical accuracy. The second effect is that the location of the objects is randomly chosen, which can give rise to configurations in which the multi-stage exhaustive search does not converge on the global minimum. This explains the decrease in error variance for higher number of observations: a larger number of random locations are more likely to be adequately spread over the room.

B. Real video

In order to prove the viability in real circumstances, an experiment was conducted in our multi-camera room. This room measures 5 by 8 metres and is equipped with a number of cameras. For this experiment, four cameras were used near the corners of the room at a height of approximately 3.5 metres. The cameras are equipped with two different types of lenses and their viewing areas completely overlap in an approximately 3.5 by 4 metre region. This multi-camera room is often used for occupancy-related experiments.

As ground truth, 10 spots are measured and marked on the ground. A 90 second four-camera video is recorded in which a person walks around the area of overlap, momentarily standing still at the marked spots. The foreground-background estimation method of Kim *et al.* [12] is used to extract the silhouette of the person in each camera view. When the person holds still, the top and bottom point of the silhouette are saved as observations. Figures 10 and 11 show the camera views,

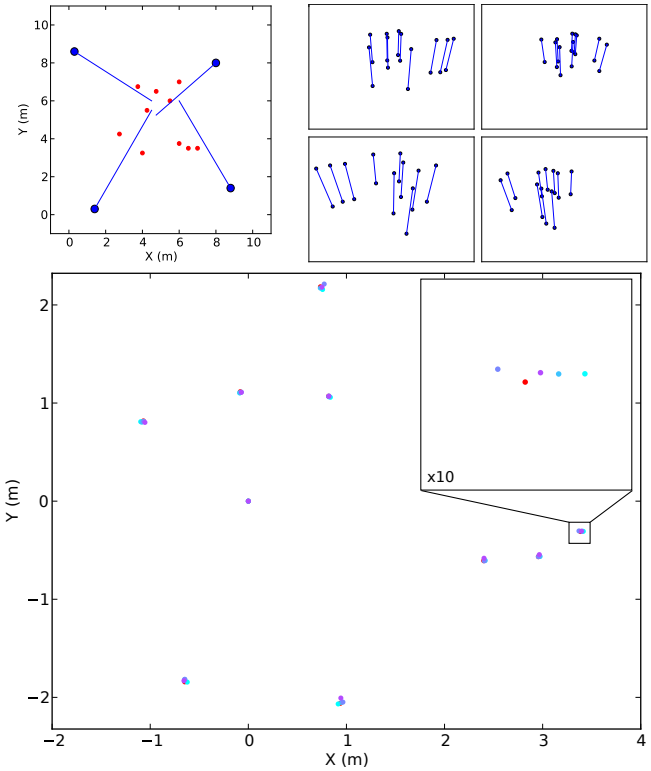

Figure 6: Example of a good reconstruction result. Simulated world is shown in top left. Camera views are shown in top right. Reconstructed world coordinates in bottom image. Red dots represent ground truth, and are completely obscured by the reconstructed positions of the individual cameras.

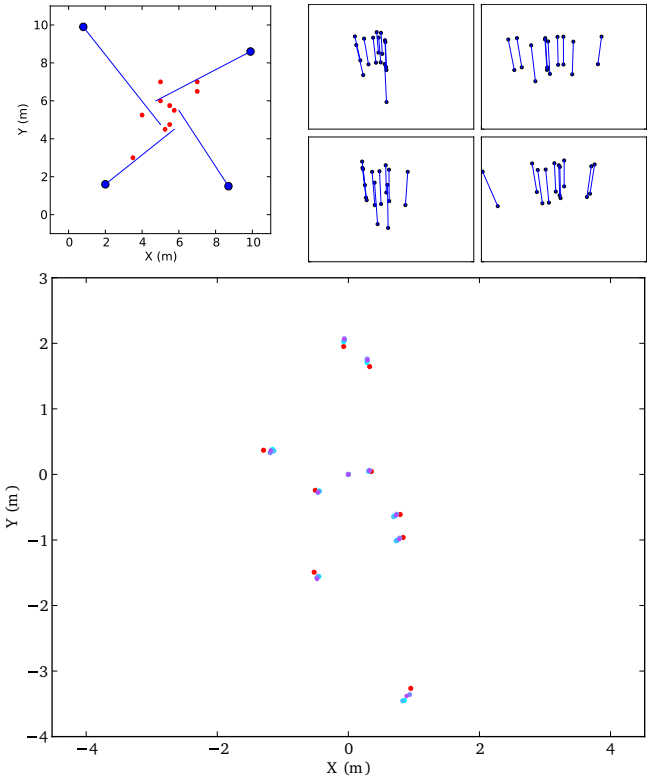

Figure 7: Example of a poor reconstruction result. Red dots represent ground truth, and show that even though the cameras agree relatively well, they do not accurately map true world coordinates. Part of the cause in this case is that the objects only span a narrow, elongated region in the overlapping space.

foreground masks and top and bottom points for a sample frame. These observations are used as input to our method to compute the image-to-world mapping for each camera. The observations corresponding to the 10 ground truth positions are then mapped by each camera, and the average of the four positions for each point is taken as the estimated location. The results of this experiment are shown in Figure 8 and table III.

For comparison, the location of the person is also estimated using a visual hull based method [13]. In this method, the four cameras are first intrinsically calibrated as per Zhang *et al.* [5] and extrinsically calibrated using a method based on the POSIT algorithm [14], [15]. The silhouette of the person in each camera is then projected as a generalized cone onto a voxel cuboid with a voxel size of 2 by 2 centimetres. The voxels that fall inside all four generalized cones are retained as the volumetric approximation of the person. The center of gravity of this voxel shape is projected onto the ground plane and taken as the estimated location. The results of this method are also shown in Figure 8 and table III.

It can be seen that the proposed method provides reasonable accuracy without requiring any specific calibration procedure other than the test person occasionally standing still. The estimation of the coordinate transforms took 21 seconds on a standard desktop workstation. Due to the noise in the measurement of the top and bottom point of the person's silhouette and the presence of lens distortion, the accuracy is lower than in simulation. The error mainly manifests itself as a skew factor on the world coordinates; manually correcting for this skew would bring the accuracy in the same range as the visual hull based method. However, as the focus of this work was to provide completely unsupervised calibration, we chose not to do this manual correction. The visual hull based method therefore boasts superior accuracy, but took over 15 minutes to calibrate and involved waving checker boards around while standing on ladders.

Once the camera-to-world transforms have been computed using our method, occupancy can be calculated by applying the transform of each camera to the image coordinates of the bottom edge of each detected object in that camera. This principle is illustrated for two cylindrical objects in a three-camera setup in Figure 9.

Table III: Summary of errors of proposed method and visual hull based method compared to ground truth.

method	proposed	visual hull
mean err (m)	0.1449	0.0875
std err (m)	0.0867	0.0516

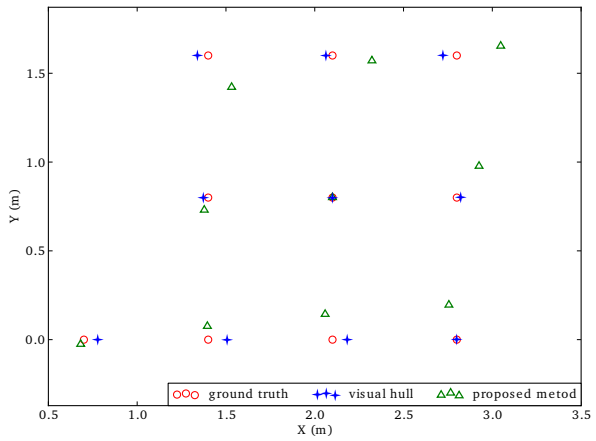

Figure 8: Comparison of the proposed method to visual hull based method and ground truth.

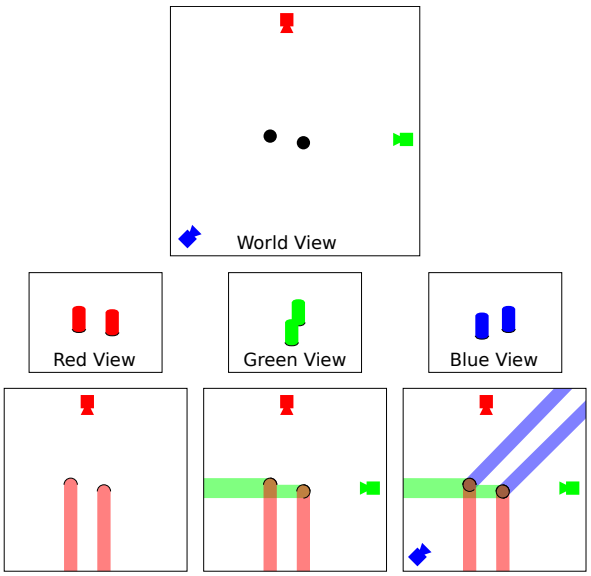

Figure 9: Principle of occupancy mapping with multiple objects and occlusion. For each camera, the bottom edge of each detected object is reprojected using the precomputed image-to-world transform for that camera. Overlaying the data from all cameras yields an occupancy evidence map from which the object positions can be recovered.

V. Conclusion

We have presented a novel approach to camera calibration for occupancy applications that does not require the knowledge of any parameters about the cameras or the scene. The method is based on comparing the projection height of observations of the same object or person in different locations, and computes a mapping transform from image coordinates to world coordinates for each camera, rather than the traditional pose and projection matrices. The concept is proven to be sufficiently accurate in simulations as well as in a real-world experiment. The results could be further improved by making the process semi-supervised instead of fully unsupervised.

Figure 10: Example of the camera views in the real experiment.

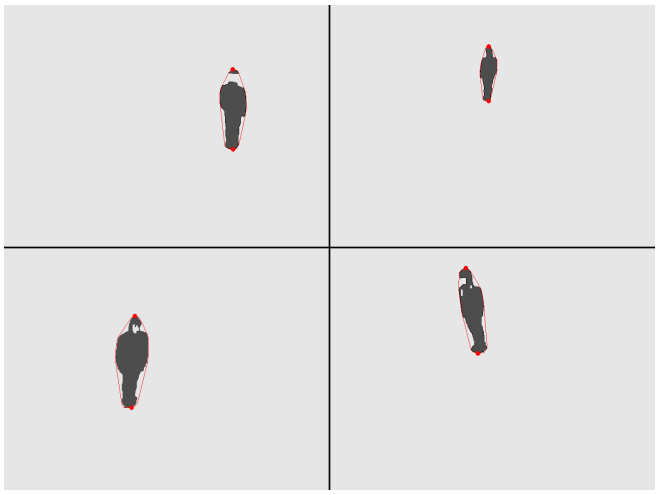

Figure 11: Example of foreground masks in the real experiment. Top and bottom point of each silhouette are indicated by red dot.

References

[1] A. Hoover and B. Olsen, "A real-time occupancy map from multiple video streams," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 3, 1999, pp. 2261–2266.

[2] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 2, pp. 267–282, 2008.

[3] X. Xie, S. Grünwedel, V. Jelača, J. Niño-Castañeda, D. Van Haerenborgh, D. Van Cauwelaert, P. Van Hese, P. Veelaert, W. Philips, and H. Aghajan, "Learning about objects in the meeting rooms from people trajectories," in *6th International Conference on Distributed Smart Cameras (ICDSC)*, Hong Kong, China, November 2012, pp. 1–6.

[4] S. Kim and J. Kim, "Building occupancy maps with a mixture of gaussian processes," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 4756–4761.

[5] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1999, pp. 666–673.

[6] S. Kawabata and Y. Kawai, "Correspondence-free multi camera cal-