



<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper presented at *IEEE International Workshop on Robotic Sensing, ROSE 2003, Örebro, Sweden, June 5-6, 2003*.

Citation for the original published paper:

Lilienthal, A J., Duckett, T. (2003)

An absolute positioning system for 100 euros

In: *ROSE 2003 - 1st IEEE International Workshop on Robotic Sensing 2003: Sensing and Perception in 21st Century Robotics: Sensing and Perception in 21st Century Robotics*, 1218705 IEEE

<https://doi.org/10.1109/ROSE.2003.1218705>

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:oru:diva-4021>

An Absolute Positioning System for 100 Euros*

Achim Lilienthal¹, Tom Duckett²

¹ University of Tübingen, WSI, D-72076 Tübingen, Germany

`lilien@informatik.uni-tuebingen.de`

² Department of Technology, AASS, Örebro University, S-70182 Örebro, Sweden

`tdt@tech.oru.se`

Abstract

This paper describes an absolute positioning system, which provides accurate and reliable measurements using low-cost equipment that is easy to set up. The system uses a number of fixed web-cameras to track a distinctly coloured object. In order to calculate the (x,y) position of this object, estimates calculated by triangulation from each combination of two cameras are combined, resulting in centimeter-level accuracy. Example applications, including tracking of mobile robots and persons, are described. An extended set-up is also introduced, which allows determination of the heading ϑ of a two coloured object from single images.

1 Introduction

For experiments in robotics it is frequently necessary (and often helpful) to possess a method that allows investigators to determine the absolute position of a robot or other moving objects.

Available positioning systems most often apply trilateration to determine 2D or 3D coordinates of the tracked object. Frequently time-of-flight measurements (of ultrasonic waves or laser light) are utilised to provide the distance measurements. While ultrasonic systems offer a low cost but not very accurate solution (see for example [8]), optical systems that use laser light are comparatively expensive. An overview of ultrasonic and optical positioning systems is given in [2]. An alternative solution that is not subject to the problem of obstructed lines of sight is provided by electromagnetic systems. Here, the magnetic field generated by fixed coils (beacons) is sampled by a mobile sensor unit to determine the distance to the coils [7]. Such systems require, however, strong magnetic fields and are susceptible to interference from metallic objects or magnetic fields.

This paper presents a vision-based positioning system that is based on triangulation. It provides reliable and ac-

curate measurements by tracking a distinctly coloured object. In order to reduce costs, the system uses a number of web-cameras to acquire images from different angles. The object chosen to be tracked was a coloured “hat” made of cardboard, which can be worn by a person or placed on top of a robot (see Fig. 1).

Only information about the instantaneous (x,y) position can be obtained using a unicoloured hat. Nevertheless, the heading can be estimated from the path tracked by derivation. However, the reliability of this technique depends on the translational speed of the object. In fact, it is not applicable if rotations without translational movement occur, because no information about the current heading can be obtained during such periods. As a possible solution to this problem, an extended set-up is introduced in Section 6, which allows determination of the instantaneous heading from single images using a hat with a two coloured pattern.

The rest of this paper is structured as follows: the experimental set-up is introduced in Section 2, followed by a brief description of the software framework that was used for the implementation in Section 3. Then, a detailed description of the applied method to determine the 2D coordinates of the tracked object is given (Section 4) and example applications are presented (Section 5). Next, the method to determine the heading is outlined in Section 6, followed by conclusions and suggestions for future work (Section 7).

2 Set-Up

The web-camera-based absolute positioning system (“W-CAPS”) was developed such that it can be utilised with an arbitrary number of cameras ($N \geq 2$). To achieve the results presented in this paper, four Philips PCVC 740K web-cameras were used with a resolution of 320×240 pixels. These cameras were mounted at a height of approximately 2 m in the corners of the 10.6×4.5 m laboratory room shown in Fig. 4. The orientation and position was adjusted to cover a large area of interest with as many cameras as possible. All the calculations were performed on a Pen-

*this is the assumed cost of two web-cameras and connectors, i.e., the minimum hardware required, excluding PC.



Figure 1: The coloured “hat” that is tracked by the absolute positioning system, worn by a person (left) and placed on a robot (right).

tium III PC, which was connected to the web-cameras by a $4 \times$ USB port.

The whole system can be arranged quickly because it uses standard components that should be easily available: first, the webcams and the (USB) connectors as well as a standard PC with a sufficient number of free (USB) ports is needed. Next, a distinctly coloured object is necessary to run the system, which can be assembled using coloured cardboard. Further, a stable support is often needed to mount the cameras. To attach the cameras, the W-CAPS installations arranged so far use either already available surfaces in the room, or supporting wooden plates, which were screwed to the wall.

3 DDFLat Framework

W-CAPS was implemented using a software framework that was originally designed to facilitate the development of robot control applications [4]. The main idea is to map the functional units of an application to objects and to model the cooperation between these objects by dynamically configurable data flow chains. A data flow chain is represented by a cascade of connected objects, which are updated downstream if a certain object changes its internal state. The possibility of adjusting the timing of the processes involved is provided by a latency period that can be assigned to each object, meaning that the object cannot trigger an update cascade before this period has elapsed. Thus, a maximum update frequency can be specified for each part of the data flow chains. Due to this feature the framework is called DDFlat (Dynamic Data Flow with Latency). A latency-based mechanism was chosen to avoid a restriction to real-time operating systems (RTOS).

DDFLat was chosen because it enables fast development while keeping the functional parts of the program clearly separated. Furthermore it supports reusability of components and provides a demonstrative way to visualise how the application works. Fig. 2 shows the mode of operation of W-CAPS in a so called DDFLat diagram. The data flow

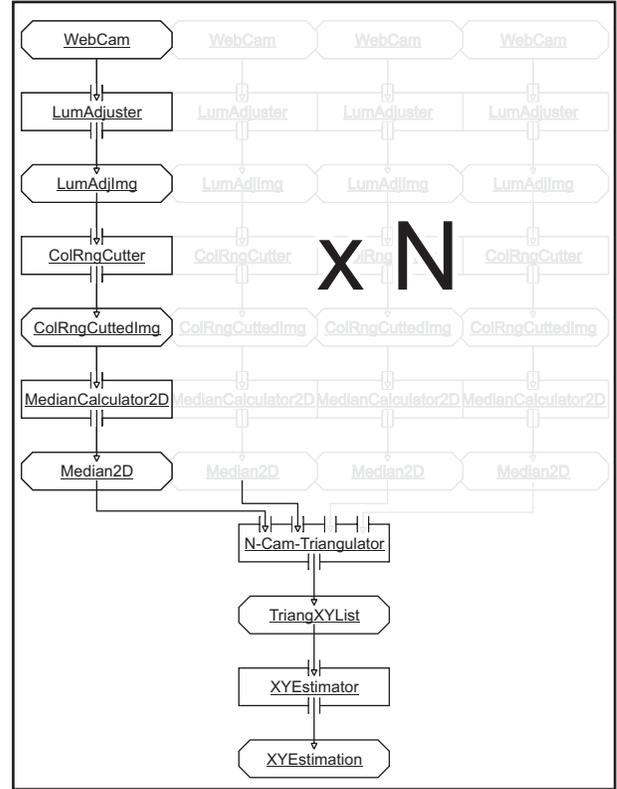


Figure 2: DDFLat diagram that shows the mode of operation of W-CAPS.

is composed of an alternating sequence of data objects (displayed by boxes with clipped corners) and algorithm objects (represented by unclipped boxes with in- and outlets that visualise possible connections to data objects needed to use this algorithm). Algorithm objects perform the intended task operating on the connected data objects.

4 Determining the 2D Coordinates

W-CAPS is based on triangulation. First, the angle φ_i at which the centre of the coloured object appears is determined for each camera. For every combination of two cameras i, j that both actually sense the whole coloured object, an estimate of the 2D position \vec{x}_{ij} is then calculated by triangulation. Using N cameras up to $N(N-1)/2$ valid position estimates result from each snapshot taken, which are combined to determine the final estimate \vec{x} . The individual steps that are executed in order to calculate a position estimate \vec{x} , can be traced in the DDFLat diagram in Fig. 2, and a sequence of images visualising the intermediate results is shown in Fig. 3.

To compensate for different lighting conditions, the orig-

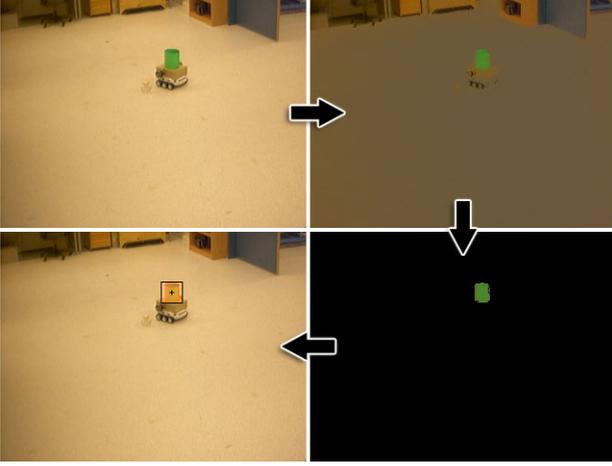


Figure 3: Sequence of images that demonstrates how the median of the tracked colour blob is calculated. Upper left: raw image; upper right: normalised colours; lower right: area that is assigned to the tracked colour; lower left: raw image with an indication of the determined median.

inal colour values (r, g, b) are first normalised by the algorithm `LumAdjuster` as

$$(r', g', b') = \begin{cases} \frac{255}{r+g+b}(r, g, b) & \text{if } r+g+b \geq B_{norm} \\ (r, g, b) & \text{if } r+g+b < B_{norm} \end{cases} . \quad (1)$$

Thus, the relative strength of the dominant colour channel is amplified. The threshold B_{norm} is used to prevent amplification of noise in dark regions (an example of a normalised image is shown in the upper right image of Fig. 3).

Next, pixels within a given contiguous rgb-colour range are selected by the algorithm `ColRngCutter` as

$$(r', g', b') \rightarrow \begin{cases} 1 & \text{if } f(r', g', b') \in \Gamma \\ 0 & \text{otherwise} \end{cases} , \quad (2)$$

where the color range Γ is given by:

$$\Gamma = [(r_{min}, g_{min}, b_{min}), (r_{max}, g_{max}, b_{max})]. \quad (3)$$

An example of an extracted area is shown in the lower right image of Fig. 3.

Then, the median values (X_i, Y_i) of the corresponding pixel-coordinates are calculated for each camera i by the algorithm `MedianCalculator2D`. In order to ensure that the centre of the coloured object is found correctly, the size of the colour blob and whether it lies completely inside the picture is checked. This is done by verifying the following heuristic conditions: first, the number n_{arr} of rows that contain at least N_{row} successive pixels with the tracked colour is evaluated. The median values are calculated if n_{arr} exceeds a certain threshold N_{arr} . This condition is checked

for rows only, because just the x-coordinate of the median X_i is used to calculate the angle φ_i at which the centre of the coloured object appears. Second, the border of the image is checked to avoid using a colour blob that is partially outside the image. Because the centre of the visible part of the coloured area would not be valid in such cases, no median values are calculated if there are less than N_{void} empty columns between the median value X_i and the border next to it. A column is considered empty if it contains less than N_{void} pixels of the tracked colour. Again, this condition is applied to the x-axis only. The position of the calculated median is indicated in the lower right of Fig. 3 and on the left side of Fig. 1 respectively. In this paper, the parameters $N_{row} = 3$, $N_{arr} = 15$, $N_{col} = 3$ and $N_{void} = 3$ were used.

In the next step, the angle of the centre of the colour blob φ_i is calculated for each camera i from the median value X_i using:

$$\varphi_i = \alpha_i - n_{X,i} \cdot \frac{\Delta\alpha_i}{n_{X,res}}, \quad (4)$$

where $n_{X,res}$ is the horizontal resolution of the web-cameras and $\Delta\alpha_i$ the angle covered by the corresponding camera.

Finally, a list of positions is triangulated for all combinations of two cameras that detected the hat. To avoid ambiguous results, only combinations φ_i, φ_j are considered for which the direction differs sufficiently. Thus, estimates \vec{x}_{ij} are calculated as

$$\vec{x}_{ij} = \frac{(C_i B_j - C_j B_i, A_i C_j - A_j C_i)}{A_i B_j - A_j B_i} \quad (5)$$

$$A_i = \sin(\varphi_i), B_i = -\cos(\varphi_i), C_i = A_i X_i + B_i Y_i, \quad (6)$$

if the difference of the direction

$$\delta_{dir}(\varphi_i, \varphi_j) = \min(|\varphi_i - \varphi_j|, |\varphi_i - \varphi_j \pm \pi|) \quad (7)$$

exceeds φ_{min} . These calculations are performed by the algorithm `N-Cam-Triangulator`, which generates a list of the estimates \vec{x}_{ij}^t for the current time step t . This list might have from zero to $N(N-1)/2$ entries.

Finally, the overall estimate \vec{x}_i is calculated by the algorithm `XYEstimator` by averaging those estimates \vec{x}_{ij}^t whose validity can be verified as follows: the position of the colour blob is propagated using the last valid estimate \vec{x}_{last} and the speed \vec{v}_{last} , which is determined from the most recent valid positions. An estimate \vec{x}_{ij}^t is believed to be valid if it lies inside a circle with radius $r(t)$ around the propagated position. The radius of the circle is increased linearly by $v_{max} \cdot (t - t_{last})$ to enable recovery in the case of lost positions, using an assumed maximum speed v_{max} and the time since the last valid estimate was detected $(t - t_{last})$.

$$(x, y)_t = \overline{(x, y)_{ij}^t}$$

$$\text{with } i, j \in \{1, \dots, N\}, i \neq j \quad (8)$$

$$\text{and } \vec{x}_{ij}^t \in \text{Circle}(\vec{x}_{last} + \vec{v}_{last} \cdot (t - t_{last}))$$

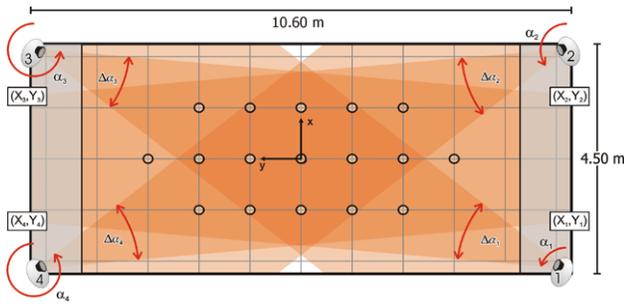


Figure 4: Floor plan of the laboratory room at Örebro University where the experiments were performed. Also shown are the positions that were used for calibration.

4.1 Calibration

The parameters of the cameras (heading α_i , coordinates X_i , Y_i and angular range $\Delta\alpha_i$) are determined by an initial calibration process. This step is crucial because the estimation performance of the whole system depends heavily on the accuracy of the camera parameters.

First, the values of the pixel coordinates $n_{X,i,k}^{(l)}$ of the centre of the hat are determined from K images taken with each camera $i \in 1, \dots, N$ for L known positions $\vec{p}^{(l)}$ of the coloured object. With this set of input data, position estimates $\vec{x}_{i,k}^{(l)}$ are calculated according to equations 4–8 using a particular set of parameters $\{\alpha_i, X_i, Y_i, \Delta\alpha_i\}$. Finally, the average distance \bar{d} between the estimated and the known positions, calculated as

$$\bar{d} = \sum_{k=1}^{k=K} \sum_{l=1}^{l=L} |\vec{x}_{i,k}^{(l)} - \vec{p}^{(l)}|, \quad (9)$$

is minimised. Any optimisation technique might be used for this purpose. Here, a hillclimbing algorithm was applied starting from a reasonable set of parameters determined by hand. It was found to be advantageous to start optimising just the heading of the cameras α_i first, considering the other parameters as fixed. Then, if no improvement is possible any more, all of the parameters are optimised.

Despite the comparatively poor horizontal resolution of 320 pixels, a good accuracy of $\bar{d} \approx 1\text{cm}$ could be achieved using the 17 positions for calibration shown in Fig. 4.

5 Example Applications

This section presents two applications where W-CAPS was successfully utilised. These examples can give, however, only an impression of the performance of our positioning system, because we did not have any other system for measuring the “ground truth” position of the tracked objects.

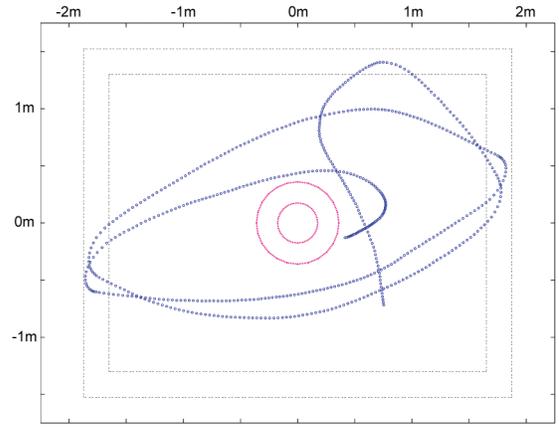


Figure 5: Positions of the Koala robot shown on the right side of Fig. 1 determined by W-CAPS using also the robots odometry.

5.1 Robot Tracking

The absolute positioning system introduced in this paper was used in a number of experiments with a Koala robot (see the right side of Fig. 1), including gas source localisation [5] and gas distribution mapping [6]. Tracking a mobile robot typically enables improvement of the accuracy of W-CAPS for two reasons: first, the assumed maximum speed v_{max} used in eq. 8 is well known and thus outliers can be detected more reliably. Second, odometry information provided by the robot can be fused. In the experiments mentioned, this was done by calculating a new position estimate (using the last estimate and relative odometry information) and adding the low pass filtered deviation of this estimate from the absolute position measured by the camera system. An example of an experiment where odometry information was used to determine the position of a comparatively slow robot (v_{max} was set to 15 cm/s) is shown in Fig. 5. Here, the robot was reactively steered to localise an odour source placed in the middle of the room (indicated in Fig. 5 by two concentric circles). At the same time the area searched by the robot was restricted by repellent walls, which were realised by assigning an artificial potential field [3] that effects a repellent pseudo-force, which increases linearly with the penetration depth. Both the borderline at which the pseudo-force starts to be effective and the one at which it reaches its maximum are indicated in Fig. 5 by broken rectangles. A detailed description of these experiments is given in [5].

Additionally, the information about the heading of the robot provided by its odometry can be improved using the positioning system. While moving with a non-zero translational speed, information about the current heading ϑ can be obtained by calculating the *measured tangent* to the robot’s

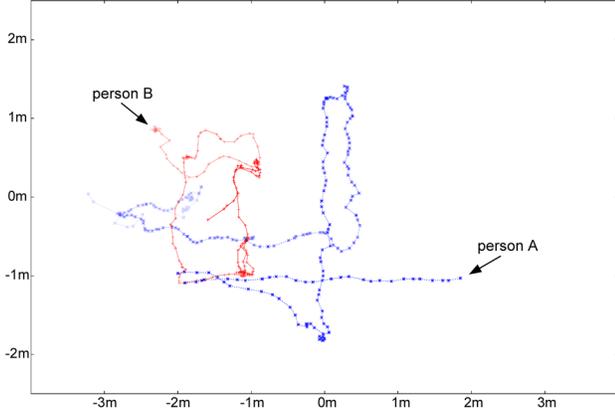


Figure 6: Path of two children running around in a crowded room, tracked during a public presentation. In order to give an impression of the time flow, earlier measurements are drawn lighter than later ones.

path as

$$\Delta \vec{p}_t^{mt} = \vec{p}_t - \vec{p}_{t-1}. \quad (10)$$

Because of its derivative character the headings calculated with eq. 10 provide a rather noisy series of estimates, especially when the robot is moving at low translational speeds. Nevertheless the calculated values can be used to compensate for long term odometry drift. This was done by fusing the heading estimates derived from the measured tangent $\hat{\vartheta}_t^{mt}$ with those from odometry $\hat{\vartheta}_t^{odo}$ using a heuristic non-linear filter described by the following equations:

$$\hat{\vartheta}_t = \frac{\gamma_t \hat{\vartheta}_t^{odo} + \hat{\vartheta}_t^{mt}}{\gamma_t + 1}, \quad (11)$$

$$\hat{\vartheta}_t^{odo} = \hat{\vartheta}_{t-1} + \vartheta_t^{odo} - \vartheta_{t-1}^{odo}, \quad (12)$$

$$\gamma_t = \begin{cases} \gamma_{min} & \text{if } \delta \hat{\vartheta}_t < \Delta \vartheta_{min} \\ \gamma_{max} & \text{if } \delta \hat{\vartheta}_t > \Delta \vartheta_{max} \\ \gamma_{min} + \frac{(\gamma_{max} - \gamma_{min}) \delta \hat{\vartheta}_t - \Delta \vartheta_{min}}{\Delta \vartheta_{max} - \Delta \vartheta_{min}} & \text{otherwise} \end{cases} \quad (13)$$

$$\delta \hat{\vartheta}_t = |\delta_{ang}(\hat{\vartheta}_t^{odo}, \hat{\vartheta}_t^{MT})|, \quad (14)$$

where the function $\delta_{ang}(\vartheta_i, \vartheta_j)$ returns the angular difference between two angles ϑ_i and ϑ_j .

This means that headings of the measured tangents that differ greatly from the odometry estimate, which is calculated by adding the last estimate and differential odometry information, are considered to be unreliable and are thus integrated with a smaller weight. During the experiments the following parameters were used: $\Delta \vartheta_{min} = 5.0^\circ$, $\Delta \vartheta_{max} = 180.0^\circ$, $\gamma_{min} = 3.0$, $\gamma_{max} = 100.0$.

5.2 Person Tracking

Compared to the example introduced in the last section, tracking of freely moving persons cannot provide the same degree of accuracy for the reasons mentioned above. This can be seen in Fig. 6, which shows the path of two simultaneously tracked children wearing differently coloured hats. The paths were recorded during a public presentation in a crowded room. In addition to the fact that the maximum speed had to be assumed to be much higher (for the measurement shown in Fig. 6 a value of $v_{max} = 1$ m/s was applied) no “odometry” information is available. Furthermore the room was extremely crowded and the children were often occluded by larger persons. Finally, the hat was so greatly demanded by the children that they started to chase after it, and thus the hat often rapidly changed its position. Considering these rough conditions, the positioning system produced reasonable results and proved to be able to recover quickly if the hat was not visible for some time. The result that is shown in Fig. 6 exhibits one child (person A) that tried to produce straight lines on a monitor that was used to display the positions online. Meanwhile the other child (person B) walked around and stood still several times.

6 Determining the Heading

To be able to determine the heading directly with W-CAPS, a differently coloured stripe was added to the cardboard hat, as shown in Fig. 7. The 2D coordinates of such an object can still be tracked as explained in Section 4 using a combined colour range as

$$\Gamma = \bigcup_{i=1}^2 [(r_{min}^i, g_{min}^i, b_{min}^i), (r_{max}^i, g_{max}^i, b_{max}^i)]. \quad (15)$$

In addition, the heading can be determined from each snapshot in three different ways. Two estimates can be calculated from the relative position of the vertical centre of both stripes in the middle of the hat. These centres are indicated in Fig. 7 by two crosses while the middle of the hat is indicated by a broken line. The vertical position relative to the height of the hat can be converted to an estimate of the heading by applying a linear transformation.

Another estimate can be calculated from the relative amount of pixels of both colours. Assuming a parallel projection, the relation between the relative number of pixels of colour 1 and colour 2 (N_1, N_2) and the heading ϑ is given by

$$\frac{N_1}{N_1 + N_2} = \frac{\int H \frac{|\varphi|}{\pi} \cdot R \cos(\varphi - \vartheta) d\varphi}{2RH} = \frac{1}{2} - \frac{\cos(\vartheta)}{\pi}, \quad (16)$$

where H indicates the height, and R the radius of the hat.

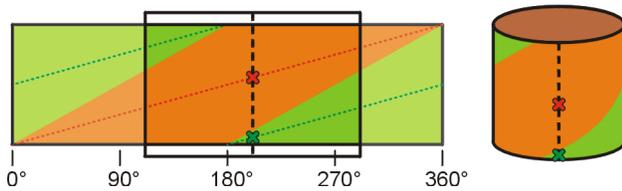


Figure 7: Design of the coloured hat used to determine the heading.

Because the quantity considered is symmetric with respect to 0 and 180, the correct branch of the *arccos*-function has to be chosen if ϑ has to be calculated from $N_1/(N_1 + N_2)$. This can be done by comparing the above mentioned positions of the centre of both colours in the middle of the hat. Considering the example given in Fig. 7, the inverse function of eq. 16 can be made non-ambiguous by restricting the range to values between 180 and 360. This follows because the centre of the stripe that starts at 0 (the orange one) is above the other stripe's centre in the middle of the hat.

7 Conclusions

The absolute positioning system W-CAPS was introduced in this paper. Based on an arbitrary number of web-cameras installed at fixed positions, it is possible to track the 2D coordinates of a coloured object. A coloured hat made of cardboard was used, which can be easily placed on top of a robot or worn by a person. Furthermore, a method was proposed to extend the positioning system in order to directly track the heading also. Therefore, the hat was augmented with a differently coloured stripe. First results are promising, but the implementation of this feature is not fully completed yet.

W-CAPS has been utilised successfully in a number of experiments with a mobile robot, including gas source localisation [5] and gas distribution mapping [6]. In addition, the system has been used to track the position of people (wearing the hat) in order to train a neural net for person tracking [1]. In addition, W-CAPS is currently being adapted by the RoboCup team TeamSweden [9], in order to provide ground truth information about the position of their legged robots. Thus, the self-localisation method applied can be verified. Moreover, together with a record of the internal states of the robots, externally logged position data are also expected to ease the analysis of the robots behaviour.

Up to now, W-CAPS calculates an estimate by averaging over those triangulated positions that are considered reliable. Future work might also investigate whether the accu-

racy of the system can be increased by assigning different weights to individual triangulation results. These weights might be related either to the total distance to the cameras used or to the angular difference between them.

References

- [1] G. Cielniak, M. Miladinovic, D. Hammarin, L. Göransson, A. Lilienthal, and T. Duckett. Appearance-based tracking of persons with an omnidirectional vision sensor. In *Proceedings of the Fourth IEEE Workshop on Omnidirectional Vision (Omnivis 2003)*, Madison, Wisconsin, 2003.
- [2] H. R. Everett. *Sensors For Mobile Robots*. A K Peters Ltd., 6th edition, 1995.
- [3] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1985)*, pages 500–505, 1985.
- [4] A. Lilienthal. Gas Source Localisation with a Mobile Robot. PhD thesis, to appear, 2004.
- [5] A. Lilienthal and T. Duckett. Experimental Analysis of Smelling Braitenberg Vehicles. In *Proceedings of the IEEE International Conference on Advanced Robotics (ICAR 2003)*, Coimbra, Portugal, 2003.
- [6] A. Lilienthal and T. Duckett. Gas Source Localisation by Constructing Concentration Gridmaps with a Mobile Robot. In *Proceedings of the European Conference on Mobile Robots (ECMR 2003)*, Warsaw, Poland, 2003.
- [7] E. Prigge and J. P. How. An Indoor Absolute Positioning System with No Line of Sight Restrictions and Building-Wide Coverage. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2000)*, pages 1015 – 1022, 2000.
- [8] C. Randell and H. L. Muller. Low Cost Indoor Positioning System. In *International Conference on Ubiquitous Computing UbiComp 2001*, pages 42 – 48, 2001.
- [9] A. Safiotti, A. Björklund, S. Johansson, and Z. Wasik. Team Sweden (Team Description). *RoboCup 2001: Robot Soccer World Cup V*, pages 725 – 729, 2002.