A Tracking Accuracy Guarantee System on Augmented Reality

Zhiqiang Bian[1], Hirotake Ishii[1] and Hiroshi Shimoda[1]

[1]Graduate School of Energy Science, Kyoto University, Uji, Japan
(Tel: +81-774-38-4405, E-mail: bianzq@uji.energy.kyoto-u.ac.jp)

Abstract: Improvement of tracking accuracy is an important issue when applying augmented reality to nuclear power plant fieldwork. Tracking accuracy depends highly on the marker arrangement when employing a tracking method using a camera and markers. For those reasons, this study develops a wheel tracking error computation method to compute the tracking error from the marker arrangement and errors in the screen coordinate. An evaluation experiment was conducted using a kind of linecode markers developed before. Experimental results show that the tracking error computation is reliable and the speed of the tracking error computation is affordable to be applied in real time error estimation in NPP field work support.

Keywords: Augmented reality, tracking accuracy, tracking error estimation method, Marker Arrangement, design system, evaluation method, industrial plant, maintenance support, linecode marker

1. INTRODUCTION

In Japan, nuclear power plants (NPPs) must undergo maintenance every 13 months. The NPPs' operation must be halted for about one month to disassemble and check equipment during the maintenance period. The NPP components are, however, much more numerous than those of a thermal power plant of similar scale. Moreover, NPP maintenance work is much harder because the NPP structure is typically more complicated. In addition, lack of expert field workers has become a widespread problem. Simultaneous improvement of maintenance work efficiency and reduction of human error are desired to improve NPPs' competitive power after the impending liberalization of the electric market.

Augmented reality (AR) technology is broadly used for entertainment, education, surgery simulations, and other applications [1]. AR offers great possibilities to support NPP fieldwork. Using AR technique, we can identify 2D and 3D positions more intuitively than when using a legacy interface such as paper instruction documents. For example, annotations, informative figures, 3D maps, and videos can be superimposed in the field worker's view. It is expected that the workers can work more accurately and rapidly using AR support.

An accurate and reliable tracking method is indispensable to apply AR technology to NPP fieldwork. To date, many tracking methods have been developed. Tracking technologies, such as global positioning systems, infrared rays, stereo view cameras, and ultrasonic and magnetic fields are popular now[1]. Although these tracking methods can realize accurate tracking at a certain level, they are difficult to apply to NPP fieldwork because these methods do not meet all requirements from the viewpoints of accuracy, stability, economics, and availability for the NPP field work[2][3]. Consequently, we sought to apply tracking methods that use cameras and markers (marker-based tracking) to NPP fieldwork. To achieve high reliability on tracking, we will develop a high reliability tracking method in this study with assurance with tracking error estimation (TEE). When tracking error is larger than a certain threshold, user will be warned because the tracking accuracy is not reliable at this time.

According to the high reliability requirement of NPP field work support, the TEE algorithm should satisfy the following requirements.

(1) Find the largest possible tracking error rather than probabilistic average error.

(2) It must be sufficiently fast for practical use.

Traditional TEE algorithms, such as [4], can not meet these requirements. Therefore, a wheel TEE algorithm is developed to meet these requirements.

2. THE WHEEL TRACKING ERROR COMPUTATION ALGORITHM

2.1 Definitions and Settings

Figure 1 shows the coordinate definitions used for this study. Screen coordinates are defined as the origin at the center of the screen D; the x axis extends left to right, and the y axis extends upward on the screen. The camera coordinate is defined as the origin at the camera's focus point C, the z direction to the line of sight of the camera. The x and y directions of the camera coordinate are the same as the screen coordinate.

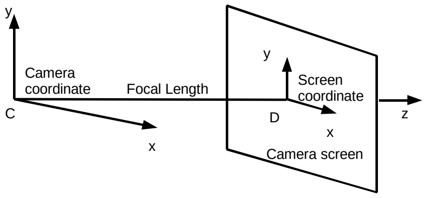

Fig. 1 Coordinate system.

The camera pose is a six-dimensional vector including the camera's three-dimensional (3D) position and 3D rotation.

Pose C' satisfies that the camera at C' can capture the same markers as the camera at C. Here the following variables are used: Q_i, i=1, 2… n are the 3D position of visible markers in the camera coordinate; n is the number of visible markers; m_i, a 2D vector, is the image point of Q_i in pose C screen coordinate; m'_i is the image point of Q_i in pose C' screen coordinate. The image difference of Q_i between the camera at C and that at C' is defined as $||m_i\text{-}m'_i||$. The largest image difference (LID) between the camera at C and that at C' is defined as the largest image difference among any two correspondent marker pairs, i.e. m_i and m'_i, in both images captured by camera at C and at C'. Equation (1) shows the definition of LID.

$$\mathrm{LID(C',C,M)}=\max_{\mathrm{i=1,2,\dots,n}} ||\mathrm{m_i}\text{-}\mathrm{m'_i}|| \qquad (1)$$

In that equation, M is an MA. The marker's position accuracy on the screen is always limited because the camera resolution is limited when the marker is recognized using image processing. The position accuracy is assumed to be Δ. When $\mathrm{LID(C', C, M)} \le \Delta$, the pose of C' and C can not be distinguished and we say that C and C' can capture the same image. The tracking error region (TER) is a set of points at which a camera can capture the same image as the real pose C, where C' is one pose in the TER.

2.2 Computation of Limitation Circles

Next, the algorithm for computing TER will be described. Presuming that n markers are captured by the camera (n should be larger than 3 for tracking¥cite{p3p}), compute TER as the intersection region of following areas in the x-z plane: (1) for each 2 markers among the n markers, the area in which the camera can capture the same x-coordinates of the two markers; (2) for each single marker, the area in which the camera can capture the same y-coordinates of the marker. These areas are enclosed by circles. These circles are called limitation circles.

2.2.1 Limitation Circles for the x Coordinate

In Fig. 2, P_A and P_B are projection points of Q_A (x_a, y_a, z_a) and Q_B (x_b, y_b, z_b) on the x-z plane in camera coordinate. A is the intersection point between CP_A and the screen x axis. B is the intersection point between CP_B and the screen x axis. A, A_1, A_2, B, B_1, B_2 are on the screen x axis and set $||A_1A||=||AA_2||=||B_2B||=||BB_1||=\Delta$. We cannot distinguish A_1, A_2 from A and B_1, B_2 from B because their differences in screen coordinate are smaller than Δ. Consequently, pose C', which can capture same x dimension images of P_A and P_B as the real camera pose, satisfies

$$\alpha_2 \le \angle P_A C' P_B \le \alpha_1 \qquad (2)$$

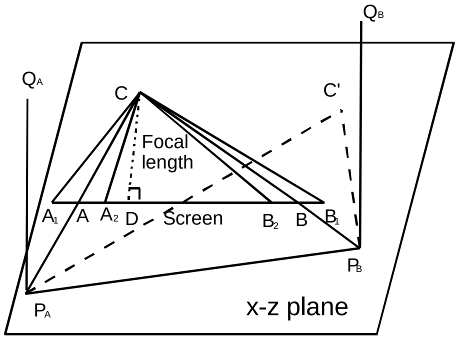

Fig. 2 Computation method for the track of C': the camera at C' can capture the same images of Q_A and Q_B in the x coordinate.

Here $\alpha_2 = \angle A_2CB_2$ and $\alpha_1 = \angle A_1CB_1$. They can be computed from

$$\tan\angle A_1CD = \tan\angle ACD + \frac{||A_1A||}{f} = \frac{x_a}{z_a} + \frac{\Delta}{f} \qquad (3)$$

$$\tan\angle A_2CD = \tan\angle ACD - \frac{||A_2A||}{f} = \frac{x_a}{z_a} - \frac{\Delta}{f} \qquad (4)$$

$$\tan\angle B_1CD = \tan\angle BCD + \frac{||B_1B||}{f} = \frac{x_b}{z_b} + \frac{\Delta}{f} \qquad (5)$$

$$\tan\angle B_2CD = \tan\angle BCD - \frac{||B_2B||}{f} = \frac{x_b}{z_b} - \frac{\Delta}{f} \qquad (6)$$

where D is the center of the screen and f is the focal length of the camera. The track of C' on $\angle P_AC'P_B = \alpha_1$ and $\angle P_AC'P_B = \alpha_2$ are circles 1 and 2, as shown in Fig. 3. The C' that satisfies (2) should be in the gray area between the two circles.

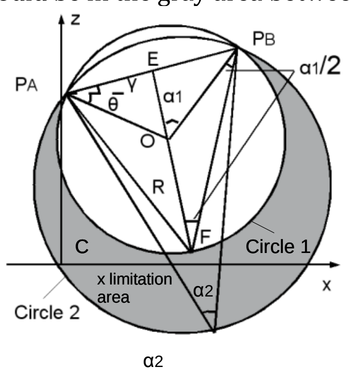

Fig. 3 Computation method of a limitation circle

The computation methods for circles 1 and 2 are the same. Here we take circle 1 for description. First assume that F is on circle 1 and satisfies $||P_AF|| = ||P_BF||$ to compute the center of circle 1, and that E is on P_AP_B and $||P_AE||=||P_BE||$. Because $||P_AF|| = ||P_BF||$, EF is midnormal of P_AP_B. Therefore, the center of circle 1, point O (x_O, 0, z_O) , is on EF. Because $||OF||=||OP_B||=R$ and $\angle P_AFP_B= \alpha_1$ (R is the radius of circle 1), $\angle OFP_B = \angle OP_BF = \alpha_1/2$. Therefore,

$$R=\frac{||P_AE||}{\sin\angle EOP_B}=\frac{||P_AP_B||}{2\sin\alpha_1} \qquad (7)$$

The slope angle of P_AP_B is $\gamma=\arctan\dfrac{z_b-z_a}{x_b-x_a}$; the slope angle of P_AO is $\theta=\alpha_1+\gamma-\pi/2$. Therefore,

$$x_O=x_a+R\cos\theta \qquad (8)$$

$$y_O=y_a+R\sin\theta \qquad (9)$$

The gray area in Fig.3 is called the x limitation area. We can compute the x limitation area for any two markers in the n markers as for Q_A and Q_B.

2.2.2 Limitation Circles for the y Coordinate

Consider the y direction in the screen coordinate, as shown in Fig. 4. Hereby, set $||A_1A||=||AA_2||=\Delta$, CP_A as the projection line of CQ_A on the x-z plane and A_2A_1 is parallel to y axis of the screen coordinate. D is the center of the screen and $||CD||=f$. E is a projection point of A on the x axis of the screen coordinate.

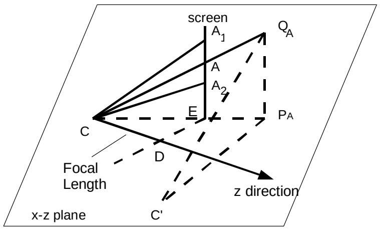

Fig. 4 Computation Method for Track of C': the camera at C' can capture the same images of Q_A in the y coordinate.

Consequently, C', which can capture the same image of Q_A in y direction as C, should satisfy

$$\beta_2\le\angle Q_AC'P_A\le\beta_1 \qquad (10)$$

Hereby $\beta_2=\angle A_2\,C\,P_A$ and $\beta_1=\angle A_1CP_A$. Centers of both circles are P_A. Radii of the circles can be computed as $||Q_AP_A||\,ctan\beta_2$ and $\big||Q_AP_A|\big|ctan\angle\beta_1$.

Here,

$$\beta_2 = atan(tan\angle Q_A\,C\,P_A-\Delta\,cos\angle DCE) \quad (11)$$

$$\beta_1 = atan(tan\angle Q_A\,C\,P_A+\Delta\,cos\angle DCE) \quad (12)$$

$$\angle DCE = atan(x_a/z_a) \quad (13)$$

The area (a ring) between the two circles is called the y limitation area. Compute y limitation areas for any marker in the n markers as for Q_A.

2.3 Computation of Tracking Error Region and Tracking Error

The intersection region of all the x and y limitation areas is TER because the camera can capture all the markers exactly as the real pose. The computation method for the TER is shown in Fig. 4. Here, to introduce the computation method easily, we use a three-marker MA. For a n-marker MA, the computation method is the same.

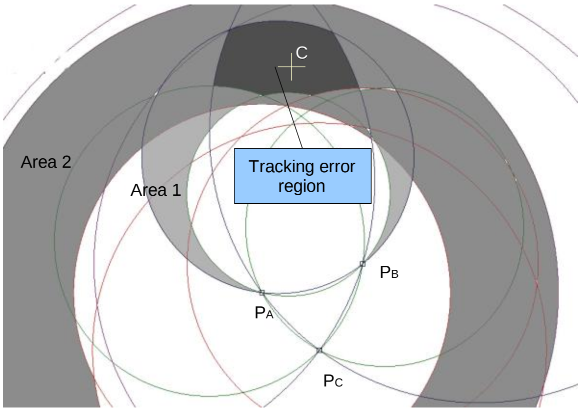

Fig. 4 Sample image of tracking error region.

The computation method is called the wheel TEE algorithm. In the following, we will explain the TER computation procedure. All of $Q_A, Q_B,\ and\ Q_C$ are respectively projected to $P_A, P_B\ and\ P_C$ in the x-z plane of the camera coordinate. As shown in Fig. 4, the camera at any position of Area 1 (x limitation area of Q_A and Q_B) can view the same x-dimension images of Q_A and Q_B; the camera at any position of Area 2 (y limitation area of Q_A) can view the same y image of Q_A. Compute x limitation areas for (Q_B, Q_C) and (Q_A, Q_C) similarly as that for (Q_A, Q_B) . Compute y limitation areas for Q_B and Q_C similarly as that for Q_A. The camera can view the same images of the three markers in the intersection region of all of the above regions: this intersection region is the TER.

We can compute the edges of the TER as a set of arcs. The most distant position in the TER from the real camera must be at one intersection point of the circles introduced above (proved in section 2.4). We must compute two circles to generate an x limitation area for each two markers and compute two circles to generate a y limitation area for each marker when the MA has n markers. We must compute n (n+1) circles and n (n+1)/2 limitation areas for n markers. Then compute the intersection points of these circles, find the farthest intersection point which can be in all limitation areas. The distance between the point and C is the tracking error.

The above method can compute tracking error on x-z plane as e_{xz}. By rotating the MA through z axis, tracking error of y direction can also be computed by above algorithm as e_{yz}. In the evaluation experiment, tracking error e is max(e_{xz}, e_{yz}).

2.4 Why Computing only Intersection Points

We need to prove that the farthest point in the TER is one intersection point of all the limitation circles. Also, the most distant point in any intersection arc to C is one of the ends of the arc.

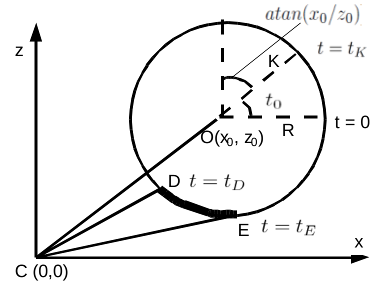

Fig. 5 Proof for the largest point is at one end of arc.

As shown in Fig. 5, C is the origin of the camera coordinate. One limitation circle O is at $(x_O, 0, z_O)$ with radius of R. Also, $z_O > 0$ because marker(s) that generated the circle O should be visible to the camera. D and E are two intersection points between circle O and other circles. Now we assume that $\widehat{DE}$ is one edge of the TER, and that K is the most distant point from C on circle O and ||CK|| = ||CO|| + R. We must prove that the greatest distance from C to any point on $\widehat{DE}$ is either ||CD|| or ||CE||. The problem can be described as

$$max\, f(t) = x^2 + z^2, \tag{14}$$

where

$$x = x_0 + R\, cos(t) \tag{15}$$

$$z = z_0 + R\, sin(t) \tag{16}$$

strict to

$$t \in [t_D, t_E] \subset [\frac{\pi}{2} - atan\frac{x_0}{z_0}, \frac{5\pi}{2} - atan\frac{x_0}{z_0}]. \tag{17}$$

Equation (17) includes the assumption that $\notin \widehat{DE}$. Because (1) there is at least one marker on circle O or at O and (2) $\widehat{DE}$ is one edge of the TER, if $\in \widehat{DE}$, then the tracking error is equal or greater than ||CK||. This tracking error is larger than the distance between the marker and the camera. Therefore, this kind of MA is not feasible and must be modified. The case in which $K \in \widehat{DE}$ can be determined and eliminated by checking whether $t_K \in (t_D, t_E)$. Therefore, we need consider only the case for which $K \notin \widehat{DE}$.

Substitute (15) (16}) into (14):

$$\max f(t') = K_1 + K_2 \sin(t') \tag{18}$$

where $t' = t + atan(x_0/z_0)$, $K_1 = x_0^2 + z_0^2 + R^2$, $K_2 = 2R\sqrt{x_0^2 + z_0^2}$, $t_D' = t_D + atan(x_0/z_0)$, $t_E' = t_E + atan(x_0/z_0)$.

We prove the problem by contradiction. Assume that $\exists\, t' \in (t_D', t_E')$ satisfies $f(t') > f(t_D')$ and $f(t') > f(t_E')$. From (18),

$$f(t') - f(t_D') = 2K_2 sin\frac{t' - t_D'}{2} cos\frac{t' + t_D'}{2} > 0 \tag{19}$$

$$f(t') - f(t_E') = 2K_2 sin\frac{t' - t_E'}{2} cos\frac{t' + t_E'}{2} > 0 \tag{20}$$

Because $sin\frac{t' - t_D'}{2} > 0$, $sin\frac{t' - t_E'}{2} < 0$ and $K_2 > 0$,

$$cos\frac{t' + t_D'}{2} > 0,\quad cos\frac{t' + t_E'}{2} < 0. \tag{21}$$

Furthermore, $\frac{t' + t_D'}{2}, \frac{t' + t_E'}{2} \in [\frac{\pi}{2}, \frac{5\pi}{2}]$. Therefore,

$$\frac{t' + t_D'}{2} > \frac{3\pi}{2} > \frac{t' + t_E'}{2}. \tag{22}$$

This contradicts $t_E' > t_D'$. Therefore, $f(t') < f(t_D')$ or $f(t') < f(t_E')$, which means that ||CD|| or ||CE|| is the longest distance from C to any point on $\widehat{DE}$.

3. EVALUATION EXPERIMENT

3.1 Linecode Marker and its Tracking Method

In this section, we present the shape design of a linecode marker and the linecode parameters (marker size, distance between markers, etc.), along with a determination method for the tracking range.

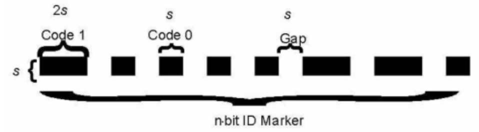

Fig. 6 Conceptual image of linecode marker

Fig. 6 shows a conceptual illustration of the linecode marker. The linecode marker is a combination of black elements: each element corresponds to one bit. The square element signifies "0". The double-sized rectangle element denotes "1".

In this study, bit number n=8 and s=30mm.

The tracking method is a traditional P3P ¥cite{p3p} method with least square optimization for other feature points except the used 3 feature points. Only the top elements and bottom elements of the linecode markers are used for tracking¥cite[linecode].

3.2 Δ Computation

Δ is computed from two types of screen errors, error in recognition Δ_r and error in tracking Δ_t.

Here Δ_r is computed from the characteristics of the linecode marker. Δ_t can be obtained from the tracking method. Δ_r can be computed from perpendicular recognition error Δ_p and inline recognition error Δ_i.

Δ_p is defined as the distance between the center of the selected element (the top element or the bottom element) and the least square line of all the elements of the linecode marker.

The distance between the top and bottom element can be computed as $(n\sum c_i + n - 2 -$

$\frac{c_0+c_1}{2})s$, where c_i the code of the ith element in the linecode marker.

Δ_i is computed as follows,

$$d_i = (i\sum_{j=0}^{i-1} c_j + i - 2 - \frac{c_0+c_{i-1}}{2})s \qquad (23)$$

$$\Delta_i = \max_i |\sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2} - d_i| \quad (24)$$

Here, (x_i, y_i) is the screen position and $(x_i^{'}, y_i^{'})$ is the projected screen position of the ith selected feature points. k is two times of the number of the linecode markers.

Δ_t is defined as the difference between the feature points on the screen and the projected feature points computed from the tracking result pose[2].

$$\Delta_t = \max_i \sqrt{(\mathrm{x_i} - \mathrm{x_i^{'}})^2 + (\mathrm{y_i} - \mathrm{y_i^{'}})^2} \qquad (25)$$

At last the Δ is decided to be

$$\Delta = \Delta_r + \Delta_t \qquad (26)$$

where $\Delta_r = \sqrt{\Delta_p^2 + \Delta_i^2}$ because these two error are perpendicular to each other.

3.3 Experimental Configuration

Figure 7 portrays the experiment configuration. The global coordinate is defined as the origin at O in Fig. 7, the x direction is set toward the reader, the y direction is set to up, and the z direction is set to the left. All markers are placed vertically. The placement error is within 1 cm. This configuration is adopted to simulate the complex and large-scale environment in the NPP field. The camera is placed before the markers and the camera is moved from 0 to 8 m in the x direction and -2 to -11 m in the z direction. The camera was rotated at every point with 0, 20, 40 degrees (to the direction in which markers can be captured). The average illumination condition in the room is 1050 lux. No linecode marker-like articles exist in the experiment room.

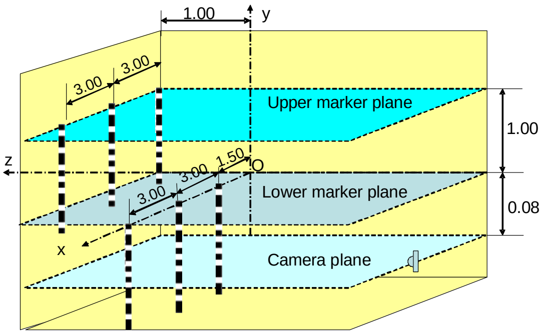

Fig. 7 Experimental setup (unit: m).

3.4 Experimental Results on Accuracy

The real tracking error when the camera is placed at 0, 20, and 40 degrees are shown from left to right in Fig. 8. The estimated tracking error is shown in Fig. 9. The distance error is illustrated using the circle's diameter and grayscale. The circle's diameter is fixed and the error (unit: mm) is drawn on the circle if the error is greater than 40 cm.

The result shows that the wheel TEE algorithm can compute tracking error correctly. The computation time is in 1 ms and therefore it is possible to implement the wheel TEE algorithm into the tracking method in real-time.

4. CONCLUSION

A wheel TEE algorithm is developed and evaluated in long distance tracking. The result shows the wheel TEE algorithm is feasible for implementing a high reliability tracking method.

REFERENCES

[1] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent Advances in Augmented Reality, IEEE Computer Graphics and Applications November/December, pp. 34-47, 2001.

[2] Z. Bian, H. Ishii, M. Izumi, H. Shimoda, and H. Yoshikawa, "Development of a tracking method for augmented reality applied to NPP maintenance work and its experimental evaluation," Proc. of ACM Symposium on Virtual Reality Software and Technology 2006, pp. 35-44, November 2006.

[3] H. Ishii, Z. Bian and T. Sekiyama, "Development and Evaluation of Tracking Method for Augmented Reality System for Nuclear Power Plant maintenance Support," Maintenology, Vol. 5, No. 4, pp. 59-68, January 2007.

[4] W. Hoff, and T. Vincent, "Analysis of head pose accuracy in augmented reality," IEEE Transactions on Visualization and Computer Graphics, vol. 6, pp 319-334, October 2000.

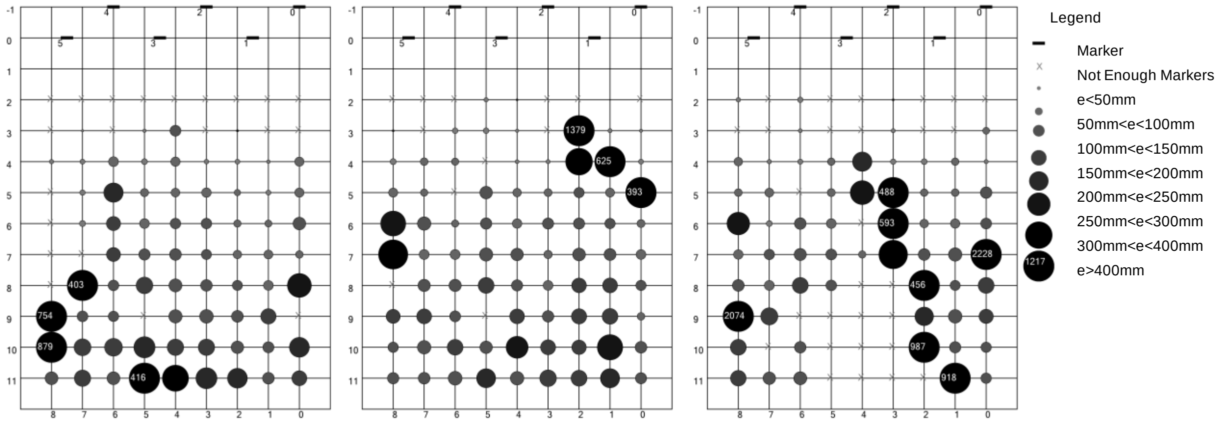

Fig. 8 Real tracking error: from left to right, camera is set at 0°, 20°, 40°

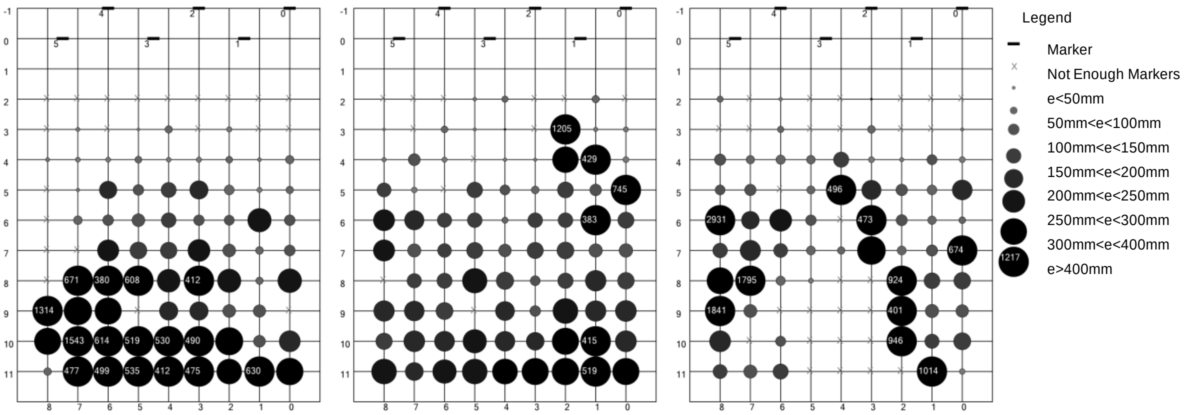

Fig. 9 Estimated tracking error: from left to right, camera is set at 0°, 20°, 40°