Optimizing Camera Placement for Localization Accuracy

Dávid Szalóki, Norbert Koszó, Kristóf Csorba, Gábor Tevesz
Department of Automation and Applied Informatics,
Budapest University of Technology And Economics,
H-1117 Budapest, Magyar Tudósok körútja 2/Q., Hungary
{David.Szaloki, Norbert.Koszo, Kristof.Csorba, Gabor.Tevesz}@aut.bme.hu

***Abstract*—This paper presents the optimization of a camera placement for improved localization accuracy. It is the basis of a localization system aiming for high localization accuracy. This goal is achieved using several cameras with very redundant field-of-views. The paper presents the calculation of the localization accuracy — which depends on the camera model and the pixel quantizing error — at one specific point. A method is introduced for handling areas in a probabilistic sense instead of examining only one point. The accuracy can be improved by adding a new camera to the system. The calculation of the optimal position of the new camera at compliance to some restrictions is demonstrated. The Smart Mobile Eyes for Localization (SMEyeL) project is open-source: the source code, all measurement input data and documentation are public available.**

I. Introduction

Motion tracking with multiple camera systems has many applications nowadays. Some examples are robotic swarm applications, human or wildlife tracking in surveillance systems etc. Regarding the field of view of the cameras, there are two main application directions: the ones with relative few overlapping, and the ones with high redundancy in fields of view. A typical application for the first is video surveillance where the coverage of the system has to be maximized. The second category consists of applications requiring 3D reconstruction, or high accuracy in positioning or timing. The proposed calculations aim this category.

A widely known field of computer vision is the PnP problem which stands for Perspective-n-Point problem meaning the estimation of the pose of a calibrated camera from n 3D-to-2D point correspondences. There are several papers about the solution of the $P3P$, $P4P$ or the generalized PnP problem. Some of them are originating the PnP problem in the simpler ones. There are given for this estimation iterative [1] and non-iterative [2] [3] methods as well. In applications using feature point-based camera tracking [4] [5] dealing with hundreds of noisy feature points is required. Detailed summary about the PnP can be found in [6]. All of these generalized methods can be improved if the nature of the noise is known. Thus our first goal is to determine the localization accuracy. The calculated localization accuracy can be used to validate the measured localization accuracy in [7].

Another popular topic is the object tracking. Assume that we have an object tracking system with fixed and moving cameras. If the movable cameras are precisely controlled by the system, their position and orientation can be determined to improve the localization accuracy. Thus our second goal is to calculate the optimal position of a newly added camera for accuracy improvement. The movable camera can be handled as it would newly added to the system containing the fixed cameras. Of course there are limitations for the placement of these cameras and these restrictions has to be applied to the model. Using the motion prediction of the tracked object during the camera movement can cause more accuracy increase. Since the prediction is every time given in the form of a density function, the calculation of the optimal position has to deal with such an inaccurate position where the localization has to be done. Thus our calculation of the optimal placement is prepared for that.

In this paper we present the calculation of the localization accuracy using several cameras. The localization error is caused by the pixel quantizing error and that one camera alone can not provide depth information. The determination of the optimal position of a newly added camera is presented. The usage of the density function is introduced. With this the problem can be handled in probabilistic sense and operate with an area instead of only one observed point.

Please note that the presented Smart Mobile Eyes for Localization (SMEyeL) system is open-source. It is written in C++ using the popular OpenCV [8] computer vison library. Its source code, documentation and all the input data for the presented measurements are available for download from our homepage [9]. The remaining part of the paper is organized as follows: section II introduces the calculation of the localization accuracy caused by the pixel quantizing error. In this section the following cases are presented: the localization accuracy of one camera at one specific point, using several cameras for one observed point, and examining an area with a density function instead of only one point. Section III demonstrates the calculation of the optimal position of the newly added camera. Section IV contains the experimental results regarding localization accuracy and optimal position, and finally, section V concludes the paper.

II. Calculation of the position accuracy

Assume that we have an object at location $\boldsymbol{x}$. An estimation of $\boldsymbol{x}$ has to be determined. For every $\boldsymbol{x}$ in the space the accuracy of the localization can be calculated.

978-1-4799-0197-5/13/$31.00 ©2013 IEEE

At first we will have some definitions. After that the accuracy will be calculated in case of using only one camera in 2D. This can be easily generalized to 3D. Next the accuracy will be calculated in case of using a camera configuration which contains several cameras for localization.

A. Definitions

Let one camera have the $\boldsymbol{R}$ rotation matrix and $\boldsymbol{t}$ translation vector. This means, that

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix}, \tag{1}$$

where $\boldsymbol{T}$ is the matrix of extrinsic parameters. A given $\boldsymbol{x}$ point can be mapped from the world coordinate system into the camera coordinate system:

$$\boldsymbol{x}^{(c,h)} = \boldsymbol{T} \cdot \boldsymbol{x}^{(w,h)}, \tag{2}$$

where $\boldsymbol{x}^{(w,h)}$ and $\boldsymbol{x}^{(c,h)}$ are the representations of the $\boldsymbol{x}$ point in homogeneous coordinates in the world and in the camera coordinate system respectively. It can be noticed that homogeneous coordinates are used only here, the upcoming calculations do not need this form of representation. To avoid further definition let the $\boldsymbol{x}$ point be represented in the world coordinate system as $\boldsymbol{x}^{(w)}$ and in the camera coordinate system as $\boldsymbol{x}^{(c)}$.

The origo centered 2D rotation with angle ϑ is represented with $\boldsymbol{R}_\vartheta$:

$$\boldsymbol{R}_\vartheta = \begin{bmatrix} cos\vartheta & sin\vartheta \\ -sin\vartheta & cos\vartheta \end{bmatrix}. \tag{3}$$

B. Calculation in 2D for one camera

First we calculate in 2D and after that it can be easily generalized to 3D.

Let one camera have the $\boldsymbol{x}_c^{(w)}$ position and α orientation in the world coordinate system. In this case the rotation matrix and translation vector can be calculated:

$$\begin{aligned} \boldsymbol{R} &= \boldsymbol{R}_\alpha^T, \\ \boldsymbol{t} &= -\boldsymbol{R}\,\boldsymbol{x}_c^{(w)}. \end{aligned} \tag{4}$$

Assume, that the pixel quantizing error has a known Gaussian distribution with standard deviation e on the image plane. The error of the position detection of an object located at point $\boldsymbol{x}$ can be derived from e. The calculation can be followed on Fig. 1.

Assume that the x axis of the camera coordinate system is fitted to the camera facing direction. The distance of the object and the camera center is:

$$d = \left\| \boldsymbol{x}^{(c)} \right\|. \tag{5}$$

The object is seen by the camera at an angle

$$\varphi = tan^{-1} \frac{x_y^{(c)}}{x_x^{(c)}}, \tag{6}$$

where $x_y^{(c)}$ and $x_x^{(c)}$ are the y and x coordinates of the $\boldsymbol{x}^{(c)}$ point respectively. $\boldsymbol{x}^{(c)}$ can be easily calculated from $\boldsymbol{x}^{(w)}$ using (2).

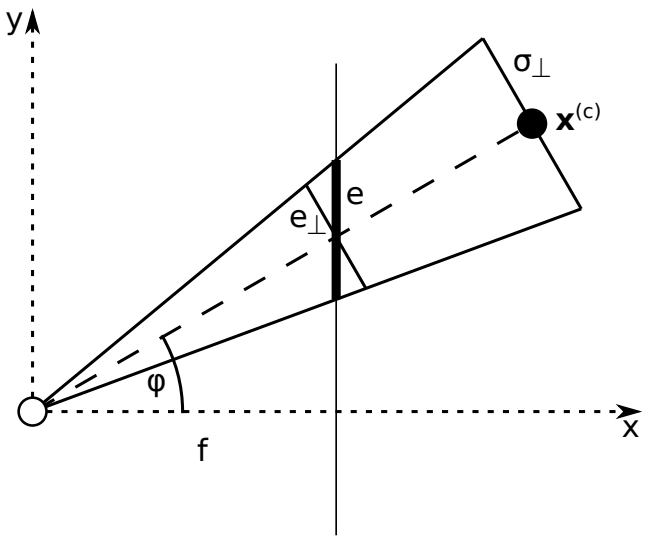

Fig. 1. The component of the localization error perpendicular to the facing direction can be calculated for one camera. The empty circle is the camera center and the filled circle is the observed point. The axes are the dotted arrows. The x axis is fitted to the facing direction of the camera. The solid vertical line is the image plane on which the pixel quantizing error e occurs. The observed point is detected at the angle φ. f is the focal length. The component of e perpendicular to the facing direction is $e_\perp$, and $\sigma_\perp$ is the perpendicular component of the localization error.

The component of e perpendicular to the facing direction is

$$e_\perp \approx e \cdot cos\varphi. \tag{7}$$

Fig. 1 is distorted for the sake of illustration, but in real the focal length is much larger than the pixel quantizing error. This means that the assumption in (7) is a good approximation. This error has to be scaled up to the object location:

$$\sigma_\perp = \frac{d \cdot cos\varphi}{f} \cdot e_\perp \approx d \cdot cos^2\varphi \cdot \frac{e}{f}, \tag{8}$$

where f is the focal length of the camera and $\sigma_\perp$ is the perpendicular component of the localization error. This means that the perpendicular component of the standard deviation is proportional with the square of the cosine of the detection angle. Thus the more the camera detects the object at the edge of the image, the smaller the positioning error derived from the pixel quantizing error is. It can be a little strange, but if we have calibrated cameras we can accept that this is true. A calibrated camera represents a model, where the distorsions are identified and the pinhole camera model is approximated during the mapping. Detailed information about the camera model and the calibration process can be found in [10].

The parallel component of the localization error is symbolized with $\sigma_\parallel$. One camera alone can not provide distance information parallel to the facing direction:

$$\sigma_\parallel \to \infty. \tag{9}$$

The discussion of this fact is presented later in this paper.

For a point in the real world the covariance matrix can be constructed using (1) – (9). This matrix is diagonal in

the coordinate system fitted to the facing direction of the camera and it can be easily transformed (rotated) into the world coordinate system:

$$\boldsymbol{\Sigma} = \boldsymbol{R}_{\alpha}^{T} \boldsymbol{R}_{\varphi} \begin{bmatrix} \sigma_{\parallel}^{2} & 0 \\ 0 & \sigma_{\perp}^{2} \end{bmatrix} \boldsymbol{R}_{\varphi}^{T} \boldsymbol{R}_{\alpha}. \tag{10}$$

The diagonal part depends only on the distance (d) and the detection angle (φ), since the focal length and the pixel quantizing error is considered to be constant. α is the orientation of the camera. $\boldsymbol{R}_{\alpha}$ and $\boldsymbol{R}_{\varphi}$ are the rotation matrices in 2D with angle α and φ respectively, as it is defined in (3).

C. Generalization from 2D to 3D

The previously presented calculations can be easily generalized to 3D. All the positions $(\boldsymbol{x}, \boldsymbol{t})$ have three coordinates instead of two. The orientation of the camera is given by three angles rather than one (α), thus the rotation matrix is the combination of three rotations. There are two different pixel quantizing errors (e), two detection angles (φ) and two perpendicular components ($e_{\perp}, \sigma_{\perp}$) as well. But there is only one parallel error component ($\sigma_{\parallel}$). Thus the covariance matrix becomes a 3-by-3 matrix rather than a 2-by-2 one.

D. Accuracy using more cameras

As we can see we can calculate the covariance matrix in every point of the world for one camera. If one single point is observed by more cameras, the resulting covariance matrix can be calculated incrementally using the product of gaussian densities:

$$\boldsymbol{\Sigma}_c = \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1}, \tag{11}$$

$$\boldsymbol{\mu}_c = \boldsymbol{\Sigma}_c \cdot \left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right), \tag{12}$$

where the two original densities are $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, while the combined density is $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$.

In Fig. 2 a camera configuration and the covariance ellipses can be seen in case of observing only one point. The filled circles are the camera centers and the little line segments symbolize the facing directions. Both of the cameras have the same parameters (pixel quantizing error and focal length), so the emplacement of the ellipses is symmetric. The covariance ellipses are drawn at a confidence level of 95 %. The two larger ones are the covariance ellipses of the localization error of the two cameras separately. In the figure, these ellipses are distorted, because the length of their major axes go to ∞. The length of the major axes in the figure are chosen so that not only two pairs of parallel lines could be seen. The resulting covariance ellipse is not distorted. It can be noticed, that the large localization error of one single camera disappeared as the resulting covariance ellipse is smaller.

A covariance matrix derived from one camera contains one eigenvalue which goes to ∞. Although its inverse can be calculated and it contains only finite elements. According to (11) the inverse of the final resulting covariance matrix is:

$$\boldsymbol{\Sigma}_r^{-1} = \sum_{i=1}^{n} \boldsymbol{\Sigma}_i^{-1}, \tag{13}$$

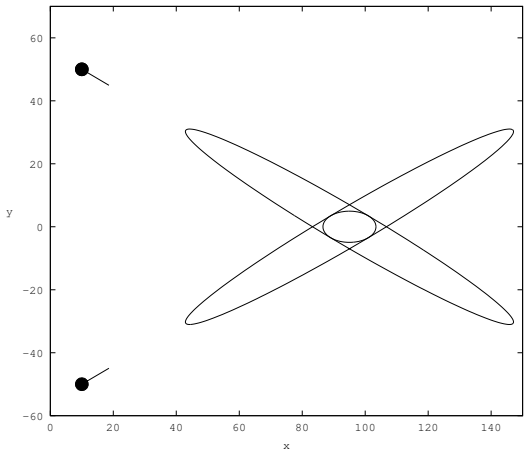

Fig. 2. The derivation of the resulting covariance ellipse. Two cameras are observing one point. The two larger ellipses are the covariance ellipses of the localization error of the two cameras separately. For better visualization, the length of the major axes of them are chosen so that not only two pairs of parallel lines could be seen.

where n is the number of the cameras used for localization and $\boldsymbol{\Sigma}_i^{-1}$ is the inverse of the covariance matrix of the i^{th} camera. Since (13) contains only the inverse of the covariance matrices the inverse of the resulting covariance matrix can be calculated and it contains only finite elements. The determinant of this matrix is zero if and only if there is a direction in space in which the localization has a variance which goes to ∞. This can occure if the observed point and the centers and the facing directions of all the cameras in the configuration fit on the same line. If the determinant of this matrix is non-zero the resulting covariance matrix can be calculated and the localization has a finite variance in any specific direction.

In Fig. 3 a camera configuration and the covariance ellipses can be seen. The notation is similar as in Fig. 2. All the cameras have the same parameters, so the emplacement of the ellipses is symmetric. It can be noticed that the errors near the cameras are smaller and they have a larger variance in the direction y, while the distant ones have this larger variance in the direction x.

The accuracy of the localization can be defined in various ways. It can be identified by the smallest eigenvalue or by the determinant of $\boldsymbol{\Sigma}_r^{-1}$.

$$q = \left|\boldsymbol{\Sigma}_r^{-1}\right|, \tag{14}$$

where q stands for quality like accuracy. In this paper we use the determinant because it has a perceptible physical meaning and it can be easily handled in analytic calculations. In Fig. 4 the accuracy is presented in case of the previous camera configuration. It can be noticed that the more closer the observed point is, the larger the accuracy becomes. This occures because the accuracy is proportional with the inverse of the distance (d^{-1}). Of course an object can not be placed any close to one of the cameras because a camera has spatial sizes and minimal focal distance.

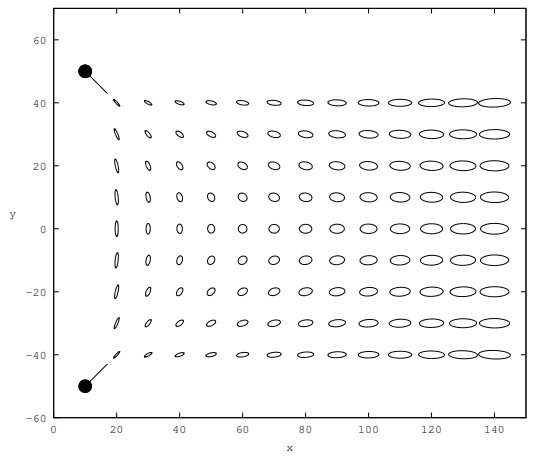

Fig. 3. The resulting covariance matrices of the localization in 2D using a specific camera configuration.

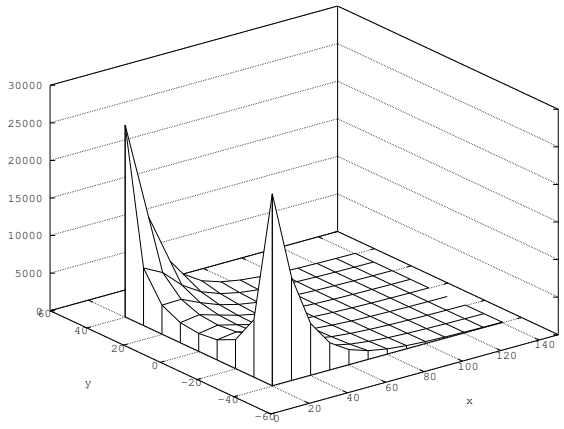

Fig. 4. The accuracy in 2D using a the same camera configuration as in Fig. 3. The accuracy is defined in (14).

In addition at this point the Field of View (FoV) is not taken into account. This can be easily added. If there is a camera with Fields of View which does not contain the observed point, then this camera has a localization accuracy equal to zero in every direction at the observed point. This means that the inverse of its covariance matrix is filled with zeros and in (11) does not count.

E. Density function

If we would like to handle the accuracy of the localization not only at one observed point then a density function can be used. It tells for a specific $\boldsymbol{x}$ point in space what is the probability that the localization has to be done at that point. In this case the expected value of the accuracy can be calculated:

$$\mathbb{E}_{\boldsymbol{x}}\left(q\left(\boldsymbol{x}\right)\right) = \int_{\Omega} q\left(\boldsymbol{x}\right) f\left(\boldsymbol{x}\right)\, d\boldsymbol{x}, \tag{15}$$

where Ω represents the whole space and $f\left(\boldsymbol{x}\right)$ is the density function.

III. Add one camera to the system

There is given a set of cameras and a point in space where the object localization is performed using these cameras. We would like to add a camera to the system to increase the accuracy. In the following sections we present a solution for this problem built from the most simple one to the generalized one.

A. Definitions

The i^{th} camera is symbolized with C_i. The set of some cameras is the camera configuration:

$$\mathbb{C}_n = \{C_i\}, \quad i = 1..n. \tag{16}$$

If the camera with index $n+1$ is added to the system the new configuration is:

$$\mathbb{C}_{n+1} = \mathbb{C}_n \cup \{C_{n+1}\}. \tag{17}$$

The position of the new camera is $\boldsymbol{x}_{c,n+1}$ and the point where the localization is performed is $\boldsymbol{x}$. The orientation of the new camera is $\boldsymbol{\alpha}_{c,n+1}$. This is a scalar value in 2D, but a three dimensional vector in 3D.

The general problem is:

$$C_{n+1}^{(opt)} = \arg\max_{C_{n+1}} \mathbb{E}_{\boldsymbol{x}}\left(q\left(\boldsymbol{x}, C_{n+1}, \mathbb{C}_n\right)\right), \tag{18}$$

where C_{n+1} contains the position and orientation of the new camera and q is the accuracy defined in (14). The expectation is defined in (15). In 3D this means that an optimal solution in the six dimensional parameter space has to be found. This problem can be simplified for the first step by calculating in 2D for one observed point with some restrictions.

The position and orientation of the new camera can be choosen in compliance to some restrictions. One of the restrictions is the spatial one which means that the camera position can be placed only within a specified area. This can prevent placing the new camera into the observed point or very close to it, although the accuracy in case of such a placement would be very high, but this placement is not feasible.

Another restriction can be the fixed or limited orientation. The fixed orientation means, that the orientation is not a variable, it is a function of the observed point and the position of the new camera. It can be a good choice to define the orientation so that the observed point is located in the center of the image. This can prevent placing the observed point on the edge of the image, although the accuracy would be higher according to (8). With the limited version the orientation is optimized taking some restrictions into account.

B. Fixed orientation

In this case the orientation is defined such that the observed point is mapped onto the center of the image. A simple analytic calculation can be performed for the accuracy. Assume that the localization at point $\boldsymbol{x}$ can be performed by the $\mathbb{C}_n$ camera set with a covariance matrix $\boldsymbol{\Sigma}_1$, and by the C_{n+1} camera

with a covariance matrix $\mathbf{\Sigma}_2$. The eigenvalues are $\sigma_{1,1}^2, \sigma_{1,2}^2$ and $\sigma_{2,1}^2, \sigma_{2,2}^2$ respectively. Assume that the eigenvalues are arranged in descending order.

According to (11) the combination of the two gaussian densities can be calculated. The accuracy is defined in (14) and it becomes the following simple form:

$$\begin{aligned} q = \sin^2\beta \cdot &\left(\sigma_{1,1}^{-2} - \sigma_{1,2}^{-2}\right)\left(\sigma_{2,1}^{-2} - \sigma_{2,2}^{-2}\right) \\ &+ \left(\sigma_{1,1}^{-2} + \sigma_{2,1}^{-2}\right)\left(\sigma_{1,2}^{-2} + \sigma_{2,2}^{-2}\right), \end{aligned} \tag{19}$$

where β is the tilt angle of the eigenvectors belonging to the first, largest eigenvalues $(\sigma_{1,1}^2, \sigma_{2,1}^2)$.

Since $\mathbf{\Sigma}_1$ is known but β and $\mathbf{\Sigma}_2$ is a function of $\boldsymbol{x}_{c,n+1}$, it is a good idea to place a coordinate system with axes aligned on the eigenvectors of $\mathbf{\Sigma}_1$ and centered on the observed point $(\boldsymbol{x})$. According to (8) and (9):

$$\begin{aligned} \sigma_{2,1} &\sim d \\ \sigma_{2,2} &\to \infty. \end{aligned} \tag{20}$$

Using (8) the accuracy becomes in the fitted coordinate system:

$$q = \frac{y_c^2}{d^4}A + \frac{1}{d^2}B + C, \tag{21}$$

where $d^2 = x_c^2 + y_c^2$ is the square of the distance of the camera and the observed point. x_c and y_c are the coordinates of the new camera in the fitted coordinate system. A, B and C are constant values, they are a function of $\sigma_{1,1}$, $\sigma_{1,2}$ and the pixel quantizing error (e) and the focal length (f) of the new camera:

$$\begin{aligned} A &= \left(\sigma_{1,2}^{-2} - \sigma_{1,1}^{-2}\right) \cdot \left(\frac{e}{f}\right)^{-2} \\ B &= \sigma_{1,1}^{-2} \cdot \left(\frac{e}{f}\right)^{-2} \\ C &= \sigma_{1,1}^{-2} \cdot \sigma_{1,2}^{-2} \end{aligned} \tag{22}$$

It can be noticed that (21) is asymmetric in the variables x_c and y_c as well as (22) is asymmetric in $\sigma_{1,1}$ and $\sigma_{1,2}$. It is caused by the fitted coordinate system placement and the descending order of the eigenvalues.

The extremal points of q can be easily calculated using its partial derivatives based on (21):

$$x_{c,max} = 0 \tag{23}$$

$$y_{c,max} = \begin{cases} \pm\sqrt{\frac{A-B}{A+B}} \cdot x & \text{if } A \geq B \\ 0 & \text{if } A < B \end{cases} \tag{24}$$

Since B is always greater than zero $y_{c,max}$ in (24) is always real.

In Fig. 5 the contours of the accuracy are presented in case of $A > B$ and $\sigma_{1,1} = 3 \cdot \sigma_{1,2}$. The curves are "iso-accuracy" curves which are centrally symmetric about the origin as we can see in (21). We can see that for a specific y_s there is only one curve which touches the $y = y_s$ line at one point as specified in (23). Similarly for a specific x_s there is only one curve which touches at two points the $x = x_s$ line as specified in (24). If there was a figure in case of $A < B$ for a specific x_s there was only one curve which touches at one point the $x = x_s$ line as specified in (24).

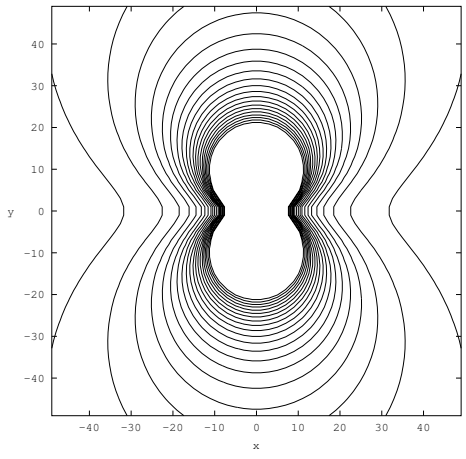

Fig. 5. The contours of the accuracy in case of adding a new camera to the system. There is given a camera configuration which localization accuracy is known at the origin. We would like to add a new camera to the system. The accuracy remains the same along the curve. The closer the new camera to the origin is, the higher the accuracy becomes.

IV. Experimental result

Our experiments are performed in 2D with the fixed orientation restriction. It can be generalized to 3D by increasing the parameter space. The fixed orientation restriction can also be avoided by optimizing the orientation like the position. This increases the dimension of the parameter space as well. The spatial restriction is also applied which prevents placing the new camera too close to the observed area. The usage of the density function makes it possible to handle the problem in probabilistic sense and operate with an area instead of only one observed point. The orientation restrictions can be also applied in this case. Fixed orientation means that the camera is facing to the direction of the expectation of the random variable $\boldsymbol{x}$ given with the density function $f(\boldsymbol{x})$.

Our main result is that using the previously mentioned calculations, methods and restrictions we can determine the best position for the new camera added to the system. The best position means where it maximizes the expected value of the accuracy.

In Fig. 6 and Fig. 7 the base camera configuration is the same as in Fig. 3. Here a new camera is added to the system. The density function is a uniform distribution on the area symbolized with a rectangle between the ellipses. The new camera is forced facing to the expected value of the position which is the center of the rectangle. The new camera position can be chosen in the rectangle above the ellipses.

In Fig. 6 the new camera is placed at the optimal position, while in Fig. 7 at some other position. The expected value of the localization accuracy is 2.5 times higher in case of the optimal placement and in both cases the accuracy is improved at all the points in space. The difference between Fig. 3 and Fig. 6 is clearly visible.

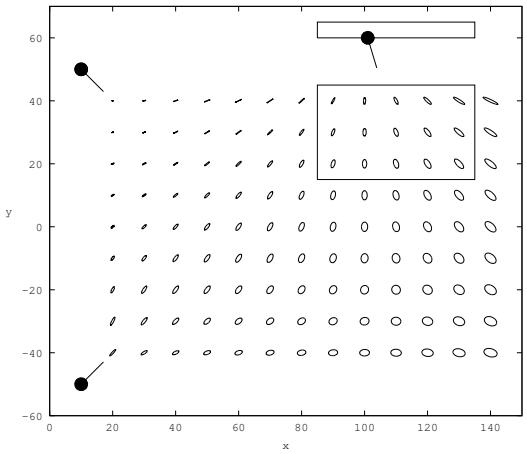

Fig. 6. The new camera placed at the optimal position. The rectangle above the ellipses symbolizes the position restriction of the new camera. The density function is a uniform distribution on the area symbolized with a rectangle between the ellipses.

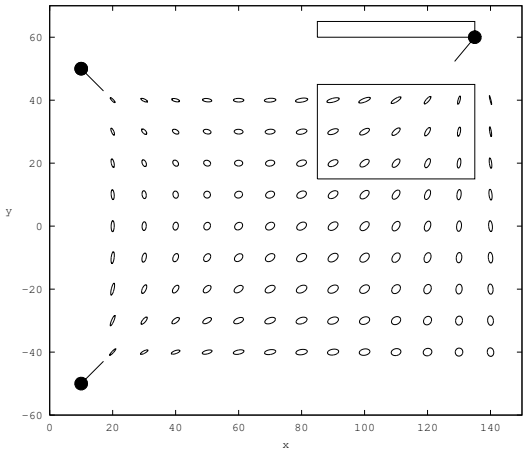

Fig. 7. The new camera placed at some position not at the optimal one. All the circumstances are similar as in Fig. 6.

V. Conclusion and future work

This paper presented the optimization of a camera placement for localization accuracy. The theorethical background of localization accuracy was discussed. The calculation of the accuracy for one camera at one observed point in 2D was introduced. It is easily generalizable to 3D. The calculation of the accuracy for more cameras was demonstrated. The use of the density function was presented which makes it possible to handle the problem in probabilistic sense and operate with an area instead of only one observed point. After that the addition of a new camera to the system was examined. The calculation of the improved accuracy achieved with the new camera was presented. At last the experimental results were introduced where the optimal placement is identified in compliance to some restrictions.

As subject of further research, the system will be enhanced to optimize the position and orientation of more than one new cameras. The Field of View (FoV) can be easily applied to the calculation of the accuracy but we would like to use it in the optimization phase as well.

Acknowledgment

This work was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013) organized by VIKING Zrt. Balatonfüred.

This work was partially supported by the Hungarian Government, managed by the National Development Agency, and financed by the Research and Technology Innovation Fund (grant no.: KMR_12-1-2012-0441).

References

[1] D. Oberkampf, D. F. DeMenthon, and L. S. Davis, "Iterative pose estimation using coplanar feature points," *Comput. Vis. Image Underst.*, vol. 63, no. 3, pp. 495–511, May 1996. [Online]. Available: http://dx.doi.org/10.1006/cviu.1996.0037

[2] J. Hesch and S. Roumeliotis, "A direct least-squares (dls) method for pnp," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 383–390.

[3] F. Moreno-Noguer, V. Lepetit, and P. Fua, "Accurate non-iterative o(n) solution to the pnp problem," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 2007, pp. 1–8.

[4] I. Skrypnyk and D. G. Lowe, "Scene modelling, recognition and tracking with invariant image features," in *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ser. ISMAR '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 110–119. [Online]. Available: http://dx.doi.org/10.1109/ISMAR.2004.53

[5] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1465–1479, Sep. 2006. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2006.188

[6] Y. Wu and Z. Hu, "Pnp problem revisited," *J. Math. Imaging Vis.*, vol. 24, no. 1, pp. 131–141, Jan. 2006. [Online]. Available: http://dx.doi.org/10.1007/s10851-005-3617-z

[7] D. Szaloki, N. Koszo, K. Csorba, and G. Tevesz, "Marker localization with a multi-camera system," in *System Science and Engineering (ICSSE), 2013 International Conference on*, 2013, pp. 135–139.

[8] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[9] Smart Mobile Eyes for Localization (SMEyeL). [Online]. Available: https://www.aut.bme.hu/Pages/ResearchEn/SMEyeL/Overview

[10] G. Bradski and A. Kaehler, *Learning OpenCV*. O'Reilly Media Inc., 2008. [Online]. Available: http://oreilly.com/catalog/9780596516130