Marker Localization with a Multi-Camera System

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 a marker localization system consisting of multiple cameras. It is the basis of a real-time robotic motion tracking system aiming for high localization accuracy and high time resolution. These goals are achieved using several cameras with very redundant field-of-views. The paper presents the theoretical background of 3D camera calibration and localization, the localization accuracy measurement setup, and its results. The measurements are obtained using a color marker and an industrial robotic arm for measurement ground truth generation. 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 virtual studios, robotic swarm applications, human tracking in surveillance systems, wildlife tracking 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 system belongs to this category. The most common approaches in this area are stereo vision systems using two cameras mounted on a common frame. The work presented in this paper aims to create a motion tracking system consisting of several cameras with a highly overlapping field of view. Our goal is to achieve a precise and fast marker tracking system for hard-realtime robotic applications, so we use the redundancy of the visual system to improve the localization accuracy and the time resolution. In stereo vision systems, especially when the two cameras are near to each other, the depth accuracy is usually worse than the accuracy in other directions. To avoid this effect, we use a set of cameras in arbitrary locations, so that they can observe the same volume from very different directions. We use a marker containing color combinations otherwise rare in the application environment to allow very fast detection in the color images.

In this paper we present the localization accuracy measurement result of the system. Position accuracies are measured agains a ground truth setup created using an industry standard robotic arm with positioning precision far below 1 mm. During the measurements, the cameras are stationary and their location and orientation is calculated during the initial phase of the measurement.

Please note that the presented Smart Mobile Eyes for Localization (SMEyeL) system is open-source. It is written in C++ using the popular OpenCV[1] computer vison library. Its source code, documentation and all the input data for the presented measurements are available for download from our homepage[2]. The remaining part of the paper is organized as follows: section II summarizes related results, section III introduces the theoretical background of the results including camera calibration and 3D geometry. Section IV presents the measurement setup including video capture parameters and ground truth generation. Section V contains the measurement results regarding localization error and relative distance measurement errors, and finally, section VI concludes the paper.

II. Related work

Object tracking consists of the following two steps: detection and spatial tracking. Both steps have a large-scale theoretical background and many proposed algorithms.

The main difference between object detection methods is the object representation: two important approaches are point or patch, and model based methods. Some methods approximate the objects with points, as objects can often be considered pointwise, having only a few pixels size. In [3], birds in the sky are detected using background subtraction. The obtained foreground objects are then assumed to be point-like. Using this method, a lot of bird-positions are generated in every frame. During the tracking of individual birds, the correspondence between the consecutive frames is formulated as a graph theoretical problem.

Other algorithms create the object representation from primitive geometrical shapes. This approach is suitable for rigid objects. If the motion of the object model is modelled using transformations such as translation, affine and projective transformation, the kernel of the object is being tracked along the images.

If only a single object is observed, the most commonly used algorithm is the template matching. This is a brute force method of searching the image for a region similar to the template. This template can contain for instance image intensity pathes or color features. Due to illumination insensitiveness, image gradients can be an even better choice than simple image patches[4]. But these templates can also be built using other object reprezentations. In [5], the color mean of eight neighbouring pixels is used. During tracking, the authors calculate for every possible location the ratio of the current color mean and the color mean of the model. The hypothesized position of the object is where this color mean ratio has the highest value.

All the template matching approaches are basically brute force: all possible locations are checked for a matching with the model. The performance can be improved using location prediction: the object location in the last frame, or the assumption of constant velocity can be used to estimate the location in the next frame. This allows limiting the search to a smaller region of interest inside the images. A related method is the iterative procedure called Mean-shift which is used in many areas of computer vision to track motion. Mean-shift is often used to search for matchings in some special feature spaces, instead of the images themselves. For example in [6], a joint spatial-color histogram is used instead of just a color histogram for the mean-shift algorithm.

The Mean-shift method works well on static probability distributions but not on dynamic ones as in a movie. The dynamic distributions can be managed by the Camshift algorithm which is based on the principles of the Mean-shift algorithm. It readjusts the window size for the next frame based on the zeroth moment of the distribution of the current frame. Camshift stands for "Continuously Adaptive Mean Shift". In [7] the Camshift is used for object tracking.

In [8] the Particle filter is used for object tracking. The filter is guided by Camshift which means that Camshift helps improve the sampling efficiency of particle filter in both position and scale space.

III. Theoretical background

This section summarizes the theoretical background of our marker localization method and the presented measurements. The main parts are the camera calibration, the search for the marker in the image, the calculation of 3D locations and the estimation of errors.

A. Camera calibration

We use the pinhole camera model which is a very effective and simple model for the projective geometry of a camera. The camera parameters can be divided into intrinsic and extrinsic parameters. The intrinsic parameters describe the projection of the camera independently of the position and orientation. They include the optical center of the image, the focal length, and lens distortion compensations. The extrinsic parameters define the position and orientation of the camera. Detailed information about the camera model and the calibration process can be found in [9].

We use the well known method of camera calibration using a printed chessboard-like pattern, as the OpenCV library already contains all the necessary processing. Using the calibration result, a 3D ray can be assigned to any pixel location in the camera images, which represents the direction of the object from the cameras focal point as starting point.

B. Marker detection

We use a color marker in our measurements as we believe that this type of marker can be detected faster. The key idea is that the marker contains two concentric circles of very distinctive colors: the inner color is blue while the outer ring is red. Every frame is scanned by a finite state machine designed for color co-occurrance detection. This finite state machine detects areas where white (background color) is followed immediately by red, and after that, blue. The center of such blue areas are considered to be the center of the marker. During our tests we found that this type of color co-occurrance based marker allows very fast detection and effective false rejection. Of cource, this approach is much more sensitive to illumination changes than a black-and-white marker, but in the current robotics lab environment, illumination can be precisely controlled.

C. Calculation of 3D locations

After identifying the center of the marker in every frame of every camera, a 3D ray is calculated where the starting point $\mathbf{a}$ is the focal point of the current camera, and the direction vector $\mathbf{v}$ points in the direction of the marker:

$$\mathbf{r}(t) = \mathbf{a} + \mathbf{v}t \tag{1}$$

where $t \geq 0$ is the free parameter.

Using two or three cameras (the marker may not be always detected in every camera image), the intersection of the rays is calculated. As the rays may not intersect exactly, we used a least squares estimation as follows.

In two dimensions, given a line with the point $\mathbf{a}$ and normal vector $\mathbf{n}$, the distance between an arbitrary $\mathbf{p}$ point and the line is $||(\mathbf{p}-\mathbf{a})^T\mathbf{n}||$. This leads to a squared error

$$d^2\left(\mathbf{p},(\mathbf{a},\mathbf{n})\right) = (\mathbf{p}-\mathbf{a})^T\mathbf{n}\mathbf{n}^T(\mathbf{p}-\mathbf{a}) \tag{2}$$

This can be generalized into 3D using $\mathbf{N} = \mathbf{I} - \mathbf{v}\mathbf{v}^T$ where $\mathbf{I}$ is the identity matrix and $|\mathbf{v}| = 1$. The generalized error function for an arbitrary number of lines is

$$E(\mathbf{p}) = \sum_i (\mathbf{p}-\mathbf{a}_i)^T\mathbf{N}_i(\mathbf{p}-\mathbf{a}_i) \tag{3}$$

The minimal error is achieved where the condition

$$\frac{\partial E(\mathbf{p})}{\partial \mathbf{p}} = 0 \tag{4}$$

is satisfied. This leads to

$$\left(\sum_i \mathbf{N}_i\right)\mathbf{p} - \sum_i \mathbf{N}_i\mathbf{a}_i = 0 \tag{5}$$

The solution is given by

$$\mathbf{p} = \left(\sum_i \mathbf{N}_i\right)^{-1}\left(\sum_i \mathbf{N}_i\mathbf{a}_i\right) \tag{6}$$

Using this equation, the least squares intersection estimation can be calculated for an arbitrary number of rays starting from the cameras.

D. Calculation of errors

In the measurement setup, there is no ground truth for the location and orientation of the cameras themselves. But there is ground truth for the marker locations, so the error is measured in two ways. The first is the standard deviation of the localization of the marker locations. The second consists of the distances between the means of consecutive marker locations.

As the error of the localization depends on the relative location with respect to the cameras, we have calculated the positioning errors along every axis separately, and then their Euclidean norm as well. While the latter is a general indicator of localization precision, the per axis standard deviations highlight the changes of errors due to relative directions.

IV. Measurement Setup

We used three PlayStation3 Eye cameras to record the movement of the marker. The cameras captured the video with 640×480 resolution at 30 fps. A few, 2–3 minutes long videos have been shot for offline analysis. As a technical detail we note that recording three videos simultaneously into separate files may be very resource consuming due to periodic jumps between the files. We used a virtual RAM disk to avoid frame losses due to the seek time of the harddrive.

The exact location of the cameras was unknown, but we ensured that the marker has been inside the field of view of the cameras during the whole recording. Each recording starts with a few seconds of calibration frames, where an OpenCV chessboard is visible. The chessboard can be used to determine the extrinsic parameters of the cameras. The calibration is only successful if all three cameras have detected the chessboard simultaneously. This is to avoid errors caused by small movements of the chessboard.

For precise and reproductible movement of the marker, we used a Mitsubishi MELFA six degrees of freedom robot. This robot arm is capable of ± 0.05mm repeatability and able to move along straight lines. Although the velocity and acceleration of the marker is unknown, as some small movements may require much faster movements in joint coordinates, our measurement setup does not take timing in account. The frames where the marker is stationary for 3 second long intervals are identified manually after the video capture.

The marker has been placed in all 27 points of a $3 \times 3 \times 3$ cuboid grid as shown in Fig. 1.

V. Measurement Results

Using the measurement setup described in the previous section, we have captured the 3 videos for the 3 cameras, and performed the marker localization on all of them. This provided 3D rays starting from every camera and indicating the direction of the marker. After merging the 3D information into single 3D locations, the results presented in Fig. 2 were obtained. The locations marked with x-es are cases where the marker was only found in two of the camera images. This was caused by one of the cameras seeing the marker in a very flat angle, mainly in the upper-left-front corner. The path of the marker can be clearly recognized.

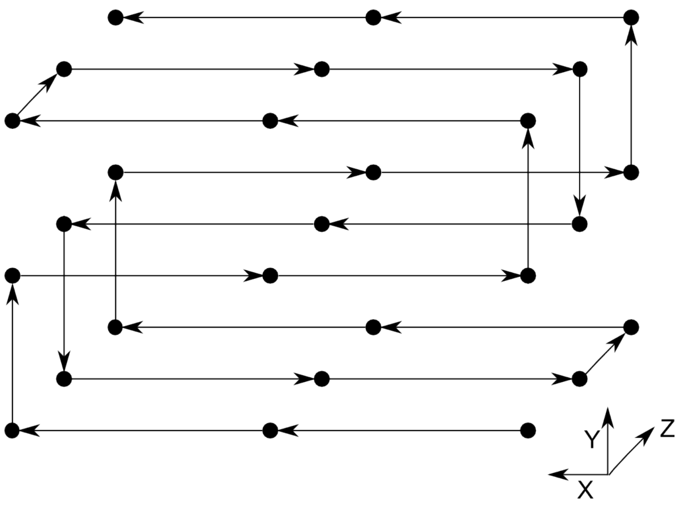

Fig. 1. Robot motion overview and the measurement coordinate system. The marker stops in every location point for 3 seconds.

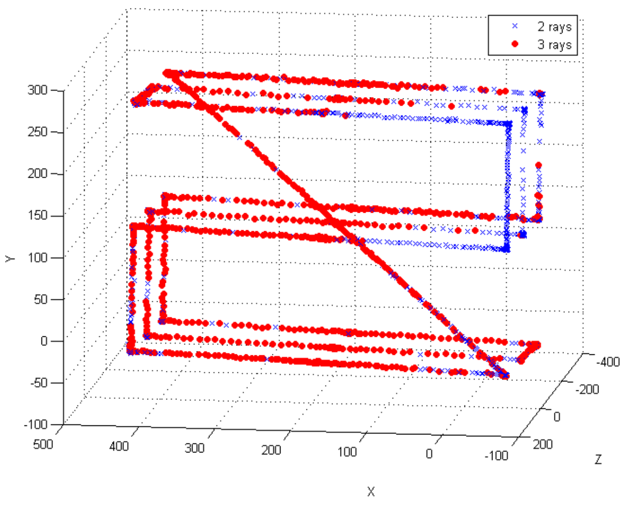

Fig. 2. Result overview with all measurement points. Locations calculated using 3 or only 2 camera images are marked with o and x respectively.

To measure the localization errors, we have marked manually those frame intervals in the videos, where the marker is not moving. This way we had several frames with marker locations known precisely from the program of the robotic arm. Fig. 3 shows only the location points calculated form these stable frames.

Although the exact location of the cameras during the measurement is not known, the localization can be evaluated using the followings. The localization accuracy can be characterized by the standard deviation of the estimated marker locations. Beside this, the distance of the consecutive marker locations has to match the distances defined in the program of the robotic arm.

We have calculated the means and standard deviations of the 3D locations for every marker position separately. The standard deviation is presented for the 3 axes separately in

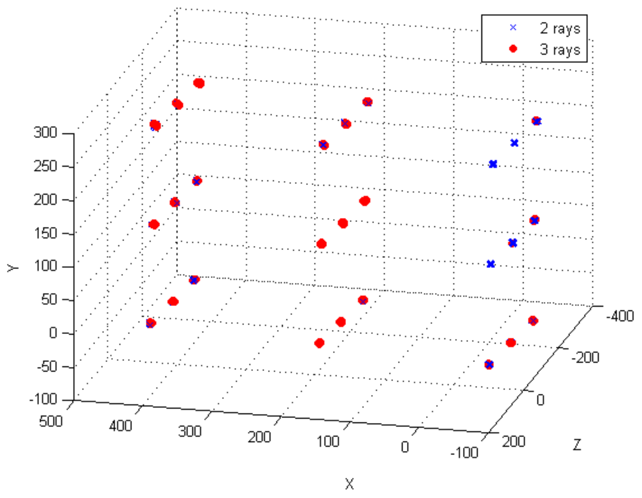

Fig. 3. Localization results in stable locations.

Fig. 4 and its norm in Fig. 5. The standard deviation of the 3D localization is around 1 mm. The two significant errors are caused by false recognition of additional marker locations. This is the same order of magnitude as the precision limited by the camera resolution which is below 2 mm for a field of view of 56° and a resolution of 640 × 480 pixels.

By observing the standard deviations for the 3 axes separately, the highest error is provided in the Z-axis which is the distance from the camera array. (See Fig. 1 for details on the coordinate system orientation.) These differences are caused by the different viewing angles of the marker from the various cameras: the nearer to orthogonal the view angle of the marker is, the more accurate the localization is.

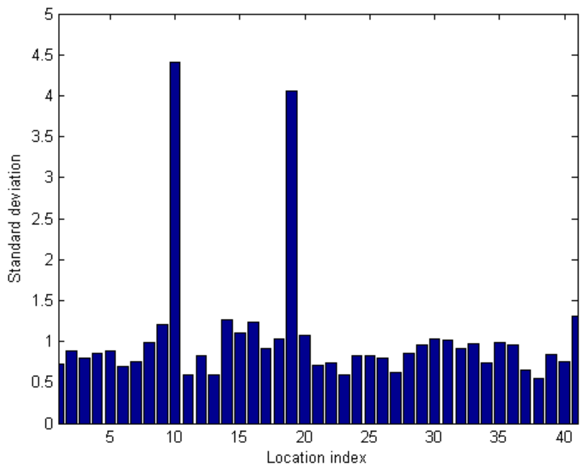

Fig. 5. Standard deviation of marker location estimations.

As we know the 3D points the marker moves to, we know the exact distance between the consecutive marker locations. Fig. 6 presents the measured distances between the estimated mean marker locations after each other. Movements have 3 length: in X direction 250 mm, in Y direction 150 mm and in Z direction 100 mm. Between two full paths, the marker moves along the diagonal of the test volume which is 616 mm long. See Fig. 2 for a comparison.

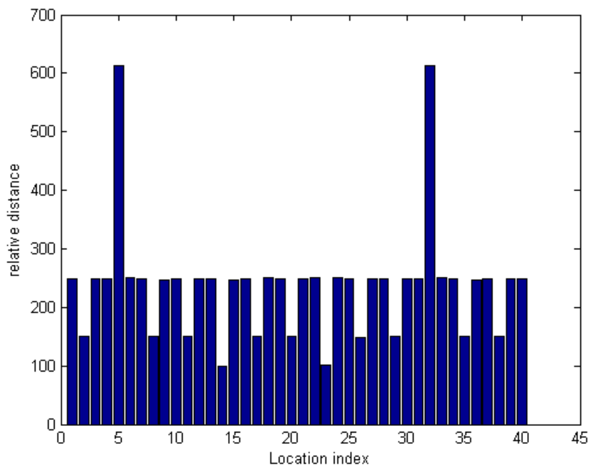

Fig. 6. Distances between marker locations. True distances are 100 mm, 150 mm, 250 mm along the grid, and 616 mm along the diagonal of the test volume. Please note that the path shown in Fig. 2 starts after the 616 mm distances.

The difference between the ground truth distances and the measured ones are presented in Fig. 7. The error is usually below 2mm. The systematic pattern is caused by the sequence if inequal distances. We believe that the negative bias is caused by the error of the camera location calibration process: most distances are smaller than expected which means that the base vectors of the world coordinate system are slightly shorter than in the real world. This is caused by localization error of the chessboard due to flat viewing angles for some cameras.

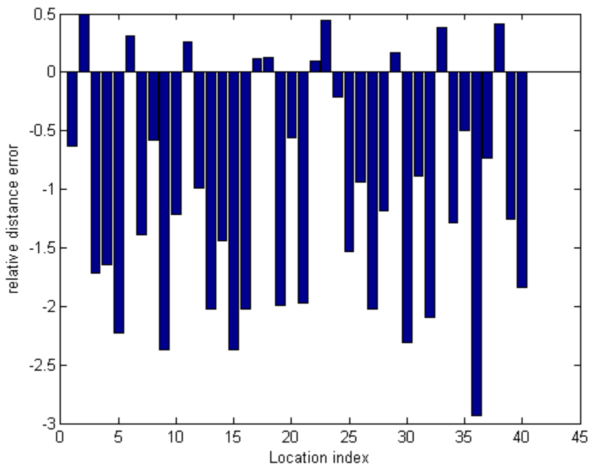

Fig. 7. Difference between estimated distances and ground truth.

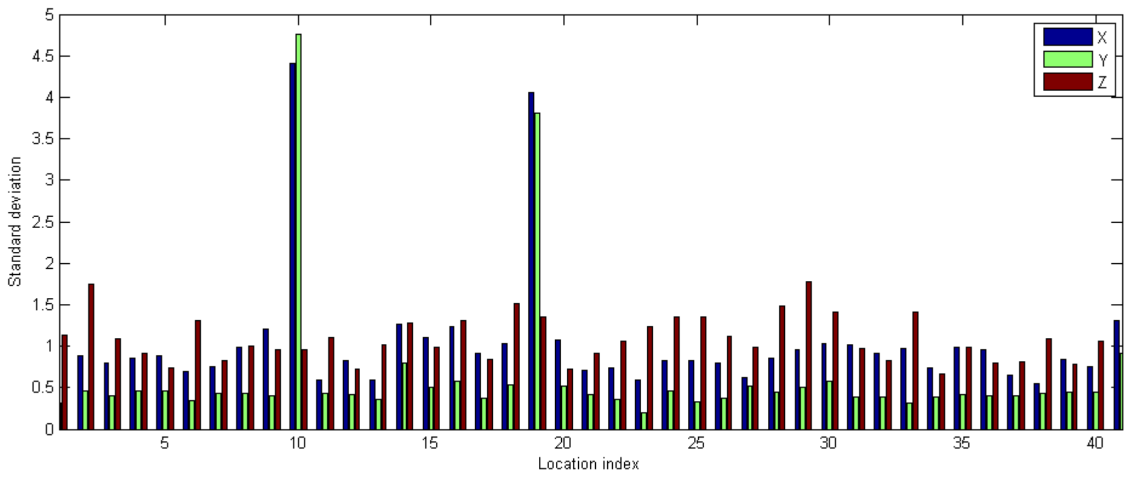

Fig. 4. Standard deviation of marker location estimations, shown for the 3 axes separately. Differences are caused by the varying view angles.

VI. Conclusion and future work

This paper presented the localization accuracy measurement results of a multiple camera system. The theoretical background of camera calibration, and location estimation using several cameras were discussed. The measurement setup utilized a fast-to-find color marker and an industry-standard robotic arm for ground truth generation. The results show a precision around 1 mm from a distance between 1-2 meters, when the cameras have an overlapping field of view and they look at the marker from very different directions.

As subject of further research, the system will be enhanced to use smartphones as cameras. This would allow the utilization of many, relative cheap smart cameras to build a visual sensor network achieving high localization precision and high time resolution.

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.

The authors would like to thank István Oláh, Ferenc Radnóti and László Tábori for their valuable help in the assembly of the measurement environment.

References

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

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

[3] K. Shafique and M. Shah, "A non-iterative greedy algorithm for multi-frame point correspondence," in *IEEE International Conference on Computer Vision (ICCV)*, 2003, pp. 110–115.

[4] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998, pp. 232–237.

[5] P. Fieguth and D. Terzopoulos, "Color-based tracking of heads and other mobile objects at video frame rates," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997, pp. 21–27.

[6] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Patt. Analy. Mach. Intell.*, pp. 603–619, 2002.

[7] J. G. Allen, R. Y. D. Xu, and J. S. Jin, "Object tracking using camshift algorithm and multiple quantized feature spaces," in *Proceedings of the Pan-Sydney area workshop on Visual information processing*, ser. VIP '05. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2004, pp. 3–7. [Online]. Available: http://dl.acm.org/citation.cfm?id=1082121.1082122

[8] Z. Wang, X. Yang, Y. Xu, and S. Yu, "Camshift guided particle filter for visual tracking," in *Signal Processing Systems, 2007 IEEE Workshop on*, 2007, pp. 301–306.

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