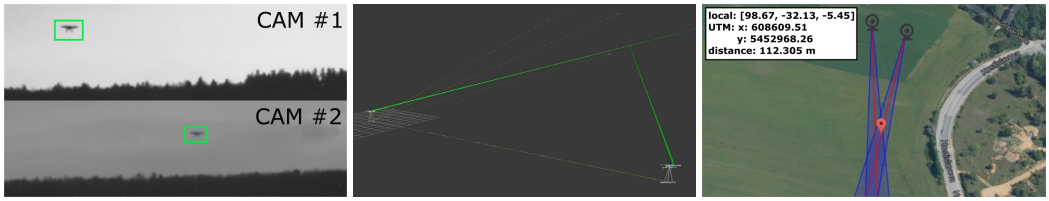
BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2016
http://excel.fit.vutbr.cz

Optical Localization of Very Distant Targets in Multi Camera System

Jan Bednařík*



Abstract
This paper presents a system for automatic optical localization of distant moving targets using multiple pan-tilt cameras. The cameras were precisely calibrated and stationed using custom designed calibration targets and methodology. The detection of the target is performed manually, while the automatic visual tracker combines the background/foreground modeling and motion model in the particle filer framework. The estimation of the 3D location is based on the N-view triangulation. A basic setup consisting of two camera units was tested against static targets and a moving terrestrial target, and the location estimation precision was compared to the theoretical model. The modularity and portability of the system allows fast deployment in a wide range of scenarios including perimeter monitoring or early threat detection in defense systems, as well as air traffic control in public space.

Keywords: multi-camera localization — visual object tracking — 2D motion prediction — particle filter based tracking — stationing and rectification — articulated model of PT unit — 3D localization using triangulation — physical simulation using Gazebo — robotic system design using ROS

Supplementary Material: Demonstration Video

*jan.bednarik@hotmail.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

An autonomous localization of arbitrary moving targets is an essential system component in multiple domains, such as air traffic control, robotic workspaces or surveillance and defense systems. If the sensory data measured by the target are available, it is straightforward to derive its location (by means of the GPS, radio multilateration, etc.). There are scenarios, however, were the target is unable (malfunctioning aircraft) or reluctant (UAV intruder) to expose its location. Then the localization estimation system is left with its own observations.

Radars, the most widely used devices for localizing distant targets, suffer from being unportable, energy-intensive and expensive. Furthermore, it might be desirable that the tracked object not find out that it is being tracked, which is the condition an actively radiating system cannot achieve.

This paper introduces a semi-autonomous passive multi-camera system for tracking and localizing the distant objects, which is based merely on ordinary RGB cameras — Optical Localization System (OLS). The system is designed to suit mobility and temporary deployment because each camera station weighs no more than twenty kilograms and the whole system is inexpensive by comparison to radars as well.

2. Related Work

The choice of how the targets are represented determines the domain of approaches used for visual detection and/or tracking. In general, two main representations are used [1]: a *shape* model which encompasses e.g. points [2], contours [3, 4] or articulated models [5, 6], and an *appearance* model which is represented by a template [7] or active appearance model [8].

Moving object detection Depending on the object model, the detection might be performed either by detecting keypoints and matching them against the pretrained model [9, 10, 11], or by dividing the image into individual patches in which the object is searched for. For each patch, the template matching is performed [12, 7] or feature set is extracted; consequently, the model presence probability is evaluated using the generative or the discriminative classifier [13, 14]. Since the exhaustive search within the whole image is computationally expensive, the cascade classifiers are applied [15, 16]. Alternatively, the moving object can be detected in the image regions yielding the highest response of frame differencing [17, 18].

Object tracking There are multiple approaches to visual tracking. Keypoint tracking represents one of the most common ones [2, 19]. Kernel approaches are based on a weighted kernel used to derive smooth distance function which can be optimized in the means of target position using traditional gradient based methods such as gradient descent [20], or even multiple collaborative kernels might be used [21, 22]. Other approaches rely on tracking-by-detection concept which heavily utilizes the detection principles in combination with motion-aware approaches to localize the object [23, 24]. To reinforce the tracker robustness, the motion models are often used, Kalman filter and particle filter being the most popular ones [25, 7].

Multi-view optical localization Multi-camera localization is mostly used in the domain of robotics, where the *intelligent space* consisting of several cameras with a priory known and fixed intrinsics and extrinsics is utilized [26]. The centralized system uses either mere visual information or enhances the localization with the help of robots' sensory data [27, 28, 29, 30]. Bound to the predefined space and using fixed cameras, those systems do not need to deal with the imprecise estimates of a current camera pose.

3. System Overview

The main building block of the OLS is a camera station (CS), a standalone unit consisting of hardware modules

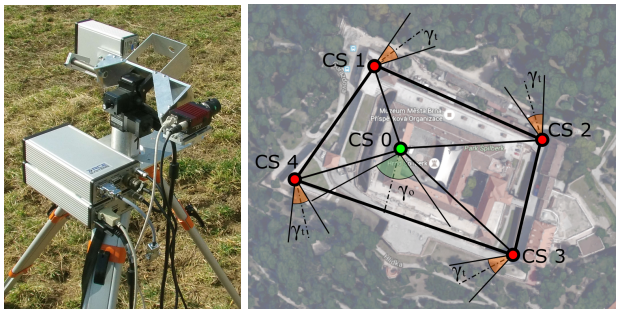


Figure 1. Tracking camera station (left) and a use case scenario (right) showing the positioning of four tracking stations (red dots) and one observation station (green dot) to protect a real world area.

necessary for capturing the images, manipulating the pose of the camera and estimating its own geographical coordinates. There are two type of CSs. The *overview station* is designed to be controlled manually by the human operator and is equipped with the zooming lens that allows achieving both a wider scanning range and a more detailed view of the farther objects. The *tracking station* consists of the fixed lens and takes part in the autonomous tracking of the moving objects.

The OLS is designed to work with an arbitrary number of CSs. Due to the geometric limitations of the multi-view systems, which affect the localization precision, the tracking stations should be positioned so as to form approximately regular polygon with long enough bases (see Section 4, see Figure 1).

The camera station itself consists of a surveying tripod, a P&T unit[1] Flir PTU-D46-70, a camera Prosilica GT 1290C (RGB, 1280×960 px, 33.3 FPS), an inclinometer and a GPS sensor. A camera unit is modeled as a kinematic chain consisting of six joints and five links corresponding to the distance between separate parts of the tripod and the manipulator (see Figure 2). The transformation between the `GROUND` and `ORIGIN` reflects the positioning and heading of the given manipulator within the local coordinate frame.

The kinematic chain is designed as composition of transformation matrices where a single joint can be located by applying the Euclidean transformation on the position of the joint which it is depending on:

$$M_{next} = M_{previous} T_{next} R_{Z_{next}} R_{X_{next}} R_{Y_{next}}, \qquad (1)$$

where M is the transformation matrix of the given joint, T is the translation between successive joints and R_a is the rotation around axis a.

4. Localization Precision

The precision of estimating the target position is subject to systematic error (miscalibration of the instru-

[1] Pan and Tilt.

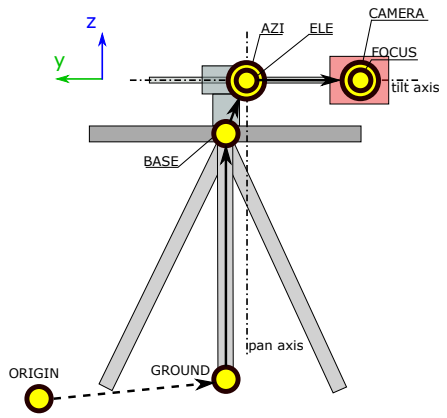


Figure 2. The model of a camera unit represented by a kinematic chain consisting of six joints (yellow dots) and 5 links (black arrows). The joints `AZI` and `ELE` share exactly the same position, the joint `CAMERA` is further along the X axis than the joint `FOCUS`.

ments) and random error (wrong measurements and disturbances in the environment) [31]. Atmospheric turbulence, refractive index fluctuations and uncertainty of the visual tracker are the main causes of the random error which is analysed in section 4.1. The systematic error was alleviated and/or measured using the custom designed stationing and rectification process.

4.1 Random Error Analysis

Stereoscopic systems are affected by a phenomenon of diminishing accuracy of depth measurement with increasing distance of the target from the cameras [32]. The depth measurement resolution for canonical stereo setup is $R = \frac{rZ^2}{fb - rZ}$, where f is the focal length, b is the base length, r is the horizontal size of one pixel and Z is the target distance. By substituting r by pr, where p is the random error represented by integer number of pixels we obtain the position estimation error function $E = \frac{prZ^2}{fb - prZ}$.

The OLS does not conform to the canonical stereo setup (all cameras can rotate freely), so the dependence of the error on the target distance is no longer quadratic (considering the setup of two cameras where only one of them makes error): $E = B\tan(\arctan(\frac{D}{B}) + \arctan(\frac{r}{f})) - D$. The cameras setup as well as the error shown as the function of the base length and target distance is depicted in Figure 3.

A more realistic scenario, where each camera makes a random error p, is depicted in Figure 4. A significant advantage of using multiple cameras is demonstrated — geometrical limitations of the two-camera setup make it impossible to precisely evaluate the position of the target placed close to the line collinear with the baseline. In the multi-camera setup, on the other hand, the subset of two cameras forming the baseline B_i is used for each position of the target, following the rule:

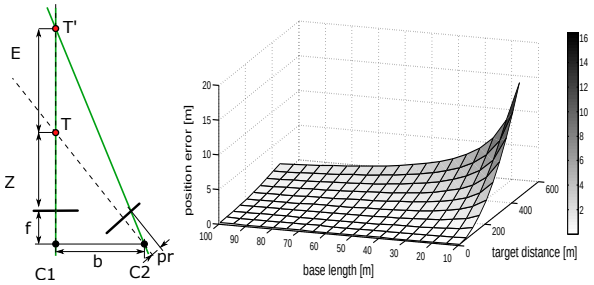


Figure 3. Left figure depicts the setup of two cameras $C1$ and $C2$, where only $C2$ makes an error worth p pixels. T represents the ground truth position of the target whereas T' is the wrongly estimated position. Right figure shows the position estimation error as the function of base size and target distance (given the random error $p = 10\ px$ and following constants: $r = 3.75e^{-6}\ m$, $f = 50e^{-3}\ m$).

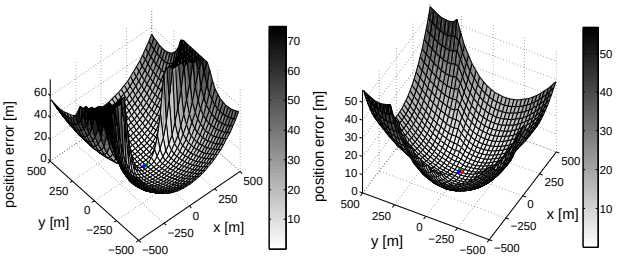


Figure 4. The position estimation error as a function of the horizontal position of the target. Two-camera (left) and three-camera (right) setup with $b = 20\ m$ are confronted, where utilization of more cameras always yields lower errors. The first two cameras are placed on the X axis with the coordinate frame center in the middle of their baseline. The third camera is placed on the Y axis, so that all the cameras form a regular triangle.

$i = \underset{i}{\text{argmax}}\ \vec{t_i}\vec{n_i}$, where $\vec{t_i}$ is the direction of the line segment linking the center of the baseline and the target and n_i is the normal vector of the baseline B_i.

4.2 Stationing and Rectification

The stationing procedure alleviates two types of systematic error: wrong heading estimation and undesirable tilt of the camera station. Since the accuracy of the commercial magnetometers is too low (hundreds to thousands of milliradians), the precise heading must be estimated visually by observing the distinctive landmarks. To achieve the horizontality of the station a digital inclinometer can be used.

The imprecision of the camera-manipulator attachment causes slight undesirable rotation of the camera coordinate frame. Three horizontally leveled rectification targets are used to alleviate and/or measure all rotation angles (around X, Y and Z axis): 5):

Rotation around optical axis The target contains parallel horizontal lines and the camera displays the blend of the original and vertically mirrored streams. The aim is to rotate the camera manually so that the

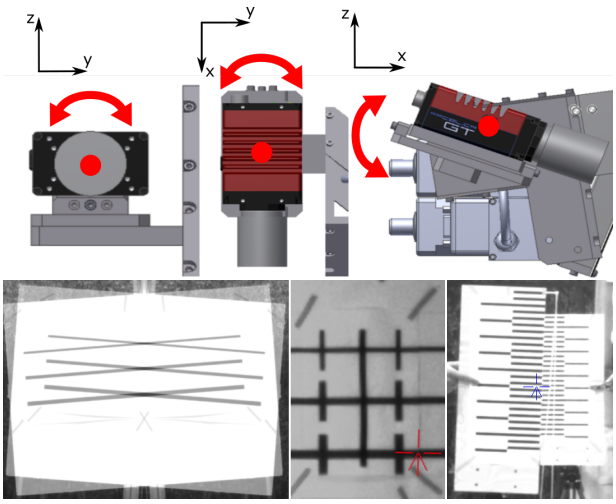


Figure 5. Three rectification targets (bottom) used to alleviate and/or measure the undesirable rotation angles of the cameras (top).

lines would appear aligned.

Rotation around azimuthal axis The target contains parallel horizontal lines and a pair of crosses whose distance equals the distance between `ELE` and `CAMERA` joint (see Figure 2). The aim is to measure the distance between the right cross and the intersection of the optical axis with the target, which translates to an error angle in the azimuth.

Default elevation angle Two targets which contain black and white lines representing a ruler are positioned in a row. The aim is to adjust the tilt of the camera so that the optical axis would intersect the same mark on both targets and the resulting elevation angle could be measured.

5. Object Tracking

The detection itself has been performed manually so far in the man-in-the-loop manner, while the autonomous tracking uses the implementation of the visual tracker combining the background subtraction, motion model and object model in the particle filter framework [7]. This approach can even cope with the moving cameras and thus is suitable for the OLS. The operation of the tracker is described below.

The target is represented as a rectangular template (consisting of gray-scale intensity values), which is normalized to the size 24×24 pixels. The advantage of the template representation is that it contains both spatial and appearance information. The template is created only once during the initialization, and thus the tracker could fail if the target changed its appearance significantly during the course of tracking. However, for very distant targets, no or merely small change is expected.

The Bootstrap particle filter (BPF) — the variant of a particle filter following the sequential importance sampling approach [33] — is used to generate and evaluate candidate positions of the target. Each particle (i.e. the state of the system) is represented as $\vec{x_n} = (x, y, v_x, v_y, h, w)$, where (x, y) represents the 2D position of the target, (v_x, v_y) represents the estimated speed of the target and (h, w) represents the bounding box size.

The perturbations in the observed position of the target caused by the moving camera are alleviated using the motion model which is applied in the *prediction* step of the BPF:

$$pos_{n+1} = pos_n + vel_n + \gamma_{pos} \sim \mathcal{N}(\mu, \sigma), \quad (2)$$

$$vel_{n+1} = vel_n + \gamma_{vel} \sim \mathcal{N}(\mu, \sigma), \quad (3)$$

$$bb_{n+1} = bb_n + \gamma_{bb} \sim \mathcal{N}(\mu, \sigma), \quad (4)$$

where scalar pos_n is the x or y position, scalar vel_n is the x or y velocity, scalar bb_n is the w or h size of the bounding box in time n, and γ is the noise drawn from the Gaussian distribution $\mathcal{N}(\mu, \sigma)$, where scalars μ and σ parameters are set empirically for each parameter.

In the *update* step, each particle is assigned a new weight w using the objective function reflecting the similarity of the template and the candidate patch:

$$w = \sum_{(x,y)\in I} e^{min(M_t^{(x,y)}, M_c^{(x,y)})} (1 - |I_t^{(x,y)} - I_c^{(x,y)}|)^2, \quad (5)$$

where M_t and M_c are the foreground masks (FM) of the template and the current candidate respectively, I is the image, t, c subscripts denote template and candidate patch respectively, and (x, y) superscript denotes indexing 2D array (an image). The FMs are estimated by subtraction of the two images where the bounding boxes denoting the position of the target do not overlap (the FM M_t is estimated only once). The resulting estimate of the target position is chosen using the Maximum a posteriori approach.

In order to enable the motion of the camera, the transformation between each pair of adjacent frames is estimated by detecting and tracking the keypoints using KLT tracker [2] and then estimating the homography using the RANSAC algorithm [34].

The homography might not be found, which often occurs if the airborne target with the uniform background of the sky is tracked or if the manipulator moves the camera too harshly. To deal with such cases in OLS, the tracker was adjusted so that in a *prediction* step of the BPF a small subset of particles would be forced to take the image positions yielding the highest response of adjacent frame differencing which is expected to contain the moving target of interest.

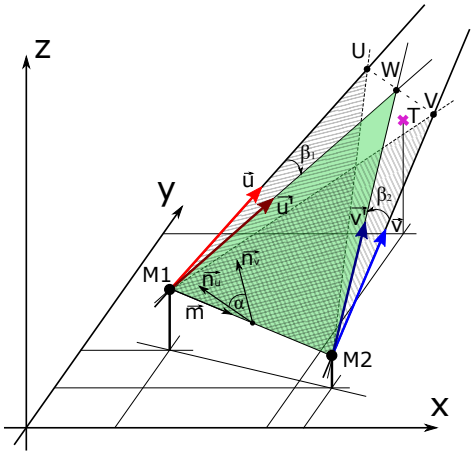


Figure 6. A schematic view of a problem of 3D position estimation using triangulation in two-cameras scenario. The camera units $M1$ and $M2$ observe the target T in the directions $\vec{u}$ and $\vec{v}$. The plane M_1M_2W is used as a common plane where the projected vectors $\vec{u'}$ and $\vec{v'}$ intersect.

6. Target Localization Using Triangulation

The hardware cameras are modeled as finite pinhole cameras based on the projection matrix P [34]:

$$P = KR[I|-C],$$
$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix},$$

where K is the intrinsics matrix and R and C are the rotation and translation matrices representing the orientation and position of the camera frame. The 3D point $\vec{X}$ projects to the 2D image point $\vec{x}$ via $\vec{x} = P\vec{X}$. If only the projection $\vec{x}$ is observed, the 3D line mapping to $\vec{x}$ can be computed using *back-projection*:

$$X(\lambda) = P^{+}\vec{x} + \lambda C,$$

where P^{+} is the pseudo-inverse of P ($P^{+} = P^{T}(PP^{T})^{-1}$).

In OLS, the intrinsics were estimated for each camera during calibration and extrinsics are known at each time due to the sensory data streamed from the manipulators. However, the rays backprojected from each camera might not intersect in the 3D space due to both systematic and random errors (see Figure 6).

The estimation of the 3D position of the target consists of the following steps. First, back-projection is used to find the vectors $\vec{u}$ and $\vec{v}$ which form the planes M_1M_2U and M_1M_2V with the angle α between them. Both vectors are then rotated around the axis $\vec{m}$ so that they lie in the same plane M_1M_2W: $\vec{u'} = R(\beta_1)\vec{u}$, $\vec{v'} = R(\beta_2)\vec{v}$. The rotation angles might be of the same value $\beta_1 = \beta_2 = \alpha/2$; however, to achieve higher precision the angles might be weighted by the trackers' beliefs b: $\beta_1 = \alpha\frac{b_2}{b_1+b_2}$, $\beta_2 = \alpha - \beta_1$. Finally, the intersection W of the vectors $\vec{u'}$ and $\vec{v'}$ is found.

If multiple camera units are used, 3D location can be estimated as the weighted centroid of the estimates computed by each pair of the camera units forming the base b_i (see Algorithm 1). The weights correspond to the angle between the baseline and the line intersecting the (estimated) position of the target and the baseline center, since this angle significantly affects the precision (see Section 4).

Since the 3D position estimation might be completely wrong occasionally, the position estimates are smoothed by the moving average computed over h consecutive estimates (h was empirically set to 10).

Algorithm 1: Estimation of the 3D position from n-views

Input: Set of bases $B = b_1, b_2, ..., b_N$.
Output: 3D position estimate T.

/* 3D location estimate disregarding weights */
foreach $b_i \in B$ **do**
  $\vec{t_i} = Estimate3DPosFrom2Views(b_i)$
end
$\vec{T'} = \frac{1}{N}\sum_{i=1}^{N}\vec{p}_i$
/* Weighted estimation of the 3D location. $\vec{bc_i}$ represents center of the baseline b_i */
foreach $b_i \in B$ **do**
  $w_i = \frac{e^{\vec{n_i}(\vec{T'}-\vec{bc_i})}}{\sum_{j=1}^{N} e^{\vec{n_j}(\vec{T'}-\vec{bc_j})}}$
end
$\vec{T} = \sum_{i=1}^{N} w_i\vec{t_i}$

7. Implementation and Experimental Results

The implementation is built on a robotic framework ROS[2] and a physical simulator Gazebo[3]. ROS was chosen for its wide support of hardware components and a seamless way to implement multi-process distributed system. The whole system was modeled and simulated in Gazebo (see Figure 7), which facilitated hardware-in-a-loop testing of the manipulators [35].

The system was tested in the real-world environment in the basic two-camera setup. The CSs were precisely positioned using the differential GPS sensor (achieving accuracy of ca 0.01 m) so that the base would be exactly 30 m long. The local heading was estimated by aiming the units on each other. The system

[2]Robot Operating System: http://www.ros.org
[3]Gazebo: http://gazebosim.org

Figure 7. A sample scene captured within the Gazebo simulator. The scenario consists of four CSs and one moving object (red ball). The simulated image streams are displayed on the right.

Figure 8. The two-camera setup, where a distant target is tracked by both CSs (left and right). The estimated position of the target is displayed in the map (center) in real time.

was tested against both static and dynamic targets, and in both cases only horizontal position was considered.

As for the static targets, nine landmarks with a priori known UTM coordinates (obtained from the cadastral map) and one target carrying an ordinary mobile GPS sensor were chosen (see Figure 8). The localization error, given as the Euclidean distance between the ground truth and the estimated locations, was compared with the estimated error (see Table 1). Note that both measured and estimated error follow the same trend (see Figure 9); however, the measured error is higher mainly due to the insufficient precision of calibration, stationing and rectification.

Table 1. The table shows the position as well as the localization error for each static target. The estimated error **est.** Δ is affected by the distance of the target and the angle α between the target and the base, and it was computed for the scenario where each CU makes random error $p = 4$ *px* (see Section 4). See also Figure 9 for graphical comparison of the estimated and the measured error.

object	dist. [m]	α [rad]	est. Δ [m]	Δ [m]
pillar1	91,92	0,38	0,20	4,41
pillar2	199,14	0,46	0,95	5,51
pillar3	285,01	0,48	1,93	11,73
pillar4	386,81	0,41	3,39	17,16
tree1	433,88	0,34	4,13	17,20
person	479,96	0,10	4,65	23,90
hide	526,86	0,77	7,57	22,77
tree2	634,46	0,35	8,56	28,33
mast	1379,67	0,33	41,24	34,21

The system was tested against one dynamic terrestrial target equipped with a mobile GPS sensor (a

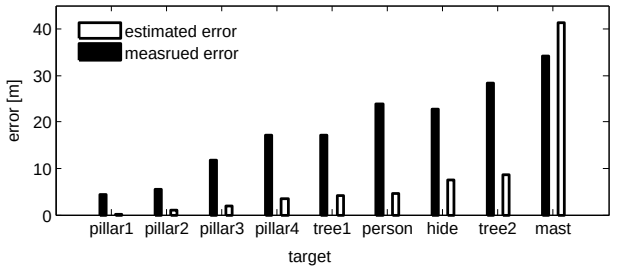


Figure 9. The plot displays both measured and estimated localization error for all static targets, which are sorted in ascending order with respect to the estimated error. The measured error is higher due to imprecise calibration, rectification and stationing.

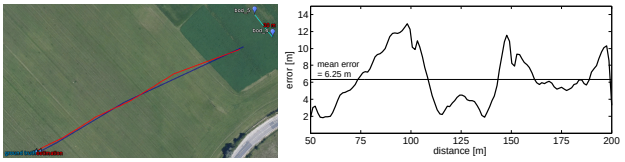


Figure 10. The comparison of the ground truth and estimated trajectory of a target moving in the distance range of ca 50–200 *m* (left). The error as the function of the distance of the target is also displayed (right). The system makes the average error of 6.25 *m*.

walking person). The target was tracked for 120 *s* and the estimated positions were captured and compared to the ground truth path (see Figure 10). On average the system achieved the precision of 6.25 *m*. Note that the position estimates oscillate around the ground truth trajectory, which is caused by the random error made by both trackers; the error, however, keeps in the specific range and reaches maximum of 13.35 *m*. The mean error is higher as compared to the estimated error (see Section 4), which is again caused by the systematic error (imprecise calibration, rectification and stationing).

8. Conclusion

This paper introduced a novel system capable of autonomous tracking and localization of distant moving targets using multiple cameras. The paper proposes precision analysis which aims on finding and alleviating the most prominent sources of error, as well as the methodology to calibrate and station all camera units.

The system utilizes a visual tracker based on the Bootstrap particle filter framework combining both visual and motion model of the target and positionable camera. The localization of the target uses the principle of triangulation, where both the belief of the tracker and the geometrical limitations given by the angle between the base and the target are incorporated into the final weighted estimate.

The system was tested in real world conditions against static and dynamic targets whose position was

known either from the cadastral map or captured by the GPS sensor. The localization precision follows the trend of diminishing accuracy of depth measurement and reaches slightly higher error then the theoretical model, namely due to the insufficiently precise calibration, rectification and stationing. This, however, can be improved by using more reliable hardware components and by performing the rectification procedure more thoroughly.

Though still in early development, the OLS system has great potential for being widely used as a passive, modular and highly portable substitute for the recently widely used radars for the applications ranging from automatic traffic control to national defense systems protecting the sensitive perimeters.

In the near future, the OLS system will be extended by the 3D environment reconstruction subsystem which should make the tracker predict occlusion and estimate more accurately the motion of the tracked target. Furthermore, more thorough tests will be carried out in order to spot the sources of error and reinforce the overall precision.

Acknowledgements

This work is supported by RCE systems s.r.o.[4] which provided the development space, necessary hardware sources, and consultations. I thank my Master's thesis supervisor, prof. Ing. Adam Herout, Ph.D., for guidance, and my adviser, doc. Ing. Vladimír Čech, CSc., for providing his insight into the project.

References

[1] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), December 2006.

[2] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical report, International Journal of Computer Vision, 1991.

[3] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4):321–331, 1988.

[4] W. Hu, X. Zhou, W. Li, W. Luo, X. Zhang, and S. Maybank. Active contour-based visual tracking by integrating colors, shapes, and motions. *IEEE Transactions on Image Processing*, 22(5):1778–1792, May 2013.

[4]Company RCE systems s.r.o. website - http://www.rcesystems.cz/

[5] Quentin Delamarre and Olivier Faugeras. 3d articulated models and multiview tracking with physical forces. *Computer Vision and Image Understanding*, 81(3):328 – 357, 2001.

[6] Cyrille Migniot and Fakhreddine Ababsa. 3d human tracking in a top view using depth information recorded by the xtion pro-live camera. In *ISVC (2)*, volume 8034 of *Lecture Notes in Computer Science*, pages 603–612. Springer, 2013.

[7] David Herman, Filip Orság, and Martin Drahanský. Object tracking in monochromatic video sequences using particle filter. In *7th Scientific International Conference - Enviromental Protection of Population*, pages 120–128. Karel Englis College Inc., 2012.

[8] Iain Matthews and Simon Baker. Active appearance models revisited. *Int. J. Comput. Vision*, 60(2):135–164, November 2004.

[9] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[10] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[11] Reza Oji. An automatic algorithm for object recognition and detection based on ASIFT keypoints. *CoRR*, abs/1211.5829, 2012.

[12] Sanjay Kumar Sahani, G. Adhikari, and B. Das. A fast template matching algorithm for aerial object tracking. In *Image Information Processing (ICIIP), 2011 International Conference on*, pages 1–6, Nov 2011.

[13] K. Wang and Z. Ren. Enhanced gaussian mixture models for object recognition using salient image features. In *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*, pages 1229–1233, Aug 2007.

[14] Li Zhang and Ramakant Nevatia. Efficient scan-window based object detection using gpgpu. *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 0:1–7, 2008.

[15] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511–I–518 vol.1, 2001.

[16] L. Bourdev and J. Brandt. Robust object detection via soft cascade. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 236–243 vol. 2, June 2005.

[17] SeungJong Noh and Moongu Jeon. *Computer Vision – ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part III*, chapter A New Framework for Background Subtraction Using Multiple Cues, pages 493–506. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[18] J. Lee, S. Lim, J. G. Kim, B. Kim, and D. Lee. Moving object detection using background subtraction and motion depth detection in depth image sequences. In *Consumer Electronics (ISCE 2014), The 18th IEEE International Symposium on*, pages 1–2, June 2014.

[19] Georg Nebehay and Roman Pflugfelder. Consensus-based matching and tracking of keypoints for object tracking. In *Winter Conference on Applications of Computer Vision*, pages 862–869. IEEE, March 2014.

[20] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–575, May 2003.

[21] Zhimin Fan, Ying Wu, and Ming Yang. Multiple collaborative kernel tracking. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 502–509 vol. 2, June 2005.

[22] M. Tang and J. Feng. Multi-kernel correlation filter for visual tracking. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3038–3046, Dec 2015.

[23] Michael D. Breitenstein and Fabian Reichlin. Robust tracking-by-detection using a detector confidence particle filter. In *IEEE International Conference on Computer Vision*, October 2009.

[24] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(7):1409–1422, July 2012.

[25] E.V. Cuevas, D. Zaldivar, and R. Rojas. *Kalman Filter for Vision Tracking*. Freie Universität Berlin, Fachbereich Mathematik und Informatik / B: Fachbereich Mathematik und Informatik. Freie Univ., Fachbereich Mathematik und Informatik, 2005.

[26] Joo-Ho Lee, Noriaki Ando, and T. Yakushi. Adaptive guidance for mobile robots in intelligent infrastructure. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 90–95 vol.1, 2001.

[27] Cristina Losada and Manuel Mazo. Multi-camera sensor system for 3d segmentation and localization of multiple mobile robots. *Sensors*, 10(4):3261, 2010.

[28] D. Pizarro, M. Mazo, E. Santiso, M. Marron, and I. Fernandez. Localization and geometric reconstruction of mobile robots using a camera ring. *Instrumentation and Measurement, IEEE Transactions on*, 58(8):2396–2409, Aug 2009.

[29] Mariana Rampinelli and Vitor Buback Covre. An intelligent space for mobile robot localization using a multi-camera system. *Sensors*, 14(8):15039, 2014.

[30] K. Morioka and H. Hashimoto. Appearance based object identification for distributed vision sensors in intelligent space. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 199–204 vol.1, Sept 2004.

[31] J.R. Taylor. *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*. A series of books in physics. University Science Books, 1997.

[32] B. Cyganek. *An Introduction to 3D Computer Vision Techniques and Algorithms*. John Wiley & Sons, 2007.

[33] Michael Isard and Andrew Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.

[34] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[35] M. Bacic. On hardware-in-the-loop simulation. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 3194–3198, Dec 2005.