Vision-based Self-contained Target Following Robot using Bayesian Data Fusion

Andrés Echeverri Guevara, Anthony Hoak, Juan Tapiero Bernal, Henry Medeiros

Department of Electrical and Computer Engineering, Marquette University
1551 W. Wisconsin Ave, Milwaukee WI, 53233
{andres.echeverri, anthony.hoak, juan.tapierobernal, henry.medeiros}@marquette.edu

Abstract. Several visual following robots have been proposed in recent years. However, many require the use of several, expensive sensors and often the majority of the image processing and other calculations are performed off-board. This paper proposes a simple and cost effective, yet robust visual following robot capable of tracking a general object with limited restrictions on target characteristics. To detect the objects, tracking-learning-detection (TLD) is used within a Bayesian framework to filter and fuse the measurements. A time-of-flight (ToF) depth camera is used to refine the distance estimates at short ranges. The algorithms are executed in real-time (approximately 30fps) in a Jetson TK1 embedded computer. Experiments were conducted with different target objects to validate the system in scenarios including occlusions and various illumination conditions as well as to show how the data fusion between TLD and the ToF camera improves the distance estimation.

1 Introduction

Target tracking using mobile robotic platforms is a well-researched problem within the computer vision and robotics communities [1,2,3,4,5,6,7]. Object following capabilities are increasingly popular for aerial platforms and wheeled vehicles alike [8,9,10,11,12]. Object/target following is typically an extension of target tracking in the sense that the tracking error is used as input to a controller that changes the position of the robotic platform with respect to the target.

In this paper, an autonomous, low-cost, computationally light target following platform is presented. The platform consists of consumer grade portable hardware and uses visual information only to estimate the relative position of the robot and generate the corresponding control signals. Specifically, the platform is comprised of an iRobot Create 2 mobile robot, a Creative Senz3D camera, and an Nvidia Jetson TK1 embedded computer. The proposed system is flexible since it imposes no constraints on the shape or color of the target. This is accomplished using the tracking-learning-detection (TLD) [13] algorithm for object detection. The output of the object detection algorithm, in addition to the depth information from the 3D camera, is used as a measurement for a standard Kalman filter which effectively tracks the target through the image sequence.

2 Related Work

There has been significant interest in robotic platforms for object or pedestrian tracking and following. The design of such platforms usually involve three main elements: 1) a tracker that is flexible enough to detect and follow different types of targets, 2) an on-board computing system that is able to perform intensive computer vision operations in real-time, and 3) a robust depth estimation mechanism. We discuss each of these elements in more detail below.

Vision-based tracking algorithms for robotic platforms must be flexible such that only a limited amount of information about the target must be known a priori. They must also be robust enough so that the platform can keep track of the target under a variety of conditions. Many existing platforms rely on defining some kind of 'unique identifier' for the system to detect and track. This could be as simple as a specific color or shape [7,8,9,10,14,15,16,17] or as intricate as using known markers such as LEDs attached to the target [18]. Although a certain level of robustness can be obtained by such approaches as long as the assumptions on the appearance of the target and the background are not violated, they lack the flexibility needed to make such systems practically useful. Flexibility can be obtained by relying on discriminative trackers that can be initialized with a target appearance at time t_0 and then updated on-the-fly [13,19,20]. These trackers can be endowed with additional robustness by integrating them with recursive Bayesian estimation methods that can effectively limit the number of opportunities for the algorithms to make mistakes [21,22,23,24].

Regarding the availability of on-board processing capabilities, image processing is notoriously computationally expensive, especially for high resolution images. In many robotic tracking systems, image processing is performed remotely, off-board [4,8,10,16,12] due to the lack of on-board processing power. There are only a few systems that present truly autonomous vehicles that perform all the processing on-board. For example in [5] a low-cost FPGA is used to increase the efficiency and speed of the image processing algorithms. FPGA-based systems are, however, intrinsically less flexible than general computing architectures and cannot, in general, benefit from the widespread dissemination of algorithms designed for general graphics processing units (GPUs). The advent of low-power embedded architectures with integrated GPUs, such as the Nvidia Tegra TK1 SOC[1], made it possible for these highly parallel algorithms to make their way into low-cost robotic applications.

The third major issue in the design of portable target following robotic systems is the availability of appropriate depth estimation mechanisms. Depth information is vital for the platform to maneuver in three dimensions and successfully follow a target. Although estimating the distance between the target and the robotic platform based on scale variations of the target is a viable option, such approach tends to be fragile in the presence of relatively small errors in the estimated target boundaries. Alternative sensing technologies can be employed in conjunction with traditional vision-based depth estimation to mitigate this

[1] http://www.nvidia.com/object/tegra-k1-processor.html

problem. Over the past few years, RGB-D sensors have been widely used for that purpose [8,25,26,27]. The ideal depth sensor needs to provide sufficient RGB image resolution and depth information at a feasible cost. Although structured light sensors tend to perform well in indoors applications, their performance under natural illumination suffers. Sensors based on time-of-flight (ToF) technology, although still not immune to illumination problems, tend to perform better.

Few autonomous target following systems [2,28] are flexible enough not to require 'unique identifiers', efficient enough to perform all processing and control operations on-board, and incorporate depth information for robust target following performance while using relatively low-cost consumer grade hardware. Similar systems to the one presented in this paper have been proposed, however, some make use of large robotic platforms [2], expensive sensors [28], or unreliable sensors [15], and may not be as robust as one based on a Bayesian framework [17]. Therefore a low-cost, computationally light, robust vision based control system for an autonomous vehicle is still subject of active research.

3 Proposed Vision-based Target Following System

This section describes the design of the proposed platform as well as the methods used to estimate the target position and to control the robot.

3.1 System description

The system is composed of the following hardware: A Creative Senz3D ToF camera that is able to capture an RGB and depth image. The camera has a depth range of approximately $1m$ and generates range images at $30fps$.[2] A Jetson TK1 embedded computer is attached to the iRobot Create 2 in order to process the information collected from the camera and send control commands to the robot. See Figure 1 for an illustration of the overall system (the cost of the components for one prototype was approximately 500 US dollars).

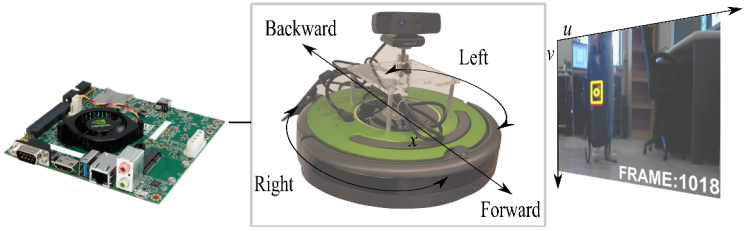

Fig. 1: Overall robotic platform architecture.

[2] Although other ToF cameras such as the classic $SR4000$ from MESA imaging, the PMD CamCube 3.0 or SoftKinetic's DS536A have ranges of up to $5m$, the low-cost and lightweight Senz3D was deemed sufficient for our purposes.

3.2 Target detection

In order to measure the position of the target at each frame, the system uses a C++ implementation of TLD [29] which computes the target position in the image plane (u and v) as well as its width and height (w and h). TLD is used because it is a discriminative tracker and hence does not require previous information about the target, which gives the system the flexibility needed to track any object of interest. The target is initially selected by the user and variations on target appearance are learned as it is tracked.

3.3 System model

The system is modeled using a linear Kalman Filter. The state vector is $\mathbf{x} = [u\ v\ z\ \dot{u}\ \dot{v}\ \dot{z}]$, where u, v are the pixel coordinates of the object, z is the distance from the sensor to the object, and $\dot{u}$, $\dot{v}$, $\dot{z}$ are the velocities in each dimension, respectively. The object tracking system is then modeled in state space form as:

$$\mathbf{x}(t) = A\mathbf{x}(t-1) + B\mathbf{u}(t) + \mathbf{w}(t) \tag{1}$$

$$\mathbf{y}(t) = C\mathbf{x}(t) + \mathbf{v}(t) \tag{2}$$

where Eq. (1) represents the system dynamics, including the state transition matrix A, the influence of the control action B and the process noise $\mathbf{w}$, and Eq. (2) is the measurement model, which includes the observation matrix C and the measurement noise $\mathbf{v}$. The process noise and measurement noise are assumed to be white, Gaussian, with variances R_{ww} and R_{vv}, respectively. That is, $\mathbf{w} \sim \mathcal{N}(0, R_{ww})$, and $\mathbf{v} \sim \mathcal{N}(0, R_{vv})$.

The object tracking system is modeled with the following state transition and measurement matrices:

$$A = \left[\begin{array}{c|c} I_3 & I_3 \\ \hline 0_{3\times3} & I_3 \end{array}\right], \quad B = \left[\begin{array}{c|c} 0_{2\times3} & \\ \hline k_1 & 0 \\ 0 & 0 \\ 0 & k_2 \end{array}\right], \quad C = \begin{bmatrix} I_2 & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & 1_{2\times1} & 0_{2\times3} \end{bmatrix} \tag{3}$$

where I_m is a $m \times m$ identity matrix and $0_{m\times n}$ and $1_{m\times n}$ are $m \times n$ matrices of zeros and ones, respectively. Matrix A above assumes that the target moves with a constant velocity such that $\dot{u}(t) = \dot{u}(t-1)$, $\dot{v}(t) = \dot{v}(t-1)$ and $\dot{z}(t) = \dot{z}(t-1)$ $\forall(t)$. Matrix B accounts for the effect of the control action of the PID controller on the velocities of the x and z axes. The rotation of the robot is accomplished by controlling the displacement in the image Δu, this relationship can be considered $\theta \approx \Delta u$ since the displacement from one frame to another is small in comparison to the distance between the robot and the target (see Figure 2). Translation is carried out by attempting to preserve the relative distance between the robot and the target at the first instant of time. The C matrix indicates that the measurements available at any given time are the current u, v coordinates of the object (the output of TLD) and z, the range from the robot to the object, which

is obtained from both the ToF camera and TLD. Matrix C is a 6×4 matrix whose first two rows correspond to the observations of u and v provided by TLD and whose last two rows correspond to the distance measurements obtained by the ToF camera and by the relative scale computed using TLD. The data fusion between the TLD and ToF measurements will be covered in detail below.

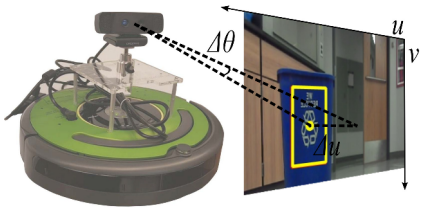

Fig. 2: Small angle approximation justification.

3.4 Data fusion approach

There are two main purposes for fusing measurements in this system. The first is to increase the overall estimation accuracy. The second is to allow the robot to follow a target even when it goes beyond the threshold of the ToF camera. The ToF camera is able to measure depth consistently and precisely when a target is located less than $1m$ away, however, it becomes very noisy and unreliable beyond this distance, generating many false measurements. A depth estimate based on relative scale changes as measured by TLD is used to compensate for these false measurements, effectively extending the operating range of the system.

The depth measurement from the ToF camera is calculated by averaging all the non-zero depth pixels inside the target bounding box (pixels whose depth cannot be estimated, such as those beyond the camera range, are read with a zero value). The height and width (h and w) provided by TLD are used to measure the scale variations of the target and hence provide an indirect depth estimate. The scale change of the target is translated to a real distance according to

$$TLD_z = K_z . \sqrt{\frac{w_{img} \times h_{img}}{w \times h}} \tag{4}$$

where K_z is a constant obtained by relating the initial depth measurement from the camera to the initial target bounding box size (w and h) and h_{img} and w_{img} are the height and width of the image.

The reliability of the ToF depth measurement is determined according to the following sigmoidal relationship

$$Rvv_\zeta = 1 - \frac{1}{1 + e^{(\eta \times r_0 - \zeta)}} \tag{5}$$

where r_0 is the percentage of zero elements in the target bounding box image, η defines the slope of the function and ζ is the value where the penalization takes place. The sigmoid function allows the Kalman filter to smoothly transition between the ToF and the TLD distance measurements using the following 4×4 covariance matrix

$$R_{vv} = diag(R_{vv_u}, R_{vv_v}, Rvv_{ToF}, Rvv_{TLD}) \tag{6}$$

where $diag(.)$ represents a diagonal matrix, R_{vv_u} and R_{vv_v} reflect the uncertainties in the observation of u and v and Rvv_{TOF} and Rvv_{TLD} represent the distance uncertainties as computed by the ToF camera and the TLD scale and are defined as follows

$$Rvv_{TOF} = \kappa \times Rvv_{\zeta} \tag{7}$$

$$Rvv_{TLD} = \kappa \times (1 - Rvv_{\zeta}) \tag{8}$$

Hence, as R_{vv_ζ} varies, the confidence level of the system is adjusted so that more weight is given to the ToF measurements or to the TLD relative scale. κ represents the penalization amplitude in the sigmoid function.

3.5 Controller design

Independent PID controllers are used for the translational and rotational velocities of the robot. As shown in Figure 1, the translational velocity allows the robot to drive forward or backward, and the rotational velocity turns it to the left or to the right.

The set-point for the translational controller is the initial distance between the target and the robot in the first measurement. We require this distance to be within the range of the ToF camera so that the TLD scale measurement can be properly initialized. The PID constants for the rotational controller are $Kp = 0.82$, $Ki = 0$ and $Kd = 0$. The set-point for the angular controller is the center of the image in the x axis. The constants for this controller are $Kp = 0.4$, $Ki = 0$ and $Kd = 0.03$. All the controller constants were found experimentally so that the robot would show a fast yet smooth response.

In order to decouple the control actions, we implemented a simple heuristic that checks for the magnitude of the error in the set points and decides whether to move forward or to turn at each frame based on the largest error. That is, if the difference between the u coordinate of the target and the corresponding set point in the center of the image is larger than the difference between the radial distance from the target to the sensor and its corresponding set point, the rotation controller is activated. Otherwise, the translation controller is activated. In order to be able to compare these distances, they are both normalized so that they range between 0 and 1.

4 Experimental Results

We qualitatively evaluated the ability of the system to track a given target by attaching an object (recycling bin) to another iRobot Create 2 which was man-

ually controlled while the autonomous robot followed it successfully through a variety of conditions. A sketch of the map illustrating the trajectory of the robot is shown in Figure 3. As the figure shows, the system autonomously followed the target for approximately $110m$. Screen captures obtained by the robot during this experiment are shown in Figure 4. The parameter values used for the experiments are the following: $\zeta = 12$, $\eta = 20$, $\kappa = 100$, $k_1 = k_2 = 0.01$, $R_{ww} = \text{diag}(0, 0, 0, 0.1, 0.1, 0.1)$, and $R_{vv_u} = R_{vv_v} = 0.3$.

In order to evaluate the ability of the system to recover from full occlusion while still carrying out smooth depth estimation, we tracked a target object (recycling bin) sliding across the ground so that another object (large trash bin) entirely occluded the target. As the screen captures in Figure 5 indicate, despite abrupt variations in depth measurements caused by the occluding object, the system is able to fully recover from severe occlusions while maintaining its distance from the target.

Figure 6 demonstrates the system's ability to respond to a fast moving target at distances beyond the range of the ToF camera. Figure 7 shows quantitative results regarding this experiments. The top left graph shows the measured and the estimated pixel positions u_{pos} and its set point s_{p_u}, which is the center of the camera field of view. The top right graph shows the reference distance from the target, s_{p_z}, the measured range from TLD_z and ToF_z as well as the fused estimate est_z. Finally, in the bottom plots of the figure we can see the control actions performed in order to move the robot in response to the position error estimates. As the figure indicates, the linear and angular speed controllers try to compensate for the estimated errors u_{pos} and est_z, respectively.[3]

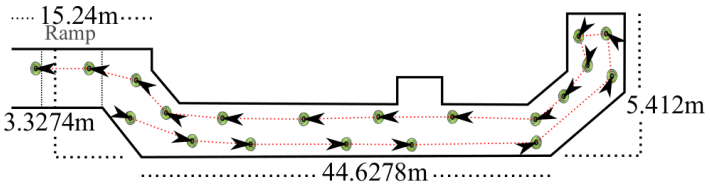

Fig. 3: Floor plan sketch showing the robot trajectory.

Figure 8 illustrates the response of the system to fast motions along the u axis. As the target moves in a certain direction, the robot moves to compensate for that. As the figure shows, when the target stops moving (from around iteration 500 to 590 and 700 to 750), the robot motion quickly stabilizes with the target near the set point. Note that the small bias in position could be easily compensated by further tuning the rotation controller.

[3] Note that the set points s_{p_u} and s_{p_z} correspond to the desired target position with respect to the robot, not to the actual robot position. The controllers use the set points to move the robot so that the difference between the estimated position and the set point is minimized.

Fig. 4: Screen captures from the qualitative experiment illustrating the system's robustness to a variety of conditions, particularly illumination changes.

Fig. 5: System recovering the target object (recycling bin) after a full occlusion by another object (large trash bin).

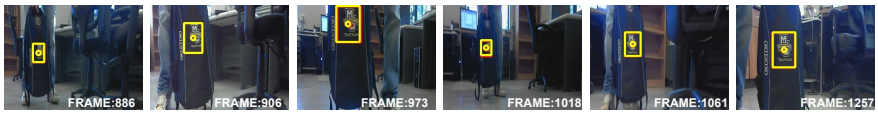

Fig. 6: The target object (backpack) is moved backwards, quickly and beyond the ToF camera's range ($> 1m$), the system is able to respond smoothly, making use of TLD_z, and successfully follows the object.

In order to show the effects of moving the target out of the range of the ToF camera, we kept the robot static and tracked a target at different distances starting well within the ToF range and progressing towards the $1m$ threshold and beyond. The results of this experiment are shown in Figure 9. The left graph shows that when the target is within the range of the ToF sensor, the estimate relies on measurements from TLD and ToF (frames $\sim 100 - 200$). When the target is farther than $1m$ (frames $\sim 350 - 650$), the estimated distance is based almost entirely on TLD. When the target is moved back to the starting position (frame ~ 650) the ToF measurements are again considered in the estimate. The right plot shows that when the target is near the $1m$ mark (frames $\sim 350 - 500$), the ToF measurements are very noisy and hence relying mostly on TLD is in fact an appropriate strategy.

We validate our choice of Rvv_ζ by illustrating that the percentage of zeros in the depth image is a viable way to determine the accuracy of the ToF sensor. In other words, the error between the ToF measurements and actual distance should increase monotonically as Rvv_ζ increases. This experiment also consisted of moving the target object progressively farther away while keeping the robot static. However, this time the focus was not on the behavior near the threshold of $1m$, but on the overall trend of the error as the distance increased. The graph in Figure 10 shows that as the target moves away from the robot, the error between ToF measurements and the ground truth increases and so does Rvv_ζ.

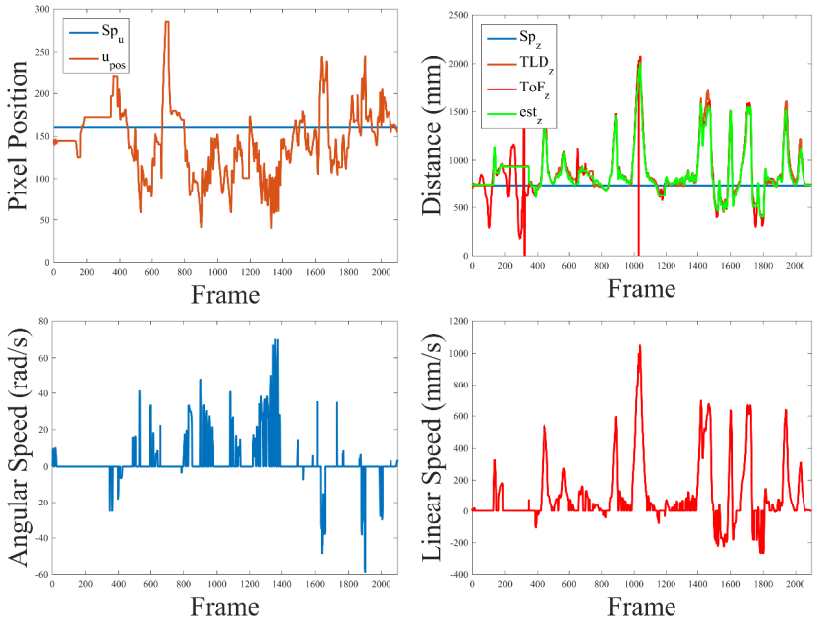

Fig. 7: Quantitative results corresponding to the backpack tracking experiment shown in Figure 6. $s_{p_u} = 160$ corresponds to the center of the image in the horizontal direction, and $s_{p_z} = 750$ corresponds to the initial distance to the target in millimeters.

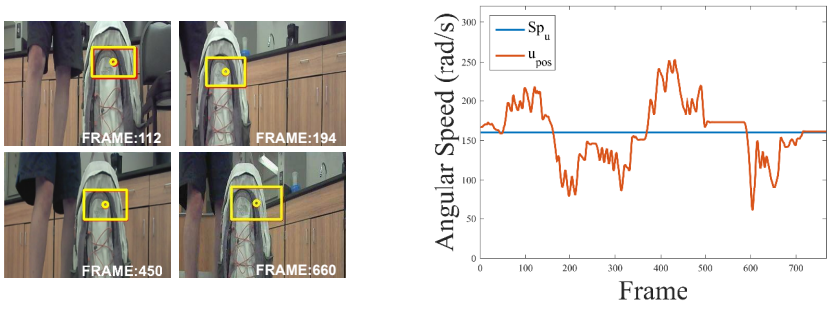

Fig. 8: Left) The target object (backpack) is moved to the left and to the right. Right) Response of the controller to the angular turn.

5 Conclusion

In this paper, an autonomous, cost effective, vision-based object following ground vehicle was proposed. The system was based on an iRobot Create 2 mobile platform, a Creative Senz3D ToF camera, and a Jetson TK1 embedded computer. Object detection was accomplished using TLD and tracking was performed by

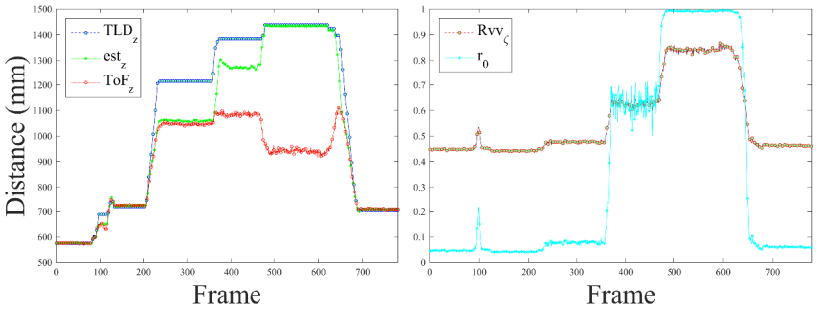

Fig. 9: Plots showing in detail how the data fusion between TLD and ToF measurements works.

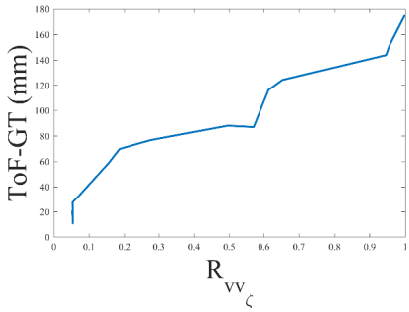

Fig. 10: Correlation between the distance and the level of confidence Rvv_ζ.

a Kalman filter. Data fusion was implemented in order to extend the operating range of the system beyond the measuring capabilities of the ToF sensor. All the processing was performed in real-time on the on-board computer. Several experiments were conducted where the system successfully followed target objects in a variety of situations, including illumination changes, full occlusions, and rapid movement at far ($> 1m$) distances. Quantitative experiments showed in detail how data fusion is accomplished.

Although the proposed approach is not restricted to Kalman filters and alternative recursive Bayesian methods such as Sequential Monte Carlo approaches [30] could be employed for increased robustness, one of our main objectives was to devise a lightweight method that could be used in portable embedded platforms. Hence, a Kalman filter seemed like the most effective choice.

There are several future directions to explore in this project. In the first place it would be beneficial to improve the control heuristics so that the decision between moving forward and turning would occur more seamlessly thereby reducing the chances of losing track of the target due to abrupt motions. Second, the ToF camera used in this project has a limited range and cannot be used outdoors, so a better camera would extend the use of the system. In addition, faster and more robust tracking can be accomplished simply by porting more of

the software implementation to the GPU in the embedded computer. Finally, exploring data association and track management mechanisms would allow for the system to perform more robustly in more complex scenarios in which multiple similar targets move in close proximity.

References

1. Bonin-Font, F., Ortiz, A., Oliver, G.: Visual navigation for mobile robots: A survey. Journal of Intelligent and Robotic Systems **53** (2008) 263–296
2. Jung, B., Sukhatme, G.S.: Real-time motion tracking from a mobile robot. International Journal of Social Robotics **2** (2010) 63–78
3. Papanikolopoulos, N.P., Khosla, P.K., Kanade, T.: Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision. Robotics and Automation, IEEE Transactions on **9** (1993) 14–35
4. Ahrens, S., Levine, D., Andrews, G., How, J.P.: Vision-based guidance and control of a hovering vehicle in unknown, GPS-denied environments. In: Robotics and Automation, 2009. ICRA '09. IEEE International Conference on. (2009) 2643–2648
5. Fowers, S.G., Lee, D.J., Tippetts, B.J., Lillywhite, K.D., Dennis, A.W., Archibald, J.K.: Vision aided stabilization and the development of a quad-rotor micro UAV. In: Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on. (2007) 143–148
6. Kwon, H., Yoon, Y., Park, J.B., Kak, A.C.: Person tracking with a mobile robot using two uncalibrated independently moving cameras. In: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. (2005) 2877–2883
7. Schlegel, C., Illmann, J., Jaberg, H., Schuster, M., Wrz, R.: Vision based person tracking with a mobile robot. In: BMVC. (1998) 1–10
8. Benavidez, P., Jamshidi, M.: Mobile robot navigation and target tracking system. In: System of Systems Engineering (SoSE), 2011 6th International Conference on. (2011) 299–304
9. Hu, C., Ma, X., Dai, X.: A robust person tracking and following approach for mobile robot. In: Mechatronics and Automation, 2007. ICMA 2007. International Conference on. (2007) 3571–3576
10. Kim, J., Shim, D.H.: A vision-based target tracking control system of a quadrotor by using a tablet computer. In: Unmanned Aircraft Systems (ICUAS), 2013 International Conference on. (2013) 1165–1172
11. Papachristos, C., Tzoumanikas, D., Alexis, K., Tzes, A.: Autonomous robotic aerial tracking, avoidance, and seeking of a mobile human subject. In: International Symposium on Visual Computing, Springer (2015) 444–454
12. Woods, A.C., La, H.M.: Dynamic target tracking and obstacle avoidance using a drone. In: International Symposium on Visual Computing, Springer (2015) 857–866
13. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34** (2012) 1409–1422
14. Ma, X., Hu, C., Dai, X., Qian, K.: Sensor integration for person tracking and following with mobile robot. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on. (2008) 3254–3259
15. Clark, M., Feldpausch, D., Tewolde, G.S.: Microsoft kinect sensor for real-time color tracking robot. In: Electro/Information Technology (EIT), 2014 IEEE International Conference on. (2014) 416–421

16. Teuliere, C., Eck, L., Marchand, E.: Chasing a moving target from a flying UAV. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. (2011) 4929–4934
17. Guerin, F., Fabri, S.G., Bugeja, M.K.: Double exponential smoothing for predictive vision based target tracking of a wheeled mobile robot. In: Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on. (2013) 3535–3540
18. Wang, W.J., Chang, J.W.: Implementation of a mobile robot for people following. In: System Science and Engineering (ICSSE), 2012 International Conference on. (2012) 112–116
19. Pieropan, A., Bergström, N., Ishikawa, M., Kjellström, H.: Robust 3d tracking of unknown objects. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE (2015) 2410–2417
20. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. (2009) 983–990
21. Rigatos, G.G.: Extended kalman and particle filtering for sensor fusion in motion control of mobile robots. Mathematics and Computers in Simulation **81** (2010) 590–607
22. Dinh, T.B., Yu, Q., Medioni, G.: Co-trained generative and discriminative trackers with cascade particle filter. Computer Vision and Image Understanding **119** (2014) 41–56
23. Medeiros, H., Park, J., Kak, A.: Distributed object tracking using a cluster-based kalman filter in wireless camera networks. IEEE Journal of Selected Topics in Signal Processing **2** (2008) 448–463
24. Medeiros, H., Holguín, G., Shin, P.J., Park, J.: A parallel histogram-based particle filter for object tracking on simd-based smart cameras. Computer Vision and Image Understanding **114** (2010) 1264–1272
25. Yoon, Y., han Yun, W., Yoon, H., Kim, J.: Real-time visual target tracking in RGB-D data for person-following robots. In: Pattern Recognition (ICPR), 2014 22nd International Conference on. (2014) 2227–2232
26. Shimura, K., Ando, Y., Yoshimi, T., Mizukawa, M.: Research on person following system based on RGB-D features by autonomous robot with multi-kinect sensor. In: System Integration (SII), 2014 IEEE/SICE International Symposium on. (2014) 304–309
27. Nakamura, T.: Real-time 3-D object tracking using kinect sensor. In: Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on. (2011) 784–788
28. Chen, C.H., Cheng, C., Page, D., Koschan, A., Abidi, M.: A moving object tracked by a mobile robot with real-time obstacles avoidance capacity. In: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. Volume 3., IEEE (2006) 1091–1094
29. Nebehay, G.: Robust object tracking based on tracking-learning-detection. Master's thesis, TU Wien (2012)
30. Loy, G., Fletcher, L., Apostoloff, N., Zelinsky, A.: An adaptive fusion architecture for target tracking. In: Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on, IEEE (2002) 261–266