2015 International Conference on Computational Science and Computational Intelligence

MRSL: Autonomous Neural Network-Based 3-D Positioning System

Hooman Hedayati
Department of Computer Science
East Carolina University
Greenville, NC
Hedayatih13@students.ecu.edu

Nasseh Tabrizi
Department of Computer Science
East Carolina University
Greenville, NC
Tabrizim@ecu.edu

***Abstract*— Stabilizing and localizing the positioning systems autonomously in the areas without GPS accessibility is a difficult task. In this paper we describe a methodology called Most Reliable Straight Line (MRSL) for stabilizing and positioning camera-based objects in 3-D space. The camera-captured images are used to identify easy-to-track points "interesting points" and track them on two consecutive images. The distance between each of interesting points on the two consecutive images are compared and one with the maximum length is assigned to MRSL, which is used to indicate the deviation from the original position. To correct this our trained algorithm is deployed to reduce the deviation by issuing relevant commands, this action is repeated until MRSL converges to zero. To test the accuracy and robustness, the algorithm was deployed to control positioning of a Quadcopter. It was demonstrated that the Quadcopter (a) was highly robust to any external forces, (b) can fly even if the Quadcopter experiences loss of engine, (c) can fly smoothly and positions itself on a desired location.**

Keywords- Artificial Intelligence; Neural Networks; Machine Learning

I. INTRODUCTION

The demands for utilizing autonomous vehicles in hazardous situations are increasing rapidly. There are plenty of examples where these vehicles are successfully used in practice, such as those inspecting the damaged nuclear reactors in Fukushima [1], robotic submarines attempting to repair the ruptured well in the Gulf of Mexico [2] and firefighters rescuing survivors in a collapsing building [3]. Among these robots, Quadcopters are becoming more popular for observational and exploration purposes in indoor and outdoor environments [4]. One of the most important advantages of Quadcopters compared to other robots is that it can move in any given 3-D direction at any time. With this unique feature, Quadcopters can maneuver extremely well in constrained indoor spaces, making them one of the best choices for indoor environments exploration [4].

In order to navigate, Quadcopters often rely on a wide variety of sensors including Inertial Measurement Unit (IMU), GPS, and gyroscopes. Although these sensors are not completely reliable individually, combinations of them arguably make them good enough for outdoor positioning. However, flying in an unknown environment without GPS signal requires alternative positioning methods that include expensive sensors like a 3-D depth scanning camera [5]. Alternatively, one can use optical cameras to collect information [6]. While cameras are cost efficient, they have some disadvantages where excessive amounts of data is collected which in turn makes processing data computationally expensive. In addition, 2-D images that are captured using cameras make it difficult to extract 3-D information.

One crucial objective of any Quadcopter is to localize and stabilize itself by maintaining its position by constantly counteracting minor randomly induced movements. Although Internal Measurement Units (IMU) help to achieve this, but a major disadvantage of using IMUs for navigation is that they typically suffer from accumulated errors, including the "Abbe Error" [7], which describes the magnification of angular error over distance. Furthermore, as stabilizing systems continually add detected changes to its previously calculated positions; any errors in measurement are accumulated leading to 'drift', or an ever-increasing deviation from the actual location, making IMUs not reliable [7].

The task of accurate localizing of a Quadcopters in previously unseen environments is widely investigated [8]. This paper presents an innovative approach for stabilizing camera-based objects in 3-D. Although accuracy and robustness of our approach is tested on a Quadcopter, this algorithm can be implemented on any 3-D positioning systems.

II. METHODOLOGY

As it is shown in Figure 1, the camera captures images that are used to identify easy-to-track points named "Landmarks" and tracks them between two consecutive images. The longest distance between the landmarks on the two consecutive

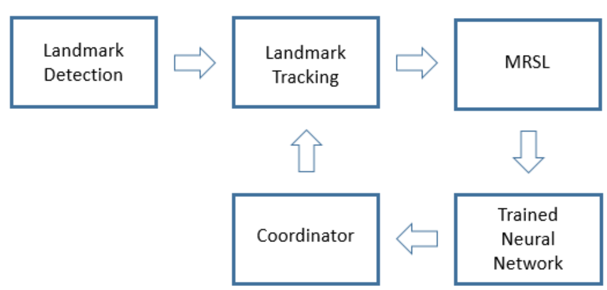

Figure 1. Schematic outline of a MRSL algorithm.

978-1-4673-9795-7/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCI.2015.88

images is assigned to MRSL, which is used to indicate the deviation from the Quadcopter's original position. Our trained algorithm is then deployed to reduce the deviation by issuing relevant commands. This action is repeated until MRSL converges to zero.

A. Landmark Detection and Tracking

Movement of the camera, which implies to movement of Quadcopter, cause changes in the position of landmark points. The movements of these points are tracked in the next step by extracting the magnitude and direction of Quadcopter movement. Some of the features that can be extracted from the images are edges, corners, blobs, or ridges. Several feature detection algorithms like SIFT (DoG) [9], SURF [10], FAST [11] and Harris corner detector [12] were considered for this study but because the high-speed requirement, it was decided to adopt the Features From Accelerated Segment Test (FAST) algorithm. FAST is a corner detection method, which can be used to extract landmarks by using a circle of 16 pixels to classify whether center pixel of the circle is aligned with a landmark. Decision is made by comparing color intensity of the center pixel with those of 16 neighboring pixels located on the perimeter of the circle. For each candidate pixel P (see Figure 2), the algorithm identifies which pixels are aligned with a circle perimeter: if a long enough sequence of continuously brighter or continuously darker pixels is found, it is classified as a landmark.

Once the landmarks are detected, the exact positions of them are extracted next. A general formula for tracking is to find parameters p of a warp function $\boldsymbol{f(x,y;p)} : \mathbb{R}^2 \times \mathbb{R}^d \to \mathbb{R}^2$ such that the difference between the original patch ***T(x,y)*** and the transformed image ***I(f(x,y ; p))*** becomes minimal, that is minimizing the sum of squared differences (SSD) [13], see Eq. 2.

$$\boldsymbol{E_{SSD}(P)} \coloneqq \sum_{x,y} \left(\boldsymbol{I(f(x,y;p)) - T(x,y)}\right)^2 \tag{1}$$

$$\boldsymbol{P^* = arg\ min\ E_{SSD}(p)} \tag{2}$$

Choosing a good warp function is critical to having a suitable degree of freedom, "Pure Translation" function selected as a warp function, which tracks a two-dimensional image patches. The resulting transformation has two degrees of freedom, the displacement in two dimensions is shown in Eq. 3.

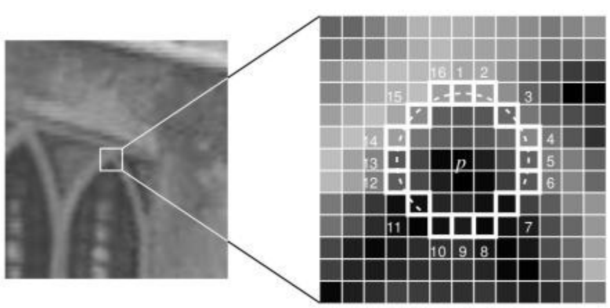

Figure 2. The FAST-16 corner detector

$$\boldsymbol{f(x,y;\delta x,\delta y)} = \begin{pmatrix} \boldsymbol{x+\delta x} \\ \boldsymbol{y+\delta y} \end{pmatrix} \tag{3}$$

B. Extracting the Movements

Let C_1 and C_2 be two consecutive images captured by the camera. First FAST algorithm is deployed on C_1 to find all landmarks coordinates $(P_1, P_2, \dots, P_n)$. Then tracking algorithm is run on C_2 to track same landmarks and find their new coordinates $(P'_1, P'_2, \dots, P'_n)$. By connecting each pair P_m to P'_m a line is formed and labeled "Reliable Straight Line" (RSL). RSLs are important because they can be used to estimate camera rotation and translation using essential matrix in epipolar geometry. The essential matrix is a 3x3 matrix that encodes the relationship between two images of the same scene from different viewpoints. The essential matrix is defined as in Eq. 4.

$$\boldsymbol{E} := \boldsymbol{R} \times [\boldsymbol{t}]_{\boldsymbol{x}} \in \mathbb{R}^{3\times 3} \tag{4}$$

Where $[\boldsymbol{t}]_x$ is the matrix corresponding to cross product of ***t*** and R, is rotation matrix. As a property of the essential matrix for each P and P$'$ Eq. 5 is held

$$\boldsymbol{(P')^T E\, P = 0} \tag{5}$$

As a result of essential matrix properties there are solutions for determining R and ***t*** based on performing Singular Value Decomposition. Using Multiple View Geometry [14] the rotation matrix R and cross product vector as shown in Eq. 6 and 7 are computed next

$$[\boldsymbol{t}]_x = \pm \mathbf{V\,W\,\Sigma\,V^T} \tag{6}$$

$$\mathbf{R = U\,W^{-1}}\boldsymbol{V^T} \tag{7}$$

$$\text{Where,}\quad \mathbf{W} = \begin{bmatrix} \mathbf{0} & \mathbf{-1} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{8}$$

Where U and V are orthogonal 3x3 matrices and Σ is 3x3 diagonal matrix, see Eq. 9.

$$\mathbf{\Sigma} = \begin{bmatrix} \boldsymbol{s} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{s} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{9}$$

There are several algorithms such as five-point and eight-point algorithms for solving R and ***t***. Five-point algorithm [15] was used in this study to estimate camera-rotation and translation from two consecutive images.

C. Most Reliable Straight Line

Let C_a and C_b be two arbitrary captured images. By assumption there is no moving object in the images if three landmarks exists such that the coordinates $P_a = P'_b$ where the position of the camera in those moments was identical. Using this fact, if the sums of all RSLs length converge to zero over time, it can be concluded that the camera is stabilized. Therefore the goal is to minimize the RSLs by issuing relevant commands to Quadcopter. Length of the RSL is directly

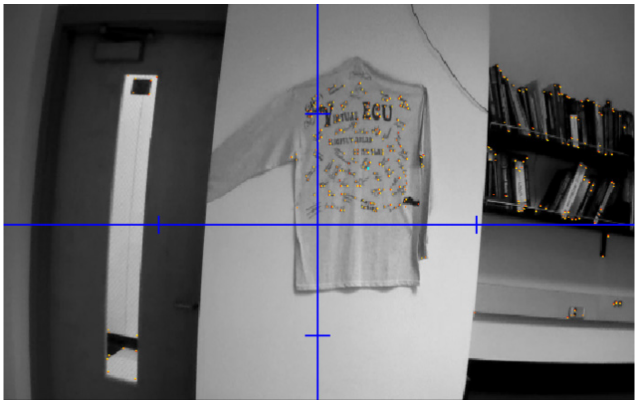

Figure 3. Landmark extraction initialization

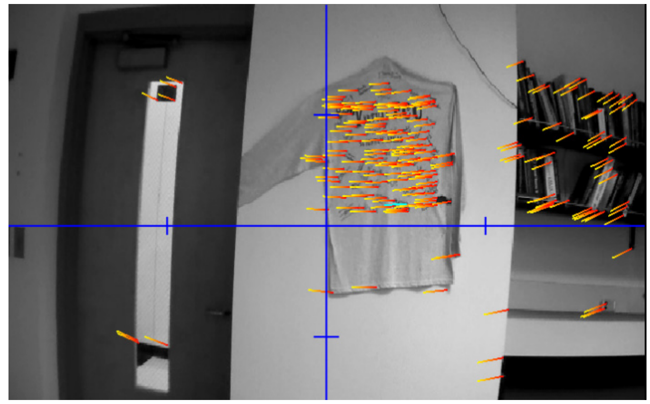

Figure 4. All detected RSLs

correlated with accuracy of algorithm. When one RSL is longer than the others, stabilization is more accurate and slope is good measure for reliability of a specific RSL. When the slopes are not the same, which means either an object is moving inside the image or camera is rotating around itself.

The MRSL is the longest of RSLs those with similar slope. For each MRSL, pp', there exist a vector named Most Reliable Straight Vector (MRSV) at starting point of p' and endpoint of p. MRSV has equal magnitude and direction that of MRSL.

Although MRSL can be implemented to stabilize Quadcopters, but to remove all the unwarranted disturbances an alternative RSL should be used in order to avoid the risk of failure. In this case the RSLs with the same slopes are grouped into 3 equal size clusters. Beside the MRSL alternative RSLs one line from first along with one from second and two from the last clusters are selected and represented by an array $[\boldsymbol{ID}, (\boldsymbol{x},\boldsymbol{y}), (\boldsymbol{x}',\boldsymbol{y}'), \boldsymbol{L}]$, where ID is unique identification number for each landmark, ***(x,y)*** is coordinate pair of C_1 , and ***(x',y')*** is coordinate pair of C_2, and L is the length of RSL.

The candidate MRSL and 4 other RSL are used as inputs to the NN, Figure-3 shows the initial step and Figure 4 shows all detected RSLs.

D. Neural Network-Based Stabilization System

The goal of using NN is that by giving the |MRSL| > 0, suitable commands are issued as output. These commands counteract and move the Quadcoter back to its initial position. The designed NN has 7 inputs, where five of them are described in the previous section and are MRSL and 4 alternative RSLs. The built-in gyroscope data and speed of each motors are used as the remaining two inputs. Current speed of the motors helps in issuing commands with respect to current state of Quadcopter.

By controlling speed of each of the motors of a Quadcopter, its movement can be controlled in 3-D environment. The four different output of NN are respective values of electrical current applied to each brushless motor. The range of each output node is [0,1]. Zero means that the motors are off and the one means that the motors are running in the full load. As a last piece of design, NN has one hidden layer consisted of 10 nodes.

In order to train NN, a training dataset is generated that includes :

$$(\boldsymbol{MRSL}, \boldsymbol{RSL_1}, \boldsymbol{RSL_2}, \boldsymbol{RSL_3}, \boldsymbol{RSL_4}, \boldsymbol{Gyro_t}, \boldsymbol{M_t}) \rightarrow (\boldsymbol{M_1}, \boldsymbol{M_2}, \boldsymbol{M_3}, \boldsymbol{M_4}) \quad (7)$$

The 5 lines, gyroscope data and current state of motors, maps to 4-tuple electrical currents controlling the speed of each motor. The interpretation is that in order to move the camera to have a specific |MRSL|, a specific electrical current should be applied to each motor.

This dataset was generated in a creative way. A program generated 4 random numbers in the range [0,1]. These numbers were sent to motors as control commands. By capturing two images (C_1 and C_2) before and after sending that command, landmarks are extracted from C_1 and tracked in C_2 . All variables regarding the MRSL algorithm were calculated and collected in a form of tuple as shown in (7). This experiment was repeated 700 times to make sure that enough data was generated for training. This experiment was supervised by an operator to avoid collisions and gather different viewpoints to make sure that the algorithm is not biased with a particular landmark. Noise and outliers were injected to the dataset to avoid over fitting. The final dataset contains 1000 samples, 90% were used for training and 10% for testing purpose.

Back propagation [16] was used to compute all the weights in the network. After training, the algorithm was validated using the test dataset. The success rate of 94.3% was observed.

III. COORDINATOR

Coordinator is an interface, which handle wireless communication between the Quadcopter and the PC. Because two different platforms are used in implementation, coordinator interpret the output of NN to suitable command and transmit them to the Quadcopter's flight control board. Also it takes images from Quadcopter and sends them as input for MRSL algorithm. Moreover, coordinator is responsible for the emergency situation e.g. when the battery is too low or the MRSL algorithm is not responding for any reason, coordinator would do an emergency landing to reduce the probable damages.

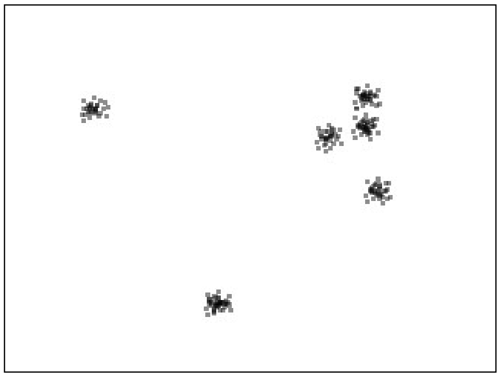

Figure 5. Landmarks change captured by camera

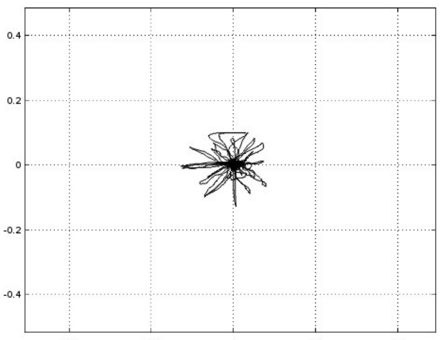

Figure 6. Movement of center of Quadcopter captured by fix camera

IV. RESULT

In this section the accuracy of stabilization and robustness of the Quadcopter using the MRSL approach is evaluated with experimental data obtained from several test flights in different location. The experiments can be categorized in two main groups. First is evaluating the stabilization without any external disturbances, second is by applying disturbances e.g. pushing or pulling Quadcopter or blowing air with blower.

In the first set of experiments the Quadcopter stabilized with supervision of a human pilot in a certain position, then the pilot stopped navigating and the algorithm began controlling Quadcopter for 60 seconds. The goal of the experiment was to evaluate the stability of the Quadcopter. This experiment was repeated 100 times. Two criteria measured for evaluation were movement of the detected landmarks and movement of the center of mass of the Quadcopter from viewpoint of external fixed camera. As is shown in Figure 5 visualization of the movement of the landmarks and the actual position of the Quadcopter in a 3-D space. Figure 6 shows the movement of the center of mass of the Quadcopter captured by a fixed camera. Accuracy of stabilization measured is expressed by the shortest diameter of a circle that includes all the landmarks. In our experiment the average diameter was about 15 inches for center of mass of quadcopter. In the second experiment after stabilization, external disturbances were introduced in order to test robustness of the MRSL algorithm. The goal was to measure convergence speed and time of the Quadcopter to return to its

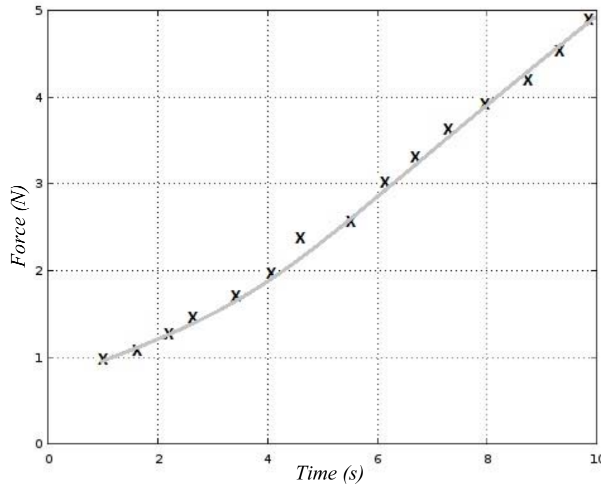

Figure 7. Convergence time by amount of applied force

initial location. Result showed that regardless of the direction of the introduced forces to disturb the stability of the Quadcopter, the convergence time remained almost constant. But the magnitude of force is inversely correlated with convergence time. The algorithm is not robust to forces that change the direction of the camera where, none of the landmarks remained within the image. This threshold is the critical point of MRSL algorithm. As Figure 7 shows forces from 1 to 5 Newton were applied and the measure of the convergence time was observed.

V. CONCLUSION

Stabilizing and localizing positioning systems autonomously in the areas without GPS accessibility is a difficult task. In this paper, we introduce an innovative methodology for stabilizing and localizing Quadcopters in 3-D environments. Most of the current methods used to positioning objects in 3-D are using expensive equipment in contrast to the methodology introduced in this paper. To prove the robustness of the algorithm, an experiment was set up using a Quadcopter to measure the reliability of the algorithm, it was observed that the algorithm can stabilize the Quadcopter effectively even in the presence of external disturbances.

ACKNOWLEDGMENT

Evaluation of the reliability of MRSL was partially supported by NSF REU grant with undergraduate student participation.

REFERENCES

[1] Javier Irizarry, Masoud Gheisari, Bruce N. Walker (2012) Usability assessment of drone technology as safety inspection tools, ITcon Vol. 17, pg. 194-212

[2] Staff writer (25 April 2010). "Robot subs trying to stop Gulf oil leak". CBC News. Retrieved 25 April 2010

[3] Brando, A (2003) Firefighter-robot interaction during a hazardous materials incident exercise, 2011 15th International Conference on Advanced Robotics

[4] Jackson, K (2012) Development and evaluation of a collision avoidance system for supervisory control of a micro aerial vehicle, MIT Thesis (p. 195-108)

[5] William Morris, Ivan Dryanovski, Jizhong Xiao (2010) 3D indoor mapping for micro-UAVs using hybrid range finders and multi-volume occupancy grids, workshop on RGB-D: Advanced Reasoning with Depth Cameras, Zaragoza, Spain

[6] Erdinç Altu˘g , James P. Ostrowski , Camillo J. Taylor (2005) Control of a Quadrotor Helicopter Using Dual Camera Visual Feedback, The International Journal of Robotics Research

[7] Siciliano B, Khatib O (2008) Handbook of Robotics, Springer

[8] C. Stachniss. (2006) Exploration and Mapping with Mobile Robots. PhD thesis, University Freiburg

[9] D. Lowe. Object recognition from local scale-invariant features. In Proc. of the International Conference on Computer Vision (ICCV), 1999.

[10] H. Bay, T. Tuytelaars, and L.V. Gool. SURF: Speeded-up robust features. In Proc. of the European Conference on Computer Vision (ECCV), 2008.

[11] E. Rosten and T. Drummond (2006) Machine learning for high-speed corner detection. In Proc. of the European Conference on Computer Vision (ECCV)

[12] Harris, C., & Stephens, M. (1988, August). A combined corner and edge detector. In Alvey vision conference (Vol. 15, p. 50).

[13] Jakob Julian Engel (2011), Autonomous Camera-Based Navigation of a Quadrocopter, TUM University Thesis

[14] Zisserman, R. H. (2004). Multiple View Geometry in Computer Vision. Cambridge University Press.

[15] H. Li and R. Hartley. Five-point motion estimation made easy. In Proc. of the International Conference on Pattern Recognition (ICPR), 2006

[16] Yves Chauvin (2008) Back-Propagation: Theory, Architecture, and Applications, Taylor & Francis