Video Processing Techniques for Mirror Detection in Monocular Visual SLAM

Julianne Marie Ruz and Jaime M. Samaniego

***Abstract*— Current methods of mirror detection rely on deep learning and set of training data to achieve their goal. While this is an effective method in itself, it may be possible to create a method based on how humans normally perceive mirrors that can act as a substitute if deep learning cannot be attempted. This paper presents a method of locating mirrors using video processing techniques and the concept of motion parallax and the perception of space.**

***Index Terms*— video processing, mirror detection, computer vision, smartphone robotics**

I. Introduction

A. Background of the Study

Robots are becoming a more familiar presence in human life with the development of technology. A basic part of what makes them useful is their ability to navigate their environment with ease. Unfortunately, robots cannot detect mirrors without either using deep learning or a combination of other sensors such as laser rangefinders and sonars. Only using one or the other will keep the robot from recognizing the reflective surface due to factors such as how light reflects ([1], [2]) which limit the sensor's ability to perceive.

B. Statement of the problem

A robot's ability to see and navigate together are essential for it be able to function. However, the unavailability of sensors such as laser rangefinders as well as the amount of time and resources needed to train a deep learning model are a hindrance to robotics engineers and developers looking to improve robot vision and mapping technology.

C. Significance of the Study

The presence of indoor robots is more widespread than ever before. For safety and navigation purposes, these robots must be able to detect mirrors, especially given how much more prevalent mirrors are in our daily lives. Mirror detection in robot SLAM (Simultaneous Localization and Mapping) focuses either on the use of other sensors such as lasers [3] alone or combined with sonar [2] or on making datatsets and training neural networks to identify features [4]. The use of lasers is more common, but due to the behavior of light when a laser is pointed at a mirror, it is possible for the mirror to go undetected [2] thus requiring a workaround in the form of an additional sensor. Deep learning (DL), while faster and more accurate, isn't as accessible because of the overall costs of training data and in the end might be overkill [5].

Agrawal et al. [6] state that a moving agent can be treated like a moving camera, making the agent naturally aware of its own motion. That, coupled with Cadena et al. [7] covering the use of cameras in the future of robotic SLAM allows for a possible alternative to the aforementioned techniques. Using traditional computer vision (TCV) to detect mirrors based on the physical behavior of light around them will not only add to the current knowledge of visual SLAM but also provide a more accessible and affordable option for robotics developers and engineers, especially those working in smartphone-related robotics (Azeta et al., 2019).

D. Objectives

The study aims to show that modified video processing techniques are a viable option for effectively identifying mirrors most commonly seen in today's modern architecture for use in simultaneous localization and mapping.

E. Scope and Delimitations

The study focuses only on the detection of flat and smooth reflective mirrors as opposed to convex or concave reflective mirrors. Additionally, due to the inconsistencies of outdoor environments, the study will be limited to indoor environments. It also does not claim that either traditional computer vision or deep learning is a better approach, only that the former can detect reflective surfaces on its own.

II. Review of Related Literature

1) Mirrors and the Behavior of Light: Mirrors are familiar objects even before their behavior is explained in the classroom. When looking into a mirror, the viewer sees a copy of themselves and whatever else is in front of the mirror. They know that this doppelganger will do whatever they do, but in reverse. This observation is part of what Croucher et al. [8] refer to as naive optics, or optics as it was understood prior to more modern theories. According to Croucher et al. [8], one of the reasons behind this naivete is that the viewer tends to see the mirror as a still image of whatever is in front of it, so they don't see the relevance of their position. Another is the ancient principle of extramission where it is believed that the viewer's sight shoots out of their eyes and towards the object [8]. In reality, light moves from the object towards the mirror where it is reflected back to the eye.

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Baños in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

What is seen inside the mirror is a virtual image formed by the reflected light. It is virtual in that the image formed by the reflected rays appears to come from the behind the mirror, when instead it comes from the points of reflection [9].

2) Egomotion in Perception: Cutting [10] describes the perception of space as being dependent on the presence and perception of objects in space. In addition to this, Cutting [10] also states that depth is not "seen", but objects in depth are seen due to depth being part of space. One of the cues used to perceive depth is the motion of objects relative to each other [11]. Gibson [11] explains further that space is seen as a gradual change in size and density in an object as the viewer moves towards or away from it. This is supported by a statement from Euclid (Burton, 1946 in Cutting, 1986) regarding the behavior of parallel lines as they move into the distance: "objects increased in size will appear to seem to approach the eye".

A mobile agent's self-motion is what Agrawal et al. [6] define as egomotion. An organism has free access to the information supplied by egomotion [6] through the brain which controls how it moves. Head motion in particular as covered in Gibson [11] is considered to be related to space perception and movement due to several reasons. One of these is that there is a higher sensitivity to motion that changes shape versus motion that changes location; Gibson [11] states that this sensitivity is seen in a camera panning from one side to another. The edges and corners seem to deform with the camera's movements, though it must be noted that the viewer cannot see these distortions in still images due to the lack of visible change over time (Arnheim, 1979 in Cutting, 1986), making it another reason for the importance of head motion in perception.

3) Visual SLAM: Simultaneous localization and mapping (SLAM) is the estimation of a robot's location on a map that is being built using the data gleaned from its sensors [7]. Visual SLAM is a subfield of this that is focused on using data from a robot's visual system for localization and mapping [12]. Sensors used in the robot's system for visual SLAM are usually 2D laser rangefinders or 3D LiDars, but nowadays laser scanners are the popular choice, limited though they may be in their sensing capacity [2]. Koch et al. [3] explain this further in their paper where they state that glass surfaces can be either reflective or transparent based on the laser's incidence angle. Besides these sensors, cameras such as the range cameras found in the Microsoft Kinect which popularized their usage are now being utilized in SLAM [7]. Robots with camera-based visual systems can interpret their environment in a way similar to how an organic agent would since the knowledge of their own movement is part of the data from the sensors [6]; [12].

In order for a robot to be able to navigate its environment, or to interact with it in general, it must have a reliable map [7]. Over the years, researchers have formulated a process for doing so. Gao et al. [12] describe the paths stemming from the procurement of sensor data. The first is through the frontend, also known as visual odometry – tracking the change of position over time using a stream of images [13]. This sounds similar to visual SLAM, however using visual odometry alone makes the map building algorithm prone to errors that cause a drift in trajectory over time if they remain unchecked since the number of errors increases with the number of images [13]; [12]. Visual SLAM fixes the problem by using the second path that Gao et al. [12] mentioned: loop closure.

Loop closures as events inform the robot that the path they are on intersects itself [7]. Cadena et al. [7] add that locating these loop closures give the robot a better understanding of its environment as well as allowing it to find shortcuts between locations. Gao et al. [12] also support the usefulness of loop closure, stating that it helps to reduce drift from the original path by checking if the robot has reached the same place. One of the methods of finding a loop closure is by checking images for similarities [12], however simply subtracting one image from another leads to problems because an image can change depending on the external factors affecting the camera sensor such as viewing angle or lighting. Remedies for these problems use visual odometry or a bag-of-words approach [12] so that the SLAM algorithm can eventually make an accurate map.

4) TCV vs Deep Learning: Traditional computer vision (TCV) techniques are algorithms built to perform the same way for any given input image [5], whilst deep learning is a subset of machine learning that uses neural networks to mimic how the human brain learns, with nodes that function as neurons ([14]; [5]). DL allows for greater accuracy as well as flexibility since neural networks can be trained and retrained with little to no supervision, however as stated in O'Mahony et al., [5] the presence of deep learning does not make traditional computer vision obsolete.

One reason for this is that using deep learning can be too much effort for simple tasks that, as mentioned previously, traditional computer vision is already built to do. Two more reasons are related to the overall costs of training a neural network. Aside from the large amount of data, time, and energy needed for training, the resulting neural network could be overfitted to the training data set, thus minimizing functionality for incoming data that strays from it [5].

Besides the problems relating to the costs and effort behind the use of deep learning, a major issue is that deep learning is unsuitable for fields like robotics and video processing due to its structure and behavior [5]. Due to nodes in a neural network being identified by where they are and what they do as opposed to more readable labels, deep learning is considered a "black box", making it harder to build and fix [15]. Marcus [15] also speaks of deep learning's incompatibility with prior knowledge. Bar a few exceptions, prior knowledge is used minimally. General knowledge such as the laws of physics are approximated based on what the neural network gathered from the given training data.

III. Methodology

The study will attempt to locate a mirror using the knowledge of its own egomotion gathered mainly through its video feed. This is to show that it is possible to detect a mirror with only one camera sensor and information available to the robot via its movement system. In order to to this, the study aims to proceed with the following:

A. Formation of the robot body

Robot body formation will follow part of the procedure used in Müller and Koltun [16] but with modifications for Android device placement as well as for the lack of a 3D printer.

1) Construct the main body
   a) Attach the motor shield to the Iduino Uno. Connect the battery container to ports on the motor shield and remove the yellow jumper to allow the micro-controller to draw power from the battery.
   b) Mount four (4) motor mounts onto the bottom of the chassis using M3 bolts and nuts.
   c) Attach the four (4) DC motors onto the motor mounts using M2 nuts and bolts. Use male-to-male jumper wires to attach all four DC motors to the motor shield.
   d) Attach the wheels to the shafts of the DC motors and turn over.
2) Construct a smartphone holder that can be rotated using a servo motor.
3) Attach the Bluetooth module, micro-controller, servo motor, and battery pack to the breadboard before attaching the breadboard and smartphone holder to the robot body.
4) Attach the Android smartphone to the Iduino Uno using a USB cable and place it on the smartphone stand.

B. Creation of the SLAM framework

The study will construct a SLAM framework based on Davison et al.'s [17] procedure using an Extended Kalman filter for real-time single camera SLAM albeit with some changes in feature initialization to account for the mirror detection software.

1) Scan the environment by rotating the Android smartphone camera 90 degrees to the right, 180 degrees to the left, then 90 degrees back to the original center position. If ten (10) seconds have passed AND the robot has moved to a new position or rotated its main body, a loop closure check will be performed using a modified version of the image-to-map procedure mentioned in Williams et al. (2008a).
2) Select the needed features (not including the features needed for the mirror detection process). The criteria for selecting these features are as follows:
3) Estimate the next location of the robot using a constant velocity, constant angular velocity model [17].
4) Move the robot to the next position.
5) Measure features and use the new information to update positions and uncertainties.
6) Repeat steps 1 through 6.

C. Creation of the mirror detection process

The requirements for the robot to have successfully located a mirror are as follows:

1) Main reference point (MRP) grows/shrinks as the robot approaches/retreats.
2) If visible, the MRP's movement is consistent with the movement of the robot.
3) Auxiliary reference blobs (ARBs) leave/enter the field of view at a rate consistent to the robot's approach/retreat.
4) ARBs grow/shrink at a rate consistent with the movement of the robot.

In order to accomplish these requirements, the study will proceed in this way:

1) Set an MRP. For the purposes of the study the MRP shall be an image of the robot's camera sensor and part of the surrounding area. The borders of the MRP will have to be hard-coded depending on the kind of Android phone being used. For this instance the MRP region is comprised of three (3) camera lenses in a line and a small flashlight bulb perpendicular the the third camera lens (Fig. 1). The region for the MRP is isolated from the rest of the source image by first converting the source image to grayscale and then sharpening it. Afterwards Gaussian blur and a Canny edge detector are used to bring out the necessary edges before locating the circles of the camera lens using Hough circle transform.

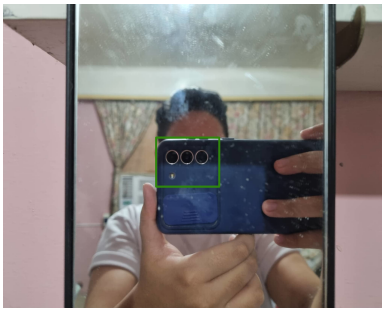


Fig. 1: The current MRP being used.

2) Do the following for each frame:
   a) Separate the frame into the largest regions using connected components.
   b) Get the contours of shapes inside of the largest regions. Separate them into triangles, rectangles, polygons with five to nine (5 - 9) corners, and blobs with more than nine (9) corners and compute their areas.
   c) Track the growth and movement of the shapes throughout the video.
3) Use these reference points as well as time and speed to successfully detect a mirror.

Note: The software will detect all required reference areas first before proceeding to any of the mentioned cases.

Case 1: Mirror in front

i) Approach the mirror then retreat to the original position, noting whether or not the MRP grows/shrinks and if any of the ARBs enter/exit the field of view.
ii) Note the robot's current position. Keeping the camera "head" facing the mirror, move the robot to the left until more than 50% of the

MRP is no longer visible and mark this new location as point A.

iii) Move to the right until more than 50% of the MRP is no longer visible and mark this new location as point B.

iv) Use the formula $d = st$ to calculate the distance between A and B to get the length of the mirror.

Case 2: Mirror to the left/right side

i) Scan the area by moving the smartphone 90 degrees to the left, then 180 degrees to the right. If at least 50% of the MRP is visible, it should move at the same rate as the smartphone stand's rotation. Mark this new location as point A.

ii) Any original ARB located should have a matching ARB flipped about the y-axis in the assumed location of the mirror that grows, shrinks, or moves at the same rate as the original ARB.

iii) Move the robot forward and scan from left to right until more than 50% of the MRP is no longer visible and mark this new location as point B.

iv) Use the formula $d = st$ to calculate the distance between A and B to get the length of the mirror.

Case 3: Mirror to the left/right side with no visible MRP

a) Scan the area by moving the smartphone 90 degrees to the left, then 180 degrees to the right. Move forward until at least 50% of the MRP is visible during the left-right scan. It should then move at the same rate as the smartphone stand's rotation. Mark this new location as point A.

b) Any original ARB located should have a matching ARB flipped about the y-axis in the assumed location of the mirror that grows, shrinks, or moves at the same rate as the original ARB.

c) Move the robot forward and scan from left to right until more than 50% of the MRP is no longer visible and mark this new location as point B.

d) Use the formula $d = st$ to calculate the distance between A and B to get the length of the mirror.

The mirror detection will be carried out with the help of the update step in the Extended Kalman filter.

D. Final assembly

Install the SLAM framework with the updated mirror detection procedure on the Android smartphone. Attach the smartphone to the robot body.

E. Testing and evaluation

The study will test the mirror detection process by allowing the robot to navigate and map a maze-like area with five (5) different-sized mirrors placed on different walls as well as other miscellaneous objects. The robot will navigate this space five (5) times, with the positions of the mirrors and the objects changing each time.

ACKNOWLEDGMENT

Many thanks to my family and friends for their support, and to Sir Jimmy for his patience and advice.

REFERENCES

[1] J. Wang and X. Wang, "Detecting glass in simultaneous localisation and mapping," *Robots and Autonomous Systems*, vol. 8, pp. 97–103, Nov. 2017.

[2] S. Yang and C. Wang, "Dealing with laser scanner failure: Mirrors and windows," in *IEEE International Conference on Robotics and Automation, ICRA 2008*, 2008.

[3] R. Koch, S. May, P. Koch, M. Kuhn, and A. Nuchter, "Detection of specular reflections in range measurements for faultless robot slam," presented at the Robot 2015: Second Iberian Robotics Conference, 2015.

[4] X. Yang, H. Mei, K. Xu, X. Wei, and R. Lau, "Where is my mirror?" in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8809–8818.

[5] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, "Deep learning vs. traditional computer vision," presented at the Computer Vision Conference (CVC) 2019, 2019.

[6] P. Agrawal, J. Carriera, and J. Malik, "Learning to see by moving," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 37–45.

[7] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard, "Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age," vol. 32, no. 6, p. 1309–1332, 2016.

[8] C. Croucher, M. Bertamini, and H. Hecht, "Naive optics: Understanding the geometry of mirror reflections," *Journal of experimental psychology: Human perception and performance*, vol. 28, pp. 546–562, July 2002.

[9] S. Ling, J. Sanny, and W. Moebs, *University Physics Volume 3*. Houston, Texas: OpenStax, 2016.

[10] J. Cutting, *Perception with an Eye for Motion*. Cambridge, Massachusetts: The MIT Press, 1986.

[11] J. Gibson, "The perception of the visual world," https://ia800702.us.archive.org/35/items/perceptionofvisu00jame/perceptionofvisu00jame.pdf, 1950.

[12] X. Gao, T. Zhang, Y. Liu, and Q. Yan, "14 lectures on visual slam from theory to practice," Publishing House of Electronics Industry, 2017, retrieved from https://github.com/gaoxiang12/slambook-en.

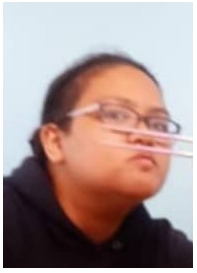
[13] M. Aqel, M. Marhaban, M. Saripan, and N. Ismail, "Review of visual odometry: types, approaches, challenges, and applications," SpringerPlus, October 2016, retrieved from https://springerplus.springeropen.com/track/pdf/10.1186/s40064-016-3573-7.pdf.

[14] N. Hordri, S. Yuhaniz, and S. Shamsuddin, "Deep learning and its applications a review," presented at the Postgraduate Annual Research on Informatics Seminar 2016, 2016.

[15] G. Marcus, "Deep learning: A critical appraisal," https://arxiv.org/abs/1801.00631v1, Jan. 2018.

[16] M. Müller and V. Koltun, "Openbot: Turning smartphones into robots," arXiv:2008.10631v2, August 2020, retrieved from https://arxiv.org/abs/2008.10631.

[17] A. Davison, I. reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," vol. 28, no. 6, 2007.



Julianne Marie Ruz
She is an undergraduate student at UPLB Institute of Computer Science. She likes trying to make robots and dissecting horror movies.