Vision based SLAM for Humanoid Robots: A Survey

Walaa Gouda1, Walid Gomaa2, Tetsuji Ogawa3,4

1Dept. of Computer Science and Engineering, Egypt-Japan University for Science and Technology University, Egypt, Alexandria, walaa.gouda@ejust.edu.eg
2Dept. of Computer Science and Engineering, Egypt-Japan University for Science and Technology University, Egypt, Alexandria, walid.gomaa@ejust.edu.eg
3Dept. of Computer Science and Engineering, Waseda University, Japan, Tokyo, ogawa@pcl.cs.waseda.ac.jp
4Dept. of Computer Science and Engineering, Egypt-Japan University for Science and Technology University, Egypt, Alexandria

Abstract— This paper is a survey work for designing a Vision based Simultaneous Localization and Mapping (VSLAM) humanoid robot to generate a map of an unknown environment. A lot of factors have to be considered while designing a VSLAM robot. Vision Sensors are very attractive for application in SLAM because of their rich sensory output and cost effectiveness. Different issues are involved in the problem of vision based SLAM and many different approaches exist in order to solve these issues. Similarly the type of environment determines the suitable feature extraction method. The main objective of this survey is to conduct a comparative study among the current vision sensing methods in terms of imaging systems used for performing VSLAM, feature extraction algorithms used in some recently published papers, and initialization of landmarks, and to figure out the best for our work.

Keywords: VSLAM, SLAM, feature extraction, landmarks

I. INTRODUCTION

In the last years a lot of researchers have spent a great effort in developing new families of algorithms, using several sensors and robotic platforms [1]. Simultaneous Localization and Mapping (SLAM) or Concurrent Mapping and Localization (CML) is a very classical problem in robotics, and a lot of solutions have been proposed. Even though the robotic field has achieved tremendous progress, modeling of environments using SLAM is still being a challenging problem [2], [3].

SLAM is the problem where the robot is being able to build a map of an unknown environment and simultaneously localize itself in it. In robotics, there are many solutions introduced to provide a robot with a robust navigation. Most of them use metric sensors such as laser range finders, sonars or 3D sensors, and some also use visual information from one or more cameras, to build either a metrical or a topological map and make the robot navigate in it [4].

Classically, laser and sonar sensors have been used for performing SLAM but during the last decade a considerable amount of research has been carried out on SLAM using vision sensors. The usage of laser sensor only may exhibit a number of important limitations, as it is difficult provide an appropriate representation of the map, the number of laser points is large and the computation cost is high and the data association is not easy especially when the loop is closed. On the other hand, cameras provide rich information about the environment enabling the detection of stable features. Furthermore vision Sensors and Cameras are low cost, light and compact, easily available, offer passive sensing, have lower power consumption and provide rich information about the environment enabling the detection of stable features. All of these features make cameras very attractive to be used for SLAM [3], [5].

However, these solutions are mostly designed for standard wheeled platforms with rich sensor information: an array of sonars, wide angle cameras, etc. Humanoid robots are very specific: due to their additional complexity, they cannot carry as many sensors as the others, and have specific problems due to their way of moving. The walk of humanoid robots poses interesting problems for SLAM, since their motion model has a much wider spread than wheeled robots. Since the walk is one of the most attractive features in a humanoid robot, it is necessary to find solutions to have a reliable way of navigating. Furthermore, the robots have many degrees of freedom, which means that estimating the pose of the robot, which is necessary to measure the angle and distances in the environment, is more
complex, and the resulting blurred image due to motion [6], [7], [8].

Different approaches exist in the field of VSLAM in order to solve different steps involved in performing SLAM. For instance, many different types of imaging systems can be used to carry out visual SLAM including single cameras, stereo camera pairs, and multiple camera rigs. Similarly different types of features can be extracted from the environment ranging from point features to edge and planar features and different approaches exist to find the correspondences between the extracted features. These multiple possibilities for solving different issues in the problem of vision based SLAM has their own pros and cons, and therefore, a technique can be more suitable for a specific applications rather than others [3].

To the best of our knowledge, only few researches have worked on VSLAM with humanoid robots. For example [7] tried and implemented a SLAM algorithm on a humanoid robot platform, the NAO robot produced by Aldebaran Robotics. They first started by testing a visual SLAM algorithm which uses keypoints as visual landmarks and tried to estimate their positions. The keypoints are adapted to the specific constraints of the platform: restricted CPU, monocular camera, low speed and drifting odometry. Their results showed that vision can be efficiently used to improve NAO’s navigation.

The research performed by [10] presented an integrated approach for robot localization, obstacle mapping, and path planning in 3D environments. All are based on the data received from an onboard consumer level depth camera. They evaluated their system with a NAO humanoid equipped with an Asus Xtion Pro Live depth camera on top of the humanoid’s head. They did navigation experiments in a multi-level environment containing static and non-static obstacles. Their approach is performed in real-time; the results demonstrated that depth camera is well-suited for robust localization and reliable obstacle avoidance in complex indoor environments.

In [11] an approach for footstep planning and collision avoidance in 3D environments is proposed. They illustrated experimentally an original real-time replanning scheme and architecture for reactive walking; the experiments are performed using HRP-2 humanoid robot. Based on a dense set of actions, their approach used a large panel of the humanoid robot capabilities and was particularly well suited for 3D collision avoidance. Their method relies on RRT combined with an approximation of the volume swept by the robot legs while walking. Their method was able to cope with 3D obstacles while maintaining real-time computation.

This study gives current state of the art on visual SLAM for humanoid robot and is organized as follows: section 2 surveys different imaging systems used for VSLAM. Section 3 presents a survey on feature extraction methods and the extracted features from the environment for VSLAM. Section 4 presents different types of map representation. The discussion and conclusions of this study is introduced in section 5.

II. VISUAL SENSING METHODS

This section surveys state of the art on vision based SLAM techniques. Different imaging systems have been used for performing VSLAM including single cameras, stereo pairs, and multiple camera rigs. The advantages and disadvantages of each of these imaging systems are discussed below.

A. Single Camera (Monocular)

Single Camera SLAM provides only information about the direction of the features exist in the robot’s environment, while as it doesn’t provide any information regarding the depth. To get the 3D location of a feature, multiple images from different viewpoints are required. Wide angle cameras (above 90° field of view) have also been used for visual SLAM as they enable the tracking of the features over wider motion ranges. Most of the single camera SLAM implementations mentioned are for indoor environments [2], [3].

The use of vision alone also means that the sensing range of the robots is severely limited, since they can only recognize obstacles in direct line of sight. However, monocular visual SLAM is still an active field of research. Satisfactory monocular visual SLAM solutions can have a big impact on many application domains, since a single camera is cheaper than a stereo rig; they are not so sensitive to calibration parameters as stereo [5].

B. Stereo Pair

A stereo can provide 3D location of the observed features in the environment; this makes stereo pair readily usable for Visual SLAM. However the feature matching problem in case of stereo is slightly more complicated than in the case of single camera. This is because in the case of stereo pair, the features have to be matched between the two images from the stereo
pair, and then between consecutive acquisitions in time [2], [3].

C. Multiple Camera Rigs

Multiple camera rigs have also been used for visual SLAM. One advantage is that the use of multiple cameras increases the field of view and enables features tracking over wider robot motion. Another advantage is that the spatial resolution over the field of view of a multiple camera rig is uniform. Multiple camera rigs can also provide better constraints for reconstruction of the environment compared to single cameras or stereo pairs. One disadvantage of using multiple camera rigs is the high computational cost [2], [3].

III. FEATURE EXTRACTION METHODS AND ENVIRONMENT’S FEATURES

A. Feature Extraction Methods

Feature-based SLAM robots make use of feature points in the scene video to track the relative motion of the robot in the environment. Different feature extraction methods can be used to extract features for a SLAM problem. The main objective of any feature extraction problem is to get features with maximum information. The suitable features detection algorithm will be different for different environments. Here our goal is to find the best suitable feature detection algorithm for being applicable to humanoid robots.

A wide variety of scale and rotation invariant feature extraction methods have already been proposed for finding correspondences between images. Harris corner detector, The Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) methods are the most popular ones. The main strength of SIFT and SURF is that they produce a feature descriptor that allows quick comparisons with other features and is rich enough to allow these comparisons to be highly discriminatory. The robust data association between their features is particularly effective.

a) Scale-Invariant Feature Transform (SIFT)

It detects distinctive key points in images and computes a descriptor for them. SIFT features are located at maxima and minima of a difference of Gaussian functions applied in scale space [12]. SIFT features are invariant to image translation, scaling and rotation, and are not sensitive to illumination changes and affine/perspective projection. Moreover, the local and multi-scale nature of SIFT features makes them insensitive to noise, clutter, and occlusion, yet the detailed local image properties enable highly selective matching. These characteristics make them ideal features for robust SLAM, as landmarks will usually be re-observed from different viewpoints and under different lighting conditions [13].

b) Speeded Up Robust Features (SURF)

It is a scale and rotation invariant descriptor for detecting features from image. The detection process is based on the Hessian matrix. SURF descriptors are based on sums of 2D Haar wavelet responses, calculated in a 4x4 subregion around each interest point. It detects region features from an image and obtains the location and the descriptor vector of each interest point [12].

c) Harris corner

It is probably the most widely interesting point detector used due to its strong invariance to scale, rotation and illumination variations, as well as image noise [14]. The basic idea behind this algorithm is to evaluate the derivative of the intensity with respect to the location. The edges are then detected where the derivative gets very large. To cover changes of intensity in each direction, the Harris Corner calculates the derivative in the x-direction and in the y-direction [14].

B. Features from the environment

In order to carry out SLAM using vision, features from the environment have to be extracted that can be used as landmarks for SLAM. These features have to be stable and observable from different viewpoints and angles. Many types of environment features have been used for VSLAM. There are also some SLAM approaches which do not extract any specific features from the environment. Pros and cons of some important feature types that can be extracted from the environment in order to perform SLAM are given below, along with a description of featureless SLAM approaches.

a) Point features: Point features are the most commonly used features for visual SLAM. Harris corner detector has widely been used in the recent years for interest point detection. Vision sensors mostly use points as geometric primitive, allowing to easily characterizing indoor or outdoor environments [4], [5].
b) Line/Edge Features: Line/Edge features exist in structured environments. This type of features is more useful for mapping than point features as they also provide some geometrical information about the environment. Moreover, these features are invariant to lighting and significant viewpoint changes. They are mostly used to characterize indoor and structured environments. For robots with laser range finders it is a normal choice because they provide a relatively dense representation of these environments [4], [5], [15].

c) Planes: Planes are generally used to represent complex structures as they provide 3D representation of the environment [4].

d) Featureless Approaches: Feature extracting methods are designed to extract salient areas from an image. There are different features, which can be extracted. Edges, corners and blobs are the most often used features. Each algorithm looks for one or more of these features [3].

IV. MAP REPRESENTATION

In order to solve the SLAM problem, some modeling of the environment is required. This is called the mapping problem. This problem has been extensively studied in the robotics community and many methods of constructing maps have been proposed [4].

The main representations of maps are classified into three categories:

- Metric Maps: Metric maps represent geometric properties of the environment in the Euclidean world.
- Topological Maps: Topological maps are usually represented as graphs that describe the connectivity between locations.
- Hybrid Maps: Based in mixed methods with topological, metric and probabilistic characteristics.

A. Metric Maps

In a metric map, the environment is represented by a set of objects; its positions are associated to a metric space. These types of maps are often related to sensors that can provide a measure of distance between the robot and the objects in the environment. One characteristic of metric maps is the use of information from exteroceptive and proprioceptive sensors. The data provided by proprioceptive sensors is used to estimate the position of the robot defining the metric space, while the data provided by exteroceptive sensors allow the detection of objects in the environment and the estimation of its relative position to the robot using the metric model of the sensors. However, there are always problems related to sensor inaccuracies, and the metric model of the sensors is not always available or easy to obtain, which is a weak point when using metric maps [4].

B. Topological Maps

Topological maps represent the environment without the use of any metric information. They are generally represented as a graph with nodes or vertices and edges which in turn are used to represent the topological properties of places as neighborhood, inclusion, connectivity and order. The nodes characterize particular places, the positions that the robot can reach, and the edges between nodes define the pathways that allow the robot to move from one place to another and to memorize how to perform this displacement. The displacement between two non-adjacent places is determined by a sequence of transitions between the intermediate nodes [4].

One of the advantages of using topological maps is that it is possible to use standard graph algorithms for high-level planning operations such as finding the shortest path between non-adjacent nodes. On the other hand, the principal weakness of topological maps is to ensure reliable navigation between places without the aid of some form of metric measure. Consequently, topological maps show good performance for small and static environments, but give a false positive for dynamic or more complex and large environments [4].

V. DISCUSSION AND CONCLUSIONS

The odometry of the robot is often not precise, so we can’t rely directly on it. Range or vision sensing of the environment can be used to correct the position of the robot. Stereo cameras use sensor for getting dense 3D information. It uses two similar cameras to capture the same scene, with a small inter camera distance. But accuracy of this camera depends on the illumination and it definitely suffers from brightness constancy problem. Kinect is the latest trend in 3D scene capture for small ranges. It uses a RGB camera and an IR depth camera together and combines the output to get the 3D point cloud of the scene. It gives highly accurate dense 3D point cloud in the range of 1 to 10 meters from it. It is cheaper than stereo camera and doesn’t suffer from brightness constancy problem.
Visual SLAM with a single camera is more challenging than using stereo vision, where the 3D geometry of the world can be recovered more easily. In contrast, in monocular visual SLAM the 3D geometry must be inferred from multiple view images. This is because it is not possible to recover the absolute scale of a 3D point due to the inherent observability problems in recovering 3D information from 2D projections using a single camera as the only sensor [3].

Another problem is that some of the images taken while the robot is walking can be very blurred. To avoid having to stop the robot too frequently, several approaches have been proposed. A first solution is to take images only when the two feet of the robot are on the ground, hence reducing the need for the head movement due to the walk. This latter head movement reduces the frame rate significantly. Another solution is to take images at a higher frame rate, and determine on the fly whether the image is blurred or not before starting the heavier processing. A light weight solution is to compute an indicator of the image sharpness by computing the maximum value of the Laplacian of the image: the higher the value, the sharper the image. By comparing the reference and current sharpness, it is possible to eliminate the most blurry images [7].

The principal problem concerning feature maps is their suitability only to environments where the observed objects can be represented by basic geometric feature models. This is not the case for unstructured environments where the objects might appear as curves rather than distinct point or lines. An alternative to construct feature maps of unstructured environments is to parameterize feature models that depict the observed objects well enough to correctly extract the features [3]. Metric maps have many advantages over topological maps. They can robustly map large scale environments, while topological maps have difficulties to construct and maintain in large scale environments when sensor’s information is ambiguous. This permits accurate and continuous estimation of the position of the robot in the environment.

Harris corner detection, SURF and SIFT are the common feature extraction algorithms in SLAM. Harris corner detector extracts the most informative points in the scene corners. These features are more effective in structured environments or in environments where there are enough corner points. In an unstructured environment it isn’t easy to expect a productive number of such feature points. SIFT feature can be effective in such environments. SIFT is a blob detector, treats blobs in a scene as features rather than corners.

In conclusion, implementing a visual SLAM on any humanoid robot is extremely hard especially when the only usable sensor is a monocular camera, which is subject to a strong motion blur, also the available CPU is restricted, which rules out heavy approaches such as Extended Kalman filter, and the odometry is not reliable due to feet slipping.

**ACKNOWLEDGEMENTS**

This research was supported by the Ministry of Higher Education (MoHE) of Egypt through Ph.D. fellowships. Our sincere thanks to EJUST University for guidance and support.

**REFERENCES**


