2018 15th Conference on Computer and Robot Vision

Indoor Localization in Dynamic Human Environments using Visual Odometry and Global Pose Refinement

Raghavender Sahdev, Bao Xin Chen, and John K. Tsotsos
Department of Electrical Engineering and Computer Science and Centre for Vision Research
York University
Toronto Canada
{*sahdev, baoxchen, tsotsos*}*@cse.yorku.ca*

***Abstract*—Indoor Localization is a primary task for social robots. We are particularly interested in how to solve this problem for a mobile robot using primarily vision sensors. This work examines a critical issue related to generalizing approaches for static environments to dynamic ones: (i) it considers how to deal with dynamic users in the environment that obscure landmarks that are key to safe navigation, and (ii) it considers how standard localization approaches for static environments can be augmented to deal with dynamic agents (e.g., humans). We propose an approach which integrates wheel odometry with stereo visual odometry and perform a global pose refinement to overcome previously accumulated errors due to visual and wheel odometry. We evaluate our approach through a series of controlled experiments to see how localization performance varies with increasing number of dynamic agents present in the scene.**

***Keywords*-localization; visual odometry; wheel odometry; humans; dynamic environments; robot pose;**

(c) (d)

Figure 1. Different situations our approach can localize in (a,b) crowded corridor with 3-4 people; (c) robot moving in a texture-less corridor (d) camera view occluded

I. Introduction

Robotics finds application in a range of different fields. A key issue in many potential application areas is the need for the robot to operate within an environment that is populated by other users (people) who execute independent motions thus complicating sensing and planning tasks. To take but one example, imagine the deployment of an autonomous robot in a hospital environment. Such a robot would have to be able to localize itself in hospital corridors with dynamic agents like moving people, beds, and other dynamic events that take place in the corridors. Although there are many computational tasks required of such a robot, one enabling capability is having the robot to be able to know its current pose in this environment, and it is this problem that is central to this work. Localization of a robot in static environments with a known map is much easier [1], than when the map is unknown and the environment is not static. Basic localization approaches for known and unknown static environments can be found in most texts on robots (e.g., [2], [3]) and for properly conditioned robots and sensors this problem can be considered solved. This work addresses a more complex version of the problem of localization of a mobile robot in a dynamic environment with a known 2D occupancy map with dynamic obstacles with unknown trajectories. The map has certain interest points which serve as important landmarks at which a global refinement is performed. Due to this refinement we overcome any previously accumulated errors introduced by visual or wheel odometry. The paper has detailed empirical analysis which show how the performance of localization varies with the different number of dynamic agents present in the scene. For this work, we make use of a standard RGBD sensor (a stereo camera) for environmental sensing and a commercial robot base (Pioneer 3AT).

In this work, we are able to localize the robot with high accuracy in challenging situations (see Figure 1) like partial or complete occlusions of the camera view, significant number of dynamic agents present in the scene, robot navigating in a texture-less corridor, robot facing blank walls, etc. A map of the environment is assumed to be known apriori. The map could be a 2D occupancy map or a floor plan of the world in which the the robot operates. As opposed to loop closure techniques for pose refinement where one needs to have visited the place in advance to perform a refinement, in our approach, we do not need to have visited the place before. The major contributions of this work are: (i) an approach which can act as a wrapper for traditional localization approaches to handle challenging

978-1-5386-6481-0/18/$31.00 ©2018 IEEE
DOI 10.1109/CRV.2018.00057

dynamic situations, (ii) empirical analysis of our approach to see how visual odometry behaves as number of dynamic agents are increased, (iii) a dataset in which the number of dynamic agents vary which can be used by others to validate their alternative approaches.

The paper is structured as follows. In section II, existing literature is presented for various approaches to localization in both static and dynamic environments. Section III presents the proposed approach. In section IV, we provide detailed empirical results for our work. Section V provides the conclusion and possible future work.

II. Relevant Work

Localization of mobile robots is the ability of the robot to know its pose at any given time instance. Essentially this requires the robot to answer the question, *"Where am I?"*. To answer this question, the robot may rely on a variety of sensors, techniques such as wheel odometry using shaft encoders [4], laser odometry using Lidars [1], inertial navigation systems using gyroscopes and accelerometers [5], visual odometry using cameras [6], global positioning systems [7] and Sonar / Ultrasonic sensors [8]. Each of these approaches have their own strengths and weaknesses. For instance lasers provide long range depth information but provide no visual context, cameras provide visual information about context but not long range depth, GPS does not work in indoor environments or its signal might degrade in city environments. Often people rely on techniques known as sensor fusion to leverage data from multiple sensors and provide an accurate estimate about the pose of the robot. One of the current state of the art techniques for localization is based on a sensor fusion approach using data from a 3D laser and monocular camera by Zhang and Singh [1]. Another interesting sensor fusion based localization technique is that of Tsotsos et al. [9] where they used data from an IMU and monocular camera and performed better than the current state of the art systems.

In this section, we focus primarily on localization using visual sensors. The process of estimating ego-motion (translation and orientation of an agent, e.g., vehicle, human, and robot) by using only the input of single or multiple cameras is called Visual Odometry (VO) [10]. The work of Aqel et al. [6] provides an overview on the different techniques for addressing localization using visual odometry. Some early works include that of Matthies and Shafer [11] in 1987, Nister et al. [12] in 2004 and Howard [13] in 2008. These earlier works form the basis of most approaches of visual odometry today. Most VO approaches today try to optimize these approaches in an efficient manner to produce optimal results. Kitt et al. [14] used an iterated sigma point Kalman Filter together with a RANSAC-based outlier rejection approach to estimate ego motion of the vehicle. Feature based approaches have been used by NASA on the Mars rovers in Maimone et al. [15]. In 2007, Klein and Murray [16] presented a SLAM approach known as PTAM (Parallel Tracking and Mapping) to create a map of the scene and in parallel estimating the pose of the camera. Following the approach of PTAM, Pire et al. [17] proposed S-PTAM in which they overcame the limitations of the PTAM approach.

Cvišić and Petrović [18] proposed a visual odometry technique SOFT to estimate vehicle pose. They extract features in an intelligent manner by selection based on its age, strength, initial descriptor, refined current position in image, etc. and track the reliable features. They also use a 3 point RANSAC scheme fused with IMU Measurements to further refine the pose. Geiger et al. [19] proposed the libviso SLAM algorithm to compute the pose of the robot and construct 3D maps from high resolution stereo images in real time. Their approach runs successfully on a CPU at 25 fps for the location part. We build on top of this localization approach in this work by making modifications to their VO approach and integrating wheel odometry and a global pose refinement based on floor plans in our approach. Their approach is able to handle sparsely populated dynamic scenes well. Localization has also been addressed using Place Recognition based techniques as in [20] and [21].

Recently in 2017, Zhu [22] proposed an approach GDVO for visual odometry using a stereo camera. They extract features in the gradient domain which makes their system robust to illumination changes. Some other interesting monocular localization approaches include ORB-SLAM [23] and LSD-SLAM [24]. However, all these approaches either work in sparsely dynamic scenes or for the most part static environments. Addressing localization in highly dynamic environments still remains an open research area.

Pink et al. [25] proposed an approach to estimate the pose of vehicle by visually matching local features with a global feature map obtained from geo-referenced aerial imagery. They matched lane markings in the global map to local lane markings to estimate ego-motion of the vehicle. Chu et al. [26] used a similar concept to estimate the pose of the vehicle by using GPS measurements and a 2D city plan. Our work uses a similar refinement stage to refine poses obtained from Visual Odometry which we correct using information from an indoor map. Chu et al. [27] used Indoor floor plans to address localization. They do matching of video streams to estimate the pose of the camera. They do piece-wise point cloud and free space matching to align the geometric structure with the given floor plan. Their localization technique is similar to that of a particle filter based localization approach [28] where initially all poses are equally likely and gradually weaker particles die out and soon the pose can be estimated with high accuracy.

In 2002, Wang and Thorpe [29] introduced the concept of detection and tracking of moving objects in SLAM. They used laser scans obtained from the objects to segment out moving objects. Yang and Wang [30] estimated ego motion of the vehicle in highly dynamic environments using laser

information. They were able to addresses the pose estimation problem even when more than 50% of the scene was covered with dynamic agents. In 2016, Sun et al. [31] proposed a localization technique for dynamic environments. Their approach was based on a Bayesian estimation process and used laser data and odometry information.They did not provide in a clear way information about the quality of the dataset they tested nor about the dynamic nature of the environment. However, these approaches use a laser scanner which may not be permitted in places like hospitals, schools, etc. Finally, laser scanners are more costly than stereo cameras, and since our target application area is institutional, these latter points are relevant for our sensor choice.

III. Our Approach

Now, we describe our proposed localization approach. We enable the robot to maintain an estimate of its pose as it moves in the presence of dynamic obstacles. Dynamic obstacles do not provide any useful information to the robot in terms of localization. Worse, their presence can degrade the quality of localization of the robot as they may obscure some of the visual landmarks required for the localization of the robot. The robot needs to find a way to make use of its wheel odometry and the information it perceives from the stereo camera about its environment to accurately localize itself in the map in the presence of these potentially intermittent visual landmarks.

Visual sensors are known to be very accurate in static environments, however a detailed analysis of their performance in terms of dynamic environments remains open. We propose to use a combination of information obtained from cameras wherever possible and use wheel odometry whenever the camera's current view is obscured by humans or dynamic objects. Wheel odometry is known to perform with good accuracy for short distances as shown in [32]. This short-term accuracy is leveraged in our approach to integrate with traditional visual odometry approaches. We additionally use a global pose refinement technique to update the pose of the robot with respect to known landmarks in the occupancy map. The input to our approach is a 2D occupancy map/floor plan and a known start point w.r.t. the global map. Mapping is assumed to be known/solved. Now we describe our approach.

A. Interest Point Selection in the Map

In this work, we use a simple form of map known as Occupancy grids [33]. A sample occupancy map we used in our approach can be seen in figure 2. Occupancy maps provide valuable information about the geometric structure of the environment. They are similar to floor plans without the semantic annotations in them. From the given occupancy maps, we mark certain points in the map as interest points. These are the points where a global refinement can be performed to accurately localize the robot in the map. In this work we mark these points manually using the occupancy map. Figure 2 shows these interest points in a sample map.

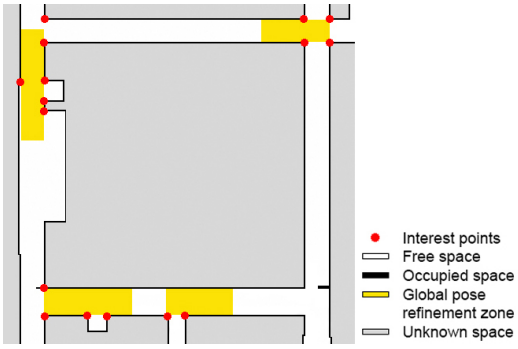

Figure 2. An occupancy grid map for the environment we deploy our robots in. Yellow zone is the global pose refinement zone.

The occupancy map in our approach is generated from a SICK tim 551 2D laser scanner[1] (the scanner is used only for initial map creation and not for any subsequent operation of the robot). A sample gmapping package in ROS[2] is used to create the map. Other open-source algorithms like the Google Cartographer [34] can also be used here. After creating occupancy map, it is cleaned manually to remove any inconsistencies in the map. Now we mark (hand select) interest points where a global refinement is performed during the localization step. Knowing the resolution of the map to be 5 cm for one pixel, we get the coordinates of each of the marked interest points in the map. These interest points serve as candidate landmarks which if detected successfully will improve the quality of localization and remove error accumulation. Similar interest point detection can also be manually done easily using a 2D floor plan of the building.

B. Localization in the presence of Dynamic Obstacles

Our approach is a hybrid approach using wheel encoders, visual odometry and a global pose refinement scheme to overcome previous accumulated errors in visual/wheel odometry. Figure 3 provides a basic overview of our approach. Now, we describe the 3 basic components involved in the Localization phase namely: (i) Visual Odometry by tracking features, (ii) Wheel Odometry using Shaft encoders, and (iii) Global Pose Refinement using Known map. Each of these components are described below:

*1) **Visual Odometry by tracking features**:* The VO component in our approach is same as that of Geiger et al. [19]. Features are extracted and then tracked to estimate ego-motion. In [19], features are matched within a set of 4 images: current left image, current right image, previous left and the previous right image. In order to find stable feature

[1]http://wiki.ros.org/sick_tim
[2]http://wiki.ros.org/gmapping

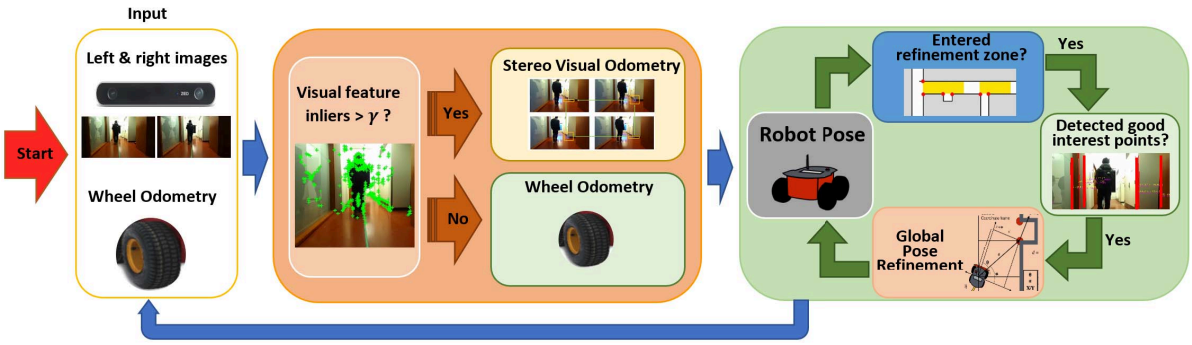

Figure 3. Overview of our proposed localization approach

locations, the input images are initially filtered with 5x5 blob and corner masks. Next non-maximum and non-minimum suppression is applied resulting in features belonging to one of the 4 classes (i.e. blob max, blob min, corner max, corner min). Features are matched only between these 4 classes. Features are matched in a circle to be qualified as a successful match. We extract features from the current left image, match it with the best point in the previous left image within a MxM search window, then in the previous right image, then the current right image and finally in the current left image again. A feature point gets accepted only if the last feature point co-incides with the first one.

A RANSAC based approach is used to estimate the transformation matrix $T = (r, t)$ which is the transformation (rotation, r and translation, t) between two subsequent images. The number of feature matches and the percentage of inliers here play a crucial role. Based on the number of matches and inliers percentage, we use wheel odometry when the inliers percentage is not promising enough.

*2) **Wheel Odometry Integration using shaft encoders**:* From the previous visual odometry component, if the percentage of inliers obtained is less than a threshold, γ, this means that the visual odometry component estimated the r, t matrices with fewer feature matches. This could happen due to lack of sufficiently good static features, tracking a dynamic consistent set of patches from a human, etc. In such cases, we rely on wheel odometry to transiently update the pose of the robot. Cases when visual odometry would not provide us with a sufficient number of feature inliers include when the robot is facing a blank featureless wall, too many moving people in front of the camera, limiting visibility of static content, motion blur, low quality of features detected, etc. In all such circumstances, we estimate and update the motion using wheel odometry. Say the robot at time, t was at position, p and upto time $t+\delta$T visual odometry cannot be relied on. So the motion of the robot during δT is computed using wheel odometry.

Using wheel odometry, we get the pose of the robot at each time instance in the form of position $P(x, y, z)$ and orientation $Q(x, y, z, w)$ in quaternion form. This is converted to a transformation matrix, T (consisting of rotation, r and translation, t) of size 4x4. Say the transformation matrix at time t_1 is given by R_{t1} and at t_2 by R_{t2} so the motion during $t_2 - t_1$ is given by $(R_{t1})^{-1} * R_{t2}$. This motion is then used to update the pose obtained from visual odometry.

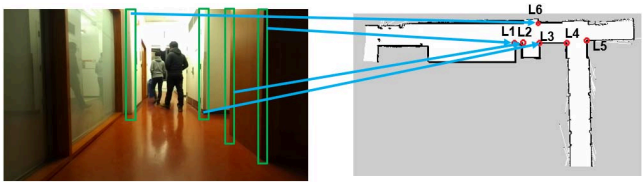

Figure 4. Interest Points detection from camera view and corresponding match in the occupancy map

It should also be noted that a standard inertial measurement unit (IMU) can also be used instead of shaft encoders in wheel odometry. However we did not use it.

*3) **Global Pose Refinement using Known Map**:* This step is used to update pose of the robot whenever the robot is near known landmarks/interest points. Interest points are unique points in the occupancy maps which the robot can use to refine its pose and reduce any previously accumulated errors in the pose estimation process. The global refinement component is only run when the robot's pose obtained from the integration of the visual and wheel odometry is within a predefined range. These ranges of robot poses form zones in which this component is run. An example of refinement zones and interest points can be seen in figures 2, 4.

The interest points are typically at the intersection of two perpendicular walls, but could also be at the intersection of two walls at an angle or a pillar. These interest points in the occupancy map are straight lines perpendicular to the ground when observed from a camera's view. Figure 4 shows a correspondence between interest points on the occupancy map and a camera view. To detect these interest points we need to detect points in the highlighted regions in Figure 4. Now, we explain the process of detection of points on the specific landmarks. As these points lie on vertical lines,

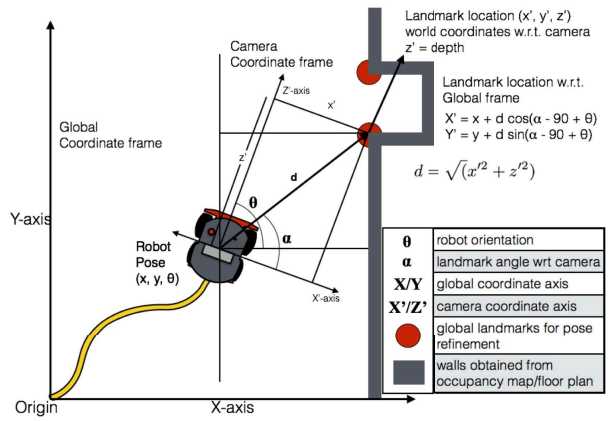

Figure 5. Estimation of the predicted landmark location (Robot pose + World Coordinates of feature point w.r.t. camera)

we need to detect points on these lines. First, we filter the images with an oriented gabor filter at 90 degrees to detect vertical edges/lines. Gabor Filters have been widely used for texture analysis, feature extraction, disparity estimation, etc. These filters are special types of filters which only allow a certain band of frequencies to pass through and reject the others. Now, we get only vertical edges from the image. Next, we employ a Line Segment Detector (LSD) [35] on the filtered image. After doing this, we retain the feature points on vertical edges only. We only detect lines greater than a specific length, at an angle of approx. 90 degrees and between 1 meter and 4 meters due to good depth available in this range from the stereo camera.

Now, we have a set of n interest points, $P = P_1, P_2, ..P_n$ from the image. Each point belongs to a vertical edge. Knowing the depth, focal length and base line, we can compute the world coordinates of each feature point in the camera coordinate system [36]. Knowing the pose of the robot obtained from visual and wheel odometry in the global coordinate frame, we can compute the global coordinates of each point detected as shown in Figure 5. Now, we have a set of n global world coordinates of the interest points P; lets call these transformed points as $W = W_1, W_2, ..W_n$, where $W_i = (x_{w_i}, y_{w_i}, z_{w_i})$. Since its a ground robot (2D case) we only care about the x and y coordinates. For the interest points as shown in Figure 2, say each of these landmarks/interest points, $L = L_1, L_2...L_m$ have world coordinates as $L_j = (x_{l_j}, y_{l_j})$. We know these location of the landmarks as we have the ground truth occupancy map, so we can estimate the absolute values of these landmarks with respect to the start position of the robot. From the set W, we find the closest point, P_i for each of the landmarks, L_j based on the distance metric $\sqrt{(x_{l_i} - x_{w_i})^2 + (y_{l_i} - y_{w_i})^2}$. Now, we have m points which are closest to each of the landmarks. We have the distance error metric for each of these points to landmark assignment. Let the error in distances be $E = (e_1, e_2, ...e_m)$. From the given map, we make a set of pairs of landmarks that are adjacent to each other. Figure 2 shows 16 landmarks and 4 zones, so we make the pairs in each zone, e.g., $(L1, L2); (L2, L3); (L3, L6); (L4, L5)$ as in Figure 4 depending on the distance between 2 landmarks in a particular zone. Now, for each pair (L_i, L_j), we compute the quality of the matched point's distance as (e_i, e_j). If both e_i and e_j are less than an empirically determined threshold, β then we consider that as a good pair and the corresponding matched points as good matches. Now, we update the absolute robot pose based on these two landmarks using triangulation [37]. Doing the update at this stage gets rid of any previously accumulated errors due to wheel and visual odometry. Algorithm 1 formulates this.

Algorithm 1 Pseudocode for Global Pose Refinement

Input :
- Set of n Key Points' world coordinates w.r.t. camera frame, $P_c = \{p_{c_1}, p_{c_2}, ..., p_{c_n}\}$; $p_{c_i} = (x_{p_i}, y_{p_i}, z_{p_i})$
- Set of landmark coordinates, $L = \{L_1, ..., L_m\}$; $L_i = (x_{l_j}, y_{l_j})$
- Pairs of adjacent Landmarks in zone k, $L_{k_{pairs}} = \{(L_1, L_2), (L_2, L_3)...(L_i, L_j)\}$
- Zone number, k
- Empirically determined threshold, β

Output :
Refinement succeeded or not
Refined robot pose, $R : (x_{refined}, y_{refined}, \theta_{refined})$

Procedure 1, Global Pose Refinement :
1. W = $GlobalCoordinatesOfPoints\,(P)$
2. C = $(C_{L1}, C_{L2}, ..C_{Lm})$, set of closest points to landmarks
3. E = $(e_{L1}, e_{L2}, ..e_{Lm})$, errors of closest points to landmarks
4. **for** $L_i \in L$ **do**
5. min = inf
6. **for** $W_j \in W$ **do**
7. e_i = $\sqrt{(x_{l_i} - x_{w_i})^2 + (y_{l_i} - y_{w_i})^2}$
8. **if** $e_i < min$
9. min = e_i
10. C_{Li} = W_j
11. **for** $(L_i, L_j) \in L_{k_{pairs}}$ **do**
12. **if** $e_i < \beta$ & $e_j < \beta$
13. **update pose wrt to** L_i,L_j **using triangulation**
14. **else**
15. **do not update robot pose**
16. **return** $robotpose$

Procedure 2, Global Coordinate of Point (P):
1. **for** $p_{c_i} \in P_c$ **do**
2. W_k = $GlobalCoordinates(P_i)$ using Figure 5
3. **return** $W = W_1, W_2, .., W_n; W_i = (x_{w_i}, y_{w_i}, z_{w_i})$

IV. Empirical System Performance

In this section, we describe our generated dataset and provide a detailed analysis of our results. Our algorithm was deployed on a mobile robot in a real world environment in a university corridor. To validate our proposed approach we developed a dataset for the purposes of localization of mobile robots in dynamic environments. We first describe

our generated dataset and later describe the localization results we obtained. The number of dynamic agents in the scene are varied and an empirical performance analysis is reported.

A. The Dataset

Several datasets exist for computing the localization of a mobile platform equipped with vision sensors. Strum et al. [38] built an RGB-D dataset in indoor environments (industrial hall and office scene) to evaluate visual odometry where they generated ground truth from motion capture systems. Their dataset was built using a handheld Kinect sensor in indoor environments, which for most of the sequences have no presence of humans/dynamic agents or are sparsely populated by one or two people. Smith et al. [39] built a SLAM dataset using a laser, stereo and omni directional cameras in a university environment outdoors. Their dataset was built while the robot was driving several kilometers through a park and university campus. It was built using a segway robot equipped with the sensors like IMU, GPS, stereo, omni-directional, panaromic cameras and Lasers. This dataset also does not have a lot of humans/dynamic agents moving in the environment. One of the most famous benchmarks for ego-motion estimation in outdoor environments (for autonomous driving) is the KITTI dataset [40] which is also sparsely dynamic. As there is not a dataset having a high number of dynamic agents in the scene, we built a new dataset to validate our approach.

Now, we describe our dataset to address the shortcomings of existing datasets. We build a dataset which has many dynamic agents (humans) navigating in the scene in an indoor office-like corridor of size 18m x 18m. Our dataset was built using a mobile ground robot in a university environment. The dataset was created using a Pioneer 3AT robot using on board stereo vision sensors (ZED stereo camera) with wheel odometry. While building this dataset, the robot was driven manually to evaluate the localization framework. Our dataset consists of wheel odometry information obtained from the mobile base, stereo image pairs and a depth image from a ZED stereo camera. The images were captured with a 720p resolution (1024 x 720) RGB stereo camera at a frame rate of 30 fps. The camera was mounted on the robot at a height of 76 cm above the ground plane. Images in the dataset were taken indoors during night time in the winter season (January 2018). We created 5 different types of sequences:

- *Type 1* is the situation of with no dynamic agent present in the scene, only the static scene
- *Type 2* indicates the situation where there is only one person in the environment
- *Type 3* implies presence of one or two people
- *Type 4* implies presence of at most 3 people, and
- *Type 5* implies presence of at most 4 people.

Each situation differs from the other in terms of the number of dynamic agents and pose changes. The data acquisition phase was spread over a week. Each sequence has 6000-8000 images. Some sample sequences from our dataset can be seen in Figure 1. We make the dataset and demo video publicly available for download at the project webpage[3]. The ground truth, map coordinates and interest points coordinates are also available at the project page.

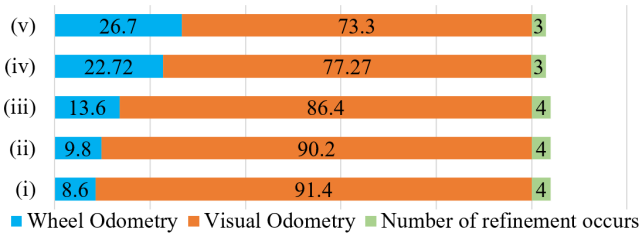

Figure 6. An analysis about the percentage of times wheel odometry is used and when visual odometry can be relied on. Experiments performed in a university corridor. (i): Static environment without any dynamic agents, (ii): Dynamic environment with at most one person in the scene, (iii) Dynamic environment with at most 2 people in the scene, (iv) Dynamic Environment with at most 3 people in the scene, and (v) Scene with at most 4 people present.

B. Results

We validate our approach through a set of controlled experiments to have a quantitative analysis using our dataset. We show how performance varies as the number of dynamic agents present in the scene are changed.

We compute the localization errors of the robot in the presence of dynamic obstacles and compare it to that when the robot moves in the static environment and with the ground truth. We report the performance of our approach on 5 different sequences in our generated dataset. The sequences differ in the number of humans present in the scene. Varying the number of dynamic agents in the scene implies varying the number of dynamic and static visual features present in the environment. As the number of dynamic agents increases the number of static visual features decreases and robot may not be able to trust its vision for estimating its pose, hence in such cases wheel odometry comes to our rescue. Wheel odometry is transiently relied on under such circumstances. On the other hand, with no dynamic agents present in the scene, the dependence of the robot on visual odometry is maximum and wheel odometry is minimally used. Some of the situations where wheel odometry is solely relied on include when a particular person blocks the view of the camera, too many moving agents in front of the camera limiting visibility of static content, robot facing a blank featureless wall/door, motion blur, etc. Figure 6 shows the proportion of times when wheel and visual odometry is relied on under varying dynamic agents. To avoid accumulation of errors, we do a global pose refinement based on landmarks from the 2D map.

[3]http://jtl.lassonde.yorku.ca/2018/03/localization-among-humans/

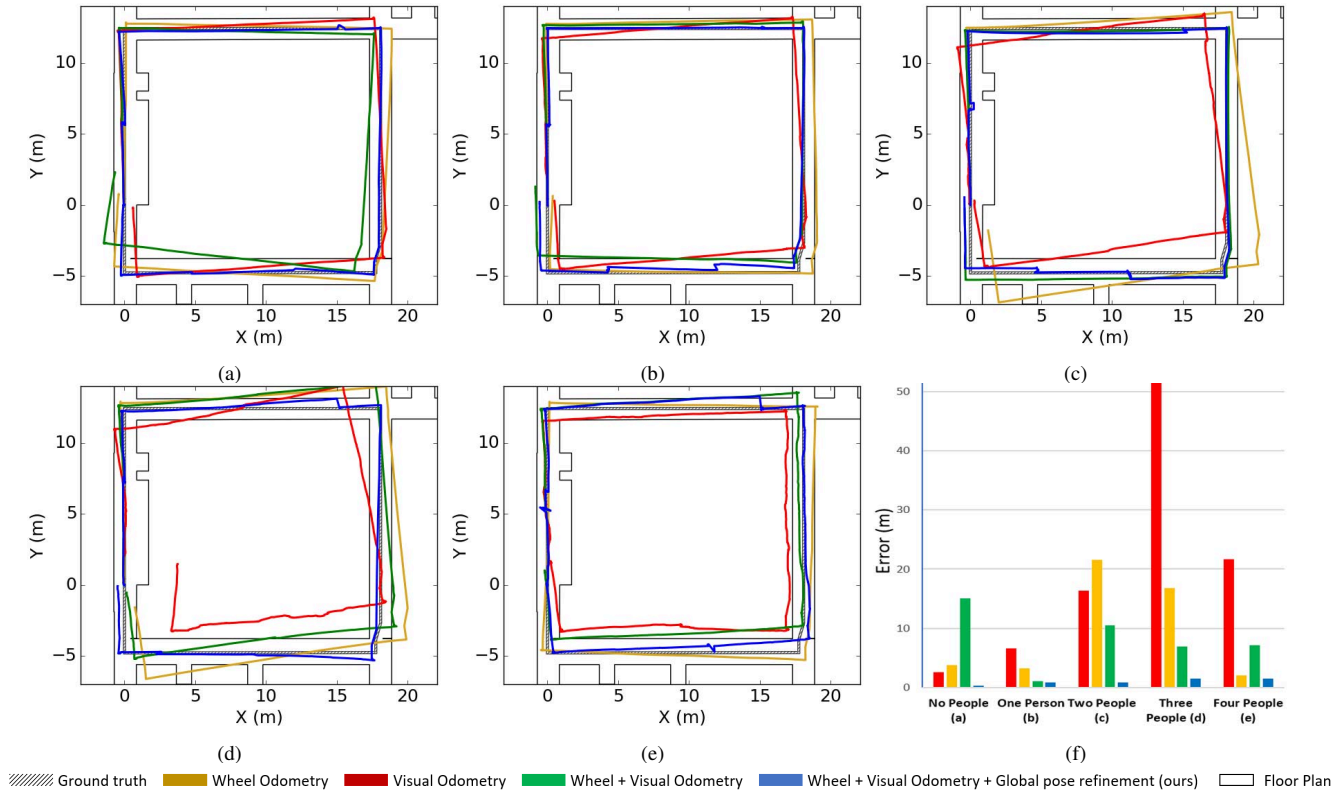

Figure 7. Trajectory of our approach against *(i)* wheel odomery, *(ii)* visual odometry (method proposed in [19]), *(iii)* visual + wheel odometry, *(iv)* visual + wheel odometry + global-refinement, and *(v)* ground truth (a) Type 1 (no people), (b) Type 2 (one person), (c) Type 3 (two people), (d) Type 4 (three people), (e) Type 5 (four people), (f) sum of squared errors of 4 corners and the terminal point of the trajectory with the ground truth. Due to global refinement we correct the pose and remove any accumulation of error due to both wheel and visual odometry which gives us a better trajectory closer to the ground truth. As can be seen as the number of dynamic agents are increased quality of traditional visual odometry approach reduces, however using our approach we maintain a good alignment with the ground truth. The performance of VO with 3 people is worse than with 4 people as in the 3 people sequence, people were closer to the camera more often thereby reducing the percentage of static features

As opposed to traditional loop closure techniques, we do not need to visit the place once to perform a refinement. Knowing the map and a few interest points, the robot knows when to perform a refinement. Our approach runs at 25 fps in real time.

We report the trajectory that the robot takes based on its visual odometry and compare it to the following: (i) Wheel Odometry alone, (ii) Visual Odometry alone, (iii) Wheel+Visual Odometry, (iv) Our Approach (Wheel+Visual+Global-Refinement), (v) Ground Truth. Figure 7 shows the trajectory under each of the approaches, it can be seen that our approach performs better than visual or wheel odometry alone. Ground truth was generated by driving the robot on a predefined path (the coordinates of which were known $\pm 7.5cm$).

V. Conclusion

In this paper, we presented an approach as to how standard localization techniques can be extended to deal with dynamic agents present in the scene. One of the existing localization algorithms was chosen and integrated with our proposed refinements. An empirical analysis was performed to see how the task of localization differs in a static environment to that of a dynamic environment as number of people in the scene are increased. Some of the future works include integrating this approach with a navigation approach to have an autonomous agent navigating among humans. In this work, only one of the current localization approach was built on top of. Our proposed additions to the localization framework can also be applied to other localization techniques and an analysis can be done on the performance of other existing algorithms as to how they perform in a dynamic context after incorporating our integrations. In this work, we showed experimental analysis in a single environment. We plan to validate our approach in different environments in future.

Acknowledgment

The authors would like to thank all the human participants in the dataset. The authors would also like to thank Michael Jenkin and Amir Rasouli for their helpful discussions during this work.

References

[1] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *Robotics and Automation*

(ICRA), 2015 IEEE International Conference on. IEEE, 2015, pp. 2174–2181.

[2] G. Dudek and M. Jenkin, *Computational principles of mobile robotics*. Cambridge university press, 2010.

[3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[4] J. Borenstein, H. Everett, L. Feng *et al.*, "Where am i? sensors and methods for mobile robot positioning," 1996.

[5] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 2008, pp. 114–123.

[6] M. O. Aqel, M. H. Marhaban, M. I. Saripan, and N. B. Ismail, "Review of visual odometry: types, approaches, challenges, and applications," *SpringerPlus*, vol. 5, no. 1, p. 1897, 2016.

[7] A. Noureldin, T. B. Karamat, and J. Georgy, *Fundamentals of inertial navigation, satellite-based positioning and their integration*. Springer Science & Business Media, 2012.

[8] A. Sanchez, A. de Castro, S. Elvira, G. Glez-de Rivera, and J. Garrido, "Autonomous indoor ultrasonic positioning system based on a low-cost conditioning circuit," *Measurement*, vol. 45, no. 3, pp. 276–283, 2012.

[9] K. Tsotsos, A. Chiuso, and S. Soatto, "Robust inference for visual-inertial sensor fusion," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5203–5210.

[10] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE robotics & automation magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[11] L. Matthies and S. Shafer, "Error modeling in stereo navigation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 3, pp. 239–248, 1987.

[12] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. Ieee, pp. I–I.

[13] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 3946–3952.

[14] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*. IEEE, 2010, pp. 486–492.

[15] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.

[16] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*. IEEE, 2007, pp. 225–234.

[17] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, J. Civera, and J. J. Berlles, "S-ptam: Stereo parallel tracking and mapping," *Robotics and Autonomous Systems*, 2017.

[18] I. Cvišić and I. Petrović, "Stereo odometry based on careful feature selection and tracking," in *Mobile Robots (ECMR), 2015 European Conference on*. IEEE, 2015, pp. 1–6.

[19] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. Ieee, 2011, pp. 963–968.

[20] M. Cummins and P. Newman, "Appearance-only slam at large scale with fab-map 2.0," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.

[21] R. Sahdev and J. K. Tsotsos, "Indoor place recognition system for localization of mobile robots," in *Computer and Robot Vision (CRV), 2016 13th Conference on*. IEEE, 2016, pp. 53–60.

[22] J. Zhu, "Image gradient-based joint direct visual odometry for stereo camera," in *International Joint Conference on Artificial Intelligence*, 2017, pp. 4558–4564.

[23] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[24] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[25] O. Pink, F. Moosmann, and A. Bachmann, "Visual features for vehicle localization and ego-motion estimation," in *Intelligent Vehicles Symposium, 2009 IEEE*. IEEE, 2009, pp. 254–260.

[26] H. Chu, A. Gallagher, and T. Chen, "Gps refinement and camera orientation estimation from a single image and a 2d map," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 171–178.

[27] H. Chu, D. Ki Kim, and T. Chen, "You are here: Mimicking the human thinking process in reading floor-plans," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2210–2218.

[28] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 1322–1328.

[29] C.-C. Wang and C. Thorpe, "Simultaneous localization and mapping with detection and tracking of moving objects," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 3. IEEE, 2002, pp. 2918–2924.

[30] S.-W. Yang and C.-C. Wang, "Multiple-model ransac for ego-motion estimation in highly dynamic environments," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 3531–3538.

[31] D. Sun, F. Geißer, and B. Nebel, "Towards effective localization in dynamic environments," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 4517–4523.

[32] J. Borenstein, H. R. Everett, L. Feng, and D. K. Wehe, "Mobile robot positioning: Sensors and techniques," 1997.

[33] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 3, pp. 249–265, 1987.

[34] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1271–1278.

[35] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: a line segment detector," *Image Processing On Line*, vol. 2, pp. 35–55, 2012.

[36] M. Kanbara, T. Okuma, H. Takemura, and N. Yokoya, "Real-time composition of stereo images for video see-through augmented reality," in *Multimedia Computing and Systems, 1999. IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 213–219.

[37] J. M. Font and J. A. Batlle, "Mobile robot localization. revisiting the triangulation methods," *IFAC Proceedings Volumes*, vol. 39, no. 15, pp. 340–345, 2006.

[38] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 573–580.

[39] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The new college vision and laser data set," *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 595–599, 2009.

[40] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.