Indoor Navigation based on Fiducial Markers of Opportunity

Mazhar Ali Lakhani, John Nielsen, Gerard Lachapelle

Position Location and Navigation (PLAN) Group, University of Calgary, 2500 University Dr. NW, Calgary, Alberta, Canada, T2N 1N4 Calgary, Canada
malakhan@ucalgary.ca, nielsenj@ucalgary.ca, Gerard.Lachapelle@ucalgary.ca

Abstract—Computer vision is becoming an important component of facilitating indoor positioning processing as applicable to a smart phone (SP). Typically such processing is in the form of ego-motion or identifying landmarks by correlating images from the SP camera via the inverse transform to pre-stored orthographic view images of the landmark. 3D ego-motion is difficult unless the feature points (fp) of opportunity are known to lie on a common plane. However, a coplanar subgroup of opportunistic fp’s can be found in the form of building features such as windows, doors, wall frames, tiles, and markers that can be assumed to be rectangular and readily useable for estimating perspective transformations. With an assumed structure of the rectangle, a useful set of constraints emerges that facilitates the perspective mapping. In this paper SLAM processing is used, starting from a known marker and moving to observed rectangles of opportunity. The method of implementing the rectangular set of FP’s into the SLAM algorithm is described.

Index Terms—Computer Vision (CV), Simultaneous Localization and Mapping (SLAM), smart phone (SP), fiducial markers (FM), fiducial marker of opportunity (FMO).

1. INTRODUCTION

There is an obvious need to reliably extend the ability of accurately locating a smartphone for indoor environments. Traditional GPS methods which operate well outdoors are not reliable for indoor applications due to the weak signal which does not adequately penetrate indoor environments. As well there is the problem of the rather narrow bandwidth of GPS signaling which results in significant errors in multipath environments [1-2]. This is further contrasted with the higher accuracy demands for indoor positioning which is really only useful if the eventual positioning error is on the order of a meter or less. For this reason other observables that are applicable for facilitating indoor location are being considered such as using MEMS based gyros, accelerometers, magnetometers and most importantly computer vision (CV) based on using video output of the small camera built into every smart phone. Unfortunately CV applied to the general 3D vision processing as required for facilitating indoor location is highly computationally intensive, ill conditioned and complex. Hence simplifying assumptions are necessary to facilitate the required computation on a handset device. One successful simplification is that of implementing 2D ego-motion based on training the camera to look for fp’s of opportunity on the floor or ceiling surface of the indoor environment. 2D ego-motion based on translation only at a reasonably constant height is computationally efficient and robust. It is certainly applicable to robotics where the camera can be held at a fixed height and the tilt of the camera is small. However, it is less applicable for the smart phone (SP) application where the user will invariably tilt the camera. Ego-motion that can accommodate the tilt is significantly more difficult as a full perspective processing is required. This increases the computational effort required as well as undermining the robustness of the location estimate as the processing is ill conditioned. To ameliorate the loss of robustness, it is necessary to use stereoscopic cameras or some form of pattern projection as is implemented into the Microsoft Kinect™. However, the additional hardware required in the smart phone to facilitate this is borderline prohibitive as the SP device form factor is so compact.

As it is desired to facilitate the ego-motion with a single web-cam quality camera it is necessary to implement the overall perspective processing such that the camera tilt issue can be accommodated. A method that has been researched and successfully implemented is that of recognizing templates or markers integrated into the floor surface. When these are observed in the camera’s FOV, the points in the 2D geometry of the marker as observed in the cameras image plane can be related the known marker geometry providing a set of constraints from which the full perspective transformation can be evaluated. Hence if the indoor environment has a set of such markers integrated into the floor surface then a robust computationally efficient means of ego-motion can be realized that is implementable in the SP. Fiducial markers (FM)’s have been extensively used for localizing in both indoor and outdoor environments [3], especially for indoor navigation. For instance, [4] illustrates a method to enable navigation by assigning markers to location and then observing a short sequence of these markers. [5] is an example of using encrypted patterns from which the 3D position information can be decoded. This can be extended to markers on an arbitrary plane surface such that any combination of wall, ceiling or floor surfaces that have some innate geometric structure can be used for such processing.
In this paper we intend to further generalize the marker such that it can be a generic rectangle that is observed as a pseudo marker or template of opportunity. The structure of the rectangular geometric shape provides useful constraints from which the perspective transformation can be directly extracted. Buildings are full of simple geometric shapes with the rectangle being the most prevalent showing up in doorways, windows, wall frames, floor tiles, picture and poster frames and so forth. The photograph of a typical indoor building environment in Figure 1 is illustrative of the large number of rectangles that can be potentially used as coplanar subsets of fp’s of opportunity that are vertices of rectangles. Of course there are geometric shapes that are not rectangles, notably triangles and trapezoidal shapes that can fool the ego-motion algorithm. However, such shapes occur far less frequent than the rectangle. The perspective transformation generated from non-rectangular shapes under the false pretense that they are rectangular will result in large outlier solutions that are identifiable by such algorithms as RANSAC such that they can be removed from the set of observable data.

The positioning method developed in this paper is based on the observation of a mix of isolated 2D markers that are known to the CV location algorithm combined with markers of opportunity that are only recognized by the algorithm to be potential rectangular shape. It is assumed that the known fiducial markers (FM) are rare relative to the plentiful FM’s of opportunity consisting of observed and assumed rectangles. To simplify the development of the underlying SLAM processing, the known FM is initially observed such that the SLAM can extract a reasonably accurate estimate of the perspective transformation matrix. As the SP is moved, the camera image changes and an FM of opportunity (FMO) is in the camera FOV, along with the known FM. Based on the extracted perspective transformation of the first FM, the coordinates of the FMO are evaluated. The assumption of the set of vertex points extracted from the contoured image of the FMO is that it is a basic rectangular shape. This is tested by the constraints of the hypothesized rectangle (i.e. equal length sides and equal length diagonals). The perspective transformation is then transferred to this new FMO as the camera is moved and the original FM leaves the camera FOV. Then the next FMO is observed in the camera FOV, is tested based on the same hypothesized rectangular shape with the perspective transformation updated. As this procedure continues, there will be a random drift in the SLAM estimates as the errors with each new FMO processed will accumulate. To ameliorate this drift issue, it is necessary to assume that there are other observables available for the SP. This could be in the form of a known FM. It could also be that the SP gets an absolute location from a recognizable landmark or perhaps and RFID registration. Alternately it can use wireless signals and perhaps GPS which are inaccurate as pointed out earlier but do not suffer from accumulated trajectory estimation drift. The SLAM algorithm systematically appropriates the weighting of these disparate observables. However, the purpose of this paper is not the details of the SLAM algorithm but rather the perspective mapping of the sequence of known FM’s and FMO’s as input observables to the SLAM algorithm.

2. Methodology

2.1 The Underlying SLAM Algorithm

The underlying algorithm for determining the trajectory of the camera as it is moved by the user through the indoor environment is a modified version of SLAM. The generic SLAM algorithm accommodates disparate observations from a number of available sensor sources contained in the SP. It is assumed here that the SLAM can access to a map outlining the location of the known anchor FM’s. However, these may be thinly distributed and not sufficient for ubiquitous location estimation of the SP. However, they provide justification for an initially known location of the SP. Note that the SLAM algorithm is more general in that it can accommodate arbitrary uncertainty as an initial belief map as to the location of the SP.

The SLAM variables typically consist of the location of observed fp’s that could be known (anchor points) or fp’s of opportunity that initially are of unknown location. In this application the perspective transformation of the FM’s is included in the set of variables tracked by the SLAM algorithm. As the SP is moved, the camera trajectory is mapped out by the SLAM algorithm as an overall joint least squares estimate of the additional FMO location and perspective transformation variables. Occasionally when a FM of known location is observed, the SLAM recalculate the complete list of variables thus correcting for the drift in the estimation of the SP trajectory. As such the SLAM algorithm provides the best representation of the posterior probability of the set of state variables. As there are a large number of non-independent jointly distributed variables with a mix of Gaussian and non-Gaussian variables, it is necessary to use a fastSLAM version of SLAM.

![Figure 1 Typical indoor environment indicating many possible rectangles](image)

The photograph of a typical indoor environment indicating many possible rectangles that can be used as FMO’s. The developed CV routine has preprocessed this image to outline one of these rectangles.
2.2 Algorithm

An illustration of the overall algorithm is shown in Figure 2. Shown in Figure 2a is the camera first sees a known FM in the camera FOV. The perspective transformation is extracted from this geometry and known image of the FM. It is assumed that the SLAM algorithm is initialized in this known state. As the SLAM has a prior map of the known locations of the FM’s, the SLAM begins with a location of the SP with small variance. In Figure 2b, the camera is moved along an unknown trajectory but the FM is contained in the FOV. Hence the initial segment of the SP trajectory is accurately calculated. As shown, an FMO which could be of variable information but not completely known to the SLAM is coincident in the camera FOV. As the SP location at this point is accurate, the perspective matrix of the new FMO can be estimated. The FMO is checked to verify that it indeed is a rectangular shape as described before. Note that the SP is assumed to move slowly relative to the frame rate of the camera such that multiple views of the simultaneous observation of the FM and FMO are obtained. SLAM accumulates the information regarding the state variables from all of these views building up an accurate estimate of the perspective transformation of the FMO as well as its absolute location.

In Figure 2c, the camera has now moved such that only the FMO is in view. However, as the transformation matrix and location of the FMO are estimated, the SP trajectory can continue to be estimated. In Figure 2d, the SP trajectory evolves further to include the first FMO and an additional new FMO in the FOV. SLAM then proceeds to determine the perspective and location of the second FMO based on the first FMO. This process continues indefinitely until an FM of known location is recognized. At this point the overall location trajectory of the SP is recalculated.

Figure 3 illustrates the image of a rectangle FMO that is observed from some arbitrary perspective. Hence, in general it appears as a general quadrilateral. The initial assumption is that the FMO is on the same plane as the current world coordinate plane inferred from SLAM. Based on this assumption, the initial guess at the perspective transformation is set to that of the previous FMO. The constraints of the sides and diagonal as shown in Figure 3 are tested. If they are reasonably satisfied then the FMO is assumed to be a rectangle shape. The perspective matrix is then corrected by SLAM based on the constraint observation. Note that at this point SLAM has to account for the possible change in position and orientation of the camera as well as the uncertainty that the FMO is a true rectangle. It achieves this with statistical constraints and observations applied in the underlying Extended Kalman filter which is used in each particle of the fastSLAM algorithm.
3. Experimental results of Algorithm

Figure 5 shows an example of the FM as it appears in the camera image plane. In this image, the CV algorithm has identified the FM and framed it with the assumed square shape such that the vertices can be extracted. The perspective transformation of the marker is then determined. As the initial map applied to SLAM gives the absolute location of the FM, the absolute location of the camera can be determined.

Figure 6 shows the FM after the perspective transformation is applied as it appears in the world coordinate frame.

Figure 7 shows the experimental setup with the initial FM of known location on the left side of the image with the FMO simultaneously in the FOV on the right. The CV algorithm has identified both figures and has contoured them for accurate estimation of the vertices.
Figure 8 shows the FMO after the inverse perspective transformation has been applied showing the perspective of the rectangle as observed in the camera image coordinates restored back into the world coordinates. (The FM has been further translated which is equivalent to transforming the world reference system to coincide with the corner of the FMO.

![Figure 8](image)

**Figure 8** Figure of opportunity as observed by the camera after performing inverse perspective transformation

Figure 9 shows the position of the camera relative to the defined world reference. The black dots represent the camera location estimation based on the successive perspective transformations that were done based on the known FM only. The red dots represent camera estimation based on the FMO of the unknown marker. These are based on the estimated perspective of the known initial FM. As noted, there is a small offset of several millimeters which is due to the estimate of the perspective transformation being slightly in error. The SLAM algorithm effectively smooths this trajectory as a best fit in the overall least squares sense. This is based on the assumptions of the motion of the camera itself which is generally expressed as a Markov model.

![Figure 9](image)

**Figure 9** Camera moving from right to left as per figure 2 setup. The trajectory in red is calibrated with respect to the initial known FM while the trajectory in black is based on the FMO with initialization based on the first FM.

4. CONCLUSIONS

Accurate and robust indoor positioning is possible with a webcam quality camera that is typically found in a SP. In this paper, the use of a sequence of FM’s and FMO’s has been demonstrated to provide useful and sufficient input to a SLAM algorithm for estimating the location of the camera relative to the position of the FM’s. The FMO’s assumed are generic rectangle shapes that are ubiquitous throughout typical indoor environments. Currently effort is being expended on fitting the CV and SLAM algorithm into an SP with limited processing capability.

5. REFERENCES