A Self-localization System with Global Error Reduction and Online Map-Building Capabilities

Karam Shaya, Aaron Mavrinac, Jose Luis Alarcon Herrera, and Xiang Chen*

Department of Electrical and Computer Engineering, University of Windsor, Windsor, Ontario, Canada
{shayak,mavrin1,alarconj,xchen}@uwindsor.ca

Abstract. An economical self-localization system which uses a monocular camera and a set of artificial landmarks is presented herein. The system represents the surrounding environment as a topological graph where each node corresponds to an artificial landmark and each edge corresponds to a relative pose between two landmarks. The edges are weighted based on an error metric (related to pose uncertainty) and a shortest path algorithm is applied to the map to compute the path corresponding to the least aggregate weight. This path is used to localize the camera with respect to a global coordinate system whose origin lies on an arbitrary reference landmark (i.e., the destination node of the path). The proposed system does not require a preliminary training process, as it builds and updates the map online. Experimental results demonstrate the performance of the system in reducing the global error associated with large-scale localization.

1 Introduction

With the growing demand for autonomous robots in industrial, medical, domestic, and other domains, a large portion of research in the robotics industry has been geared toward the development and improvement of localization systems. For the purposes of this paper, existing localization systems in the literature will be divided into two categories: those that obtain their data from multiple sensors (e.g., [8], [1], [5]) and those that obtain them from a single sensor (e.g., [11], [15], [16]). The former type of system takes a sensor fusion approach. One major advantage of sensor fusion is the availability of multiple sources of data, through which the robot may verify the readings of its individual sensors and reduce the overall error of its pose estimates. The disadvantages of multiple sensors are added complexity (in the localization algorithm and hardware design), larger form factor, and increased cost. Conversely, systems that use single sensors tend to be simpler, smaller, and less expensive; however, they do not have the redundancy and fusion of multiple independent sensor measurements and must therefore use internal methods to reduce estimation error. The focus of our work is on reducing the estimation error of a single sensor localization system.

* This research is supported in part by NSERC-Discovery Grant to Prof. Xiang Chen.

C.-Y. Su, S. Rakheja, H. Liu (Eds.): ICIRA 2012, Part III, LNAI 7508, pp. 13–22, 2012.

Compared to the sensing modalities used in other solutions (e.g., odometry, sonar, and laser), two dimensional images provide a robot's localization algorithm with more data about the environment [9]. They can be used by a localization algorithm to detect, identify, and estimate the pose of objects in a scene.

The nature of the map-building process of visual localization yields two particular types of pose estimation errors: *local error* – which originates from error and noise in image capture and affects pose estimations made with respect to coordinate systems in the image – and *global error* – which arises from the accumulation of local error and affects pose estimations made with respect to a global coordinate system that may not necessarily be in the image. Due to the influence local error has on global error, reducing the former would result in a reduction of the latter. This may be achieved, for example, through markers (i.e., artificial landmarks) that can be accurately detected and discerned to help keep local error at a minimum [7], [4], [10].

In this paper, the problem will be approached from a different perspective. While local error reduction can be characterized as a low-level means of reducing global error, the proposed method will take on a more direct approach by representing the surrounding environment as a mathematical graph whose edge weights reflect the effects of local error. Graphical representation of similar vision-based pose estimation problems has previously been applied to such areas as multiview registration of 3D scenes [14] and large-scale extrinsic calibration of camera networks [3]. By applying a shortest path algorithm, the graph can be optimized to yield the paths of minimum global error (i.e., accumulated local error). To the best of the authors' knowledge, the minimization of global error in a localization system using the graphical approach outlined herein has not been presented in literature.

The remainder of this paper is organized as follows: Section 2 defines the preliminary concepts relating to pose composition, graphs, and the shortest path algorithm; Section 3 states the necessary assumptions and presents the proposed self-localization system's method and algorithm; Section 4 demonstrates the performance of the system in reducing the effect of global error; and Section 5 concludes the paper by providing reflections on the results.

2 Preliminaries

2.1 Pose Composition

A pose $P_{\alpha\beta}$ is a rigid three dimensional Euclidean transformation from the coordinate system of object α to the coordinate system of object β. This may be referred to as the pose of object α with respect to object β.

The inverse of pose $P_{\alpha\beta}$ may be denoted $P_{\alpha\beta}^{-1}$ or $P_{\beta\alpha}$. The former notation will be used here to emphasize that $P_{\alpha\beta}$ is the available direct estimate.

Successive pose transformations may be composed into a single pose:

$$P_\alpha\gamma(\mathbf{p}) = (P_{\alpha\beta} \circ P_{\beta\gamma})(\mathbf{p}) = P_{\beta\gamma}(P_{\alpha\beta}(\mathbf{p})) \tag{1}$$

Note that a left-composition convention is used to better illuminate the sequence of pose transformations.

The details of pose inversion and composition vary depending on the representation used. The reader is directed to any of the numerous texts on Euclidean geometry for a treatment appropriate to his or her working representation.

2.2 Relative Pose Calculation

Refering to Fig. 1, the relative pose transformation of a marker α with respect to another marker β is determined through pose composition to be

$$P_{\alpha\beta} = P_{\alpha c} \circ P_{\beta c}^{-1} \tag{2}$$

where $P_{\alpha c}$ and $P_{\beta c}$ are the poses of α and β with respect to the camera c (assumed to be available). Note that this calculation is made possible by the fact that both markers are in the camera's FOV (shaded area in Fig. 1) at the same time; more specifically, pose estimates are taken from the same captured camera frame.

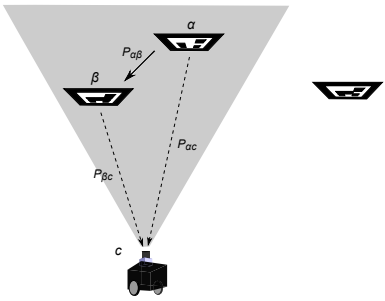

Fig. 1. Relative Pose Calculation - the camera can calculate the relative pose transformation between multiple markers by capturing them concurrently in its FOV

2.3 Marker Graph

The *marker graph* (based on the *calibration graph* introduced by Mavrinac et al. [12]) is a method of representing a set of markers as a topological map. It is a weighted, undirected graph $\mathcal{G}_M = (\mathcal{M}, E_M, \mathcal{W}_M)$, where $\mathcal{M}$ is the set of detected markers in the system, E_M is a set of edges, and $\mathcal{W}_M$ is the set of weights corresponding to the edges in E_M. The existence of an edge $\{\alpha, \beta\} \in E_M$ indicates that a relative pose transformation from marker α to marker β (or vice versa) is available.

Since it is trivial to invert a pose, the availability of $P_{\alpha\beta}$ implies availability of $P_{\beta\alpha}$. The edge weight $(w_{\alpha\beta} \in \mathbb{R}^+) \in \mathcal{W}_M$ is the estimation uncertainty of $P_{\alpha\beta}$.

A path $p = \langle \alpha, \ldots, \beta \rangle$ in $\mathcal{G}_M$, from node α to node β, represents a sequence of pose transformations which may be composed to yield $P_{\alpha\beta}$. If $p = \langle v_1, v_2, \ldots, v_n \rangle$,

$$P_{1,n} = P_{1,2} \circ P_{2,3} \circ \cdots \circ P_{n-1,n} \tag{3}$$

where $P_{i,j}$ is the pose transformation from v_i to v_j. If any $P_{i,j}$ is not available, $P_{i,j} = P_{j,i}^{-1}$. The aggregate error associated with this pose is

$$w_{1,n} = \sum_{2}^{n} w_{k-1,k} \tag{4}$$

which is the length of path p.

2.4 Localization Graph

The *localization graph* is essentially a marker graph that includes the camera c as an additional node. It is a weighted, undirected graph $\mathcal{G}_L = (\mathcal{L}, E_L, \mathcal{W}_L)$, where $\mathcal{L} = c \cup \mathcal{M}$, E_L is the set of edges between nodes $\mathcal{L}$, and $\mathcal{W}_L$ is, again, the set of associated edge weights. The localization graph is incrementally updated as the camera c moves through the environment. Note that $\mathcal{G}_L \supset \mathcal{G}_M$.

2.5 Shortest Path Algorithm

Shortest path algorithms can be applied to graphs to find the minimum topological path between two nodes. The famous shortest path algorithm by Dijkstra [6] solves the single-source shortest path problem of undirected graphs with non-negative edge weights. Since globally localizing a camera involves obtaining its pose with respect to a single source (i.e., the global coordinate system), Dijkstra's algorithm is appropriate for computing the shortest path of edges from said source to the camera. It returns the path yielding the minimum aggregate error defined in (4).

3 Global Localization

3.1 Assumptions

Internal Calibration and Pose Estimation. It is assumed that there exists some means by which a camera may estimate, from a single view, its relative three dimensional pose with respect to a calibration target of known structure [17], [2]. This normally implies that the camera is internally calibrated.

Marker Constraints. It is assumed that (i) if a marker $m \in \mathcal{M}$ is connected to an edge, it remains fixed in its position, and (ii) the selected reference node R corresponds to a marker that is available and detectable in the environment.

Map Updates. It is assumed that any operation that updates $\mathcal{G}_M$ simultaneously updates $\mathcal{G}_L$, and vice versa.

3.2 Problem Definition

The problem of global localization using computer vision is formalized as follows:

> Given a monocular camera c, a set of markers $\mathcal{M}$, and an arbitrary global reference frame $R \in \mathcal{M}$, find P_{cR} as c traverses the environment.

Let us further define a set of markers $\mathcal{V}$ which are in the camera's current FOV (e.g., shaded area in Fig. 1). Then, it can be noted that P_{cR} may either be obtained directly from R (when $R \in \mathcal{V}$) or indirectly from another marker $v \in \mathcal{V}$ (when $R \notin \mathcal{V}$), assuming there exists a path in $\mathcal{G}_L$ from c to R.

It is additionally desirable to decrease the global error by using the path p yielding the minimum aggregate error (as defined in (4)) for each estimated P_{cR}.

3.3 Self-localization Method

The method will be explained with the aid of an example. Suppose it is desired to find the pose of a monocular camera c within an environment consisting of markers $\mathcal{M} = \{W, X, Y, Z, R\}$, as shown in Fig. 2. In this case, R is selected as the global reference frame, so the problem is to find P_{cR}. As mentioned previously, there are two ways of finding P_{cR}: either directly through R (when R is in the FOV) or indirectly through intermediate markers W, X, Y, Z (when R is not in the FOV).

As shown in Fig. 3, the localization graph is connected assuming that the direct pose estimates in Fig. 2 are all available. Thus, the camera c can be localized with respect to the common reference frame R using any of the markers in the map.

The minimum requirement to achieve global localization is that there must exist a path from the current position of c to the reference frame R in $\mathcal{G}_L$. Additional edges may yield shorter paths (i.e., pose compositions with lower aggregate error). The positioning of the set of markers $\mathcal{M}$ should be chosen appropriately. Note that there is no disadvantage, aside from additional effort positioning and obtaining pose estimates, to increasing the size of $\mathcal{M}$.

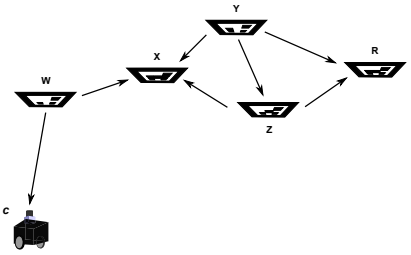

Fig. 2. Camera Pose Estimation - arrows indicate the direct pose estimate of one object (camera or marker) with respect to another, where the pose of the object on the arrow's tail is given with respect to that on the arrow's head

In this example, direct pose estimates P_{Wc}, P_{WX}, P_{YX}, P_{ZX}, P_{YZ}, P_{YR}, and P_{ZR} are obtained, along with their respective pose uncertainties. The availability of direct pose estimates is encapsulated in the localization graph of Fig. 3.

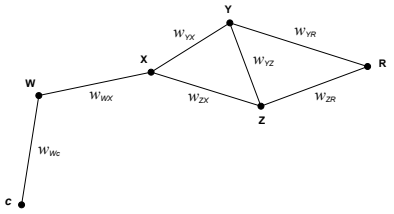

Fig. 3. Localization Graph for System in Fig. 2 - edge weights represent pose estimation uncertainty values

The solution is obtained through composition of the estimated poses, according to (3), where the shortest paths are computed using Dijkstra's algorithm or similar. As an example, suppose the shortest path from c to R in $\mathcal{G}_L$ is $\langle c, W, X, Y, Z, R\rangle$. Then,

$$P_{cR} = P_{Wc}^{-1} \circ P_{WX} \circ P_{YX}^{-1} \circ P_{YZ} \circ P_{ZR} \tag{5}$$

as per (3). The associated aggregate error is $w_{cR} = w_{Wc} + w_{WX} + w_{YX} + w_{YZ} + w_{ZR}$.

Map Updating. When there are multiple markers in the FOV, the relative poses between all possible pairs of markers in the frame are calculated and connecting edges (with associated weights) are created between them. If in a subsequent frame an edge is re-detected and is found to have a lower weight than the existing edge, its relative pose transformation and weight overwrite those corresponding to the existing edge. In this way, the edges of the graph maintain the minimum weights (and thus the relative poses with the least uncertainty) at all times.

3.4 Self-localization Algorithm

In the formal expression of the algorithm (Algorithm 1), let primed ($'$) variables represent calculations made in the current frame (e.g., P'_{ij} represents the relative pose between markers i and j as calculated from the current frame). Three previously undefined functions are used in this algorithm: The first, $calcw(\{\alpha, \beta\})$, calculates the weight of the edge connecting nodes α and β based on an appropriately derived error metric; the second, $con(\mathcal{G}, s, d)$, returns $True$ if there exists a path from s to d in $\mathcal{G}$; the third, $sp(\mathcal{G}, m, R)$, returns the shortest path p_s from m to R in $\mathcal{G}$ (using Dijkstra's algorithm) in the form defined in Section 2.3. The boolean variable n will be used to indicate that a map update has occurred.

Algorithm 1. Proposed Self-Localization Algorithm (Finding P'_{cR})

$\mathcal{M} \leftarrow \{R\}$
$E_M, \mathcal{W}_M, \mathcal{V} \leftarrow \emptyset$
$n \leftarrow False$
loop
  Capture frame
  if $\mathcal{V} \neq \emptyset$ **then**
    $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{V}$
    if $|\mathcal{V}| > 1$ **then**
      for all $\{v_i, v_j\} \in \binom{\mathcal{V}}{2}$ **do**
        $P'_{ij} \leftarrow P'_{ic} \circ P'^{-1}_{jc}$
        $P'_{ji} \leftarrow P'^{-1}_{ij}$
        if $\{v_i, v_j\} \notin E_M$ **then**
          $E_M \leftarrow E_M \cup \{v_i, v_j\}$
          $w_{ij}, w_{ji} \leftarrow \infty$
          $\mathcal{W}_M \leftarrow \mathcal{W}_M \cup w_{ij}, w_{ji}$
        end if
        if $calcw'(\{v_i, v_j\}) < w_{ij}$ **then**
          $w_{ij}, w_{ji} \leftarrow calcw'(\{v_i, v_j\})$
          $P_{ij} \leftarrow P'_{ij}$
          $P_{ji} \leftarrow P'_{ji}$
          $n \leftarrow True$
        end if
      end for
      if $n = True$ **then**
        for all $\{m \in \mathcal{M} \mid con(\mathcal{G}_M, m, R)\}$ **do**
          $p_s \leftarrow sp(\mathcal{G}_M, m, R)$
          $P_{mR} \leftarrow P_{p_{s,1}, p_{s,2}}$
          $w_{mR} \leftarrow w_{p_{s,1}, p_{s,2}}$
          for $k = 2 \rightarrow |p_s| - 1$ **do**
            $P_{mR} \leftarrow P_{mR} \circ P_{p_{s,k}, p_{s,k+1}}$
            $w_{mR} \leftarrow w_{mR} + w_{p_{s,k}, p_{s,k+1}}$
          end for
        end for
      end if
    end if
    if $R \in \mathcal{V}$ **then**
      $P'_{cR} = P'^{-1}_{Rc}$
    else if $\exists \{v \in \mathcal{V} \mid con(\mathcal{G}_M, v, R)\}$ **then**
      $v_m \leftarrow \underset{v \in \mathcal{V}}{\operatorname{argmin}}(calcw'(\{v, c\}) + w_{vR})$
      $P'_{cR} \leftarrow P'^{-1}_{v_m c} \circ P_{v_m R}$
    end if
  end if
  $n = False$
end loop

4 Experimental Results

An experiment was performed in an indoor environment using an internally calibrated ICube NS4133BU Camera, a set of markers, a marker detection algorithm, and the proposed self-localization algorithm. The purpose of the experiment is to demonstrate the proposed algorithm's performance in reducing global error.

The metric used for quantifying the edge weights of the localization graph is based on the findings of Schweighofer and Pinz [13]. In their paper, they demonstrate that pose ambiguity of planar targets is affected by the change in position and/or orientation of the target in the scene. For instance, the pose ambiguity was found to increase with the distance between the camera and the target. Based on this, a simple function $f(d)$, where d is the perpendicular distance between the camera and a marker, can be used as a metric for the edge weights. When the edge is defined by two markers, the function is applied to each marker individually and the two results are averaged to obtain the weight of the connecting edge.

The general procedure is outlined as follows:

- The markers are attached to a wall in the configuration shown in Fig. 4(a).
- The camera is traversed through the room (facing the markers throughout) while the marker detection and self-localization algorithms are running.
- A map with reference R is built and updated online by the self-localization algorithm (refer to Fig. 4(b)).
- The camera is positioned such that only marker Y is visible in its FOV.
- From this known position, the localization accuracy was compared between the shortest path obtained from our algorithm and the 35 other possible (simple) paths between markers Y and R.

Three types of functions for edge weights were tested on the map: linear ($f(d) = d$), quadratic ($f(d) = d^2$), and exponential ($f(d) = 3^d - 1$). Each function was applied to the 36 possible paths from Y to R. In all the cases, the shortest path corresponding to the minimum aggregate error (obtained from our algorithm)

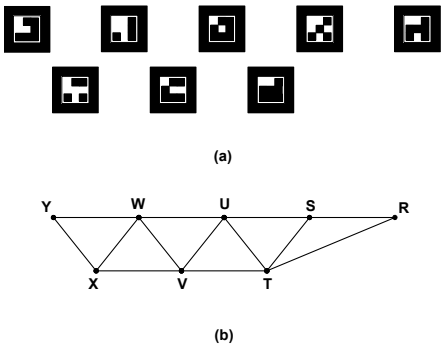

Fig. 4. Experimental Map - (a) arrangement of markers in the environment, (b) topological representation of the map

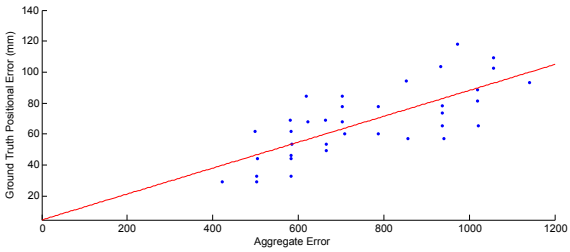

Fig. 5. Ground Truth vs. Aggregate Error - finding a correlation using exponential edge weight function $f(d) = 50^d - 1$

yielded the lowest ground truth positional error compared to the 35 other paths. The average, minimum, and maximum ground truth errors for the 36 paths were identical for all three functions: 67.9, 29.1, and 117.8 mm, respectively. Furthermore, it was observed that compared to the other two, the exponential function showed the best correlation between the aggregate error and the ground truth positional error. Fig. 5 shows this correlation in the form of a line of best fit with correlation coefficient $R \approx 0.74$, using the exponential function $f(d) = 50^d - 1$. The bottom left-most data point in this figure graphically shows how the minimum aggregate error corresponds to the minimum ground truth positional error of the 36 paths for the given metric.

The experiment was repeated using the same exponential error function, except the three lower markers were moved closer to the camera and tilted horizontally. In this case, the minimum ground truth error corresponded to the *third* least aggregate error and there was a *negative* correlation of $R \approx 0.47$ between the aggregate and ground truth errors. This indicates that this particular error function may not be suitable for all configurations of markers.

5 Conclusion

A proposed topological mapping approach is applied to a self-localization system to reduce global error and eliminate the need for an offline map-building mode. The map is incrementally built and updated as the camera moves through the environment. A shortest path algorithm is applied to the map to find the path of least aggregate error based on an appropriate edge weight metric. Experiments were done with online map-building and using a weight metric based on the perpendicular distance between the camera and each marker. The results demonstrate the effectiveness of the system in reducing global error but emphasize the importance of deriving an appropriate error metric for the edge weights.

References

1. Anjum, M., Park, J., Hwang, W., il Kwon, H., Hyeon Kim, J., Lee, C., Soo Kim, K., il Danr Cho, D.: Sensor data fusion using unscented kalman filter for accurate localization of mobile robots. In: 2010 International Conference on Control Automation and Systems (ICCAS), pp. 947–952 (October 2010)
2. Ansar, A., Daniilidis, K.: Linear pose estimation from points or lines. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(5), 578–589 (2003)
3. Brand, M., Antone, M., Teller, S.: Spectral Solution of Large-Scale Extrinsic Camera Calibration as a Graph Embedding Problem. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3022, pp. 262–273. Springer, Heidelberg (2004)
4. Chen, X., Li, R., Wang, X., Tian, Y., Huang, Q.: A novel artificial landmark for monocular global visual localization of indoor robots. In: 2010 International Conference on Mechatronics and Automation (ICMA), pp. 1314–1319 (August 2010)
5. Choi, H., Kim, D.Y., Hwang, J.P., Kim, E., Kim, Y.O.: Cv-slam using ceiling boundary. In: 2010 the 5th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 228–233 (June 2010)
6. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1(1), 269–271 (1959)
7. Fiala, M.: Artag, a fiducial marker system using digital techniques. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 590–596 (June 2005)
8. Hwang, S.Y., Park, J.T., Song, J.B.: Autonomous navigation of a mobile robot using an upward-looking camera and sonar sensors. In: 2010 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), pp. 40–45 (October 2010)
9. Kitanov, A., Bisevac, S., Petrovic, I.: Mobile robot self-localization in complex indoor environments using monocular vision and 3d model. In: 2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1–6 (September 2007)
10. Lim, H., Lee, Y.S.: Real-time single camera slam using fiducial markers. In: ICCAS-SICE, pp. 177–182 (August 2009)
11. Lv, Q., Zhou, W., Liu, J.: Realization of odometry system using monocular vision. In: 2006 International Conference on Computational Intelligence and Security, vol. 2, pp. 1841–1844 (Novemebr 2006)
12. Mavrinac, A., Chen, X., Tepe, K.: An automatic calibration method for stereo-based 3d distributed smart camera networks. Computer Vision and Image Understanding 114(8), 952–962 (2010)
13. Schweighofer, G., Pinz, A.: Robust pose estimation from a planar target. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(12), 2024–2030 (2006)
14. Sharp, G., Lee, S., Wehe, D.: Multiview registration of 3d scenes by minimizing error between coordinate frames. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(8), 1037–1050 (2004)
15. Van Hamme, D., Veelaert, P., Philips, W.: Robust monocular visual odometry by uncertainty voting. In: 2011 IEEE Intelligent Vehicles Symposium (IV), pp. 643–647 (June 2011)
16. Yu, Y., Pradalier, C., Zong, G.: Appearance-based monocular visual odometry for ground vehicles. In: 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 862–867 (July 2011)
17. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(11), 1330–1334 (2000)