Coordination of Multiple Agents for Probabilistic Object Tracking

Roozbeh Mottaghi and Shahram Payandeh
School of Engineering Science
Simon Fraser University
rmottagh, shahram@cs.sfu.ca

Abstract

In this paper, we develop a new tracking approach which is based on cooperation and coordination of multiple agents which are pan-tilt-zoom cameras to optimize the cost of tracking and communication while simultaneously focus on the details of the object of interest. Each agent is able to track the object individually but the problem arises when the object goes suddenly out of the field of view of one agent because of an occlusion or an unexpected event. So each agent has to decide to take an action among a set of finite possible actions to overcome this situation in a way that optimizes the task of tracking.

Index Terms – Cooperative Tracking, Particle Filtering, Pan-Tilt-Zoom Tracker.

1. Introduction

As the demand for reliable, fault-tolerant and fast systems has increased, many researchers have been attracted to Multi Agents Systems. Inherent parallelism which results in better performance as well as distribution of intelligent components and overall reliability and robustness has given more popularity to these kinds of systems in the research labs and industries.

The areas of application of these systems vary greatly but in general a solution need to be sought to coordinate the agents to optimize the costs of doing the assigned task or to share information among the agents for higher efficiency. A novel planning method is proposed by Li et al in [1] for multi-agent dynamic manipulation where a single agent is not capable of doing the task individually. A game theory approach has been presented for solving the coordination task. As an example of a multi agent system consisting agents with different capabilities, Grabowski et al proposes the design of a team of heterogeneous robots which can be coordinated to provide real-time surveillance and reconnaissance [2]. Each group of robots has its own type of sensor and the robot team exploits modular sensing, processing and mobility to achieve a wide range of tasks that include mapping and exploration. Burgard et al also consider the problem of exploring an unknown environment by a team of robots which provides a faster and more reliable approach rather than traditional approaches [3].

One of the applications which has attracted many researchers in the field of multi-agent systems is multi-sensor tracking. This means determining the position of one or multiple objects of interest and tracking their movements according to the data from multiple sensors. Kang et al presents an adaptive background generation and moving region detection in [4] for a single pan-tilt-zoom camera. Jung et al has solved the problem of fixed cameras in [5] and has implemented a robust real-time algorithm for moving object detection for an outdoor robot carrying a single camera. The proposed methods for any single camera have a number of drawbacks such as not being robust against failure and occlusion and also poor depth estimation which are the major deficiencies of the above examples that use a single camera. These problems have been overcome by switching to multiple camera approaches. Collins et al present a mean-shift tracker that adjusts the pan, tilt, zoom and focus parameters of multiple active cameras for tracking a person in the scene [6]. Their design emphasizes modularity and robustness of each individual camera, so they have to broadcast a large amount of data in a period of time (once per second) which can degrade the performance of the whole system. An automated surveillance system is proposed by Lim et al in [7]. They use multiple pan-tilt-zoom cameras to track people in the scene. First, a master camera finds the object of interest in its field of view and assigns a camera to track it by using a Kalman filter tracker. This approach also can suffer from the lack of robustness. If the master camera fails, the system will not work at all.

In this paper, we present a new tracking approach which is based on cooperation and coordination of

multiple agents which are pan-tilt-zoom cameras to optimize the cost of tracking and communication while simultaneously focus on the details of the object of interest. Each camera is able to track the object individually which can result in a robust system against failure. The cooperative behavior of the cameras also results in the better performance of the system in case of loosing track of the object due to occlusion or limited field of view. Another feature of the proposed method is that it focuses on reducing the cost of communication and searching space.

The architecture of each agent consists of three main modules: Object Detection module, Tracking module and Coordination module. In the next section we present a fast and inexpensive method developed for object detection. Section III provides the description for CONDENSATION (Conditional Density Propagation) [8] algorithm for tracking which represents arbitrary multi-modal densities and unlike [8] is not based on image plane coordinates. In section IV, the cooperative action selection according to multiple degrees of freedom of each camera and optimization strategies are discussed. Also we describe our extension to the probabilistic tracker for improvement of the performance. After that, the experimental results are shown for single and multiple camera cases.

2. Moving object detection

The first processing step which is done by each agent individually is searching and detecting the object of interest in the field of view at the current degree of pan and tilt.

A. Image segmentation and finding connected regions

A fast, inexpensive and robust color-based method that is a modified version of [9] is implemented for the detection of the objects. The modification is made mainly on reducing the memory needed for processing and also removing the noise after the processing.

A data structure is defined for storing the information of the objects which are found in the image. This information includes coordinates of the bounding boxes around the objects (blobs), the size of the object which is the number of pixels that are surrounded by the rectangular region mentioned above, color and object number.

B. Noise filtering

Before we pass the information from Object Detection module to the Tracking module we should eliminate the noise objects from the list of the detected objects. Two fast filtering methods are being adopted for performing this task. First we search in the object list and if the size (number of pixels in the blob) is less than an expected value, we remove that object from the list. The second filtering method is based on evaluating the height to width ratio (or width to height ratio). If this ratio is also less than a specified value, that object should be considered as a noise. Since the number of found objects is usually small, the search is done quickly. These filters are adjusted considering the physical specification of the environment such as shape and size of the objects and frequent errors happened in the vision system due to camera noise and so on. After the processing is finished, we pass the information from this module that is the coordinates of the center of mass of the objects and width of the objects to the tracking module for localization and prediction of the object state in a global coordinate frame.

3. Probabilistic tracking using a single pan-tilt-zoom camera

A probabilistic approach has been applied for tracking the objects in the scene. We have developed our tracking algorithm for a single pan-tilt-zoom camera based on the Condensation algorithm. This algorithm which is based on the Bayes' rule is not only computational efficient it has also shown better performance compare to similar trackers. For instance, unlike a Kalman filter, it is able to represent almost arbitrary distributions and no functional assumptions (linearity, Gaussianity, unimodality) are made and it is simpler compare to a Kalman filter due to the absence of computationally complex Riccati equations. A detailed comparison of the Condensation with Mean-Shift and Kalman Filter trackers can be found in [10].

A general assumption which is made here is that the environment is Markovian which means the current state of the object only depends on the previous state. The observations are also mutually independent and they are not related to the dynamic process. We can express these assumptions by the following equation where probability distribution of previous measurements and current state given the previous states is shown:

$$p(z_1,\ldots z_{t-1}, x_t | x_1,\ldots x_{t-1}) = p(x_t | x_1,\ldots, x_{t-1}) \prod_{i=1}^{t-1} p(z_i | x_i) \quad (1)$$

where x_t and z_t are state and measurement in time t.

In our implementation, the state vector x defines a complement of spherical coordinates of the object, i.e. $x = [q \quad \dot{q} \quad f \quad \dot{f}]$. If we project the line that connects the center of projection of the camera (at an initial degree of pan and tilt) and center of mass of the visible part of the object on the x-y plane of the reference coordinate frame

(to be defined next and shown in Fig. 1), q is defined as the angle between that line and positive direction of the **x** axis. f is also defined as the angle between the projection of that line onto y-z plane and positive direction of the **z** axis. Let us define the coordinate frame C like what is shown in Fig. 1. The origin of this frame is located at the center of projection of the camera in an initial state and its ***z*** axis is perpendicular to the plane that the camera resides on and the ***y*** axis is perpendicular to the center of the image plane at an initial position of the camera. $\dot{q}$ and $\dot{f}$ are representing the rate of change of those angles in each time step. Since objects with the same q and f but different distances to the origin has the same projection in the image plane, for efficiency in computations we have eliminated the distance component from the spherical coordinates. Because we have different zoom values during the processing, we should map the image plane data to a world coordinate frame that is the complement of spherical coordinates in our solution.

At each time step the pan and tilt degree of the camera (q_c and f_c) are known. To compute the complement of spherical coordinates of the object that is being tracked we should consider the deviation of the center of the visible part of the object from the center of the image plane at the current degree of pan and tilt considering the current zoom value (Fig. 1). So q and f, the angular components of object state are evaluated according to following equations:

$$q = q_c - q_I = q_c - (a_h . o_{cx})/(z.w_I)$$
$$f = f_c - f_I = f_c - (a_v . o_{cy})/(z.h_I) \qquad (2)$$

where a_h and a_v are horizontal and vertical angle of view of the camera (without zooming), o_{cx} and o_{cy} are the differences (in pixel) of the object center and ***x*** and ***y*** axis of the coordinate frame which is shown in the image plane in Fig. 1. w_I and h_I are width and height of the image plane in pixel and z is the zoom ratio at that time step.

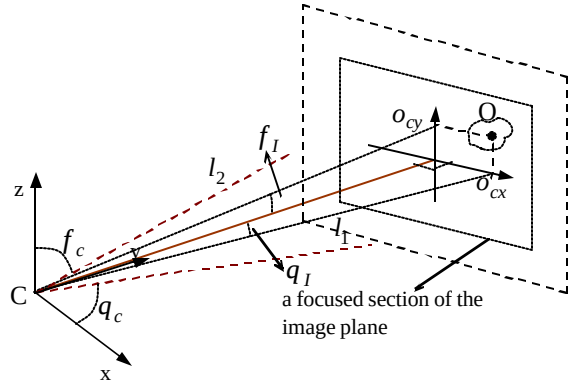


Figure 1. l_1 and l_2 are the projection of the optical axis of the camera in the current degree of pan and tilt (The solid red line) on the x-y and y-z plane respectively. q_c is the angle between l_1 and the x axis and f_c is the angle between l_2 and the z axis.

Since the whole object is not visible in the zoom-in state, we do not have the real coordinate of the center of the object and we assume that there is not a large difference between the real center of the object and the center of the visible part of the object.

The whole emphasis of the above methods or the methods in the next section is to minimize the amount and complexity of the computation needed considering the fact that a good estimate is gained through the computations without a need for the intrinsic camera parameters.

The goal of the Condensation tracker is to find a density function for approximating the state of the object which means finding of the current state given the measurements from the beginning of the processing till now (i.e. $\boldsymbol{p(x_t | z_1,\ldots,z_t)}$ where $\boldsymbol{x_t}$ and $\boldsymbol{z_t}$ are state and measurement in time *t)*. Using the Bayes' rule we have:

$$\boldsymbol{p(x_t | z_1,\ldots,z_t) = \frac{p(z_t | x_t, z_1,\ldots,z_{t-1})\, p(x_t | z_1,\ldots,z_{t-1})}{p(z_t | z_1,\ldots,z_{t-1})} =}$$
$$\boldsymbol{k p(z_t | x_t, z_1,\ldots,z_{t-1})\, p(x_t | z_1,\ldots,z_{t-1}) =}$$
$$\boldsymbol{k p(z_t | x_t)\, p(x_t | z_1,\ldots,z_{t-1})} \qquad (3)$$

where k is a normalization factor which does not depend on $\boldsymbol{x_t}$ and simplification has been made using the fact that observations are independent.

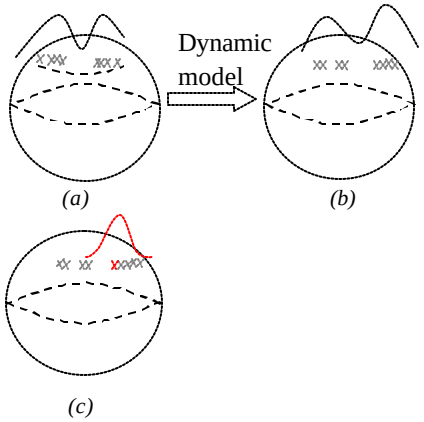


Figure 2. (a) The density distribution for the previous time step (b) particles' distribution after applying the dynamic model (c)The red cross shows the real measurement.

The one dimensional case is shown in Fig. 2 and the assumption is that the object moves on a path parallel to the equator of the sphere which is an approximation for the image planes in all degrees of pan and tilt. First, we choose a set of N samples randomly with a probability proportional to $p(x_{t-1}|z_1,\ldots,z_{t-1})$ which is known from the previous time step (Fig. 2a). Then we apply the stochastic dynamic model of the object movement to the set of samples to get a new set of samples (Fig. 2b). The density distribution at this step is proportional to $p(x_t|z_1,\ldots,z_{t-1})$. After the measurement is done according to the data from the processing of the image, we re-weigh each particle according to a Gaussian whose mean is located on the measurement and the set of samples with their new distribution is propagated to the next step. Since we have a four dimensional state vector we need a four dimensional Gaussian and the dynamic model should be applied in two dimensions (The real measurement is shown in Fig.3). The particle with the median of the probabilities is considered as the estimated belief of the agent in each time step.

At the beginning or in the single camera case, we use a first order linear stochastic differential equation as the dynamic model due to mechanical limitation of the camera($x_t = Ax_{t-1} + \textit{stochastic part}$), but we change this model over time for a better cooperative tracking performance in the multiple camera case. In section IV we will discuss the cases in which the dynamic model changes.

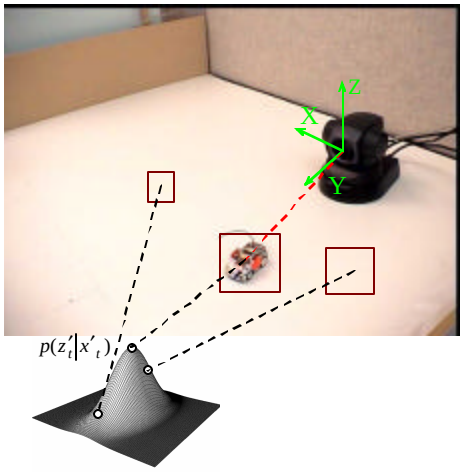


Figure 3. Two particles (rectangular boxes) and the real state measurement are shown in the picture. Less difference between the measurement and the estimated state results in a higher chance for that state to be selected in the next time step. A two dimensional Gaussian is shown as the density function for two position components of the state vector.

4. Cooperative action selection

The goal of the system is to track the objects of interest by relying on cooperation and information sharing and each agent tries to maximize the number of the cameras that track the object. At the beginning, the tracking is being done with a maximum focus on the object but the problem arises when the object of interest goes suddenly out of the field of view of the camera (it can be an occlusion or moving with a higher velocity than the speed of the camera) and the camera misses the track of the object. So the camera should decide what to do in the next time step to maximize the performance of the system.

We have a set of agents $A=\{A_1, A_2, ..., A_n\}$ which can choose an action from a finite set of possible actions. Currently, we have defined three actions for each agent: Tracking, Communication and Zoom-out. Each agent should select a joint action in a way to maximize the total utility of the system and minimize the cost function which is defined to be:

$$f = \sum_i c_{comm_i} + c_{zoom_i} \qquad (4)$$

provided that the number of cameras that can see the object need to be maximized. c_{comm_i} and c_{zoom_i} are the costs of communication and zooming for agent *i*, respectively. At each time step we assign a weight to the following actions and based on the rules and the weighted actions each agent makes a decision. The rules for assigning a weight to each action are as follows:

Tracking: This action has the highest weight if the object of interest is visible and there is no request from the other agents for sharing information about the state of the object. In this case, a relative degree of pan and tilt is selected according to the agent's belief about the state of the object and the appropriate pan-tilt command is sent to the camera. It should be noted that, the camera has the maximum focus when choosing this action.

Communication and Information Sharing: If agent *i* looses the track of the object which means it doesn't have any measurement about the object in a single frame, it sends a request command to agent *j*, an agent that sees the object partially or completely and asks for information and agent *j* sends back its belief (b_j) which is the angular components (q and f) of the particle with the median of weights in the response to the request. Then agent *i* adjusts its belief (b_i) according to the agent *j*'s. If this action is selected camera *i* transforms the coordinates of the object from the reference coordinate frame whose origin is the center of projection of the other camera in the initial state (like the coordinate frames in figures 1 and 3) to its own

reference frame. The procedure of this approximate transformation is described in Appendix A.

In this case we modify the Condensation algorithm by considering another dynamic model for the object. Since in this case the state of the object and the measurements has a large difference, we increase the effect of velocity components in the dynamic model.

Zooming out: The other strategy that an agent can take is to widen its field of view to cover a larger area for searching for the object. This process is considered as a costly process because of the zooming out and zoom in again imposes a lot of delay due to the mechanical movement of the lens and each agent prefers to minimize usage of this action. But in the cases that all of the agents have no idea about the state of the object this action has the highest weight. So two agents are selected among n agents and we assign a unique weight to this action for both of the cameras. The weight which is assigned to this action is proportional to the difference between $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$, the believes of agent i and agent j from the view of an agent:

$$w \propto \left\| \boldsymbol{b}_i - \boldsymbol{b}_j \right\| \qquad (5)$$

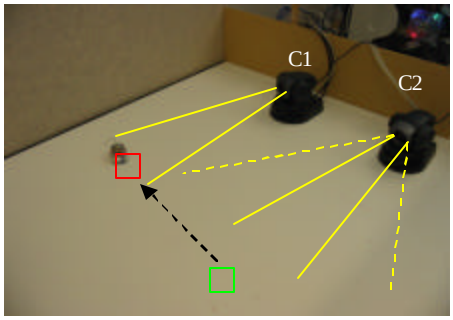
The idea is that if there is a large difference between the believes of the two agents, the chance of finding the object would be higher if both of the cameras zooms out. The other option is that only one camera zooms out and the other agents wait for the result of the camera which has zoomed out and get the data through communication. The weight of this action is selected according to the following equation (the weights are summed to one).

$$\boldsymbol{w} = 1 - \boldsymbol{k} \left\| \boldsymbol{b}_i - \boldsymbol{b}_j \right\| \qquad (6)$$

where k is a normalization constant.

5. Experimental results

We have used Sony EVI-D100 Pan-Tilt-Zoom cameras for conducting the experiments. These cameras can pan 200 degrees with the maximum speed of 300 degrees per second and tilt 50 degrees with the maximum speed of 125 degrees per second. The horizontal angle of view of these cameras in normal zoom is 65 and it varies to 6.6 degrees in the maximum zoom state. First, we used a single camera for tracking a small remote controlled car which moves randomly on a table. In this case, the camera that is placed on the table, estimates the position of the car by using the Condensation algorithm and adjusts its pan and tilt degree according to the state of the tracker and when it fails to track the car it zooms out to search a larger area for the car and when it finds the car it zooms in again. We have defined the failure as the invisibility of the object in a single frame for a camera. The result has been gathered from 20 trails of one minute of tracking and the percentage of the failure is 53.2%. The failures are mainly due to the movement of the car in the invisible areas (behind the camera, for instance).



(a)

(b)

Figure 4. (a) The red box shows the belief of the camera that sees the car. (b) Camera 2 adjusts its belief according to Camera 1's belief.

Next, we assigned two pan-tilt-zoom cameras to track the car cooperatively (Fig. 4) by following the algorithm and weighting scheme which is presented in section IV. The results which are shown in table 1 are gained from 20 trails of one minute of cooperative tracking. It should be noted that the car moves with the speed of two meters per second.

The first column shows the average percentage of failures of each camera during the tracking, the second column represents the average percentage of times when the camera takes the communication strategy and the third column is the representative of the average percentage of number of taking the zoom-out action.

TABLE I
Result of the cooperative tracking

Camera No.	Fail (%)	Communication (%)	Zoom-out (%)
Camera 1	32.6	38.3	13.6
Camera 2	33.4	38.6	9.5

As the results show, the number of failures has been decreased greatly compare to a single camera tracker and

the performance and robustness has been improved with the algorithm presented for cooperative tracking.

6. Conclusion and future works

A cooperative multiple pan-tilt-zoom camera system was introduced to track objects of interest in the environment. First, a fast and robust color-based object detector was implemented to detect the objects of interest in the field of view of the camera. Then, the result of the object detector is passed to the Condensation tracker module which is able to model non-linear and non-Gaussian motions where the state of the tracker consists of angular components of the spherical coordinates of the object. Then we defined a finite set of actions for each camera, consisting tracking, communication and zooming-out and the agents decide at each time step based on the weight of each action.

We tested the algorithm for a single and multiple pan-tilt-zoom cameras and we showed that the cooperative tracker has a better performance and robustness compare to a single camera tracker.

We are planning to implement an adaptive background modelling technique to detect any moving object in the scene. For example we can use this system for tracking athletes in the sports fields while zooming on them. The other thing that we plan to do is to share the information between the agents and assign a level of reliability to each agent to have a better estimation of the depth of the object. Also a planning algorithm should be considered for optimizing the task of tracking in the cases when the number of objects of interest exceeds the number of cameras.

Appendix A: *Coordinate transformations between two cameras*

First, the Cartesian coordinates of the center of the object (x_j, y_j and z_j) in the coordinate frame attached to the camera j are found (the assumption here is that the object is only visible to camera j). For computing the Cartesian world coordinate system, we need the distance of the object from the camera. This distance is currently approximated by the use of width of the object. Then we do the transformation to find the coordinates of the center of the object relative to the other coordinate frame which is attached to the other camera. If $d_{ji} = [d_x \quad d_y \quad d_z]^T$ be the vector from the origin of camera j to the origin of camera i, the new coordinates, $[x_i \quad y_i \quad z_i]^T$, are computed by using the following equation:

$$\begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x_j \\ y_j \\ z_j \\ 1 \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \qquad (7)$$

If camera j has been also rotated relative to the coordinate frame of camera i, we should apply the rotation matrix to the above equation. In the next step, we compute the spherical coordinates from the new Cartesian coordinates because the measurement and the state vector of the Condensation tracker is based on the spherical coordinates. Then camera i adjusts its belief according to these new coordinates in the next time step.

References

[1] Q. Li and S. Payandeh, "Multi-agent Cooperative Manipulation with Uncertainty: A Neural-Net-Based Game Theoretic Approach," *IEEE International Conference on Robotics and Automation 2003*, pp. 3607-3612, Sep. 2003, Taiwan.

[2] R. Grabowski, L. E. Navarro-Serment, C. J. J. Paredis and P. K. Khosla, "Heterogeneous Teams of Modular Robots for Mapping and Exploration," *Autonomous Robots*, 8(3):293-308, 2000.

[3] W. Burgard, M. Moors, D. Fox, R. Simmons, S. Thrun, "Collaborative Multi-Robot Exploration," *Proceedings of IEEE International Conference on Robotics and Automation 2000*, 476-481, San Francisco, April 2000.

[4] S. Kang, J. Paik, A. Koschan, B. Abidi and M. A. Abidi, "Real-time Video Tracking using PTZ Cameras" Proc. of SPIE 6th International Conference on Quality Control by Artificial Vision, Vol. 5132, pp 103-111, Gatlinburg, TN, May 2003.

[5] B. Jung and G. S. Sukhatme, "Detecting Moving Objects Using a Single Camera on a Mobile Robot in an Outdoor Environment," *8th conference on Intelligent Autonomous Systems*, pp 980-987, Amsterdam, The Netherlands, March 2004.

[6] R. Collins, O. Amidi, T. Kanade, "An Active Camera System for Acquiring Multi-view Video," *International Conference on Image Processing*, Rochester, NY, Sep. 2002.

[7] S. Lim, A. Elgammal and L. S. Davis, "Image-based Pan-tilt Camera Control in a Multi-Camera Surveillance Environment," *IEEE ICME 2003, Special Session on Visual Surveillance*, Jul 6-9, Baltimore, Maryland.

[8] M. Isard and A. Blake, "Condensation – Conditional Density Propagation for Visual Tracking," *International Journal on Computer Vision*, 29(1): 5-28, 1998.

[9] J. Bruce, T. Balch, and M. Veloso. "Fast and Inexpensive Color Image Segmentation for Interactive Robots," *In Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)*, volume 3, pages 2061, 2000.

[10] K. Nummiaro, E. Koller-Meier, L. Van Gool, "An Adaptive Color-based Particle Filter, " *Image and Vision Computing*, pp. 99-110, 2002.