Target Trajectory Prediction in PTZ Camera Networks

Vahab Akbarzadeh, Christian Gagné, and Marc Parizeau
Laboratoire de vision et systèmes numériques
Département de génie électrique et de génie informatique
Université Laval, Québec (Québec), Canada G1V 0A6
vahab.akbarzadeh.1@ulaval.ca, christian.gagne@gel.ulaval.ca
marc.parizeau@gel.ulaval.ca

Abstract

In this paper we compare different mechanisms for the prediction of targets position inside a PTZ camera network. The goal is to predict the next location of each target with higher accuracy, in order to better plan the movements of the cameras at the next time step. For that purpose, we are proposing a probabilistic multimodal approach, and show that this prediction method can improve the total coverage of a camera network compared to other probabilistic prediction methods.

1. Introduction

The use of Pan-Tilt-Zoom (PTZ) cameras for automatic surveillance systems have recently increased due to advances in the field of multi-camera multi-target surveillance camera networks [2]. In classical surveillance systems, a limited number of CCTV cameras are controlled by humans, while in a smart network, the PTZ cameras are automatically controlled to obtain a better coverage of the targets inside the field of interest.

The challenging objective in these networks is the control mechanism of the cameras' pan, tilt, and zoom capabilities for an overall better coverage of the targets. For that purpose, the targets should first be detected in the environment, their movement should then be modelled to predict the next locations, and finally the PTZ parameters of the cameras adjusted for optimal coverage at the next time steps. The focus of this paper is on the targets' movement prediction and camera adjustment steps.

Rahimi *et al*. [6] were among the first to propose a system to track a target with a network of non-overlapping cameras. They proposed an algorithm for simultaneous estimation of targets' trajectories (current location and directional velocity) and camera calibration. Later, Lim *et al*. [3] proposed a system for active control of cameras to obtain an non-occluded view of each target. For that purpose, the position of the targets are predicted as a straight line with uncertainty modelled as a Gaussian function.

Krahnstoever *et al*. [2] proposed a probabilistic objective function for coverage estimation of a camera over a target considering the capture distance, view angle, target reachability, and trackability. Using a similar relevance model between cameras and targets, Qureshi and Terzopoulos [5] presented the optimal collaboration between cameras as a planning problem which was solved using the greedy best-first search mechanism. Their target location prediction is calculated by the static cameras, although no more details are provided on the prediction mechanism.

Recently, Natarajan *et al*. [4] presented a decision theoretic approach based on Markov Decision Process to maximize coverage of the camera network. Their work also has a probabilistic model for prediction of targets position, although, as we will mention later, their work is categorized under the unimodal Gaussian movement assumption.

To the best of our knowledge, in all the previous work that has been done on target trajectory prediction, the next location was predicted with a combination of the target's velocity and uncertainty on the velocity (we refer to this approach as the unimodal approach in this paper). In these approaches, at each time step, the uncertainty has a unimodal shape and cannot represent simultaneous alternative hypotheses. For example, if a target is moving in a zig-zag pattern, the system does not have the capacity to represent the two possible next locations simultaneously.

In this paper, we are proposing to predict the next location of a target using a multimodal probabilistic model. So that the prediction for the next movement is modelled probabilistically based on the previous movements of each target. The effectiveness of our proposed model is shown by comparing its performance with three other prediction models.

For the rest of the paper, we present the smart camera network model and the trajectory prediction problem in Section 2. Different prediction methods are presented in Section 3. Section 4 reports the simulation setup for the experiments and the results before concluding the paper in Section 5.

2. Problem Statement

Assume that we have a camera network $\mathbf{C} = \{\mathbf{C}_i | i = 1, \ldots, N\}$ consisting of N Pan-Tilt-Zoom (PTZ) cameras $\mathbf{C}_i$. The state of each camera at time t is defined as

$\mathbf{C}_i^{(t)} = (\mathbf{p}_i, \theta_i^{(t)}, \xi_i^{(t)}, f_i^{(t)})$, where $\mathbf{p}_i = (x_i, y_i)$ is the position of the camera in the coordinate system, $\theta_i^{(t)}$ is the pan angle, $\xi_i^{(t)}$ is the tilt angle, and $f_i^{(t)}$ is the focal length of the camera. It is also assumed that the cameras are positioned τ meters above the ground. The environment Ξ is defined by a Digital Elevation Model (DEM), where the elevation of each location (x, y) is given as the function $k(x, y)$. Therefore, the elevation of each camera is $z_i = k(x_i, y_i) + \tau$. From this definition it is clear that the position of each camera is static over time. Each camera has a set of characteristic parameters which define the maximum and minimum value of each PTZ parameters, along with their associated speed of change:

$$\langle(\theta_{min}, \theta_{max}, v_\theta), (\xi_{min}, \xi_{max}, v_\xi), (f_{min}, f_{max}, v_f)\rangle.$$

The goal of the network $\mathbf{C}$ is to optimally cover an unknown number of moving targets $\mathbf{T} = \{\mathbf{T}_j | j = 1, 2, \dots\}$. Each target has a history of past locations $\mathbf{T}_j = \{\mathbf{T}_j^{(s)} | s = t_j, \dots, t\}$, starting at time t_j, when the target was first detected in the environment, up to the current time t. Here, $\mathbf{T}_j^{(s)} = (x_j^{(s)}, y_j^{(s)})$ is the location of target j at time s.

The strategy the network follows for camera $\mathbf{C}_i$ through time is defined by $\mathbf{S_i} = \{s_i^{(t)} | t = 1, \dots, T\}$, where T is the total simulation time and $s_i^{(t)} = (\Delta_\theta, \Delta_\xi, \Delta_f)$ is the command given to camera $\mathbf{C}_i$ at time t for updating its PTZ values. The set of strategies for the N cameras is summarized as $\mathbf{S} = \{\mathbf{S}_i | i = 1, \dots, N\}$. It is clear that the given command at each time step should satisfy each camera's movement constraints. More precisely we have as constraints:

$$|\Delta_\theta| \le v_\theta, \quad \theta_{min} \le \underbrace{\theta_i^{(t)} + \Delta_\theta}_{\theta_i^{(t+1)}} \le \theta_{max},$$

$$|\Delta_\xi| \le v_\xi, \quad \xi_{min} \le \underbrace{\xi_i^{(t)} + \Delta_\xi}_{\xi_i^{(t+1)}} \le \xi_{max},$$

$$|\Delta_f| \le v_f, \quad f_{min} \le \underbrace{f_i^{(t)} + \Delta_f}_{f_i^{(t+1)}} \le f_{max}.$$

Now the coverage $C(\mathbf{C}_i^{(t)}, \mathbf{q})$ of camera $\mathbf{C}_i$ on point $\mathbf{q} = (x_\mathbf{q}, y_\mathbf{q})$ in the environment at time t is defined as a function of distance $d(\mathbf{C}_i^{(t)}, \mathbf{q}) = \|\mathbf{p}_i - \mathbf{q}\|$, pan angle $p(\mathbf{C}_i^{(t)}, \mathbf{q}) = \angle_p(\mathbf{q} - \mathbf{p}_i) - \theta_i^{(t)}$, tilt angle $t(\mathbf{C}_i^{(t)}, \mathbf{q}) = \angle_t(\mathbf{q} - \mathbf{p}_i) - \xi_i^{(t)}$, and visibility $v(\mathbf{C}_i, \mathbf{q})$ of the camera:

$$C(\mathbf{C}_i^{(t)}, \mathbf{q}) = f[\mu_d(\|\mathbf{p}_i - \mathbf{q}\|), \mu_p(\angle_p(\mathbf{q} - \mathbf{p}_i) - \theta_i^{(t)}), \mu_t(\angle_t(\mathbf{q} - \mathbf{p}_i) - \xi_i^{(t)}), v(\mathbf{p}_i, \mathbf{q})], \quad (1)$$

where $\angle_p(\mathbf{q} - \mathbf{p}_i) = \arctan(y_\mathbf{q} - y_{\mathbf{p}_i}, x_\mathbf{q} - x_{\mathbf{p}_i})$ is the pan angle between camera $\mathbf{C}_i$ and point $\mathbf{q}$, and $\angle_t(\mathbf{q} - \mathbf{p}_i) = \arctan(z_\mathbf{q} - z_{\mathbf{p}_i}, \|\mathbf{p}_i - \mathbf{q}\|)$ is the tilt angle between camera $\mathbf{C}_i$ and point $\mathbf{q}$. In other words, for $\mathbf{q}$ to be covered by camera $\mathbf{C}_i$, we need to take into account its range, viewing angles, and visibility. The focal point $f_i^{(t)}$ indirectly affects the coverage through pan and tilt angles, and range. This effect will be later discussed in Section 2.1. Let $\mu_d, \mu_p, \mu_t \in [0, 1]$ represent some membership functions of the mentioned coverage conditions, then (1) can be rewritten as a multiplication of these memberships:

$$C(\mathbf{C}_i^{(t)}, \mathbf{q}) = \mu_d(\|\mathbf{p}_i - \mathbf{q}\|) \cdot \mu_p(\angle_p(\mathbf{q} - \mathbf{p}_i) - \theta_i^{(t)}) \cdot \mu_t(\angle_t(\mathbf{q} - \mathbf{p}_i) - \xi_i^{(t)}) \cdot v(\mathbf{p}_i, \mathbf{q}).$$

Function $v(\mathbf{p}_i, \mathbf{q})$ is usually binary. Given a camera position $\mathbf{p}_i$, if the line of sight between camera $\mathbf{C}_i$ and $\mathbf{q}$ is obstructed, then we assume that the visibility cannot be achieved ($v(\mathbf{p}_i, \mathbf{q}) = 0$), otherwise the visibility is fully attained ($v(\mathbf{p}_i, \mathbf{q}) = 1$).

Value $C = 1$ means full coverage, and $C = 0$ indicates no coverage. If more than one camera covers $\mathbf{q}$, a way to compute the local network coverage C_l of network $\mathbf{C}$ over location $\mathbf{q}$ at time t is:

$$C_l(\mathbf{C}^{(t)}, \mathbf{q}) = 1 - \prod_{i=1,\dots,N} \left(1 - C(\mathbf{C}_i^{(t)}, \mathbf{q})\right).$$

The speed of the cameras is limited (defined by parameters v_θ, v_ξ, and v_r), therefore, at each time step, the system should predict the location of targets in the next time step and plan for movement of the cameras appropriately. More precisely, let us assume that $\widehat{\mathbf{T}}_j^{(t+1)}$ is the predicted position of target $\mathbf{T}_j$ at time $t + 1$, and $\widehat{g}(\mathbf{T}_j, \mathbf{q}, t + 1)$ is the probabilistic displacement model of target $\mathbf{T}_j$ for time $t + 1$. In Section 3, we will explain how this probabilistic model is learned from the previous positions of each target. The probabilistic displacement model assigns a probability $\widehat{g}(\mathbf{T}_j, \mathbf{q}, t + 1) = p(\widehat{\mathbf{T}}_j^{(t+1)} = \mathbf{q})$ to all positions $\mathbf{q}$ in the environment which are within a limited neighbourhood of the target.

Each position $\mathbf{q}$ is also attributed to another parameter $w_\mathbf{q}^{(t)}$. This parameter defines the importance of location $\mathbf{q}$ for the coverage task at time t. Therefore, higher values of $w_q^{(t)}$ represent higher importance of the location $\mathbf{q}$ in the goal coverage problem. Targets are modelled as the weight they assign to the locations in which they reside. Therefore, if we predict that target $\mathbf{T}_j$ moves into a location $\mathbf{q}$ at time $t+1$ then it contributes the amount of $\widehat{g}(\mathbf{T}_j, \mathbf{q}, t+1)$ to the weight of $\mathbf{q}$ and we have: $w_\mathbf{q}^{(t+1)} = \sum_{\mathbf{T}_j \in \mathbf{T}} \widehat{g}(\mathbf{T}_j, \mathbf{q}, t+1)$.

The goal of the system is to find an optimal strategy for the cameras, so that targets are detected as soon as they arrive inside the environment and they stay under the maximal possible coverage while they traverse the environment. We define the following objective function for a given strategy as the coverage it provides for all of the targets through time:

$$O(\mathbf{S}) = \sum_{t=1}^{T} \sum_{\mathbf{q} \in \Xi} w_\mathbf{q}^{(t)} C_l(\mathbf{C}^{(t)}, \mathbf{q}),$$

and the goal is to find a strategy $\mathbf{S}^*$ which maximizes this objective function:

$$\mathbf{S}^* = \underset{\mathbf{S}}{\operatorname{argmax}}\ O(\mathbf{S}).$$

The defined objective function provides an estimate on the actual coverage that the network provides over the targets. The actual coverage is calculated using the following formula:

$$C_g(\mathbf{S}) = \sum_{t=1}^{T} \sum_{\mathbf{q}\in\Xi} 1(\mathbf{T}_j^{(t)}, \mathbf{q})\ C_l(\mathbf{C}^{(t)}, \mathbf{q}),$$

where the function $1(\mathbf{T}_j^{(t)}, \mathbf{q})$ returns one if target $\mathbf{T}_j$ is at location $\mathbf{q}$ at time t, and zero otherwise.

2.1. Camera Coverage Model

The following real-valued membership functions are used in our camera model. These functions provide a monotonically decreasing membership value over distance and relative angle of position to the camera.

We propose to use the following function, based on the well-known sigmoid function, to evaluate the distance membership:

$$\mu_{di}^{(t)} = \mu_d(\underbrace{\|\mathbf{p}_i - \mathbf{q}\|}_{d_i}) = 1 - \frac{1}{1 + \exp\left(-\beta_d(d_i - \alpha_{di}^{(t)})\right)},$$

with $\alpha_{di}^{(t)}$ and β_d as the parameters configuring the membership function. These parameters can be estimated using experimental observations on camera behaviours (e.g. object recognition rate as a function of distance). Parameter β_d controls the slope of the function and $\alpha_{di}^{(t)}$ determines the distance where the camera has 50% of its maximum coverage.

As for the pan angle membership functions, we propose the following function also based on the sigmoid function:

$$\mu_{pi}^{(t)} = \mu_p(\underbrace{\angle_p(\mathbf{q} - \mathbf{p}_i) - \theta_i}_{p_i}) = \frac{1}{1 + \exp\left(-\beta_p(p_i + \alpha_{pi}^{(t)})\right)} - \frac{1}{1 + \exp\left(-\beta_p(p_i - \alpha_{pi}^{(t)})\right)},$$

where $\alpha_{pi}^{(t)}$ controls the "width" of the function and β_p controls the slope of the function at the boundaries. Note that the proposed function has a range of $p_i \in [-180, 180]$ degrees. Therefore, any calculated angle should be brought into this range accordingly. Similarly, membership function μ_t is defined as:

$$\mu_{ti}^{(t)} = \mu_t(\underbrace{\angle_t(\mathbf{q} - \mathbf{p}_i) - \xi_i}_{t_i}) = \frac{1}{1 + \exp\left(-\beta_t(t_i + \alpha_{ti}^{(t)})\right)} - \frac{1}{1 + \exp\left(-\beta_t(t_i - \alpha_{ti}^{(t)})\right)},$$

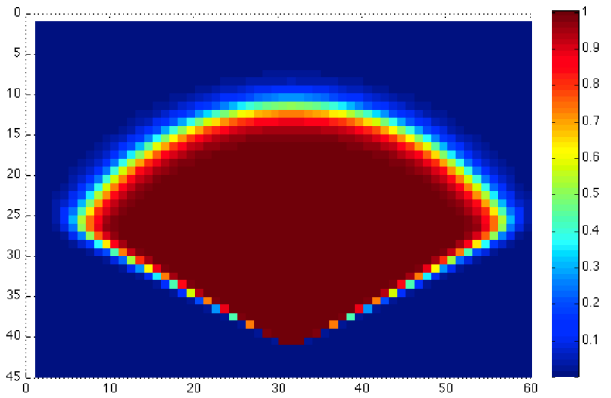


Figure 1: Probabilistic coverage model of a camera. Assuming that a camera is positioned at (30,40) heading upward, the colour shows different degrees of coverage for points inside the map. Note that in this figure only the distance and pan membership functions are visible. The camera has the same coverage over the tilt as the pan angle.

which has the range $t_i \in [-90, 90]$. The pan and tilt membership functions are related to the focal length value through the $\alpha_{pi}^{(t)}$ and $\alpha_{ti}^{(t)}$ parameters. More precisely, these parameters define the angle of view for the camera, so we have:

$$\alpha_{pi}^{(t)} = 2\arctan(L_h, 2f_i^{(t)}),$$
$$\alpha_{ti}^{(t)} = 2\arctan(L_v, 2f_i^{(t)}),$$

where L_h and L_v are the horizontal and vertical sizes of the camera's sensor, respectively.

The distance membership function is also related to the focal length value $f_i^{(t)}$ through the following formula:

$$\alpha_{di}^{(t)} = \left[\frac{\alpha_{d_{max}} - \alpha_{d_{min}}}{f_{max} - f_{min}}(f_i^{(t)} - f_{min})\right] + \alpha_{d_{min}},$$

where $\alpha_{d_{max}}$ and $\alpha_{d_{min}}$ are the maximum and minimum value that $\alpha_{di}^{(t)}$ can take. In practice, the value of these parameters depends on the camera type, size of the targets, the minimum number of pixels inside the images needed for accurate detection of target, and so forth.

3. Target's Displacement Prediction

We are now looking for predicting the next position of each target considering the previous movements of the targets in the environment.

The displacement vector of each target $\mathbf{T}_j$ at time t is represented by $d_j(t) = \mathbf{T}_j^{(t)} - \mathbf{T}_j^{(t-1)}$. Therefore, each target is attributed with a list $d_j = [d_j(s)|s = t_j + 1, \ldots, t]$. The goal is to predict the next displacement $\widehat{d_j}(t+1)$ using the history of past movements d_j. A simple approach

would be to take the average of all the previous movements:

$$\widehat{d}_j(t+1) = \frac{1}{n} \sum_{s=t_j+1}^{t} d_j(s),$$

where $n = t - t_j$. There are several problems related to this approach. First, it gives equal weight to all previous movements. For example, if a target has moved towards the west and then suddenly it changed direction and moved towards the east, this method keeps predicting the next position within the west side, until the east movements can outweigh the west movements. Therefore, it is clear that more recent movements are more important. This problem can be solved using an exponential moving average, as follows:

$$\widehat{d}_j(t+1) = \frac{1}{\sum_{k=0}^{k=n-1} \rho^k} \sum_{s=t_j+1}^{t} \rho^{t-s} d_j(s), \qquad (2)$$

where $\rho \in [0,1]$ is the weighting factor. The second problem is related to the shape of the final result, which is a vector, so this method gives high importance to one location and all neighbour locations obtain zero probability. Here we propose a probabilistic model for the next location of each target. More precisely, for a target $\mathbf{T}_j$ at time t we define a discrete random variable $X_j^{(t)}$ over the sample space of all the locations $\mathbf{q} \in \Xi$. $X_j^{(t)}$ is defined by a discrete probability mass function $g(\mathbf{T}_j, \mathbf{q}, t)$ which returns the probability of target $\mathbf{T}_j$ at location $\mathbf{q}$ at time t. The goal is to incrementally learn the function g as each target moves within the environment. So far we had a simple binary probability function where $g(\mathbf{T}_j, \mathbf{q}, t) = 1$ for only one location, but a more interesting approach is to use a bivariate Gaussian distribution around the predicted location:

$$\widehat{g}(\mathbf{T}_j, \mathbf{q}, t+1) = \iint_{\mathcal{A}_\mathbf{q}} \mathcal{N}_2(\widehat{d}_j(t+1), \mathbf{\Sigma}),$$

where $\widehat{g}(\mathbf{T}_j, \mathbf{q}, t+1)$ is the predicted probability that the target moves to location $\mathbf{q}$ in the next time step, $\mathcal{A}_\mathbf{q}$ is the cell inside the discretized environment around location $\mathbf{q}$, $\widehat{d}_j(t+1)$ is the mean of the bivariate distribution which we calculated in (2), and $\mathbf{\Sigma}$ is the covariance matrix. Here, for simplicity we assume $\mathbf{\Sigma} = \sigma_p^2 \mathbf{I}$.

For the third problem consider the following example. Assume the target has moved to the west for some time and then moved towards the north. After the transition, the predicted location would be towards the north-west for some time until it is gradually directed towards the north. While the target has never moved towards the north-west, and still, after the prediction has moved north, there is no probability of the target moving towards the west, while it is more reasonable that the probability of a west movement would be larger than a probability of a movement in other directions (*e.g.* east or south). To solve this problem, we define the probabilistic measure for all the previous movements and then sum up their values for all the locations $\mathbf{q}$. Therefore, in each time step, the probabilistic model of the last movement $g(\mathbf{T}_j, \mathbf{q}, t)$ for location $\mathbf{q}$ is added to the prediction of probability calculated for that location $\widehat{g}(\mathbf{T}_j, \mathbf{q}, t)$. We call this approach the multimodal approach and it is formalized as follows:

$$\widehat{g}(\mathbf{T}_j, \mathbf{q}, t+1) = \eta\, g(\mathbf{T}_j, \mathbf{q}, t) + (1-\eta)\, \widehat{g}(\mathbf{T}_j, \mathbf{q}, t), \qquad (3)$$

$$g(\mathbf{T}_j, \mathbf{q}, t) = \iint_{\mathcal{A}_\mathbf{q}} \mathcal{N}_2(d_j(t), \mathbf{\Sigma}), \qquad (4)$$

where $\eta \in [0,1]$ is the learning rate. As can be seen, each movement is creating a Gaussian distribution (4), and the effect of these distributions is incrementally added through time (3). This way, if the target has already moved toward a direction before, the probability of that direction would always be higher than other directions towards which the target has never moved.

So far, we have assumed that the covariance matrix is constant for all the movements. Another approach is to estimate the mean and covariance matrix of the next movement using the previous movements. This method is the most widely used method for target trajectory estimation in the literature [4, 6]. In this approach, all the previous displacements are weighted and the parameters are calculated using the maximum likelihood estimation. We call this approach the unimodal approach. More precisely, assuming that $\theta(t+1) = (d_j(t+1), \mathbf{\Sigma}_j(t+1))$ are the parameters of the bivariate normal distribution that we want to estimate, we have:

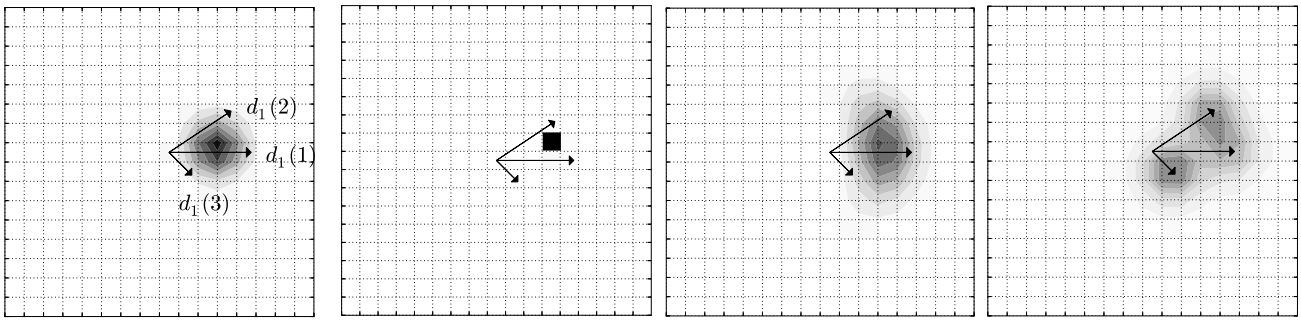
$$\widehat{\theta}(t+1) = \underset{\theta(t+1)}{\operatorname{argmax}} \prod_{s=t_j+1}^{s=t} \mathcal{N}_2(\rho^{t-s} d_j(s) | \theta(t+1)),$$

$$\widehat{g}(\mathbf{T}_j, \mathbf{q}, t+1) = \iint_{\mathcal{A}_\mathbf{q}} \mathcal{N}_2(\widehat{\theta}(t+1)).$$

Note that the estimation is done in polar coordinate systems. In other words, the distance r and angle θ between target position $\mathbf{T}_j^{(t)}$ and location $\mathbf{q}$ are used for calculations in this space.

Another possible approach for next movement prediction is the classical Kalman Filter method. In this approach the mean and covariance matrix is updated at each time step using the following formulas:

$$\begin{aligned}
\mathbf{K} &= \widehat{\mathbf{\Sigma}_j}(t)(\widehat{\mathbf{\Sigma}_j}(t) + \mathbf{\Sigma})^{-1}, \\
\widehat{d}_j(t+1) &= \widehat{d}_j(t) + \mathbf{K}(d_j(t) - \widehat{d}_j(t)), \\
\widehat{\mathbf{\Sigma}_j}(t+1) &= (\mathbf{I} - \mathbf{K})\widehat{\mathbf{\Sigma}_j}(t), \\
\widehat{g}(\mathbf{T}_j, \mathbf{q}, t+1) &= \iint_{\mathcal{A}_\mathbf{q}} \mathcal{N}_2(\widehat{d}_j(t+1), \widehat{\mathbf{\Sigma}_j}(t+1)),
\end{aligned}$$



(a) Weighted probabilistic (b) Kalman filter (c) Unimodal (d) Multimodal

Figure 2: Assume that target $\mathbf{T}_1$ was detected in the first time step $t_1 = 0$, and currently we are at $t = 3$. Therefore, we have the history of target $\mathbf{T}_1$ movements for the past three time steps $d_1 = \{d_1(1), d_1(2), d_1(3)\}$. The goal is to predict the next movement. The probability of next movement is shown for different prediction methods.

Figure 3: Part of the Université Laval map, chosen for the experiments. The map used for the experiments has 300 rows and 300 columns with a resolution of 1 meter per cell.

where $\mathbf{\Sigma} = \sigma_p^2\mathbf{I}$ is the covariance of the $d_j(t)$ displacement, and $\mathbf{K}$ is called the Kalman gain, which measures the relative certainty between the new measurement and the current estimate. All these prediction methods are presented in Fig. 2.

4. Experiments

In the following, the performance of the proposed target displacement prediction approach (multimodal) is compared with other prediction methods on the PTZ camera movement problem.

4.1. Simulation Setup

Simulation settings are as follows.

Map: Algorithms were tested over a map of Université Laval campus, in Québec, Canada. The map of the area is shown in Fig. 3.

Table 1: The parameter values for a realistic model of a camera.

Parameter	Value	Parameter	Value
L_h	5.37×10^{-3}	L_v	4.04×10^{-3}
$\alpha_{d_{max}}$	50	$\alpha_{d_{min}}$	25
f_{min}	4.7×10^{-3}	f_{max}	9.4×10^{-3}
β_d,β_p,and β_t	1	τ	1
ξ_{min}	-90^o	ξ_{max}	90^o
v_θ	$\pm 30^{o/t}$	v_ξ	$\pm 10^{o/t}$
v_f	$\pm 1.33 \times 10^{-3}$		

Cameras: Cameras are modelled following a description given in Sec. 2.1. For a reasonable model of a camera, we propose to use the parameters shown in Table 1. As can be seen, the maximum and minimum pan angles $(\theta_{min}, \theta_{max})$ are not defined in the table, meaning that cameras can freely rotate around the $\mathbf{Z}$ axis.

Targets: In our experiments each target is a simulation of a pedestrian walking on the campus. The target enters the environment from one gate of a building, walks inside the environment and exits from another gate (see Fig. 4). The trajectory of each target is generated based on the current location and the temporary goal location it is willing to reach. The temporary goal is generated based on the shortest path between the initial gate and the final gate on the pedestrian path of the campus. Here, the pedestrian path is represented as a graph with intersections on the path being the vertices of the graph and the paths themselves being the edges. Each intersection on the shortest path between the two gates can be a temporary goal for the target.

At each time step, the next location of a target is randomly chosen from a distribution which itself is produced by the multiplication of two other distributions, namely the beta distribution and the Gaussian distribution. More precisely, at each time step, a Gaussian distribution is applied on the angle between the target's location and its temporary goal and a beta distribution is applied on the distance to determine the step size. The multiplication between these two

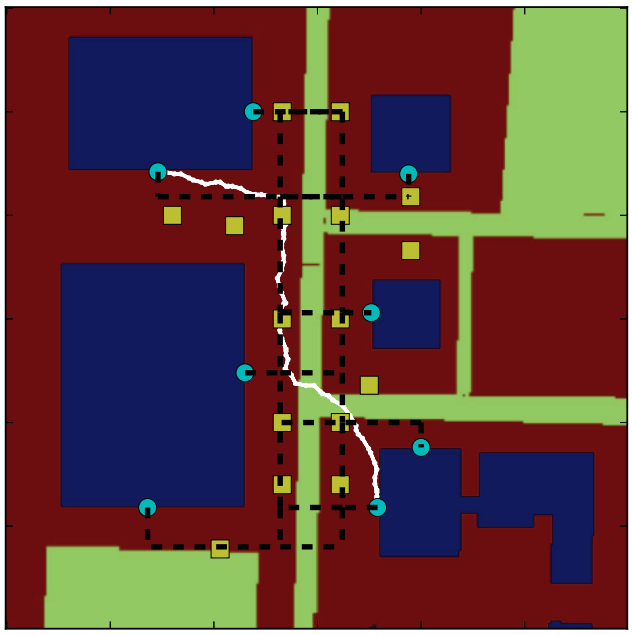


Figure 4: There are eight gates on the campus map. Each gate is presented by a cyan circle. Targets can enter from any of the eight gates, walk around the campus and exit the environment through another gate. The trajectory of a sample target is also shown using the white line. The pedestrian paths inside the campus are shown using the dashed black line. The target follows the path to move between gates. The position of the cameras is presented by the yellow squares.

Table 2: The parameter values for the trajectory generation.

Parameter	Value	Parameter	Value
α	2	β	2
μ	0	σ_a^2	125

Table 3: The parameter values for the trajectory prediction methods.

Parameter	Value	Parameter	Value
ρ	0.95	σ_p^2	4
δ	7	η	0.05

distributions is normalized to sum up to one and then used to determine the probability of each location in the environment to be selected as the next location of the target. Beta distribution has two parameters α and β and the Gaussian distribution has the parameters μ and σ_a. The maximum step size of the target is also shown with parameter δ (values for trajectory generation summarized in Table 2 and the values for different trajectory prediction methods in Table 3).

Table 4: The parameter values for the CMA-ES optimization method.

Parameter	σ_{CMA}	gen	λ
Value	0.33	100	14

4.2. Coverage optimization

To evaluate the performance of the multimodal prediction method explained in Section 3, we compare its performance with three other prediction methods, namely weighted probabilistic, Kalman filter, and unimodal. We use Covariance Matrix Adaptation Evolution Strategy (CMA-ES) as the optimization method to determine the optimal strategy at each time step, based on the prediction made by each of the four prediction methods.

CMA-ES is an optimization method that belongs to the class of evolutionary computation methods. Like classical quasi-Newton optimization methods, CMA-ES tries to estimate a second order model of the objective function in an iterative procedure. In contrast to quasi-Newton methods, CMA-ES does not need the gradient of the objective function. The algorithm's parameters that require manual adjustment are the mutation factor (σ_{CMA}), the number of generations through which the algorithm runs (gen), and the number of offsprings in each generation (λ) [1]. The value of these parameters is summarized in Table 4.

For the experiments, at each time step, the optimization strategy (CMA-ES) is applied on the camera directions to determine the directions and zoom level of each camera for the next time step. Next, all the targets move to their new positions, the combined coverage of the camera network is calculated as the weighted average of the coverage for each target $C_l(\mathbf{C}_i^{(t)}, \mathbf{T}_i^{(t)})$. The average is weighted by the number of targets present in the environment at each time step. The final result is the average coverage provided by each prediction method for all the targets through their presence in the environment.

To test different scenarios, we compare the performance of each strategy with different ratios between the number of targets and simulation time. Therefore, at each time step, a number of targets equal to the ratio between the simulation time to the total number of targets enter the environment. The final results of the experiments are presented in Table 5.

As can be seen in Table 5, the Kalman filter produced the worst results. The main reason for its inferior performance is that the Kalman filter cannot cope well with targets changing directions within their path. To improve the performance of this method there is a need for an extra mechanism to estimate if the target has changed direction or not. Therefore, each time that the target has changed directions the method should be reset in order to estimate the new temporary goal. That will be addressed in future work.

Another problem with the Kalman filter approach is that it is assuming a Gaussian noise for the displacement of the target. Although this might be the case, the multimodal approach can adapt with non-gaussian noise models and

Table 5: Coverage percentage of the targets with various numbers of targets. For each scenario, 32 sets of random paths were generated and the CMA-ES method was applied on each one of them, with average coverage of the network and corresponding standard deviation reported. Results in bold denote best results that are statistically significant according to a Student's t-test (p-value of 0.05).

No. of cameras	16	16	16
No. of targets	50	150	300
Simulation time	200	200	200
Kalman filter ave.	73.29%	75.89%	76.57%
Kalman filter std.	2.7	1.6	1.7
Weighted prob. ave.	78.50%	79.49%	79.28%
Weighted prob. std.	2.3	2.1	1.2
Unimodal ave.	79.05%	79.40%	79.18%
Unimodal std.	1.6	1.7	1.5
Multimodal ave.	**80.46%**	**81.28%**	**80.33%**
Multimodal std.	1.8	1.5	1.5

predict the displacement with good accuracy.

The weighted probabilistic approach produced stable results, which could be related to the simple assumptions of this method. The only problem with this approach is its unimodal movement assumption. Although this might not be the case for pedestrians, but in another application where targets may often change directions, the multimodal approach is the only method which has the capability to present both possibilities with sufficient probabilities.

The unimodal approach produced results close to the weighted probabilistic approach, which is not surprising considering that both methods produced similar predictions in Fig. 2. The strong point of the unimodal approach is that it is using almost the same assumptions as those that we are using for target trajectory generation in the environment. The weak point of the approach concerns the direction changes. When a target changes direction, it takes some time for the method to adapt to the new goal.

5. Conclusion

We have presented and compared different methods for prediction of targets position inside a camera network. The multimodal approach produced better results compared to other prediction methods. The strong point of the multimodal approach is its capability to handle several assumptions at the same time (*e.g.* zig-zag movement) and to adapt to non-gaussian movement models.

For future work, we will continue to improve the performance of the multimodal method. One possible approach is to use the history of past targets who passed a specific location in the environment to help predict the trajectory of future targets.

Acknowledgements

The project was supported by funding from Natural Sciences and Engineering Research Council (NSERC) of Canada. Computation resources have been provided by Calcul Québec / Compute Canada. We also thank Annette Schwerdtfeger for proofreading the manuscript.

References

[1] N. Hansen. The CMA evolution strategy: A tutorial. Available online at http://www.lri.fr/~hansen/cmatutorial.pdf (last checked Feb 26th, 2013), 2011. 6

[2] N. Krahnstoever, T. Yu, S.-N. Lim, K. Patwardhan, and P. Tu. Collaborative real-time control of active cameras in large scale surveillance systems. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2)*, 2008. 1

[3] S.-N. Lim, L. S. Davis, and A. Mittal. Constructing task visibility intervals for video surveillance. *Multimedia systems*, 12(3):211–226, 2006. 1

[4] P. Natarajan, T. N. Hoang, K. H. Low, and M. Kankanhalli. Decision-theoretic approach to maximizing observation of multiple targets in multi-camera surveillance. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 1, pages 155–162, 2012. 1, 4

[5] F. Z. Qureshi and D. Terzopoulos. Planning ahead for ptz camera assignment and handoff. In *Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–8. IEEE, 2009. 1

[6] A. Rahimi, B. Dunagan, and T. Darrell. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 187–194. IEEE, 2004. 1, 4